Flatness-based nonlinear control strategies for trajectory tracking of quadcopter systems under faults

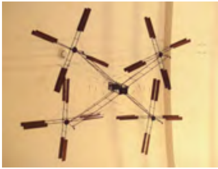
Ngoc Thinh Nguyen* Ionela Prodan* Florin Stoican** Laurent Lefèvre*

*Univ. Grenoble Alpes, LCIS, F-26902, Valence, France,
Email: {ngoc-thinh.nguyen,ionela.prodan,laurent.lefevre}@lcis.grenoble-inp.fr
**Univ. Politehnica of Bucharest, Automatic Control and Systems Engineering Department, Bucharest, Romania,
Email: florin.stoican@acse.pub.ro

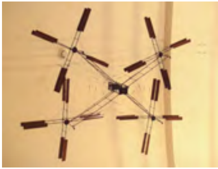
November 15, 2016

Journée GDR MACS, Lille, France

Quadcopters have a long history

Breguet-Richet Gyroplane No.1 1907	Oemichen No.2 1920	DJI Phantom 4
		

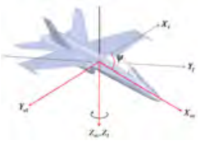
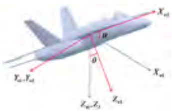
Quadcopters have a long history

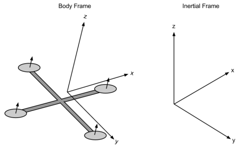
Breguet-Richet Gyroplane No.1 1907	Oemichen No.2 1920	DJI Phantom 4
		
Bitcraze Crazyfile 2.0	Bell Boeing V–22 Osprey	Ambulance Drone
		

Outline

Kinematics of quadcopter

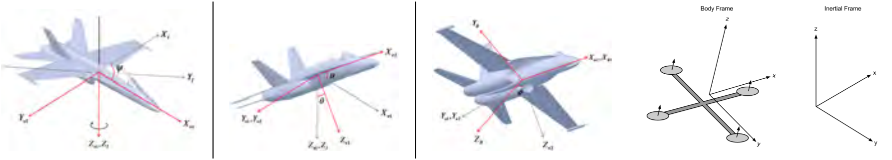
Euler ZYX (ψ, θ, ϕ) rotation sequence:



Kinematics of quadcopter

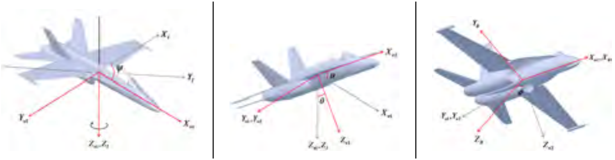
Euler ZYX (ψ, θ, ϕ) rotation sequence:

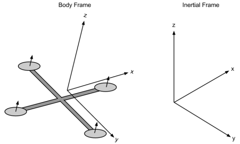


$$
{}^{I}_{B}R = R_Z(\psi)R_Y(\theta)R_X(\phi) = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}
$$

Kinematics of quadcopter

Euler ZYX (ψ, θ, ϕ) rotation sequence:



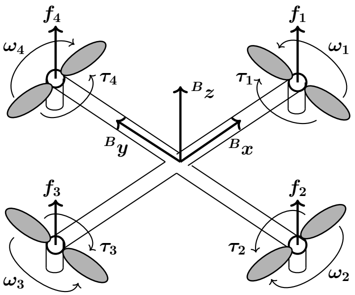


$$ {}^{I}_{B}R = R_Z(\psi)R_Y(\theta)R_X(\phi) = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} $$

The angular velocity ${}^{B}\overrightarrow{\omega}$ physically measured by the gyroscope ${}^{B}\overrightarrow{\omega} \triangleq (\omega_x \; \omega_y \; \omega_z)^T$ is expressed in term of 3 Euler angles $\eta \triangleq (\phi, \theta, \psi)^T$ as:

$$ {}^{B}\overrightarrow{\omega} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W\dot{\eta} $$

Forces and torques created by the 4 propellers



Aerodynamic forces and torques (Spakovszky (2007)):

$$f_i = K_T \omega_i^2$$
$$\tau_i \approx (-1)^i b \omega_i^2$$

Total thrust force and torques acting on the quadcopter (Formentin and Lovera (2011)):

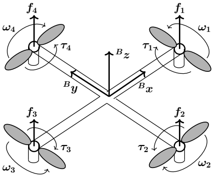
$$T = \sum_{i=1}^{4} f_i = K_T \sum_{i=1}^{4} \omega_i^2$$
$$\tau_\phi = Lf_4 - Lf_2 = LK_T \left(\omega_4^2 - \omega_2^2\right)$$
$$\tau_\theta = Lf_3 - Lf_1 = LK_T \left(\omega_3^2 - \omega_1^2\right)$$
$$\tau_\psi = \sum_{i=1}^{4} \tau_i = b\left(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2\right)$$

