POLITECNICO
MILANO 1863

Department of Aerospace Science & Technology

Multibody System Dynamics
-
Prof. Pierangelo Masarati

MSD Project: flap extraction and retraction system

Group 6 Project Report - A.Y. 2022/2023

Authors:

10619664 - Alberto Carnevali
10914788 - Alvaro Fernandez
10905975 - Guillem Landa

Contents

1 Project description **2**
1.1 Assumptions . . . 2

2 Kinematics **3**
2.1 First kinematic equation: (A - O) + (G - A) = (C - O) + (G - C) . . . 3
2.2 Second kinematic equation: (D - B) + (G - D) = (C - B) + (G - C) . . . 5
2.3 Third kinematic equation: (G - C) + (E - G) + (H - E) = (F - C) + (H - F) . 6

3 Dynamics **7**
3.1 Equation of motion . . . 7
3.2 DC Motor . . . 9

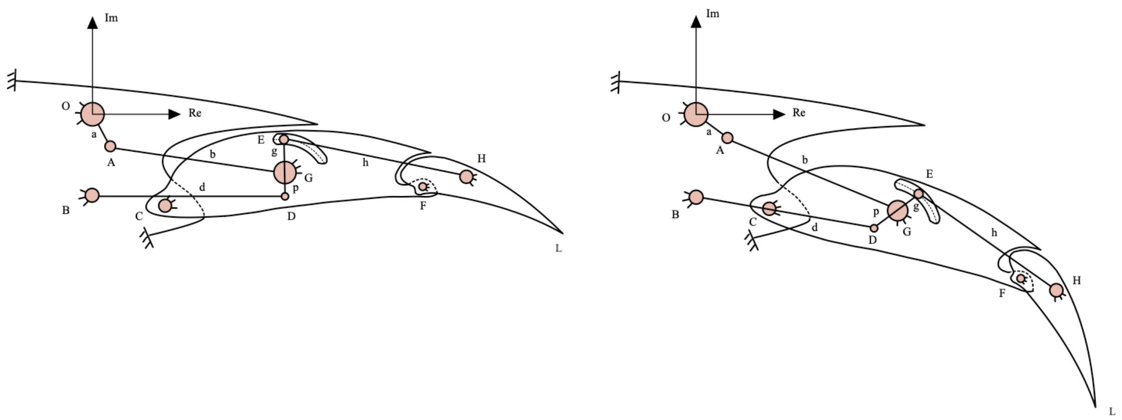
4 Matlab simulation **10**
4.1 Non-linear equations . . . 11
4.2 Final results . . . 12

5 Multibody solver simulation: MBDyn **13**
5.1 Modelling process . . . 13
5.2 Post-processing . . . 14

6 Comparison of the results **14**
6.1 Kinematic results . . . 14
6.2 Dynamic results . . . 16
6.3 Filter implementation . . . 17

1 Project description

A double flap extraction system prototype is here presented in Fig. 1:



(a) Rest position. **(b)** Extraction position.

Figure 1: Flap activation mechanism.

It consists of a 2D five-bars mechanism, in which the main flap is connected to a wing's rib through a double beam connection OAG, allowing it to rotate around hinge C. A second double beam connection BDG causes hinge E to slide into its rail as the main flap rotates, triggering the second flap to revolve around hinge F.

The whole structure is driven by a torque T_m applied over hinge O, anti-clockwise direction, forcing the main crank $\overline{OA}$ to rotate at constant speed $\dot{\alpha}$, resulting in a single degree of freedom system: 1 DOF $= \alpha(t)$.

The main goal of this project is to describe the correct motion of the system, while obtaining a total deflection angle of 60° of the second flap.

In order to do so, the whole system is analytically investigated using Gauss plane reference formulation for the kinematic study, while the dynamic functioning is analyzed through the Work-Energy Principle.

Finally, two numerical simulations are carried out using a standard programming platform (Matlab) and a Multibody software (MBDyn).

1.1 Assumptions

The physical model taken into account implies some approximations and ideal assumptions in order to simplify the real system, including:

1. RIGID BODY: all bodies, beams are assumed as rigid bodies.
2. IDEAL HINGES: all hinges, guide-trails are considered as ideal.
3. LUMPED APPROACH: mass and inertia moment are considered as lumped, thus concentrated in the bodies' centers of mass.
4. LINEARIZED AERODYNAMICS: aerodynamic forces are linearized over each flap's deflection angle, and applied on the aerodynamic centers ($x_{AC} = \frac{1}{4}c$).

2 Kinematics

Kinematics is investigated through the use of complex analysis:

$$\begin{cases}(A-O)=ae^{i\alpha}\\ (G-A)=be^{i\beta}\\ (C-O)=l-ir\\ (G-C)=ce^{i\gamma}\\ (C-B)=s-it\\ (D-B)=de^{i\phi}\end{cases}\qquad \begin{cases}(F-C)=c_1e^{i(\gamma-\frac{\pi}{2})}\\ (G-D)=pe^{i\psi}\\ (E-G)=ge^{i\psi}\\ (H-E)=he^{i\xi}\\ (H-F)=ze^{i\tau}\\ (L-F)=c_2e^{i(\tau-\frac{\pi}{2})}\end{cases}$$

Where:

$$\begin{cases}\gamma=\frac{\pi}{2}-\delta \quad \longrightarrow \quad (G-C)=ce^{i\gamma}=ice^{-i\delta}\\ \tau=\frac{\pi}{2}-\theta \quad \longrightarrow \quad (H-F)=ze^{i\tau}=ize^{-i\theta}\\ \text{F is assumed to be placed at the end of main flap's chord } c_1\\ \text{L is assumed to be placed over the trailing edge of the secondary flap}\end{cases}$$

Important : Every angle measurement starts from the positive real axis of the Gauss plane, counterclockwise direction.

Three main kinematic chain equations are highlighted in order to better describe every translation, rotation of each body and beam as a function of the main crank rotating angle $\alpha(t)$. In particular, the dependency between the deflection angle of first and second flap δ and θ respectively, needs to be obtained so that the aerodynamic force acting on the two surfaces can be measured.

- **First kinematic chain equation:** $(A-O)+(G-A)=(C-O)+(G-C)$
- **Second kinematic chain equation:** $(D-B)+(G-D)=(C-B)+(G-C)$
- **Third kinematic chain equation:** $(G-C)+(E-G)+(H-E)=(F-C)+(H-F)$

2.1 First kinematic equation: (A - O) + (G - A) = (C - O) + (G - C)

This kinematic bond provides the dependency between β,δ and α (and their derivatives).

- Position equation: $ae^{i\alpha}+be^{i\beta}-l+ir-ice^{-i\delta}=0$

$$\begin{cases}acos(\alpha)+bcos(\beta)-l-csin(\delta)=0\\ asin(\alpha)+bsin(\beta)+r-ccos(\delta)=0\end{cases}\longrightarrow\begin{cases}Real(\beta,\delta)=0\\ Imm(\beta,\delta)=0\end{cases}$$

  This is a non linear algebraic system of equations: $\beta(\alpha)$ and $\delta(\alpha)$ can be obtained using Matlab function `fsolve`, or by implementing Newton-Raphson method for multiple equations.

- Velocity equation: $ia\dot{\alpha}e^{i\alpha} + ib\dot{\beta}e^{i\beta} - c\dot{\delta}e^{-i\delta} = 0$

$$\begin{cases} a\dot{\alpha}sin(\alpha) + b\dot{\beta}sin(\beta) + c\dot{\delta}cos(\delta) = 0 \\ a\dot{\alpha}cos(\alpha) + b\dot{\beta}cos(\beta) + c\dot{\delta}sin(\delta) = 0 \end{cases}$$

$\beta(\alpha)$ and $\delta(\alpha)$ are already known from the previous step, turning the velocity kinematic equation into a linear algebraic system of equations:

$$\mathbf{A}\dot{\mathbf{x}} = \mathbf{b} \qquad \longrightarrow \qquad \begin{bmatrix} bsin(\beta) & ccos(\delta) \\ bcos(\beta) & csin(\delta) \end{bmatrix} \begin{bmatrix} \dot{\beta} \\ \dot{\delta} \end{bmatrix} = -a \begin{bmatrix} sin(\alpha) \\ cos(\alpha) \end{bmatrix} \dot{\alpha}$$

$$\dot{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b} \qquad \longrightarrow \qquad \begin{bmatrix} \dot{\beta} \\ \dot{\delta} \end{bmatrix} = \frac{a}{cbcos(\beta+\delta)} \begin{bmatrix} csin(\delta) & -ccos(\delta) \\ -bcos(\beta) & bsin(\beta) \end{bmatrix} \begin{bmatrix} sin(\alpha) \\ cos(\alpha) \end{bmatrix} \dot{\alpha}$$

