Convertible aircraft dynamic modelling and flatness analysis

Tudor-Bogdan Airimitoaie, Gemma Prieto Aguilar, Loïc Lavigne, Christophe Farges, and Franck Cazaurang

Univ. Bordeaux, Bordeaux INP, CNRS, IMS, UMR 5218, 33405 Talence, France

February 21, 2018

9^{th} Vienna International Conference on Mathematical Modelling, February 21-23, 2018, Vienna, Austria

What is a convertible aircraft ?

What is a convertible aircraft ?

A combination between:

- a multicopter

What is a convertible aircraft ?

A combination between:

- a multicopter

- a fixed-wing aircraft

What is a convertible aircraft ?

A combination between:

- a multicopter

- a fixed-wing aircraft

➜ hover at fixed point

➜ take-off and landing vertically

  ➥ reduced environmental footprint (no need for a runway)

What is a convertible aircraft ?

A combination between:

- a multicopter

- a fixed-wing aircraft

➜ hover at fixed point

➜ take-off and landing vertically

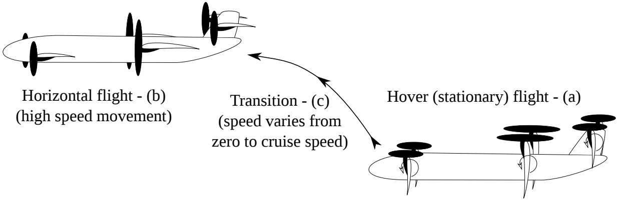
  ➥ reduced environmental footprint (no need for a runway)

➜ improve aerodynamic efficiency using wings

  ➥ reduced energy consumption

  ➥ increased flight length

Proposed convertible aircraft concept

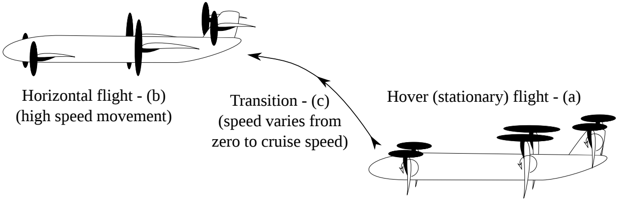


- 6 propeller-engines: 4 for VTOL and 2 for forward flight
- 3 pairs of wings: canard, main and elevator
- tilt-wing design: wings and propellers are fixed together

Project goal: design autonomous pilot for safe transition

1. modelling of the dynamic behaviour
   - initial work on the modelling (gimbal lock problem)
2. robust and fault tolerant control
   - flatness analysis of the model

Proposed convertible aircraft concept



- 6 propeller-engines: 4 for VTOL and 2 for forward flight
- 3 pairs of wings: canard, main and elevator
- tilt-wing design: wings and propellers are fixed together

Project goal: design autonomous pilot for safe transition

1. modelling of the dynamic behaviour
   - initial work on the modelling (gimbal lock problem)
2. robust and fault tolerant control
   - flatness analysis of the model

Part I – Dynamic model

Dynamic model – reference frames

- inertial frame ($\mathcal{I}$)
- vehicle-carried normal Earth frame ($\mathcal{O}$)
- body frame ($\mathcal{B}$)
- aerodynamic ($\mathcal{A}$) and kinematic ($\mathcal{K}$) frames are supposed identical (zero wind speed)

Euler angles are used to represent vector rotations between frames:

- from $\mathcal{B}$ to $\mathcal{O}$:

$$R_b^o = R_z(\psi(t))R_y(\theta(t))R_x(\phi(t))$$
$$= \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

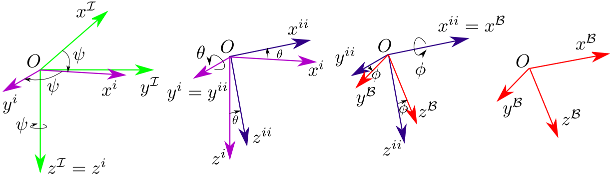
- from $\mathcal{A}$ to $\mathcal{O}$: $R_a^o = R_z(\chi(t))R_y(\gamma(t))R_x(\mu(t))$
- from $\mathcal{A}$ to $\mathcal{B}$: $R_a^b = R_y(-\alpha(t))R_z(\beta(t))$

Dynamic model – reference frames

- inertial frame ($\mathcal{I}$)
- vehicle-carried normal Earth frame ($\mathcal{O}$)
- body frame ($\mathcal{B}$)
- aerodynamic ($\mathcal{A}$) and kinematic ($\mathcal{K}$) frames are supposed identical (zero wind speed)

