Optimized control law for quadrotor position control

Deterministic constrained optimization methods

Final presentation – Optimization methods for engineering problems (01RGBRV)

D.Celestini, D.Ruggiero, M.Rinaldi - May 2022

Politecnico di Torino

OUTLINE

- **Problem formulation:**
  - Introduction on quadrotor dynamics and control.
  - Description of Model Predictive Control.

- **Optimization methods:**
  - Methods for constrained optimization.
  - Penalty Function method.
  - Augmented Lagrange Multiplier.
  - Quadratic Programming.

- **Results:**
  - Comparison of the methods for the first optimization step.
  - Complete simulation of quadrotor motion.

QUADROTOR DYNAMICS AND CONTROL

- **Quadrotor dynamics:**
  - Formulation of position dynamics in the inertial reference frame:

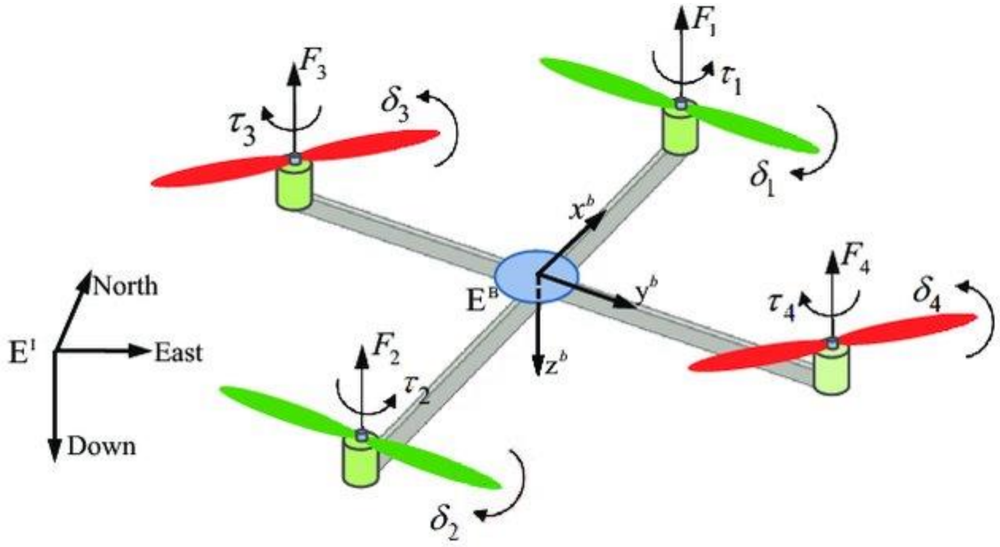
$$\ddot{x} = -\frac{F}{m} = -\frac{1}{\mathrm{m}}\begin{bmatrix} F_x \\ F_y \\ F_z - mg \end{bmatrix}$$

  - Formulation of attitude dynamics in body reference frame:

$$\dot{\omega} = J^{-1}[\tau - \omega \times J\omega]$$

  - Forces and torques acting on the system and correlated with the propellers thrust according to:

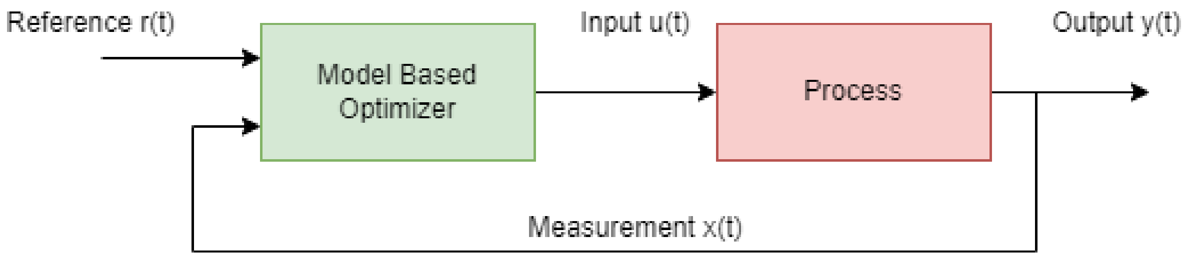
$$\begin{bmatrix} f \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & -lk_f & 0 & lk_f \\ lk_f & 0 & -lk_f & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \Omega^2$$