Where: $\quad det(\mathbf{A}) = -cbcos(\beta+\delta) \neq 0 \qquad \longleftrightarrow \qquad (\beta+\delta) \neq \frac{\pi}{2} + k\pi, \quad with \quad k \in Z$

The values of $\dot{\beta}, \dot{\delta}$ are finally reported showing the explicit dependency from α (this dependency will turn very useful when analyzing the dynamics of the system):

$$\begin{cases} \dot{\beta} = \frac{a}{cbcos(\beta+\delta)} c\left[sin(\alpha)sin(\delta) - cos(\alpha)cos(\delta)\right]\dot{\alpha} = -\frac{acos(\alpha+\delta)}{bcos(\beta+\delta)}\dot{\alpha} = \frac{\partial\beta}{\partial\alpha}\dot{\alpha} \\ \\ \dot{\delta} = \frac{a}{cbcos(\beta+\delta)} b\left[-cos(\beta)sin(\alpha) + cos(\alpha)sin(\beta)\right]\dot{\alpha} = \frac{asin(\beta-\alpha)}{ccos(\beta+\delta)}\dot{\alpha} = \frac{\partial\delta}{\partial\alpha}\dot{\alpha} \end{cases}$$

- Acceleration equation: $-a\dot{\alpha}^2e^{i\alpha} + ib\ddot{\beta}e^{i\beta} - b\dot{\beta}^2e^{i\beta} - c\ddot{\delta}e^{-i\delta} + ic\dot{\delta}^2e^{-i\delta} = 0$
  The $\ddot{\alpha}$ term is neglected since the dynamic investigation will assume a constant angular velocity $\dot{\alpha}$ later on.

$$\begin{cases} a\dot{\alpha}^2cos(\alpha) + b\ddot{\beta}sin(\beta) + b\dot{\beta}^2cos(\beta) + c\ddot{\delta}cos(\delta) - c\dot{\delta}^2sin(\delta) = 0 \\ -a\dot{\alpha}^2sin(\alpha) + b\ddot{\beta}cos(\beta) - b\dot{\beta}^2sin(\beta) + c\ddot{\delta}sin(\delta) + c\dot{\delta}^2cos(\delta) = 0 \end{cases}$$

As well as for the velocity equation, $\ddot{\beta}, \ddot{\delta}$ can be obtained through solving a linear algebraic system of equations:

$$\begin{bmatrix} -bsin(\beta) & ccos(\delta) \\ bcos(\beta) & csin(\delta) \end{bmatrix} \begin{bmatrix} \ddot{\beta} \\ \ddot{\delta} \end{bmatrix} = a \begin{bmatrix} -cos(\alpha) \\ sin(\alpha) \end{bmatrix} \dot{\alpha}^2 + b \begin{bmatrix} -cos(\beta) \\ sin(\beta) \end{bmatrix} \dot{\beta}^2 + c \begin{bmatrix} sin(\delta) \\ -cos(\delta) \end{bmatrix} \dot{\delta}^2$$

$$\begin{bmatrix} \ddot{\beta} \\ \ddot{\delta} \end{bmatrix} = \frac{-a}{cbcos(\beta+\delta)} \begin{bmatrix} csin(\delta) & -ccos(\delta) \\ -bcos(\beta) & bsin(\beta) \end{bmatrix} a \begin{bmatrix} -cos(\alpha) \\ sin(\alpha) \end{bmatrix} \dot{\alpha}^2 + b \begin{bmatrix} -cos(\beta) \\ sin(\beta) \end{bmatrix} \dot{\beta}^2 + c \begin{bmatrix} sin(\delta) \\ -cos(\delta) \end{bmatrix} \dot{\delta}^2$$

Since $\mathbf{A}$ is the same matrix of the velocity equation, the former considerations regarding $det(\mathbf{A})$ are still valid for every kinematic chain.

2.2 Second kinematic equation: (D - B) + (G - D) = (C - B) + (G - C)

This kinematic bond provides the dependency between ϕ, ψ and $\delta(\alpha)$, thus their bond with α. As previously investigated in the latest section, the position relationship will be a non-linear algebraic system of equations, which can be solved numerically using Matlab function `fsolve` or `newtonSys`, while the velocity and acceleration relationships will be linear equations.

- Position equation: $de^{i\phi} + pe^{i\psi} - s + it - ice^{-i\delta} = 0$

$$\begin{cases} dcos(\phi) + pcos(\psi) - s - csin(\delta) = 0 \\ dsin(\phi) + psin(\psi) + t - ccos(\delta) = 0 \end{cases} \longrightarrow \begin{cases} \phi(\delta) \\ \psi(\delta) \end{cases}$$

- Velocity equation: $id\dot{\phi}e^{i\phi} + ip\dot{\psi}e^{i\psi} - c\dot{\delta}e^{-i\delta} = 0$

$$\begin{cases} d\dot{\phi}sin(\phi) + p\dot{\psi}sin(\beta) + c\dot{\delta}cos(\delta) = 0 \\ d\dot{\phi}cos(\phi) + p\dot{\psi}cos(\psi) + c\dot{\delta}sin(\delta) = 0 \end{cases}$$

$$\mathbf{A\dot{x}} = \mathbf{b} \qquad \longrightarrow \qquad \begin{bmatrix} dsin(\phi) & psin(\psi) \\ dcos(\phi) & pcos(\psi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\psi} \end{bmatrix} = -c \begin{bmatrix} cos(\delta) \\ sin(\delta) \end{bmatrix} \dot{\delta}$$

$$\mathbf{\dot{x}} = \mathbf{A}^{-1}\mathbf{b} \qquad \longrightarrow \qquad \begin{bmatrix} \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \frac{-c}{dpsin(\phi+\psi)} \begin{bmatrix} pcos(\psi) & -psin(\psi) \\ -dcos(\phi) & dsin(\phi) \end{bmatrix} \begin{bmatrix} cos(\delta) \\ sin(\delta) \end{bmatrix} \dot{\delta}$$

Where $det(\mathbf{A}) = dpsin(\phi - \psi) \neq 0 \quad \longleftrightarrow \quad (\phi - \psi) \neq k\pi, \quad with \quad k \in Z$

$$\begin{cases} \dot{\phi} = \frac{-c}{dpsin(\psi-\phi)} p[cos(\psi)cos(\delta) - sin(\psi)sin(\delta)]\dot{\delta} = \frac{ccos(\psi+\delta)}{dsin(\psi-\phi)}\dot{\delta} = \frac{\partial\phi}{\partial\delta}\dot{\delta} = \frac{\partial\phi}{\partial\delta}\frac{\partial\delta}{\partial\alpha}\dot{\alpha} \\ \\ \dot{\psi} = \frac{-c}{dpsin(\psi-\phi)} d[-cos(\phi)cos(\delta) + sin(\phi)sin(\delta)]\dot{\delta} = \frac{ccos(\phi+\delta)}{psin(\phi-\psi)}\dot{\delta} = \frac{\partial\psi}{\partial\delta}\dot{\delta} = \frac{\partial\psi}{\partial\delta}\frac{\partial\delta}{\partial\alpha}\dot{\alpha} \end{cases}$$

- Acceleration equation: $id\ddot{\phi}e^{i\phi} - d\dot{\phi}^2e^{i\phi} + ip\ddot{\psi}e^{i\psi} - p\dot{\psi}^2e^{i\psi} - c\ddot{\delta}e^{-i\delta} + ic\dot{\delta}^2e^{-i\delta} = 0$

$$\begin{cases} d\ddot{\phi}sin(\phi) + d\dot{\phi}^2cos(\phi) + p\ddot{\psi}sin(\psi) + p\dot{\psi}^2cos(\psi) + c\ddot{\delta}cos(\delta) - c\dot{\delta}sin(\delta) = 0 \\ d\ddot{\phi}cos(\phi) - d\dot{\phi}^2sin(\phi) + p\ddot{\psi}cos(\psi) - p\dot{\psi}^2sin(\psi) + c\ddot{\delta}sin(\delta) + c\dot{\delta}cos(\delta) = 0 \end{cases}$$

$$\begin{bmatrix} dsin(\phi) & psin(\psi) \\ dcos(\phi) & pcos(\psi) \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\psi} \end{bmatrix} = \mathbf{b}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\psi} \end{bmatrix} = \frac{c}{dpsin(\phi-\psi)} \begin{bmatrix} pcos(\psi) & -psin(\psi) \\ -dcos(\phi) & dsin(\phi) \end{bmatrix} \mathbf{b};$$

With: $\mathbf{b} = d \begin{bmatrix} -cos(\phi) \\ sin(\phi) \end{bmatrix} \dot{\phi}^2 + p \begin{bmatrix} -cos(\psi) \\ sin(\psi) \end{bmatrix} \dot{\psi}^2 + c \begin{bmatrix} sin(\delta) \\ -cos(\delta) \end{bmatrix} \dot{\delta}^2 - c \begin{bmatrix} cos(\delta) \\ sin(\delta) \end{bmatrix} \ddot{\delta}$

2.3 Third kinematic equation: (G - C) + (E - G) + (H - E) = (F - C) + (H - F)

This kinematic bond provides the dependency between θ, ξ and $\psi(\delta)$, thus their bond with α. Again, the position relationship will be a non-linear algebraic system of equations, which can be solved numerically using Matlab function `nfsolve` or `newtonSys`, while the velocity and acceleration relationships will be linear equations.

