Differential flatness

Consider a dynamical system with

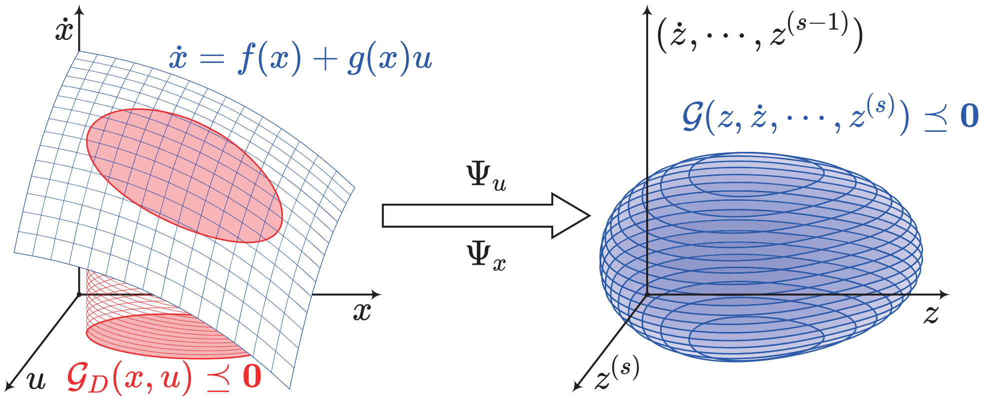
$$\dot{x} = f(x) + g(x)u,$$
$$f\colon \mathbb{R}^n \mapsto \mathbb{R}^n,\ g\colon \mathbb{R}^n \mapsto \mathbb{R}^n,\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m,\ \mathrm{rank}(g) = m.$$

The system is *differentially flat*, if there exists $z \in \mathbb{R}^m$ which can be determined by the state $x \in \mathbb{R}^n$ and finite derivatives of $u \in \mathbb{R}^m$.

$z \in \mathbb{R}^m$ is called the *flat output*, whose finite derivatives can uniquely determine all state and input:

$$\boxed{\begin{aligned} x &= \Psi_x(z, \dot{z}, \cdots, z^{(s-1)}), \\ u &= \Psi_u(z, \dot{z}, \cdots, z^{(s)}). \end{aligned}}$$

Differential flatness eliminates differential constraints.



by Zhepei Wang

Multicopter dynamics and differential flatness

Multicopter State

$$x = \{r, v, R, \omega\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{R}^3$$

position, velocity, attitude, body rate

Multicopter Control Input (After Input Mapping)

$$u = \{f, \tau\} \in \mathbb{R}_{\geq 0} \times \mathbb{R}^3$$

collective thrust, torque

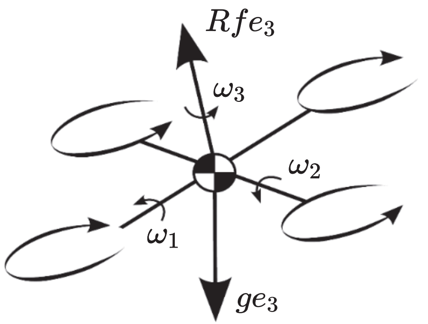
Multicopter Nonlinear Dynamics

$$\begin{cases} \dot{r} = v, \\ m\dot{v} = -mge_3 - RDR^{\mathrm{T}}\sigma(\|v\|)v + Rfe_3, \\ \dot{R} = R\hat{\omega}, \\ M\dot{\omega} = \tau - \omega \times M\omega - A(\omega) - B(R^{\mathrm{T}}v). \end{cases}$$

Multicopter Flat Output

$$z = \{r, \psi\} \in \mathbb{R}^3 \times \mathrm{SO}(2)$$

position, yaw heading



$m \in \mathbb{R}_{\geq 0}$	vehicle mass
$g \in \mathbb{R}_{\geq 0}$	gravitational acceleration
$e_3 = (0, 0, 1)^{\mathrm{T}}$	unit vector
$D = \mathrm{Diag}(d_h, d_h, d_v)$	drag force coefficients
σ: $\mathbb{R}_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$	user-defined nonlinear term
$M \in \mathbb{R}^{3\times 3}_{\geq 0}$	moment of inertia
A: $\mathbb{R}^3 \mapsto \mathbb{R}^3$	user-defined torque induced body rate
B: $\mathbb{R}^3 \mapsto \mathbb{R}^3$	user-defined torque induced velocity

Wang et al., Robust Trajectory Planning for Spatial-Temporal Multi-Drone Coordination in Large Scenes, arXiv 2109.08403

Multicopter dynamics and differential flatness

Flatness Transformation?