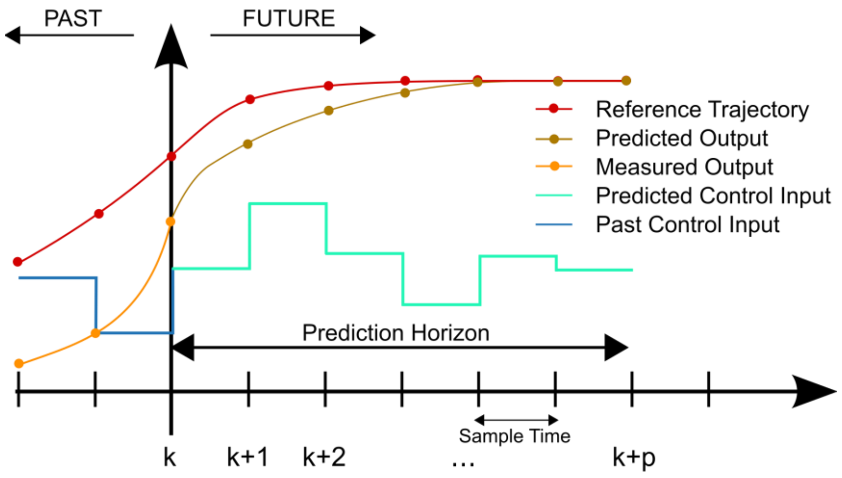


$$\begin{cases} F_x = f[\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta)] \\ F_y = f[\sin(\phi)\cos(\psi) - \cos(\phi)\sin(\psi)\sin(\theta)] \\ F_z = f cos(\phi)\cos(\theta) \end{cases}$$

MODEL PREDICTIVE CONTROL [1]

- **Model Predictive Control (MPC):**
  - MPC concepts date back to 60's.
  - It use a simplified dynamical model of the system to predict its future evolution, evaluating the "best" control action.
  - **MPC problem:** Find the best control sequence over a future horizon of N steps.
  - Algorithm:
    1. Estimate system state evolution x(t).
    2. Find the control sequence u(t) minimizing the cost function J(x,u,t).
    3. Apply the first optimal input u to the process.
    4. Repeat at all time steps.





MODEL PREDICTIVE CONTROL [2]

- **MPC design:**
  - The position dynamics of the quadrotor is a linear function.
  - It can be rewritten in the space state form, with respect to the hovering condition, as

$$\dot{x} = Ax + Bu \quad \text{with} \quad \mathrm{x} = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} \quad \mathrm{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 - \mathrm{mg} \end{bmatrix}$$

  - It can be discretized according with the controller sample Δt as

$$x_{k+1} = A_k x_k + B_k u_k$$

  - Considering a prediction horizon p and a control horizon m the system evolution is predicted for the time interval $[t, t + p\Delta t]$, as function of a sequence of control inputs u.

MODEL PREDICTIVE CONTROL [3]

- **MPC design:**
  - Considering a control horizon $m = 2$ leads to the prediction function

$$x_k = A_d^k x_0 + A_d^{k-1} B_d u_1 + A_d^{k-2} u_2$$

  - Finally, the optimization problem consists in the minimization of the cost function given as

Control minimization

$$J = \int_{t_0}^{t_0+p\Delta t} [(x - x_t)^T Q (x - x_T) + u^T R u] dt + \left(x_p - x_t\right)^T S (x_p - x_t)$$

Error minimization

and subjected to the following constraints

$$LB \leq u \leq UB$$

- **Methods for constrained optimization:**
  - Constrained optimization problems aim at finding the minimizer or maximizer of a function (i.e., cost function) subject to constraints
  - Constrained optimization problem in standard form:

$$\begin{aligned} &minimize && f(\boldsymbol{x}) && \\ &subject\ to && g_i(\boldsymbol{x}) \leq 0 && i = 1 \dots N \\ & && h_j(\boldsymbol{x}) = 0 && j = 1 \dots M \end{aligned}$$

$f(\boldsymbol{x})$: cost function

$\boldsymbol{x}$: optimization variables' vector [*nx1*]