- Position equation: $(ic - c_1)e^{-i\delta} + ge^{i\psi} + he^{i\xi} - ize^{-i\theta} = 0$

$$\begin{cases} csin(\delta) - c_1cos(\delta) + gcos(\psi) + hcos(\xi) - zsin(\theta) = 0 \\ ccos(\delta) + c_1sin(\delta) + gsin(\psi) + hsin(\xi) - zcos(\theta) = 0 \end{cases} \longrightarrow \begin{cases} \xi(\psi) \\ \theta(\psi) \end{cases}$$

- Velocity equation: $(c + ic_1)\dot{\delta}e^{-i\delta} + ig\dot{\psi}e^{i\psi} + ih\dot{\xi}e^{i\xi} - z\dot{\theta}e^{-i\theta} = 0$

$$\begin{cases} -c\dot{\delta}cos(\delta) - c_1\dot{\delta}sin(\delta) + g\dot{\psi}sin(\psi) + h\dot{\xi}sin(\xi) + z\dot{\theta}cos(\theta) = 0 \\ -c\dot{\delta}sin(\delta) + c_1\dot{\delta}cos(\delta) + g\dot{\psi}cos(\psi) + h\dot{\xi}cos(\xi) + z\dot{\theta}sin(\theta) = 0 \end{cases}$$

$$\mathbf{A\dot{x}} = \mathbf{b} \quad \longrightarrow \quad \begin{bmatrix} hsin(\xi) & zcos(\theta) \\ hcos(\xi) & zsin(\theta) \end{bmatrix} \begin{bmatrix} \dot{\xi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} ccos(\delta) + c_1sin(\delta) \\ csin(\delta) - c_1cos(\delta) \end{bmatrix} - g \begin{bmatrix} sin(\psi) \\ cos(\psi) \end{bmatrix} \dot{\psi}$$

$$\begin{bmatrix} \dot{\xi} \\ \dot{\theta} \end{bmatrix} = -\frac{1}{hzcos(\theta+\xi)} \begin{bmatrix} zsin(\theta) & -zcos(\theta) \\ -hcos(\xi) & hsin(\xi) \end{bmatrix} \left(\begin{bmatrix} ccos(\delta) + c_1sin(\delta) \\ csin(\delta) - c_1cos(\delta) \end{bmatrix} \dot{\delta} - g \begin{bmatrix} sin(\psi) \\ cos(\psi) \end{bmatrix} \dot{\psi} \right)$$

Where $det(\mathbf{A}) = -hzcos(\theta + \xi) \neq 0 \quad \longleftrightarrow \quad (\theta + \xi) \neq \frac{\pi}{2}k, \quad with \quad k \in Z$

$$\begin{cases} \dot{\xi} = \frac{csin(\delta-\theta) - c_1cos(\theta-\delta)}{hcos(\theta+\xi)}\dot{\delta} - \frac{gcos(\theta+\psi)}{hcos(\theta+\xi)}\dot{\psi} = \frac{\partial\xi}{\partial\delta}\dot{\delta} + \frac{\partial\xi}{\partial\psi}\dot{\psi} \\ \\ \dot{\theta} = \frac{ccos(\delta+\xi) + c_1sin(\xi+\delta)}{zcos(\theta+\xi)}\dot{\delta} + \frac{gsin(\xi-\psi)}{zcos(\theta+\xi)}\dot{\psi} = \frac{\partial\theta}{\partial\delta}\dot{\delta} + \frac{\partial\theta}{\partial\psi}\dot{\psi} \end{cases}$$

- Acceleration equation:
$(c + ic_1)\ddot{\delta}e^{-i\delta} + (c_1 - ic)\dot{\delta}^2e^{-i\delta} + ig\ddot{\psi}e^{i\psi} - g\dot{\psi}^2e^{i\psi} + ih\ddot{\xi}e^{i\xi} - h\dot{\xi}^2e^{i\xi} - z\ddot{\theta}e^{-i\theta} + iz\dot{\theta}^2e^{-i\theta} = 0$

$$\begin{cases} Re: & -c\ddot{\delta}cos(\delta) - c_1\ddot{\delta}sin(\delta) - c_1\dot{\delta}^2cos(\delta) + c\dot{\delta}^2sin(\delta) + g\ddot{\psi}sin(\psi) + g\dot{\psi}^2cos(\psi) \\ & +h\ddot{\xi}sin(\xi) + h\dot{\xi}^2cos(\xi) + z\ddot{\theta}cos(\theta) - z\dot{\theta}^2sin(\theta) = 0 \\ \\ Im: & -c\ddot{\delta}sin(\delta) + c_1\ddot{\delta}cos(\delta) - c_1\dot{\delta}^2sin(\delta) - c\dot{\delta}^2cos(\delta) + g\ddot{\psi}cos(\psi) - g\dot{\psi}^2sin(\psi) \\ & +h\ddot{\xi}cos(\xi) - h\dot{\xi}^2sin(\xi) + z\ddot{\theta}sin(\theta) + z\dot{\theta}^2cos(\theta) = 0 \end{cases}$$

$$\begin{bmatrix} hsin(\xi) & zcos(\theta) \\ hcos(\xi) & zsin(\theta) \end{bmatrix} \begin{bmatrix} \ddot{\xi} \\ \ddot{\theta} \end{bmatrix} = \mathbf{b} \quad \longrightarrow \quad \begin{bmatrix} \ddot{\xi} \\ \ddot{\theta} \end{bmatrix} = -\frac{1}{hzcos(\theta+\xi)} \begin{bmatrix} zsin(\theta) & -zcos(\theta) \\ -hcos(\xi) & hsin(\xi) \end{bmatrix} \mathbf{b}$$

$$With: \quad \mathbf{b} = \begin{bmatrix} ccos(\delta) + c_1sin(\delta) \\ csin(\delta) - c_1cos(\delta) \end{bmatrix} \ddot{\delta} + \begin{bmatrix} c_1cos(\delta) - csin(\delta) \\ c_1sin(\delta) + ccos(\delta) \end{bmatrix} \dot{\delta}^2 + \\ - g \begin{bmatrix} sin(\psi) \\ cos(\psi) \end{bmatrix} \ddot{\psi} + g \begin{bmatrix} -cos(\psi) \\ sin(\psi) \end{bmatrix} \dot{\psi}^2 + h \begin{bmatrix} -cos(\xi) \\ sin(\xi) \end{bmatrix} \dot{\xi}^2 + z \begin{bmatrix} sin(\theta) \\ -cos(\theta) \end{bmatrix} \dot{\theta}^2$$

Every rotation, translation is now reported in form of a dependency from the main crank rotation angle $\alpha(t)$ and its derivatives.