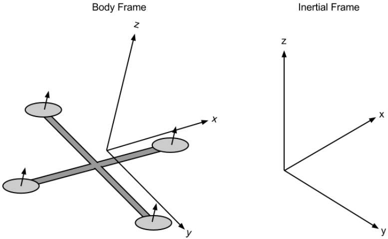
Newton-Euler formalism

Translation equation:

$$m\ddot{\xi} = m\overrightarrow{g} + ({}^{I}_{B}R)\overrightarrow{T} + \overrightarrow{F_D},$$

where $\xi \triangleq (x, y, z)^T$ represents the position of the quadcopter in the IF, m is the system mass, $\overrightarrow{T}$ is the thrust force in BF and $\overrightarrow{F_D}$ is the external perturbation force.





Rotation equation:

$$^{B}I^{B}\dot{\overrightarrow{\omega}} + {}^{B}\overrightarrow{\omega} \times ({}^{B}I^{B}\overrightarrow{\omega}) = \tau_\eta,$$

where ^{B}I is a diagonal inertial matrix, $\tau_\eta \triangleq (\tau_\phi, \tau_\theta, \tau_\psi)^T$ gathers the roll, pitch and yaw torques and '$\times$' denotes the cross-product of two vectors.

Outline

Flat systems and their trajectories

Consider the continuous nonlinear system

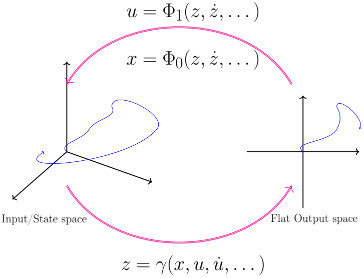
$$\dot{x}(t) = f(x(t), u(t)),$$

it is called differentially flat if there exist $z(t)$ s.t. the states and inputs can be algebraically expressed in terms of $z(t)$ and a finite number of its derivatives:

$$x(t) = \Phi_0(z(t), \dot{z}(t), \cdots, z^{(q)}(t)),$$
$$u(t) = \Phi_1(z(t), \dot{z}(t), \cdots, z^{(q+1)}(t)),$$

where

$$z(t) = \gamma(x(t), u(t), \dot{u}(t), \cdots, u^{(q)}(t))$$



- For any linear and nonlinear flat system, the number of flat outputs equals the number of inputs Lévine (2009), Fliess et al. (1995)
- For linear systems, the flat differentiability is implied by the controllability property Sira-Ramírez and Agrawal (2004)

B-spline curve generation

Considering a collection of *control points*

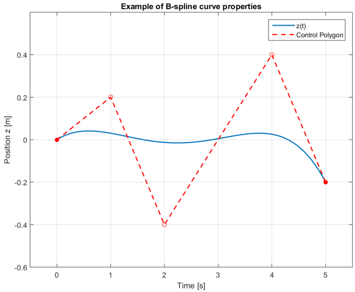
$$\mathbb{P} = \{p_0, p_1, \ldots, p_n\},$$

we rewrite the knot-vector as

$$\mathbb{T} = (\underbrace{\tau_0, \tau_1, \ldots, \tau_{d-1}}_{d \text{ equal knots}}, \underbrace{\tau_d, \tau_{d+1}, \ldots \tau_{n-1}, \tau_n}_{n-d+1 \text{ internal knots}}, \underbrace{\tau_{n+1}, \tau_{n+d}}_{d \text{ equal knots}})$$

and define a *B-spline curve* as a linear combination of the control points and the B-spline basis functions

$$z(t) = \sum_{i=0}^{n} B_{i,d}(t) p_i = \mathbf{P}\mathbf{B}_d(t)$$



Constrained parametrization – I

Let us consider a collection of $N+1$ way-points and time stamps associated to them Prodan (2012):

$$\mathbb{W} = \{w_k\} \text{ and } \mathbb{T}_{\mathbb{W}} = \{t_k\},$$

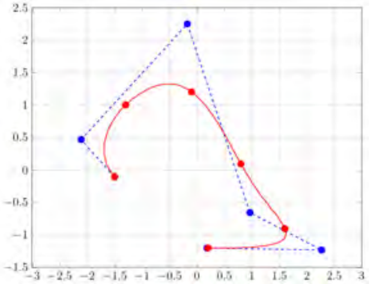
for any $k = 0, \ldots, N$.