Euler angles are used to represent vector rotations between frames:

- from $\mathcal{B}$ to $\mathcal{O}$:



$$R_b^o = R_z(\psi(t))R_y(\theta(t))R_x(\phi(t))$$
$$= \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{c}\phi \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\psi\mathrm{s}\theta\mathrm{s}\phi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\psi\mathrm{s}\theta\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix}$$

- from $\mathcal{A}$ to $\mathcal{O}$: $R_a^o = R_z(\chi(t))R_y(\gamma(t))R_x(\mu(t))$
- from $\mathcal{A}$ to $\mathcal{B}$: $R_a^b = R_y(-\alpha(t))R_z(\beta(t))$

Dynamic model – Newton's second law

Newton's second law in a non-inertial frame $\mathcal{A}$:

$$\dot{\xi} = R_a^o V_a^a, \tag{1}$$

$$m\frac{\mathrm{d}V_a^a}{\mathrm{d}t} + \Omega_{ao}^a \wedge mV_a^a = F^a + (R_a^o)^T G^o. \tag{2}$$

- $\xi(t) = [x(t)\ y(t)\ z(t)]^T$: position of the aircraft in frame $\mathcal{I}$
- $V_a^a(t) = [v_a(t)\ 0\ 0]^T$: speed vector in frame $\mathcal{A}$
- $\Omega_{ao}^a(t)$: rotational speed of frame $\mathcal{A}$ with respect to frame $\mathcal{O}$
- m: mass of the aircraft

$$F^a(t) = \begin{bmatrix} X^a(t) \\ Y^a(t) \\ Z^a(t) \end{bmatrix} = F_a^a(t) + \left(R_a^b\right)^T F_p^b(t) \tag{3}$$

- $F_a^a(t)$: aerodynamic forces
- $F_p^b(t)$: propulsion forces

Dynamic model – Newton's second law

Newton's second law in a non-inertial frame $\mathcal{A}$:

$$\dot{\xi} = R_a^o V_a^a, \tag{1}$$

$$m\frac{\mathrm{d}V_a^a}{\mathrm{d}t} + \Omega_{ao}^a \wedge m V_a^a = F^a + (R_a^o)^T G^o. \tag{2}$$

- $\xi(t) = [x(t)\ y(t)\ z(t)]^T$: position of the aircraft in frame $\mathcal{I}$
- $V_a^a(t) = [v_a(t)\ 0\ 0]^T$: speed vector in frame $\mathcal{A}$
- $\Omega_{ao}^a(t)$: rotational speed of frame $\mathcal{A}$ with respect to frame $\mathcal{O}$
- m: mass of the aircraft

$$F^a(t) = \begin{bmatrix} X^a(t) \\ Y^a(t) \\ Z^a(t) \end{bmatrix} = F_a^a(t) + \left(R_a^b\right)^T F_p^b(t) \tag{3}$$

- $F_a^a(t)$: aerodynamic forces
- $F_p^b(t)$: propulsion forces

Dynamic model – Euler's second law

Euler's rotation equations:

$$\frac{\mathrm{d}\left(I\Omega_{bo}^{b}(t)\right)}{\mathrm{d}t} + \Omega_{bo}^{b}(t) \wedge I\Omega_{bo}^{b}(t) = \tau_b(t). \tag{4}$$

- I: inertia matrix
- $\Omega_{bo}^{b}(t) = [p(t)\ q(t)\ r(t)]^T$: rotational speed of frame $\mathcal{B}$ wrt frame $\mathcal{O}$

$$\tau^b(t) = \begin{bmatrix} L^b(t) \\ M^b(t) \\ N^b(t) \end{bmatrix} = \tau_a^b(t) + M_p^b(t) + \tau_d^b(t) \tag{5}$$

- $\tau_a^b(t)$: aerodynamic moments
- $M_p^b(t)$: moments from propulsion forces
- $\tau_d^b(t)$: drag couples

Dynamic model – Euler's second law

Euler's rotation equations:

$$\frac{\mathrm{d}\left(I\Omega_{bo}^{b}(t)\right)}{\mathrm{d}t} + \Omega_{bo}^{b}(t) \wedge I\Omega_{bo}^{b}(t) = \tau_b(t). \qquad (4)$$

- I: inertia matrix
- $\Omega_{bo}^{b}(t) = [p(t)\ q(t)\ r(t)]^T$: rotational speed of frame $\mathcal{B}$ wrt frame $\mathcal{O}$

$$\tau^b(t) = \begin{bmatrix} L^b(t) \\ M^b(t) \\ N^b(t) \end{bmatrix} = \tau_a^b(t) + M_p^b(t) + \tau_d^b(t) \qquad (5)$$