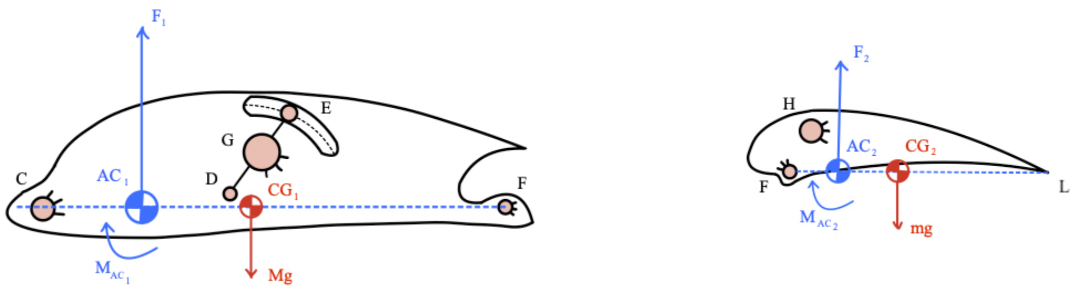
3 Dynamics

3.1 Equation of motion

The equation of motion can be developed using several methods. Since this is a 1 DOF system with scleronomic constrains only, the Work-Energy Principle can be applied (using the differential formula):

$$\frac{dT}{dt} = \Pi$$

Where T is the kinetic energy of all bodies, while Π is the sum of the active powers acting on the system as reported on Fig.2.



(a) Main flap dynamics.

(b) Second flap dynamics.

Figure 2: Active forces on the whole system.

$$\begin{cases} T = \frac{1}{2}M\left|\dot{\mathbf{x}}_{\mathbf{G_1}}\right|^2 + \frac{1}{2}J_M\dot{\delta}^2 + \frac{1}{2}m\left|\dot{\mathbf{x}}_{\mathbf{G_2}}\right|^2 + \frac{1}{2}J_m\dot{\theta}^2 \\ \\ \Pi = T_m\dot{\alpha} + \mathbf{F_1}\dot{\mathbf{x}}_{\mathbf{AC_1}} + \mathbf{F_2}\dot{\mathbf{x}}_{\mathbf{AC_2}} + M_{AC_1}\dot{\delta} + M_{AC_2}\dot{\theta} - Mg\dot{y}_{G_1} - mg\dot{y}_{G_2} \end{cases}$$

Where:

$$\begin{cases} M = \text{main flap's mass} \\ m = \text{second flap's mass} \\ J_M = \text{main flap's inertia} \\ J_m = \text{second flap's inertia} \end{cases} \qquad \begin{cases} F_1 = \text{main flap's aerodynamic force} \\ F_2 = \text{second flap's aerodynamic force} \\ M_{AC_1} = \text{main flap's aerodynamic moment} \\ M_{AC_2} = \text{second flap's aerodynamic moment} \end{cases}$$

The flap's centers of gravity and aerodynamic centers are placed at 50% and 25% of each chord, respectively; both aerodynamic forces are applied over the aerodynamic centers, with their vectors always being perpendicular to chords c_1 and c_2, while overall drag is neglected as the aerodynamic load will bend backwards along the flaps' deflection angles:

$$\begin{cases} F_1 = \frac{1}{2}\rho U^2 S_1 C_{L/\delta}\delta \\ F_2 = \frac{1}{2}\rho U^2 S_2 C_{L/\theta}\theta \end{cases} \qquad \begin{cases} M_{AC_1} = \frac{1}{2}\rho U^2 S_1 c_1 C_{MAC_1} \\ M_{AC_2} = \frac{1}{2}\rho U^2 S_2 c_2 C_{MAC_2} \end{cases}$$

Air density value ρ, landing speed U, flap surfaces S_1 S_2 and aerodynamic coefficients C_L C_{MAC}, along other aero-mechanical data, are listed in section 4.

Positions and velocities can be expressed as:

$$\begin{cases} x_F = x_C + c_1 cos(\gamma - \frac{\pi}{2}) = x_C + c_1 cos(\delta) \\ y_F = y_C + c_1 sin(\gamma - \frac{\pi}{2}) = y_C - c_1 sin(\delta) \end{cases}$$

$$\begin{cases} x_{G_1} = x_C + \frac{c_1}{2} cos(\delta) \\ y_{G_1} = y_C - \frac{c_1}{2} sin(\delta) \end{cases} \longrightarrow \begin{cases} \dot{x}_{G_1} = -\frac{c_1}{2} \dot{\delta} sin(\delta) \\ \dot{y}_{G_1} = -\frac{c_1}{2} \dot{\delta} cos(\delta) \end{cases}$$

$$\begin{cases} x_{G_2} = x_F + \frac{c_2}{2} cos(\theta) \\ y_{G_2} = y_F - \frac{c_2}{2} sin(\theta) \end{cases} \longrightarrow \begin{cases} \dot{x}_{G_2} = -c_1 \dot{\delta} sin(\delta) - \frac{c_2}{2} \dot{\theta} sin(\theta) \\ \dot{y}_{G_2} = -c_1 \dot{\delta} cos(\delta) - \frac{c_2}{2} \dot{\theta} cos(\theta) \end{cases}$$

$$\begin{cases} x_{AC_1} = x_C + \frac{c_1}{4} cos(\delta) \\ y_{AC_1} = y_C - \frac{c_1}{4} sin(\delta) \end{cases} \longrightarrow \begin{cases} \dot{x}_{AC_1} = -\frac{c_1}{4} \dot{\delta} sin(\delta) \\ \dot{y}_{AC_1} = -\frac{c_1}{4} \dot{\delta} cos(\delta) \end{cases}$$

$$\begin{cases} x_{AC_2} = x_F + \frac{c_2}{4} cos(\theta) \\ y_{AC_2} = y_F - \frac{c_2}{4} sin(\theta) \end{cases} \longrightarrow \begin{cases} \dot{x}_{AC_2} = -c_1 \dot{\delta} sin(\delta) - \frac{c_2}{4} \dot{\theta} sin(\theta) \\ \dot{y}_{AC_2} = -c_1 \dot{\delta} cos(\delta) - \frac{c_2}{4} \dot{\theta} cos(\theta) \end{cases}$$

While aerodynamic forces can be divided into horizontal and vertical components:

$$\begin{cases} F_{1x} = F_1 cos(\gamma) = F_1 sin(\delta) \\ F_{1y} = F_1 cos(\delta) \end{cases} \qquad \begin{cases} F_{2x} = F_2 cos(\tau) = F_2 sin(\theta) \\ F_{2y} = F_2 cos(\theta) \end{cases}$$

Finally, the equation of motion can be expressed highlighting the dependency from $\dot{\alpha}$, from which $T_m(\alpha, \dot{\alpha})$ can be achieved:

$$T = \frac{1}{2}\left(M\frac{{c_1}^2}{4} + J_M + mc_1^2\right)\dot{\delta}^2 + \frac{1}{2}\left(m\frac{{c_2}^2}{4} + J_m\right)\dot{\theta}^2 + \frac{1}{2}mc_1c_2cos\left(\theta - \delta\right)\dot{\delta}\dot{\theta}$$

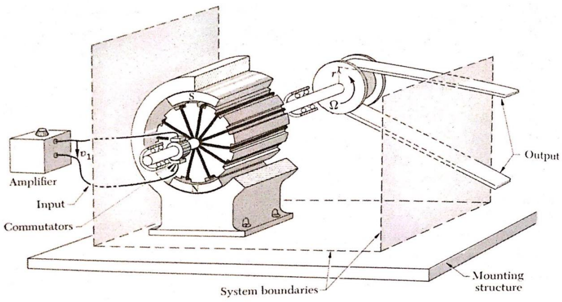
$$\begin{aligned} \frac{dT}{dt} = &\left[\ddot{\delta}(M\frac{{c_1}^2}{4} + J_M + mc_1^2)\frac{\partial\delta}{\partial\alpha} + \ddot{\theta}(m\frac{{c_1}^2}{4} + J_m)\frac{\partial\theta}{\partial\alpha}\right]\dot{\alpha} \\ &+ \frac{1}{2}mc_1c_2\left[cos(\theta-\delta)(\ddot{\theta}\frac{\partial\delta}{\partial\alpha} + \ddot{\delta}\frac{\partial\theta}{\partial\alpha}) - \frac{\partial\delta}{\partial\alpha}\frac{\partial\theta}{\partial\alpha}sin(\theta-\delta)(\frac{\partial\theta}{\partial\alpha} - \frac{\partial\delta}{\partial\alpha})\dot{\alpha}^2\right]\dot{\alpha} \end{aligned}$$

$$\begin{aligned} \Pi = &\left[M_{AC_1} - \frac{c_1}{4}F_1 - F_2c_1cos(\theta-\delta) + Mg\frac{c_1}{2}cos(\delta) + mgc_1cos(\delta)\right]\frac{\partial\delta}{\partial\alpha}\dot{\alpha} \\ &+ \left[\left(M_{AC_2} - \frac{c_2}{4}F_2 + mg\frac{c_2}{2}cos(\theta)\right)\frac{\partial\theta}{\partial\alpha} + T_m\right]\dot{\alpha} \end{aligned}$$