The goal is to construct a flat trajectory which passes through each way-point w_k at the time instant t_k, i.e., find a flat output $z(t)$such that

$$x(t_k) = \Theta(z(t_k), \ldots z^{(r)}(t_k)) = w_k, \ \forall k = 0 \ldots N.$$

$$\tilde{\Theta}(\mathbf{B}_d(t_k), \mathbf{P}) = w_k, \ \forall k = 0 \ldots N,$$

where $\tilde{\Theta}(\mathbf{B}_d(t), \mathbf{P}) = \Theta(\mathbf{P}\mathbf{B}_d(t), \ldots, \mathbf{P}M_rL_r\mathbf{B}_d(t))$ is constructed along property (P5).



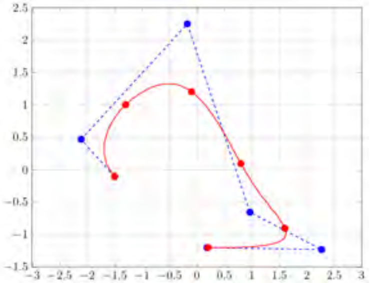
Constrained parametrization – II

Solve an optimization problem Stoican et al. (2016), De Doná et al. (2009), Suryawan (2012):

$$\mathbf{P} = \arg\min_{\mathbf{P}} \int_{t_0}^{t_N} ||\tilde{\Xi}(\mathbf{B}_d(t), \mathbf{P})||_Q \, dt$$
$$\text{s.t. } \tilde{\Theta}(\mathbf{B}_d(t_k), \mathbf{P}) = w_k, \ \forall k = 0 \ldots N$$

with Q a positive symmetric matrix.

- The cost $\tilde{\Xi}(\mathbf{B}_d(t), \mathbf{P}) = \Xi(\tilde{\Theta}(\mathbf{B}_d(t), \mathbf{P}), \tilde{\Phi}(\mathbf{B}_d(t), \mathbf{P}))$ can impose any penalization we deem to be necessary (length of the trajectory, input variation, input magnitude, etc).
- In general, such a problem is nonlinear (due to mappings $\tilde{\Theta}(\cdot)$ and $\tilde{\Phi}(\cdot)$) and hence difficult to solve.



Outline

Flat output description of the quadcopter system

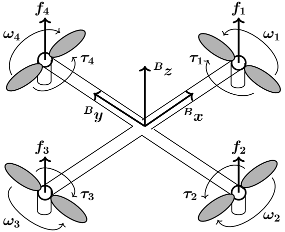
The flat output vector is considered as:

$$\mathbf{z} = \left[z_1\ z_2\ z_3\ z_4\right]^\top = \left[x\ y\ z\ \tan\left(\frac{\psi}{2}\right)\right]^\top,$$

The states and inputs described in term of $\mathbf{z}$:

$$\mathbf{x} = \left[\xi^\top\ \eta^\top\right]^\top = \left[x\ y\ z\ \phi\ \theta\ \psi\right]^\top$$

$$\mathbf{u} = \left[T\ \tau_\phi\ \tau_\theta\ \tau_\psi\right]^\top$$



$$\phi = arcsin\left(\frac{2z_4\ddot{z}_1 - (1-z_4^2)\ddot{z}_2}{(1+z_4^2)\sqrt{\ddot{z}_1{}^2 + \ddot{z}_2{}^2 + (\ddot{z}_3+g)^2}}\right)$$

$$\theta = arctan\left(\frac{(1-z_4^2)\ddot{z}_1 + 2z_4\ddot{z}_2}{(1+z_4^2)(\ddot{z}_3+g)}\right)$$

$$\psi = 2arctan(z_4)$$

$$T = m\sqrt{\ddot{z}_1{}^2 + \ddot{z}_2{}^2 + (\ddot{z}_3+g)^2}$$

$$\tau_\eta =^B I\left(W\ddot{\eta} + \dot{W}\dot{\eta}\right) + (W\dot{\eta}) \times (^B I W\dot{\eta})$$

Intensively studied trajectory generation problem

Specifications which need to be taken into account at the off-line and on-line stages:

- internal dynamics of the system
- state and input constraints fulfillment
- optimization problem such that a certain objective is minimized/maximized (e.g, length curve, total energy, dissipating energy, wind effects)
- trajectory reconfiguration mechanisms
- obstacle avoidance specifications
- multi–trajectory generation