- $\tau_a^b(t)$: aerodynamic moments
- $M_p^b(t)$: moments from propulsion forces
- $\tau_d^b(t)$: drag couples

Dynamic model – Euler angles derivatives

Time evolution of Euler angles between frames $\mathcal{A}$ and $\mathcal{O}$ is obtained using:

$$\dot{R}_a^o = R_a^o \left[\Omega_{ao}^a\right]_\times \tag{6}$$

Skew-symetric operator $[\cdot]_\times$, of a vector $W = [w_1 \ \ w_2 \ \ w_3]^T \in \mathbb{R}^3$:

$$[W]_\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \tag{7}$$

Using that

$$\Omega_{ao} = \Omega_{ab} + \Omega_{bo} \implies \Omega_{ao}^a = \Omega_{ab}^a + R_b^a \Omega_{bo}^b, \tag{8}$$

while

$$\dot{R}_a^b = R_a^b \left[\Omega_{ab}^a\right]_\times \implies \left[\Omega_{ab}^a\right]_\times = \left(R_a^b\right)^T \dot{R}_a^b, \tag{9}$$

which implies

$$\Omega_{ab}^a = \begin{bmatrix} -\dot{\alpha} \mathrm{s}\beta & -\dot{\alpha} \mathrm{c}\beta & \dot{\beta} \end{bmatrix}^T. \tag{10}$$

Dynamic model – Euler angles derivatives

Time evolution of Euler angles between frames $\mathcal{A}$ and $\mathcal{O}$ is obtained using:

$$\dot{R}_a^o = R_a^o \left[\Omega_{ao}^a\right]_\times \tag{6}$$

Skew-symetric operator $[\cdot]_\times$, of a vector $W = [w_1 \ w_2 \ w_3]^T \in \mathbb{R}^3$:

$$[W]_\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \tag{7}$$

Using that

$$\Omega_{ao} = \Omega_{ab} + \Omega_{bo} \implies \Omega_{ao}^a = \Omega_{ab}^a + R_b^a \Omega_{bo}^b, \tag{8}$$

while

$$\dot{R}_a^b = R_a^b \left[\Omega_{ab}^a\right]_\times \implies \left[\Omega_{ab}^a\right]_\times = (R_a^b)^T \dot{R}_a^b, \tag{9}$$

which implies

$$\Omega_{ab}^a = \begin{bmatrix} -\dot{\alpha}\mathrm{s}\beta & -\dot{\alpha}\mathrm{c}\beta & \dot{\beta} \end{bmatrix}^T. \tag{10}$$

Dynamic model – system of equations

Newton's 2$^{\text{nd}}$ law:

$$\begin{cases} \dot{x}(t) = \mathrm{c}\chi\mathrm{c}\gamma v_a, \\ \dot{y}(t) = \mathrm{s}\chi\mathrm{c}\gamma v_a, \\ \dot{z}(t) = -\mathrm{s}\gamma v_a, \\ \dot{v}_a(t) = \frac{X^a}{m} - \mathrm{s}\gamma g, \\ \dot{\beta}(t) = \mathrm{s}\alpha p - \mathrm{c}\alpha r + \frac{\mathrm{c}\gamma\mathrm{s}\mu m g + Y^a}{m v_a}, \\ \dot{\alpha}(t) = q - (\mathrm{c}\alpha p + \mathrm{s}\alpha r)\mathrm{t}\beta + \frac{\mathrm{c}\gamma\mathrm{c}\mu}{\mathrm{c}\beta}\frac{g}{v_a} + \frac{Z^a}{\mathrm{c}\beta m v_a} \end{cases}$$

Euler angles derivatives:

$$\begin{cases} \dot{\chi}(t) = \frac{-Z^a\mathrm{s}\mu + Y^a\mathrm{c}\mu}{v_a m\mathrm{c}\gamma}, \\ \dot{\gamma}(t) = \frac{-\mathrm{c}\gamma g m - Y^a\mathrm{s}\mu - Z^a\mathrm{c}\mu}{v_a m}, \\ \dot{\mu}(t) = \frac{-\mathrm{c}\mu\mathrm{c}\gamma\mathrm{s}\beta g}{v_a\mathrm{c}\beta} + \frac{p\mathrm{c}\alpha + r\mathrm{s}\alpha}{\mathrm{c}\beta} - \frac{Z^a\mathrm{s}\beta}{v_a m\mathrm{c}\beta} + \frac{\mathrm{s}\gamma(Y^a\mathrm{c}\mu - Z^a\mathrm{s}\mu)}{v_a m\mathrm{c}\gamma}, \end{cases}$$