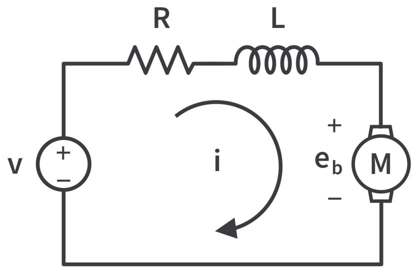
$$\begin{aligned} T_m = &\left[\ddot{\delta}(M\frac{{c_1}^2}{4} + J_M + mc_1^2) - M_{AC_1} + F_1\frac{c_1}{4} + F_2c_1sin(\theta-\delta) - (Mg\frac{c_1}{2} + mgc_1)cos(\delta)\right]\frac{\partial\delta}{\partial\alpha} \\ &+ \left[\ddot{\theta}(m\frac{{c_2}^2}{4} + J_m) - M_{AC_2} + F_2\frac{c_2}{4} - mg\frac{c_2}{2}cos(\theta)\right]\frac{\partial\theta}{\partial\alpha} \\ &+ \frac{1}{2}mc_1c_2\left[(\ddot{\theta}\frac{\partial\delta}{\partial\alpha} + \ddot{\delta}\frac{\partial\theta}{\partial\alpha})cos(\theta-\delta) - \frac{\partial\delta}{\partial\alpha}\frac{\partial\theta}{\partial\alpha}sin(\theta-\delta)(\frac{\partial\theta}{\partial\alpha} - \frac{\partial\delta}{\partial\alpha})\dot{\alpha}^2\right] \end{aligned}$$

3.2 DC Motor

The torque T_m applied on the main crank $\overline{OA}$ can be obtained introducing a DC electric motor, sketched in Fig.3.



(a) DC motor illustration.



(b) Physical model of a DC motor.

Figure 3: Sketch of a DC motor.

The behavior of the DC motor can be characterized by a RL circuit, coupled with the mechanical system of the rotor shaft:

$$\begin{cases} v_i = R_a i + L_a \frac{di}{dt} + e_b \\ T_m = K_m i \\ e_b = K_m \omega \end{cases}$$

Both the motor torque T_m and the voltage drop across the motor's coil v_b are proportional to the torque constant $K_m = \frac{4}{\pi} NBRL$, which depends on the geometry of the magnetic coil itself:

- N = number of coils.
- B = magnetic field modulus.
- R = semi-width of a single coil (radius).
- L = length of a single coil.

Since the coupled mechanical-electrical system have different dynamics with different velocities, an A-stable integration scheme needs to be selected in order to properly solve the ODE problem, such as Matlab function `ode15s` (stiff solver).

However, the velocity of the main crank $\omega = \dot{\alpha}$ is taken as a constant throughout the entire flap motion, resulting in no transitory stages: the electric current derivative acting on the inductor can be neglected, generating a simple algebraic equation which indicates the necessary input voltage v_i to activate the flap extraction.

$$\begin{cases} T_m = T_m(\dot{\alpha}) \\ v_i = R_a \frac{T_m}{K_m} + K_m \dot{\alpha} \end{cases}$$

The required voltage input trend over $\alpha(t)$ is reported in Fig. 6.

4 Matlab simulation

Matlab was chosen as the primary software to carry out the simulation of the flap extraction system. All input data were taken from Boeing 747 technical documentation and from statistical experience [2]:

```
%% DATA:

% alpha vector:
alpha0 = deg2rad(275);                      % [rad] Initial alpha position
alpha1 = deg2rad(320);                      % [rad] Final alpha position
alpha = alpha0 : pi/1800 : alpha1;          % [rad * e-01] alpha vector interval
omega = 1;                                  % [rad/s] d_alpha (constant)

% kinematics data:
K = 17;                                     % [-] meter - model conversion factor
a = 3.5/K;                                  % [m] (A - O) length
b = 9/K;                                    % [m] (B - A) length
c = 5/K;                                    % [m] (G - C) length
l = 9/K;                                    % [m] Re(C - O) length
r = 6.5/K;                                  % [m] Im(C - O) length

d = 9/K;                                    % [m] (D - B) length
p = 2.25/K;                                 % [m] (G - D) length
s = 7/K;                                    % [m] Re(C - B) length
t = 8/K;                                    % [m] Im(C - B) length

g = 3/K;                                    % [m] (E - G) length
h = 17.75/K;                                % [m] (H - E) length
z = 4/K;                                    % [m] (H - F) length

% aero-mechanical dynamics data:
U = 240/3.6;                                % [m/s] landing speed
rho = 1.225;                                % [kg/m^3] air density (zero level)
c1 = 1;                                     % [m] main flap chord
c2 = 0.5;                                   % [m] second flap chord
S1 = 1.5*c1;                                % [m^2] main flap surface
S2 = 1.5*c2;                                % [m^2] second flap surface
M = 20;                                     % [kg] main flap mass
m = 10;                                     % [kg] secondary flap mass
J_M = M*c1*3.5/K;                           % [kg*m^2] main flap inertia
J_m = m*c2*2.5/K;                           % [kg*m^2] secondary flap inertia
G = 9.81;                                   % [m/s^2] gravitional acceleration

Cl_delta = 1.4;                             % [1/rad] main flap lift coefficient
Cl_theta = 1.3;                             % [1/rad] second flap lift coefficient
Cm_delta = -0.3;                            % [-] main flap moment coefficient
Cm_theta = -0.3;                            % [-] second flap moment coefficient
F1_delta = 0.5*rho*U^2*S1*Cl_delta;         % [N/rad] main flap lift (derivated over delta)
F2_theta = 0.5*rho*U^2*S2*Cl_theta;         % [N/rad] second flap lift (derivated over theta)
MAC1 = 0.5*rho*U^2*S1*c1*Cm_delta;          % [Nm] main flap aerodynamic moment
MAC2 = 0.5*rho*U^2*S2*c2*Cm_theta;          % [Nm] second flap aerodynamic moment

% electrical dynamics data:
R_a = 2;                                    % [Ohm] electric circuit resistance
K_m = 20;                                   % [Nm/A] DC motor constant
```

4.1 Non-linear equations

Each non-linear algebraic system of equations was solved by implementing a Newton-Raphson method based function called `newtonSys.m` which provides the solution of the follwing problem:

$$\begin{cases} \mathbf{x} = \mathbf{x_0} - \mathbf{J_0}^{-1}\mathbf{f}(\mathbf{x_0}) \\ \texttt{[x, it] = newtonSys(x0, nmax, toll, f, J)} \end{cases}$$

It's important to remark that some conditions must be verified for the correct implementation of this numerical method:

1. The initial guess $\mathbf{x_0}$ must be sufficiently close to the real zeros of the function.
2. The tolerance parameter of the method must be chosen properly.
3. The determinant of $\mathbf{J}$ must be different from zero:
   $det(\mathbf{J}) \neq 0$

The first kinematic chain's position results are here reported as an example in Fig. 4: the outcomes of `newtonSys.m` are compared with the solutions obtained by using Matlab function `fsolve`, showing a negligible relative error between the two solutions.

$$\mathbf{x} = \begin{bmatrix} \beta(\alpha) \\ \delta(\alpha) \end{bmatrix}; \quad \mathbf{x_0} = \begin{bmatrix} \beta_0 \\ \delta_0 \end{bmatrix}; \quad \mathbf{f} = \begin{bmatrix} Real(\beta,\delta) \\ Imm(\beta,\delta) \end{bmatrix}; \quad \mathbf{J} = \frac{\partial f_i}{\partial x_j} = \begin{bmatrix} -bsin(\beta) & -ccos(\delta) \\ bcos(\beta) & csin(\delta) \end{bmatrix};$$