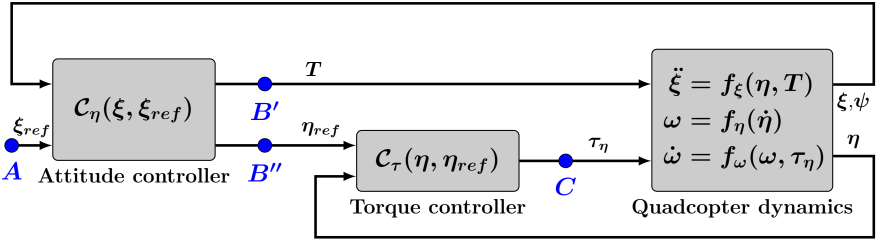
Stoican et al. (2016), Prodan et al. (2013a), Chamseddine et al. (2012), Suryawan et al. (2011), De Doná et al. (2009), Formentin and Lovera (2011); Sydney et al. (2013)

For further use, the references for states and inputs are denoted as:

$$\bar{\mathbf{x}} = \left[\bar{\xi}^\top \; \bar{\eta}^\top\right]^\top = \left[\bar{x} \; \bar{y} \; \bar{z} \; \bar{\phi} \; \bar{\theta} \; \bar{\psi}\right]^\top$$
$$\bar{\mathbf{u}} = \left[\bar{T} \; \bar{\tau_\phi} \; \bar{\tau_\theta} \; \bar{\tau_\psi}\right]^\top$$

Outline

General control scheme



Method:

- feedback linearization control via flatness
- computed torque control

Computed torque based control law

Consider the general dynamics of mechanical system:

$$M(\Phi)\ddot{\Phi} + V(\Phi, \dot{\Phi}) = \tau_{\Phi}$$

Computed torque concept Craig (2005) considers:

$$\tau_{\Phi} = \alpha\tau' + \beta$$

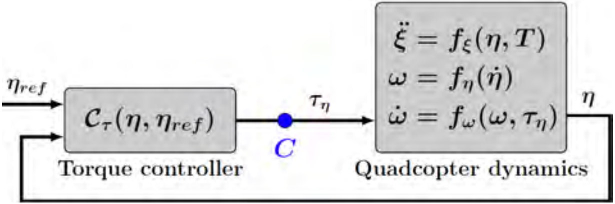
where the model-based portions α, β and the servo portion τ' are given as:

$$\alpha = M(\Phi), \; \beta = V(\Phi, \dot{\Phi})$$
$$\tau' = \ddot{\Phi}_r + K_p E + K_d \dot{E} + K_i \int E dt$$

with $E = \Phi_r - \Phi$.
The controlled system becomes:

$$M(q)\ddot{\Phi} + V(\Phi, \dot{\Phi}) = M(q)\left(\ddot{\Phi}_r + K_p E + K_d \dot{E} + K_i \int E dt\right) + V(\Phi, \dot{\Phi})$$
$$\Rightarrow \ddot{E} + K_d \dot{E} + K_p E + K_i \int E dt = 0$$

Torque controller



Consider the rotational equation of the quadcopter:

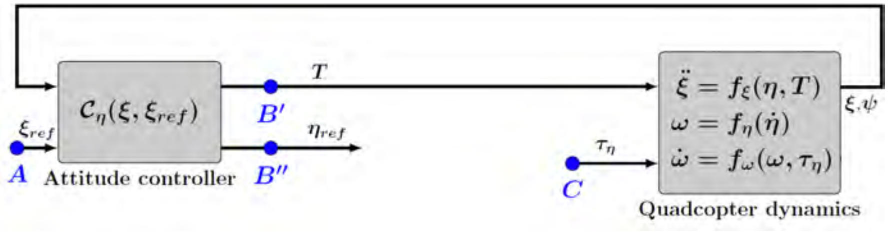
$$^{B}IW\ddot{\eta} + ^{B}I\dot{W}\dot{\eta} + (W\dot{\eta}) \times (^{B}IW\dot{\eta}) = \tau_\eta$$

Applying the computed torque based control law, the input angle torques can be considered as:

$$\tau_\eta = ^{B}IW\left(\ddot{\eta}_r + K_{p\eta}E_\eta + K_{d\eta}\dot{E}_\eta + K_{i\eta}\int E_\eta dt\right) + ^{B}I\dot{W}\dot{\eta} + (W\dot{\eta}) \times (^{B}IW\dot{\eta})$$

$$\Rightarrow \ddot{E}_\eta + K_{d\eta}\dot{E}_\eta + K_{p\eta}E_\eta + K_{i\eta}\int E_\eta dt = 0$$

Attitude controller



Consider the roll, pitch, yaw angles and input thrust T in terms of the flat output $\mathbf{z}$:

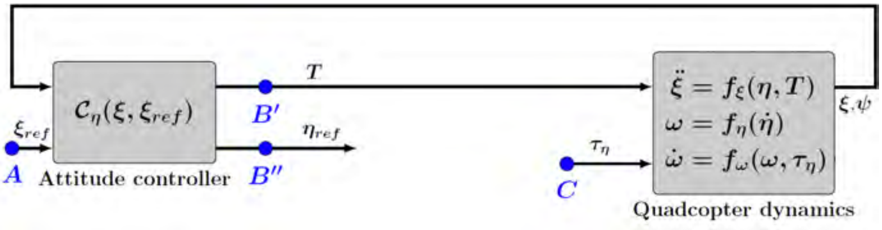
$$\phi = \arcsin\left(\frac{2z_4\ddot{z}_1 - (1 - z_4^2)\ddot{z}_2}{(1 + z_4^2)\sqrt{\ddot{z}_1{}^2 + \ddot{z}_2{}^2 + (\ddot{z}_3 + g)^2}}\right) = \Gamma_\phi(\ddot{z}_1, \ddot{z}_2, \ddot{z}_3, z_4)$$

$$\theta = \arctan\left(\frac{(1 - z_4^2)\ddot{z}_1 + 2z_4\ddot{z}_2}{(1 + z_4^2)(\ddot{z}_3 + g)}\right) = \Gamma_\theta(\ddot{z}_1, \ddot{z}_2, \ddot{z}_3, z_4)$$

$$\psi = 2\arctan(z_4) = \Upsilon_\psi(z_4)$$

$$T = m\sqrt{\ddot{z}_1{}^2 + \ddot{z}_2{}^2 + (\ddot{z}_3 + g)^2} = \Gamma_T(\ddot{z}_1, \ddot{z}_2, \ddot{z}_3)$$

Attitude controller (cont.)



The feedback linearization based control law for attitude controller:

$$\begin{aligned}
\phi_{ref} &= \Gamma_\phi(\ddot{z}_1{}^*, \ddot{z}_2{}^*, \ddot{z}_3{}^*, z_4), \\
\theta_{ref} &= \Gamma_\theta(\ddot{z}_1{}^*, \ddot{z}_2{}^*, \ddot{z}_3{}^*, z_4), \\
\psi_{ref} &= \Upsilon_\psi(\bar{z}_4), \\
T &= \Gamma_T(\ddot{z}_1{}^*, \ddot{z}_2{}^*, \ddot{z}_3{}^*),
\end{aligned}$$

where the corrective term $\ddot{\xi}^* \triangleq \left[\ddot{z}_1{}^* \;\; \ddot{z}_2{}^* \;\; \ddot{z}_3 *\right]^\top$ is given as:

$$\ddot{\xi}^* = \ddot{\xi}_{ref} + K_{d\xi}\dot{\epsilon_\xi} + K_{p\xi}\epsilon_\xi + K_{i\xi}\int \epsilon_\xi dt,$$

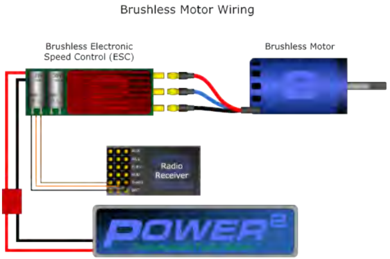
and z_4 is the real value of ψ angle.

Stuck rotor fault modeling

The rotor speed under stuck fault consideration is modeled through (Qi et al. (2013)):

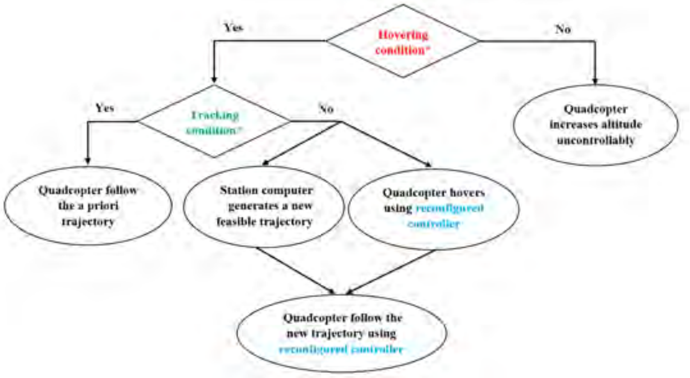
$$\omega_i^f(t) = f_i(t)\omega_i(t) + (1 - f_i(t))\alpha_i\omega_{max},$$

where $f_i(t)$ is the binary fault signal and $\alpha_i \in (0,1)$ is the constant fault magnitude.



Fault tree analysis

A FDD (Fault Detection and Diagnosis) module is used to detect, isolate, and identify the fault magnitude α_i.