Euler's 2$^{\text{nd}}$ law:

$$\begin{cases} \dot{p}(t) = \frac{(I_{xz}(I_{xx} - I_{yy} + I_{zz})p - (I_{xz}^2 - I_{zz}(I_{yy} - I_{zz}))r)q}{I_{xx}I_{zz} - I_{xz}^2} + \frac{I_{xz}N^b + I_{zz}L^b}{I_{xx}I_{zz} - I_{xz}^2}, \\ \dot{q}(t) = \frac{-I_{xz}p^2 - r(I_{xx} - I_{zz})p + I_{xz}r^2 + M^b}{I_{yy}}, \\ \dot{r}(t) = \frac{((I_{xz}^2 + I_{xx}(I_{xx} - I_{yy}))p - I_{xz}(I_{xx} - I_{yy} + I_{zz})r)q}{I_{xx}I_{zz} - I_{xz}^2} + \frac{I_{xx}N^b + I_{xz}L^b}{I_{xx}I_{zz} - I_{xz}^2}. \end{cases}$$

How to handle the gimbal lock?

Euler angles: gimbal lock at $\gamma(t) = 90°$

- ➥ problem during vertical take-off and landing

Solution: use a second set of Euler angles $\mu_v(t)$, $\gamma_v(t)$, and $\chi_v(t)$ known as vertical Euler angles (Castillo et al. (2005))

$$R^o_{a_v} = R_y(\pi/2)R_z(\chi_v(t))R_y(\gamma_v(t))R_x(\mu_v(t)) \tag{11}$$

Part II – Flatness analysis of the model

Flatness analysis - recall

Recall on flat systems (J. Lévine, 2009):

$$\dot{x} = f(x, u),\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,\ m \leq n$$

is differentially flat iff $\exists\ y \in \mathbb{R}^m$, $y = f_y(x, u, \dot{u}, \ldots, u^{(p)})$, s.t.

$$x = f_x(y, \dot{y}, \ddot{y}, \ldots, y^{(q)}),$$
$$u = f_u(y, \dot{y}, \ddot{y}, \ldots, y^{(q+1)}).$$

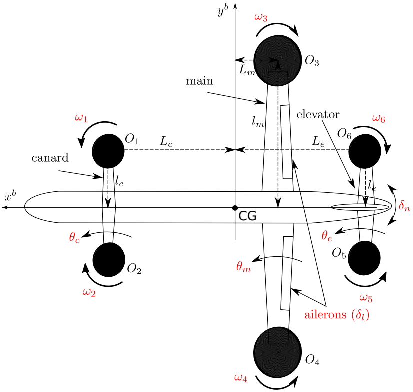
➜ y are called the flat outputs.

Applications of flatness: feedforward control, fault detection and isolation (FDI), ...

Flatness analysis – control inputs

The convertible aircraft has a total of 11 control inputs:

- 3 wing rotation angles (θ_c, θ_m, and θ_e),
- rotation speeds of 6 propeller-engines (ω_1, ω_2, ω_3, ω_4, ω_5, and ω_6),
- main wing aileron deflection (δ_l),
- rudder tilt angle (δ_n)



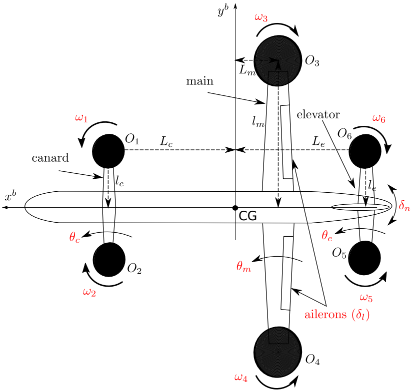
There are only 5 independent control inputs:

- Take-off, landing, and hover
  - ➥ ω_1, ω_2, ω_5, ω_6, and $\theta_c = \theta_e$
- Fast forward flight
  - ➥ ω_3, ω_4, δ_l, δ_n, and θ_c

Flatness analysis – control inputs

The convertible aircraft has a total of 11 control inputs:

- 3 wing rotation angles (θ_c, θ_m, and θ_e),
- rotation speeds of 6 propeller-engines (ω_1, ω_2, ω_3, ω_4, ω_5, and ω_6),
- main wing aileron deflection (δ_l),
- rudder tilt angle (δ_n)