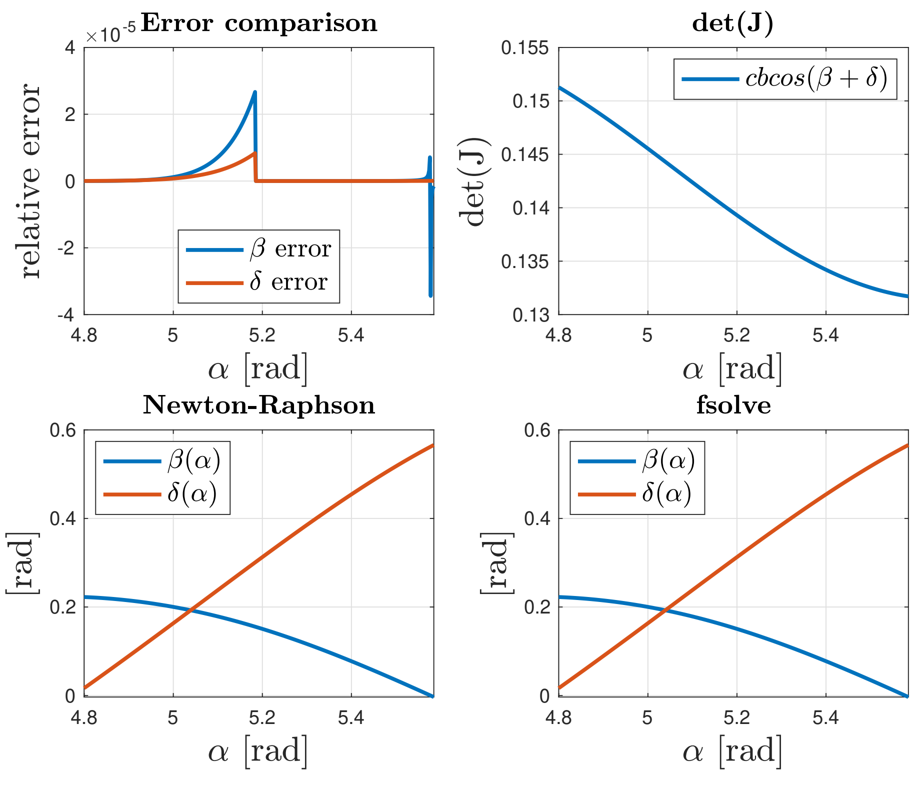


Figure 4: Comparison between newtonSys.m and fsolve

4.2 Final results

Final kinematic and dynamic results of the Matlab simulation are presented in figures 5 6 below.

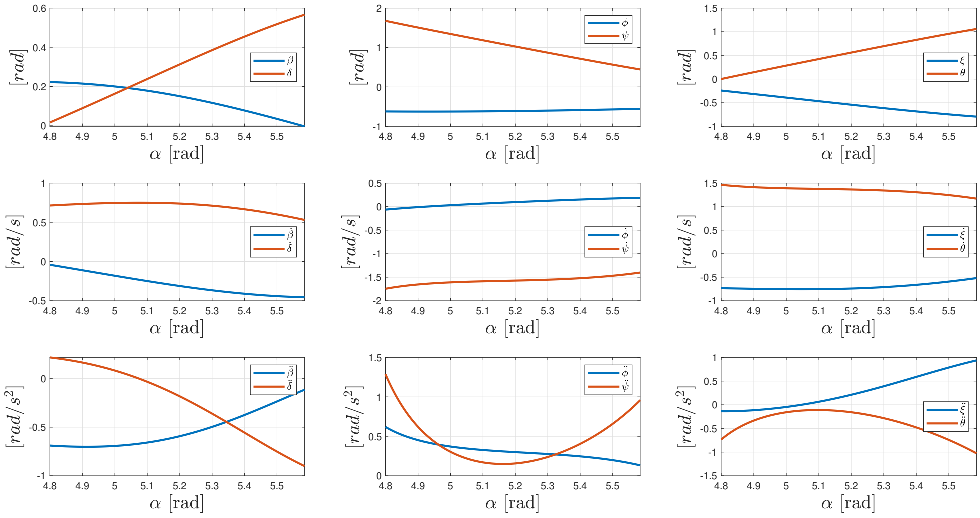


Figure 5: Matlab kinematic results

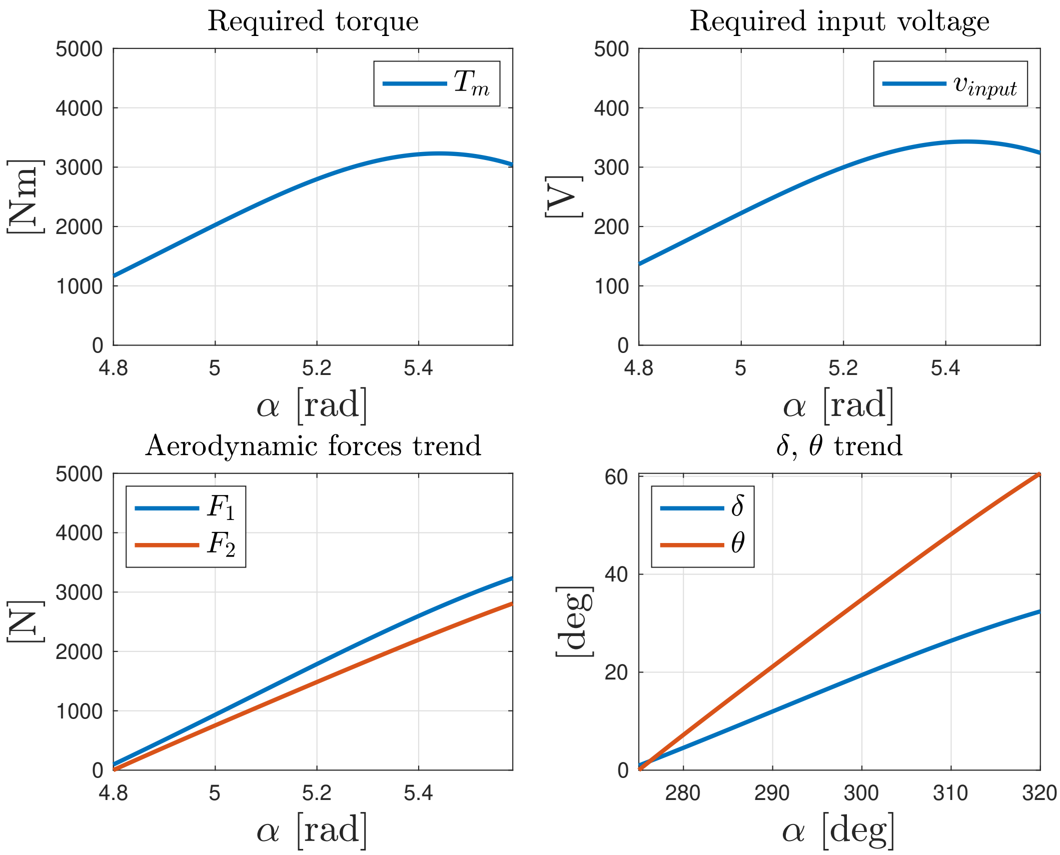


Figure 6: Matlab dynamic results

5 Multibody solver simulation: MBDyn

MBDyn is an open-source multibody solver software particularly useful for simulating and studying the dynamic behaviour of complex system mechanisms, such as the one being modelled in this project. It employs numerical integration methods to solve the equations of motion and accurately simulate the dynamic response of the mechanism over time.

5.1 Modelling process

The physical model sketched in Fig. 1 was implemented in MBDyn representing both flaps and bars as rigid bodies, with the latest ones characterized by negligible mass and inertia (placed over midpoint) in order to be identified as cranks.

For this purpose, several reference systems were established over every strategic hinge shown in Fig. 1, identified as structural nodes. The connections between each body of the system have been created using joints (specifically, with total joints that allow to arbitrarily constrain specific components of the relative position and orientation of two nodes [1]). It is from these joints that the angular variations will be exported to obtain the results that will be presented later.

For the modeling of the two flaps, a body has been established at point G for the main flap and another body at point H for the second flap. Unlike the bars, these bodies do have calculated mass and inertia (see section 4). Additionally, the centre of gravity has been positioned based on the coordinates of these points.

Within the dynamic analysis of the mechanism, the aerodynamic forces F_1 and F_2 have been modeled in the problem. For this purpose, two structural nodes have been incorporated at the aerodynamic centers of each flap, imposing the respective aerodynamic forces on them (which should vary with the angle of attack of the flap and thus with the flap orientation).

To enhance the visualization of the results, structural nodes with bodies of negligible mass and inertia have also been established at points C, F, and L. The aim was to export the displacements of these points (except for point C, which remains static): they are purely for representation purposes and do not contribute any physical value to the analysis.

The angular variation of the single DOF has been imposed with constant angular velocity $\dot{\alpha} = 1$ rad/s, varying from 275° (4.7997 rad) to 320° (5.5851 rad) (see section 4). To achieve this, a ramp-type drive has been established with a slope of 1 rad/s: since this ramp is abruptly introduced from the initial time instance of the simulation, Fig. 7, the angular velocities and accelerations obtained from MBDyn present significant oscillations at the beginning and at the end of the time interval.