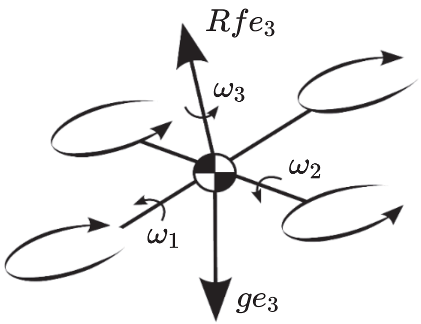
$$z = \{r, \psi\} \in \mathbb{R}^3 \times \mathrm{SO}(2)$$

$$x = \Psi_x(z, \dot{z}, \cdots, z^{(s-1)}),$$
$$u = \Psi_u(z, \dot{z}, \cdots, z^{(s)}).$$

?

$$x = \{r, v, R, \omega\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{R}^3$$

$$u = \{f, \tau\} \in \mathbb{R}_{\geq 0} \times \mathbb{R}^3$$



Obviously, we have

$$r = r \qquad v = \dot{r}$$

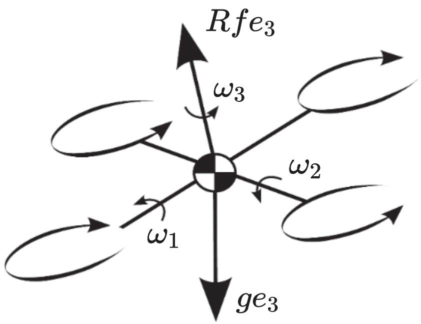
Dot product the Newton equation by X and Y axes of the body frame:

$$x_b = Re_1,\ y_b = Re_2$$
$$(Re_i)^{\mathrm{T}} m\dot{v} = (Re_i)^{\mathrm{T}}\left(-mge_3 - RDR^{\mathrm{T}}\sigma(\|v\|)v + Rfe_3\right),\ \forall i \in \{1, 2\}$$

These yield

$$(Re_i)^{\mathrm{T}}(\dot{v} + \frac{d_h}{m}\sigma(\|v\|)v + ge_3) = 0,\ \forall i \in \{1, 2\}.$$

Multicopter dynamics and differential flatness



What does it mean?

$$(Re_i)^{\mathrm{T}}(\dot{v} + \frac{d_h}{m}\sigma(\|v\|)v + ge_3) = 0,\ \forall i \in \{1, 2\}.$$

Geometrically, we have

$$x_b \perp (\dot{v} + \frac{d_h}{m}\sigma(\|v\|)v + ge_3)$$
$$y_b \perp (\dot{v} + \frac{d_h}{m}\sigma(\|v\|)v + ge_3)$$

Since the thrust force and body Z axis share the same direction,

$$z_b = \mathcal{N}(\dot{v} + \frac{d_h}{m}\sigma(\|v\|)v + ge_3),\ \mathcal{N}(x) := x/\|x\|_2$$

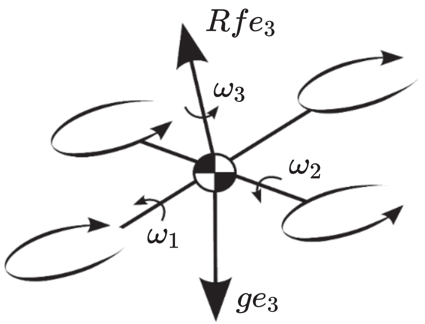
Dot product the Newton equation by Z axis of the body frame:

$$(Re_3)^{\mathrm{T}} m\dot{v} = (Re_3)^{\mathrm{T}}\left(-mge_3 - RDR^{\mathrm{T}}\sigma(\|v\|)v + Rfe_3\right)$$

We obtain

$$f = z_b^{\mathrm{T}}(m\dot{v} + d_v\sigma(\|v\|)v + mge_3)$$

Multicopter dynamics and differential flatness



Now that the yaw heading and body Z axis are both known, the yaw rotation quaternion is $q_\psi = (\cos(\psi/2), 0, 0, \sin(\psi/2))^{\mathrm{T}}$

The tilt rotation quaternion is $q_z = \frac{1}{\sqrt{2(1+z_b(3))}}(1+z_b(3), -z_b(2), z_b(1), 0)^{\mathrm{T}}$

The tilt rotation has no component for inertial Z axis , since it is decomposed using Hopf fibration. The attitude quaternion of vehicle is thus

$$q = q_z \otimes q_\psi$$

Expanding it yields

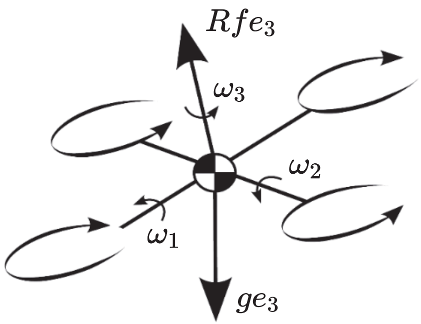
$$q = \frac{1}{\sqrt{2(1+z_b(3))}} \begin{pmatrix} (1+z_b(3))\cos(\psi/2) \\ -z_b(2)\cos(\psi/2) + z_b(1)\sin(\psi/2) \\ z_b(1)\cos(\psi/2) + z_b(2)\sin(\psi/2) \\ (1+z_b(3))\sin(\psi/2), \end{pmatrix}$$

Thus the attitude rotation matrix is uniquely determined, $R = \mathcal{R}_{quat}(q)$

Quaternion product, inverse, and rotation matrix formula are detailed by Vince.