* used to ensure the quadcopter's capability of maintaining the position while waiting for the new trajectory and after finishing the tracking mission;

** used to ensure the tracking capability for the a priori trajectory or to generate a new feasible trajectory.

Conditions under stuck rotor fault

Hovering condition*: considering a unique i^{th} rotor stuck, the quadcopter can still assume hovering if the following constraint is respected:

$$\alpha_i^2 \leq \frac{mg}{2K_T\omega_{max}^2}.$$

Tracking condition**: considering a unique i^{th} stuck rotor we have that a sufficient condition for tracking the position component of the reference trajectory is:

$$0 \leq min\left((MP_i)^{-1}\begin{bmatrix}\bar{T}\\ \bar{\tau}_\phi\\ \bar{\tau}_\theta\end{bmatrix} - (MP_i)^{-1}MK_i\alpha_i^2\omega_{max}^2\right),$$

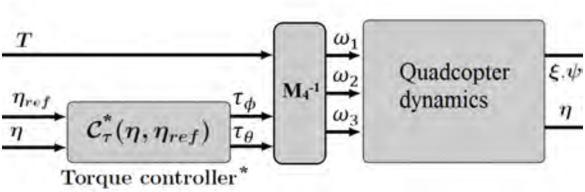
$$max\left((MP_i)^{-1}\begin{bmatrix}\bar{T}\\ \bar{\tau}_\phi\\ \bar{\tau}_\theta\end{bmatrix} - (MP_i)^{-1}MK_i\alpha_i^2\omega_{max}^2\right) \leq \omega_{max}^2,$$

where $K_i \in \mathbb{R}^4$ is the i^{th} column of the identity matrix I_4 of size 4, the matrix $P_i \in \mathbb{R}^{4\times 3}$ is composed of the other columns of I_4 and

$$M = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -LK_T & 0 & LK_T \\ -LK_T & 0 & LK_T & 0 \end{bmatrix}.$$

Control reconfiguration under stuck rotor

Reconfigured controller under fault of the 4^{th} stuck rotor:



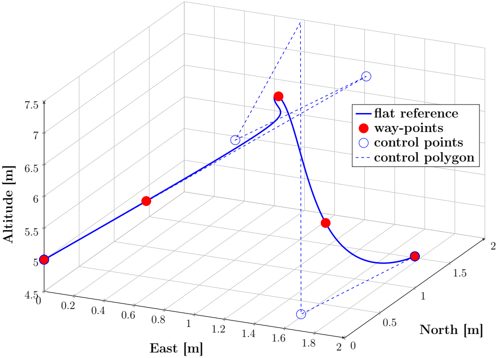
where the matrix M_4 is calculated as:

$$M_4 = \begin{array}{c} \begin{array}{cccc} & & & 4^{th} \text{ column} \end{array} \\ \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -LK_T & 0 & LK_T \\ LK_T & 0 & -LK_T & 0 \\ b & b & b & b \end{bmatrix} \end{array}$$

Outline

Waypoint trajectory generation

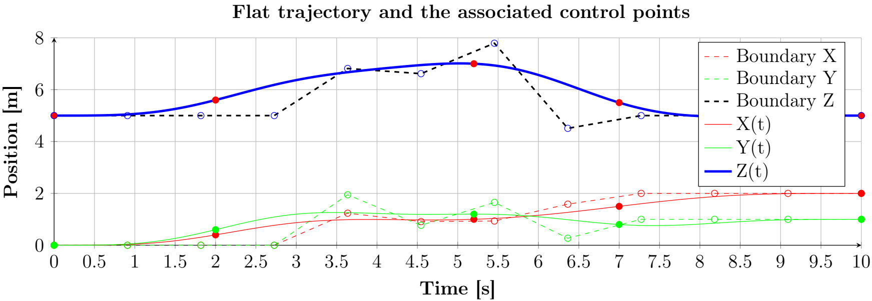
Generate the trajectory passing through 5 way points at specific time instants taking into account the operating constraints while minimize the length of the trajectory.

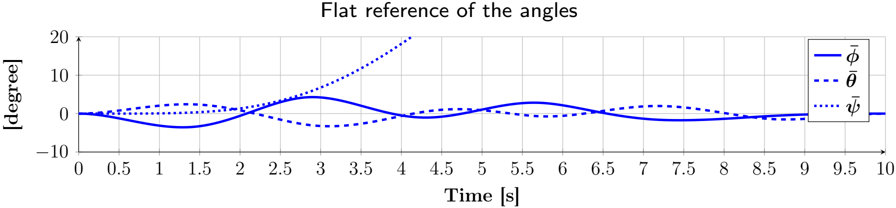


Simulation of wind disturbances:

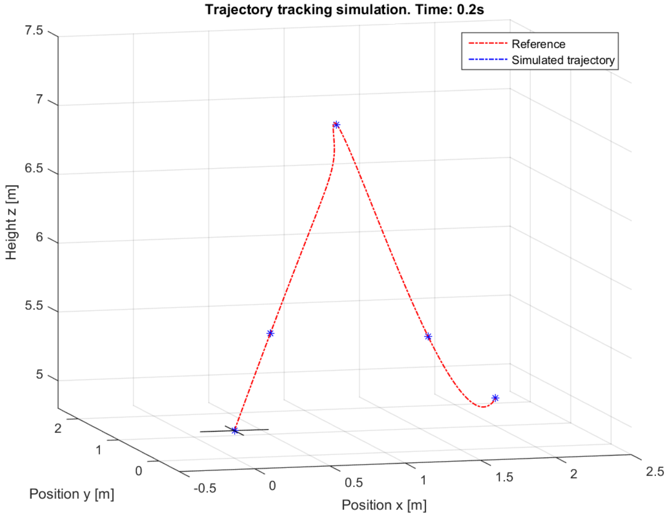
Nominal	Moderate	Strong
0 km/h	15.3 km/h	20.5 km/h

Waypoint trajectory generation



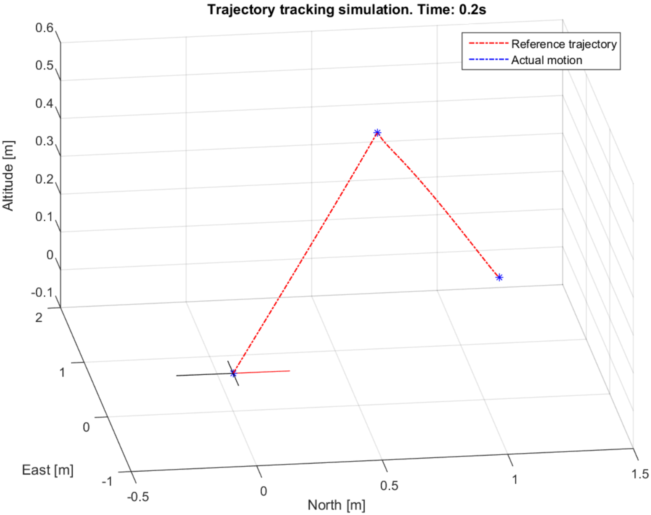


Trajectory tracking



Simulation of combined flat angle and position tracking strategy in strong wind condition (20.5 km/h)

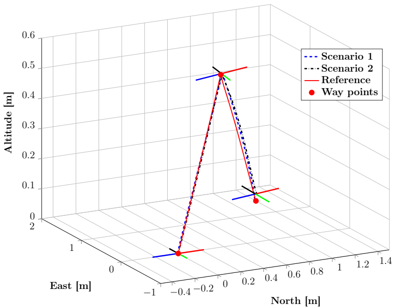
Trajectory tracking under stuck rotor fault



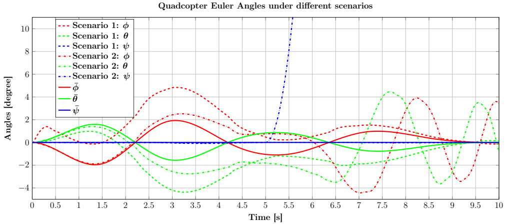
Simulation of way-point tracking under fault in strong wind conditions.
The 4^{th} rotor is permanently stuck from $t = 5$ sec.

Trajectory tracking under fault of stuck rotor

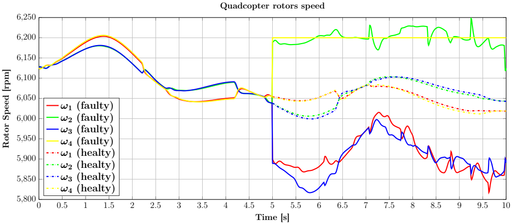
- Scenario 1: Healthy rotors
- Scenario 2: the 4^{th} stuck rotor



Actual motions under different scenarios



Angles under different scenarios



Rotors speed under different scenarios

Simulation parameters and IAE results

- Controller parameters:

Table: Controllers tuning parameters.

Controller	$K_p = diag\{.\}$	$K_d = diag\{.\}$	$K_i = diag\{.\}$
Attitude	$16, 16, 16$	$8, 8, 8$	$1, 1, 0.3$
Torque (nominal)	$25, 25, 25$	$10, 10, 10$	$0, 0, 0$
Torque (fault)	$25, 25$	$10, 10$	$0, 0$

- Yalmip in Matlab/Simulink Löfberg (2004)
- Fixed sampled time of 0.01s and solver ode4
- Integral of Absolute magnitude of the Error (IAE) over the position:

$$IAE_{\xi} = \int_{t_0=0}^{t_f=10} ||\xi_r - \xi|| dt$$

	Nominal Wind	Moderate Wind	Strong Wind
Nominal case	0.0278	0.3750	0.5980
Faulty case	0.2111	0.4320	0.6221