This behavior can be contained by applying a filter at the beginning of the simulation, in order to reduce the slope of the step input introduced by the ramp (see section 6.3).

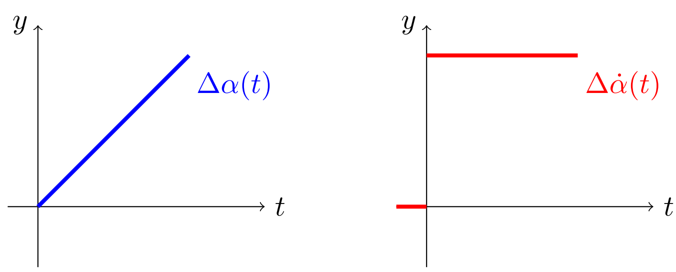
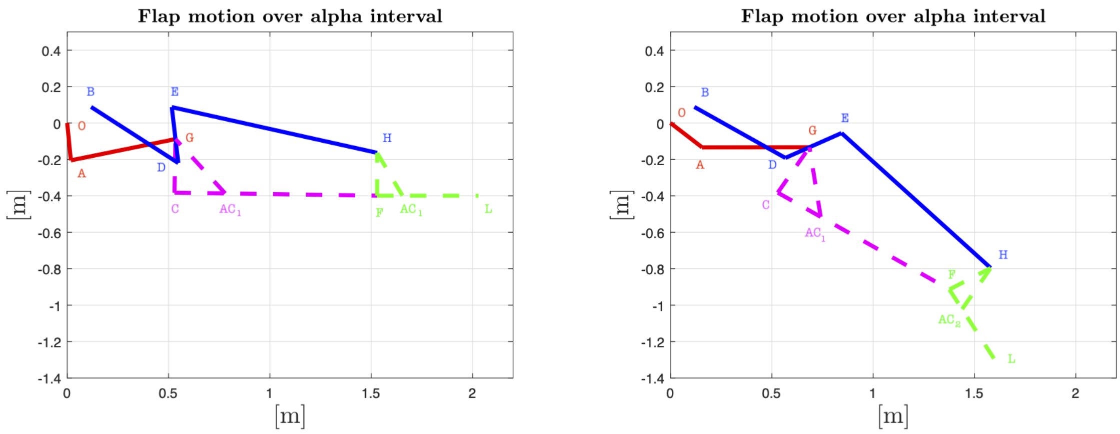


Figure 7: α, $\dot{\alpha}$ trend

5.2 Post-processing

In a representative manner, Fig. 8 displays both initial and final configuration of the mechanical system, with rest flap position at t = 0 s and extracted flap position at t = 0.79 s. For this representation, Matlab functions `VideoWriter.m` and `Drawnow.m` were employed: the red bars represent the OAG mechanism, while the blue ones represent the BDEH mechanism, which is also connected to point G through bar DE.

Lastly, the flaps' chords are represented by dashed lines: it's important to remark that both the main and secondary chords always remain perpendicular to vectors $\overline{GC}$ and $\overline{HF}$, respectively. The connections between point G and the aerodynamic center of the first flap, as well as between point H and the aerodynamic center of the second flap, are also illustrated to better understand the moving geometry of the system.



(a) Diagram of the model at the beginning of the simulation: t = 0 s, α = 275 degrees.

(b) Diagram of the model at the end of the simulation: t = 0.79 s, α = 320 degrees.

Figure 8: Diagram of the model at different time instances

6 Comparison of the results

6.1 Kinematic results

A comparison between Matlab and MBDyn kinematic results is reported in Figs. 9, 10, 11: all angles initial conditions were extracted from the Matlab simulation, along with the overall data from section 4.

Important : Angular accelerations' data were obtained by implementing a finite difference scheme function `finite_deriv.m` over the angular velocities provided by MBDyn, since according to the MBDyn manual [1], the usage of total joints in prevents the users from extracting accelerations using *NetCDF Output Format*. However, these values are only reported in order to highlight the comparison between Matlab and MBDyn kinematic results: in order to obtain the required torque, the reaction moment generated over joint O were exported directly.