$g(\boldsymbol{x})$: set of N inequality constraints

$h(\boldsymbol{x})$: set of M equality constraints

- Maximization problems and inequality constraints of the form $g^*(\boldsymbol{x}) \geq 0$ can be expressed in standard form
- Often, numerical methods exploit unconstrained formulation of the constrained problem and find local optimal solutions

PENALTY FUNCTION

- **Penalty Function method (PF):**
  - Unconstrained optimization algorithm is applied to a PF formulation of the constrained problem
  - The method can deal with any objective function and both equality and inequality constraints
  - Definition of Penalty Function:

$$PF(X, \alpha, \beta) = f(x) + \sum_{i=1}^{M} \alpha_i h_i^2(x) + \sum_{j=1}^{N} \beta_j g_j^2(x)$$

$$\rho \gg 0$$

$$\alpha = \rho$$

$$\beta_j = 0 \;\; if \;\; g_j(x) \leq 0, \qquad otherwise \; \beta_j = \rho$$

  - Pseudocode of the algorithm:

step k=0 : choose tolerances, X_0, p_0

$\rho_k = 10 * \rho_{k-1}$

X_k*: minimizer of* $PF(X_{k-1}, \alpha_{k-1}, \beta_{k-1})$

If (convergence criteria are met) , stop at step k

(the last three lines are repeated for) $k=1 \ldots N_{max\ steps}$

AUGMENTED LAGRANGE METHOD

- **Augmented Lagrange Multiplier (ALM):**
  - The ALM method combines the classical Lagrange Method with the Penalty Function method.
  - It is used to track inequalities constraints:

$$\begin{array}{ll} \textit{minimize} & f(\boldsymbol{x}) \\ \textit{subject to} & g_j(x) \leq 0 \end{array}$$

  - One possible definition of the augmented Lagrangian function is

$$L(x, \lambda, \rho) = f(x) + \sum_{i=1}^{p} [\max(0.5\lambda_i + \rho g_i(x), 0)]^2$$

    where λ is the Lagrange Multiplier and ρ is an adjustable penalty parameter.
  - Perform unconstrained optimization of L to get x_k^*.
  - Iteratively λ is updated as

$$\lambda_{k+1} = \max(\lambda_k + 2\rho g(x_k^*), 0)$$

  - Iteratively, ρ is updated as

$$\rho_{k+1} = 2\rho_k \qquad if \ \left| |\lambda_k - \lambda_{k+1}| \right| < 0.5$$

- **Quadratic Programming (QP):**
  - Requires quadratic objective function and linear constraints:

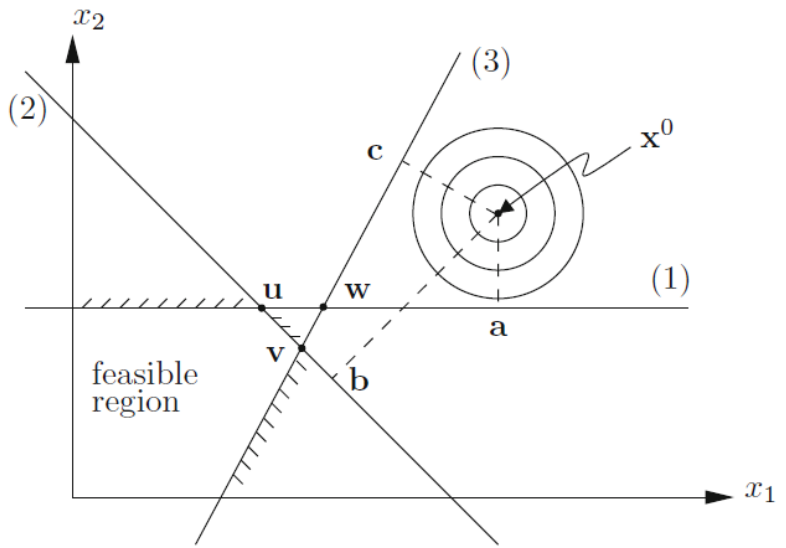
$$\begin{aligned} &\textit{minimize} \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} \\ &\textit{subject to} \quad \sout{C\boldsymbol{x} = \boldsymbol{d}} \\ &\qquad\qquad\quad D\boldsymbol{x} \le \boldsymbol{e} \end{aligned}$$