Table: IAE results of 3 control strategies

Outline

Conclusions and future developments

Conclusions:

- Quadcopter modeling using Newton-Euler formalism
- Novel flat output representation
- Trajectory generation problem
- Feedback linearization based control designs for trajectory tracking
- Robustness under bounded wind perturbations
- Control reconfiguration analysis under stuck rotor fault
- Extensive simulations for different wind conditions

Future development:

- MPC/NMPC implementations
- Bounded/stochastic disturbances considerations
- Trajectory reconfiguration mechanisms
- Experiments on the Crazyflie platform

[0] Prodan I., Stoican F., Olaru S. and Niculescu S-I. (2016): Mixed-Integer Representations in Control Design, SpringerBriefs in Control, Automation and Robotics Series, Springer.

References I

A. Chamseddine, Y. Zhang, C.A. Rabbath, C. Join, and D. Theilliol. Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):2832–2848, 2012.

J.J. Craig. *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.

J. De Doná, F. Suryawan, M. Seron, and J. Lévine. A flatness-based iterative method for reference trajectory generation in constrained NMPC. *Int. Workshop on Assesment and Future Direction of Nonlinear Model Predictive Control*, pages 325–333, 2009.

M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.

Simone Formentin and Marco Lovera. Flatness-based control of a quadrotor helicopter via feedforward linearization. In *CDC-ECE*, pages 6171–6176, 2011.

J. Lévine. *Analysis and control of nonlinear systems: A flatness-based approach*. Springer Verlag, 2009.

J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://users.isy.liu.se/johanl/yalmip`.

I. Prodan. *Control of Multi-Agent Dynamical Systems in the Presence of Constraints*. PhD thesis, Supélec, 2012.

I. Prodan, F. Stoican, S. Olaru, and Niculescu S.I. Enhancements on the Hyperplanes Arrangements in Mixed-Integer Techniques. *Journal of Optimization Theory and Applications*, 154(2):549–572, 2012a. doi: 10.1007/s10957-012-0022-9.

I. Prodan, F. Stoican, S. Olaru, C. Stoica, and S-I. Niculescu. Mixed-integer programming techniques in distributed mpc problems. In *Distributed MPC made easy*, number 69, pages 273–288. Springer, 2012b.

I. Prodan, S. Olaru, R. Bencatel, J. Sousa, C. Stoica, and S.I. Niculescu. Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Control Engineering Practice*, 21(10):1334–1349, 2013a. doi: 10.1016/j.conengprac.2013.05.010.

I. Prodan, S. Olaru, C. Stoica, and S.I. Niculescu. Predictive control for trajectory tracking and decentralized navigation of multi-agent formations. *International Journal of Applied Mathematics and Computer Science*, 23(1):91–102, 2013b. doi: 10.2478/amcs-2013-0008.

I. Prodan, S. Olaru, F. Fontes, F. Pereira, C. Stoica Maniu, and S.-I. Niculescu. Predictive control for path following. from trajectory generation to the parametrization of the discrete tracking sequences. In *Topics in optimization based control and estimation*, pages 161–181. Springer, 2015.

Xin Qi, Didier Theilliol, Juntong Qi, Youmin Zhang, and Jianda Han. A literature review on fault diagnosis methods for manned and unmanned helicopters. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 1114–1118. IEEE, 2013.

H. Sira-Ramírez and S. Agrawal. *Differential Flatness*. Marcel Dekker, New York, 2004.

Z. S. Spakovszky. Unified: Thermodynamics and propulsion, 2007. http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/notes.html.

F. Stoican, I. Prodan, and D. Popescu. Flat trajectory generation for way-points relaxations and obstacle avoidance. In *Proceedings of the 23th IEEE Mediterranean Conference on Control and Automation*, pages 695–700. IEEE, 2015.

Florin Stoican, Vlad-Mihai Ivanusca, Ionela Prodan, and Dan Popescu. Obstacle avoidance via b-spline parameterizations of flat trajectories. In *Proceedings of the 24th Mediterranean Conference on Control and Automation*, pages 1002–1007, Athens, Greece, 2016.

F. Suryawan. *Constrained Trajectory Generation and Fault Tolerant Control Based on Differential Flatness and B-splines*. PhD thesis, 2012.

F. Suryawan, J. De Doná, and M. Seron. Minimum-time trajectory generation for constrained linear systems using flatness and b-splines. *International Journal of Control*, 84(9):1565–1585, 2011.

Nitin Sydney, Brendan Smyth, and Derek A Paley. Dynamic control of autonomous quadrotor flight in an estimated wind field. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3609–3616. IEEE, 2013.