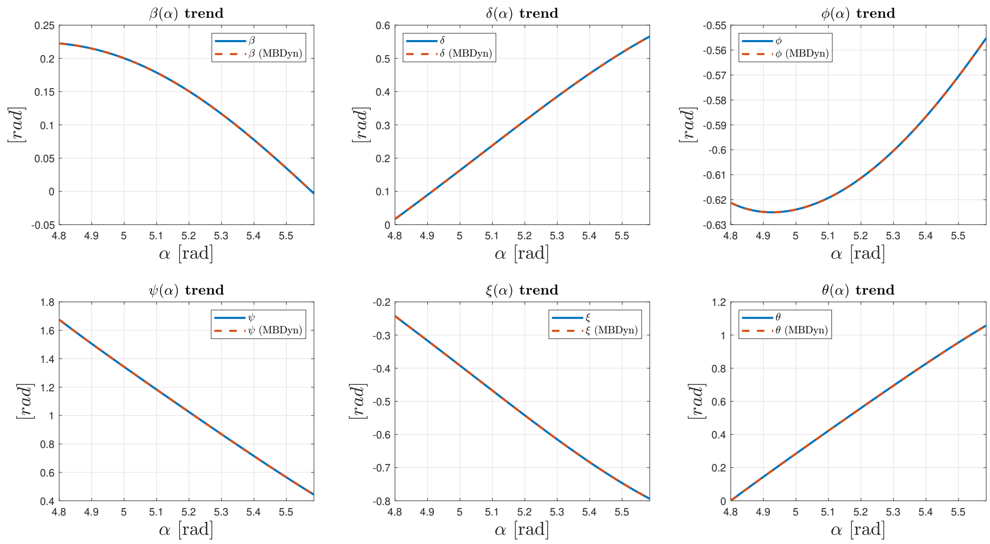


Figure 9: Comparison of angular variations trend over alpha interval

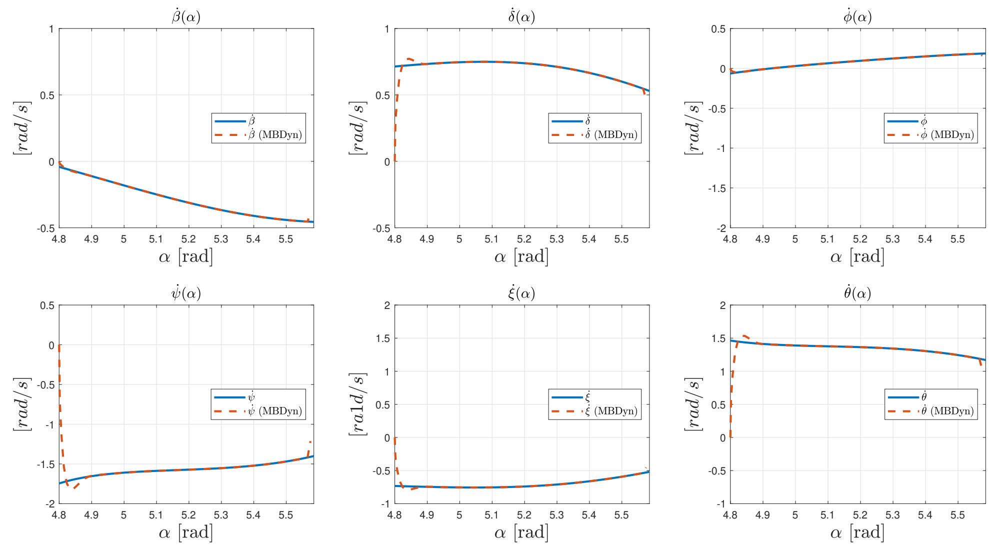


Figure 10: Comparison of angular velocities trend over alpha interval

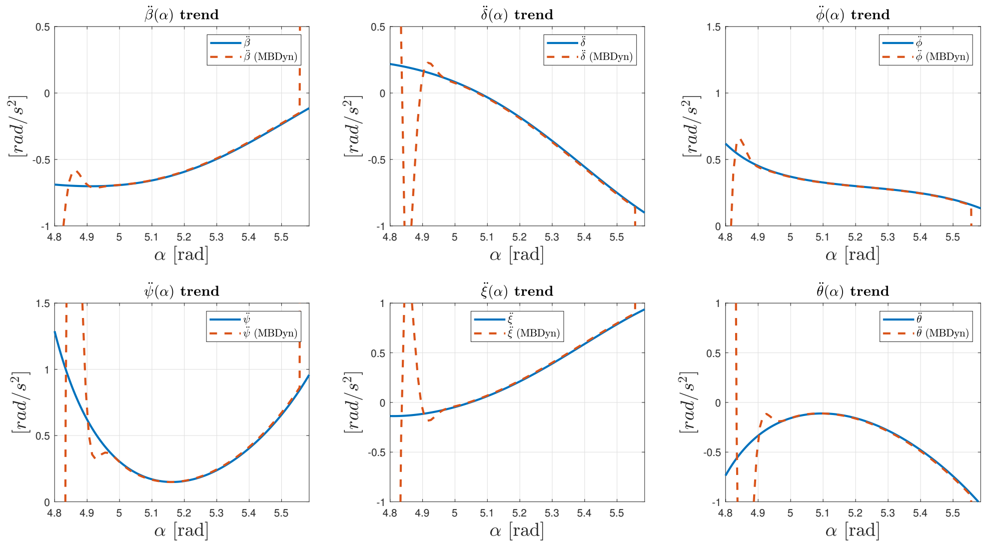
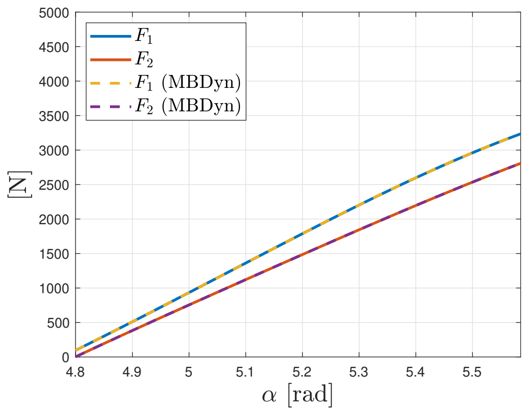


Figure 11: Comparison of angular accelerations as a function of α

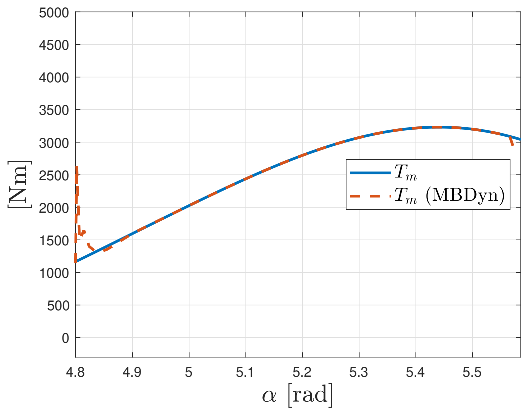
6.2 Dynamic results

A comparison between Matlab and MBDyn dynamic results is reported in Fig. 12.

The torque trend presented in Fig. 12b was obtained by imposing a variation of α from 275° to 320° and thus calculating the resulting reaction moment at point O under this condition.



(a) Aerodynamic forces trend over α



(b) Torque trend over α

Figure 12: MBDyn dynamic results

The trend clearly shows a proportionate correlation between the deflection angles and the required torque input, as the latest has to compensate the increment of the aerodynamic forces and their direct dependency from $\delta(\alpha)$, $\theta(\alpha)$

6.3 Filter implementation

As already anticipated in section 5.1, significant oscillations can be seen at the beginning and at the end of the simulation as a consequence of the ramp-drive input for α vector, which inevitably turns into a step function for the main driving link angular velocity $\dot{\alpha}(t)$.

This phenomenon can be diminished by applying a discrete filter at the beginning of the simulation in order to gradually modify the slope of the ramp input, avoiding abrupt variations in angular velocity and angular acceleration. In other words, this filter aims to achieve a smoother input by realizing a more realistic system, due to the fact that, in a real electric motor, a desired rotational speed is not achieved immediately, yet following a transition period.

Fig. 13 shows the new $\dot{\alpha}(t)$ trend after a 2nd order Butterworth filter was applied in MBDyn, with cut frequency 10 Hz and time step dt $=$ 0.01:

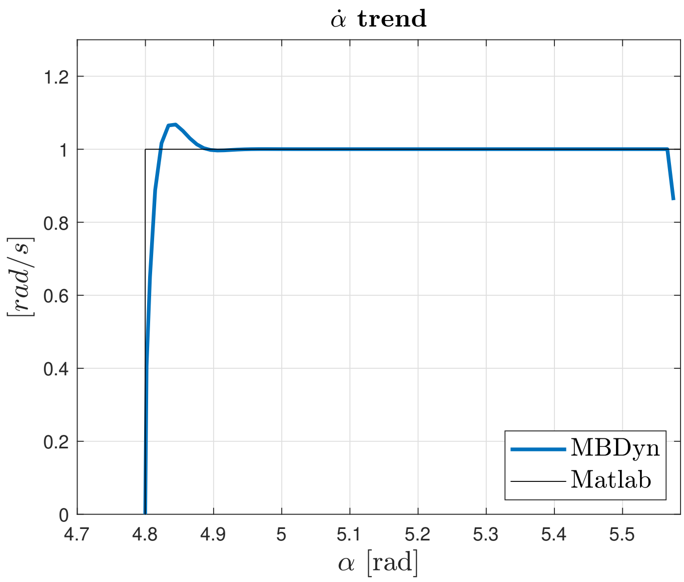
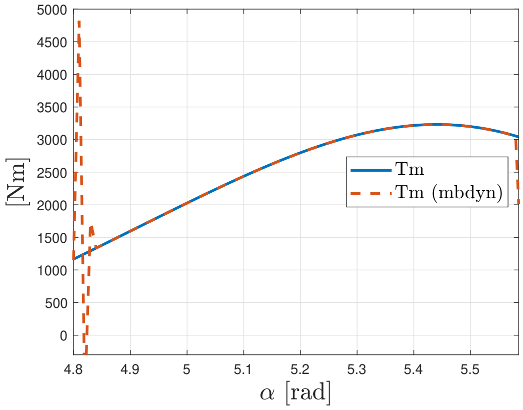


Figure 13: Comparison of angular accelerations as a function of α

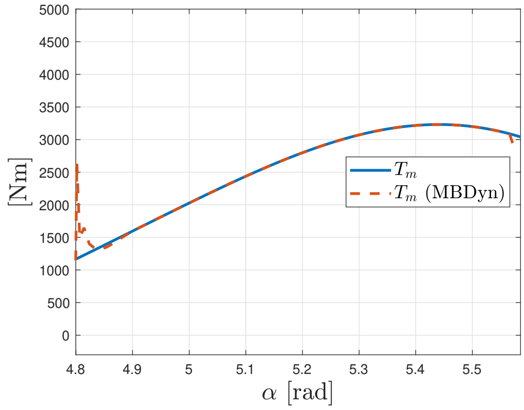
The coefficient of the filter were computed using Matlab function `butter.m`:

```
[B, A] = butter(2, 10./(1/(2*dt)))
a_1 = -A(2);     1st regression coefficient
a_2 = -A(3);     2nd regression coefficient
b_0 = B(1);      direct transmission coefficient
b_1 = B(2);      1st input coefficient
b_2 = B(3);      2nd input coefficient
```

Even with a filter application, the angular oscillations at the beginning and at the bottom of the time interval cannot be completely erased, as Fig. 13 shows a comparison between the torque trend before and after the filter implementation; the same obviously applies to the angular velocities and accelerations since torque input directly depends on kinematics formulation.



(a) No filter applied.



(b) Filter applied

Figure 14: 2nd order Butterworth filter comparison over torque trend.

In order to realize a more realistic transition without the application of a filter, the motion can be imposed as a function of time. It should be noted that the equation of motion has to meet certain requirements. On the one hand, it must have zero derivative at the beginning of the simulation and on the other hand, it must quickly tend to the desired velocity of 1 rad/s. An example of this can be eq. (1) where t_0 is the initial time, and k is a parameter that can be modified and will determine the velocity at which the desired speed is reached.

$$\Delta\alpha(t) = (t - t_0) + \frac{(\exp(-k * (t - t_0)) - 1)}{k} \tag{1}$$

References

[1] *Input manual mbdyn*, `https://www.mbdyn.org/userfiles/documents/mbdyn-input-1.7.2.pdf`.

[2] Alex Zanotti Giuseppe Quaranta, Gerardus Janszen, *Istituzioni di ingegneria aerospaziale*, McGraw-Hill Education.