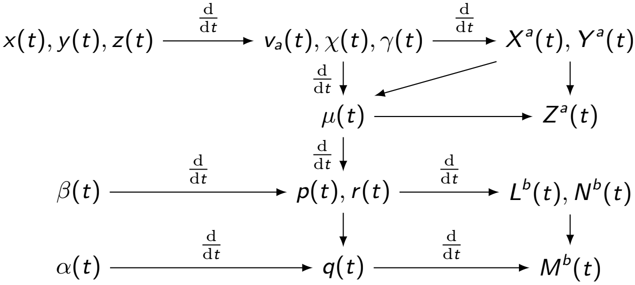


There are only 5 independent control inputs:

- Take-off, landing, and hover
  - ➥ ω_1, ω_2, ω_5, ω_6, and $\theta_c = \theta_e$
- Fast forward flight
  - ➥ ω_3, ω_4, δ_l, δ_n, and θ_c

Flatness analysis – flat outputs

Proposed set of flat outputs: $x(t)$, $y(t)$, $z(t)$, $\alpha(t)$, and $\beta(t)$.



What happens if $v_a(t)$ is small?

Problem: model written using vertical Euler angles becomes singular.

Proposed solution: write Newton's second law in frame $\mathcal{B}$

$$\dot{\xi} = V_o^o, \tag{12}$$

$$m\frac{\mathrm{d}V_o^o}{\mathrm{d}t} = R_b^o F_p^b + G^o, \tag{13}$$

$$\dot{R}_b^o = R_b^o \left[\Omega_{bo}^b\right]_\times, \tag{14}$$

$$\frac{\mathrm{d}\left(I\Omega_{bo}^b(t)\right)}{\mathrm{d}t} = -\Omega_{bo}^b(t) \wedge I\Omega_{bo}^b(t) + \tau_b(t). \tag{15}$$

where $F_p^b(t) = R_y(\theta_e(t))\left[F_p^e(t)\ 0\ 0\right]^T$.

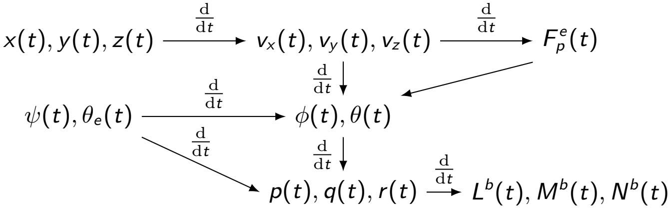
$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{z}(t)\end{bmatrix} = \begin{bmatrix}v_x\\ v_y\\ v_z\end{bmatrix} \tag{16a}$$

$$\begin{bmatrix}\dot{v}_x(t)\\ \dot{v}_y(t)\\ \dot{v}_z(t)\end{bmatrix} = R_z(\psi)R_y(\theta)R_x(\phi)R_y(\theta_e)\begin{bmatrix}F_p^e\\ 0\\ 0\end{bmatrix} \tag{16b}$$

$$\begin{bmatrix}\dot{\phi}(t)\\ \dot{\theta}(t)\\ \dot{\psi}(t)\end{bmatrix} = \begin{bmatrix}1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta\\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi\\ 0 & \frac{\mathrm{s}\phi}{\mathrm{c}\theta} & \frac{\mathrm{c}\phi}{\mathrm{c}\theta}\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{16c}$$

$$I\begin{bmatrix}\dot{p}(t)\\ \dot{q}(t)\\ \dot{r}(t)\end{bmatrix} = -\begin{bmatrix}p\\ q\\ r\end{bmatrix} \wedge I\begin{bmatrix}p\\ q\\ r\end{bmatrix} + \begin{bmatrix}L^b\\ M^b\\ N^b\end{bmatrix} \tag{16d}$$

Flat outputs: $x(t, y(t), z(t), \psi(t)$, and $\theta_e(t) = \theta_c(t)$.



Conclusions and perspectives

- In this paper: convertible aircraft modelling with focus on the flatness property
- Other models can be obtained (e.g., using quaternion rotation)
  - need to analyse flatness
  - might be useful for simulation
- Further work is necessary to determine the sizes the convertible (3D modelling and CFD)
- Fault tolerant guidance of the convertible aircraft to be adressed next

Thank you for your attention!

Project web page: http://tudor-bogdan.airimitoaie.name/mica/

Conclusions and perspectives

- In this paper: convertible aircraft modelling with focus on the flatness property
- Other models can be obtained (e.g., using quaternion rotation)
  - need to analyse flatness
  - might be useful for simulation
- Further work is necessary to determine the sizes the convertible (3D modelling and CFD)
- Fault tolerant guidance of the convertible aircraft to be adressed next

Thank you for your attention!

Project web page: http://tudor-bogdan.airimitoaie.name/mica/