Watterson et al., Control of Quadrotors Using the Hopf Fibration on SO(3), ISRR 2017.
Vince, Quaternions for Computer Graphics, Springer 2011.

Multicopter dynamics and differential flatness



Now the rotation is known.

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

This implies

$$\omega = (R^{\mathrm{T}}\dot{R})^{\vee}$$

Equivalently,

$$\omega = 2(q_z \otimes q_\psi)^{-1} \otimes \left(\dot{q}_z \otimes q_\psi + q_z \otimes \dot{q}_\psi\right)$$

Substituting the tilt rotation quaternion and yaw rotation quaternion gives

$$\omega = \begin{pmatrix} \dot{z}_b(1)\sin(\psi) - \dot{z}_b(2)\cos(\psi) - \dot{z}_b(3)(z_b(1)\sin(\psi) - z_b(2)\cos(\psi))/(1 + z_b(3)) \\ \dot{z}_b(1)\cos(\psi) + \dot{z}_b(2)\sin(\psi) - \dot{z}_b(3)(z_b(1)\cos(\psi) + z_b(2)\sin(\psi))/(1 + z_b(3)) \\ (z_b(2)\dot{z}_b(1) - z_b(1)\dot{z}_b(2))/(1 + z_b(3)) + \dot{\psi}, \end{pmatrix}$$

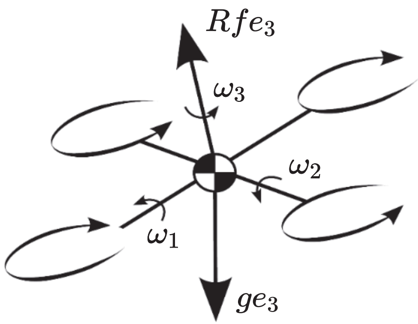
Differentiate body Z axis gives

$$\dot{z}_b = \frac{d_h}{m}\mathcal{DN}\left(\dot{v} + \frac{d_h}{m}\sigma(\|v\|)v + ge_3\right)^{\mathrm{T}}\left(\frac{m}{d_h}\ddot{v} + \sigma(\|v\|)\dot{v} + \dot{\sigma}(\|v\|)\frac{v^{\mathrm{T}}\dot{v}}{\|v\|}v\right),$$

$$\mathcal{DN}(x) := \frac{1}{\|x\|}\left(\mathbf{I} - \frac{xx^{\mathrm{T}}}{x^{\mathrm{T}}x}\right).$$

Now we have expressed the body rate using finite derivatives of the flat output.

Multicopter dynamics and differential flatness



Now all states and inputs are known except the torque.

We trivially differentiate the body rate to get its expression using $r^{(4)},\ \psi^{(2)}$

Consequently,

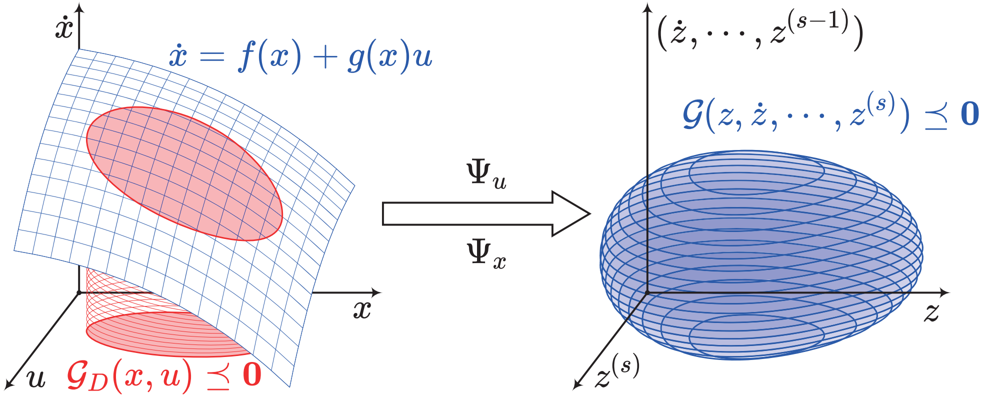
$$\tau = M\dot{\omega} + \omega \times M\omega + A(\omega) + B(R^{\mathrm{T}}v)$$

Finally, we have finished the derivation of **flatness transformation**.

$$z = \{r, \psi\} \in \mathbb{R}^3 \times \mathrm{SO}(2)$$

$$x = \Psi_x(z, \dot{z}, \cdots, z^{(s-1)}),$$
$$u = \Psi_u(z, \dot{z}, \cdots, z^{(s)}).$$

$$x = \{r, v, R, \omega\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{R}^3$$
$$u = \{f, \tau\} \in \mathbb{R}_{\geq 0} \times \mathbb{R}^3$$



Planning flat-output trajectories with high-order continuity suffices for the dynamics.