  - Able to immediately find the global minimum if it is inside the feasibility region as $\boldsymbol{x}^0 = -A^{-1}\boldsymbol{b}$
  - If the global minimum is not inside the feasibility region the solution is on the border
  - Identification of the *active-set of constraints* evaluating all the possible combinations of active constraints from the set of inequality constraints broken

Possible active-set $D'\boldsymbol{x}^k = \boldsymbol{e}'$

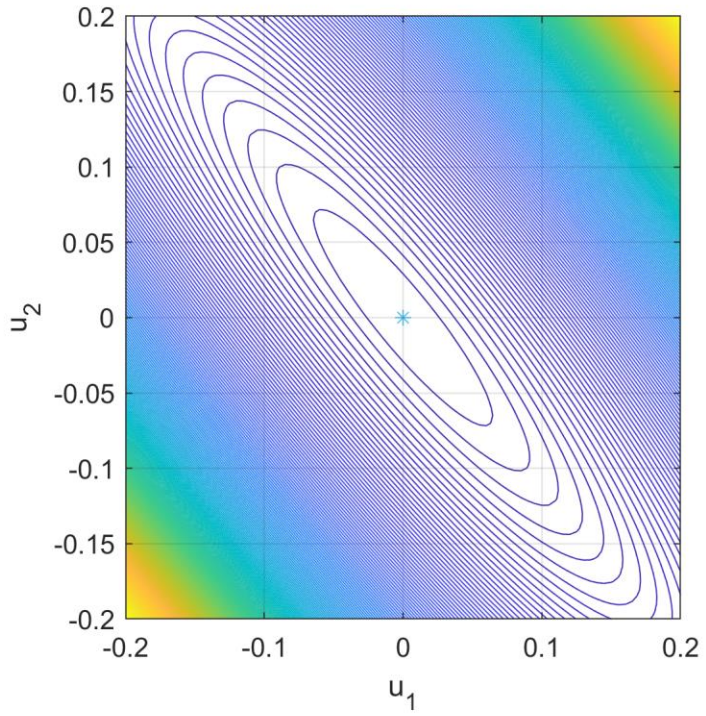
Solution with that active-set $\begin{bmatrix} A & D'^T \\ D' & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^k \\ \boldsymbol{\lambda}^k \end{bmatrix} = \begin{bmatrix} -\boldsymbol{b} \\ \boldsymbol{e}' \end{bmatrix}$

If $D\boldsymbol{x}^k \le \boldsymbol{e}$ and $\boldsymbol{\lambda}^k \ge 0$, $\boldsymbol{x}^k$ is the solution of the problem.

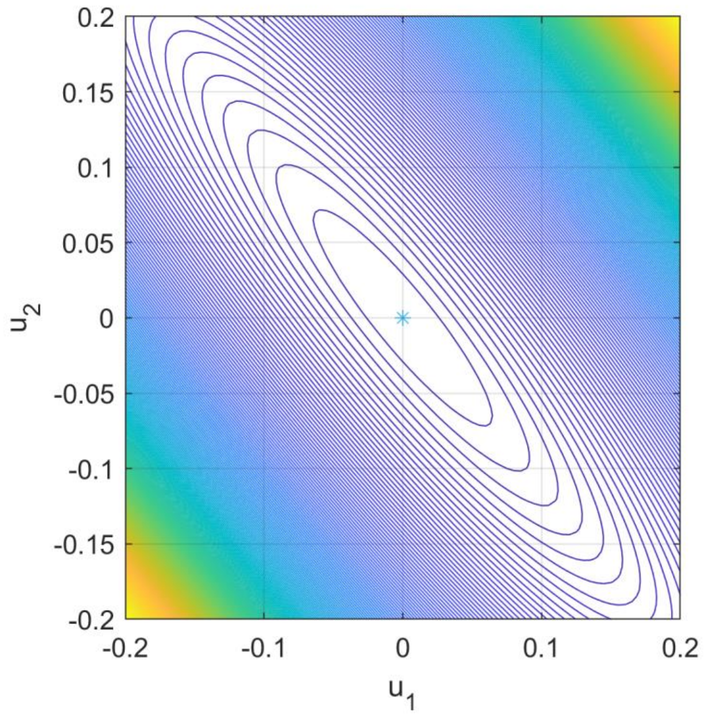


OPTIMIZATION METHODS EVALUATION [1]

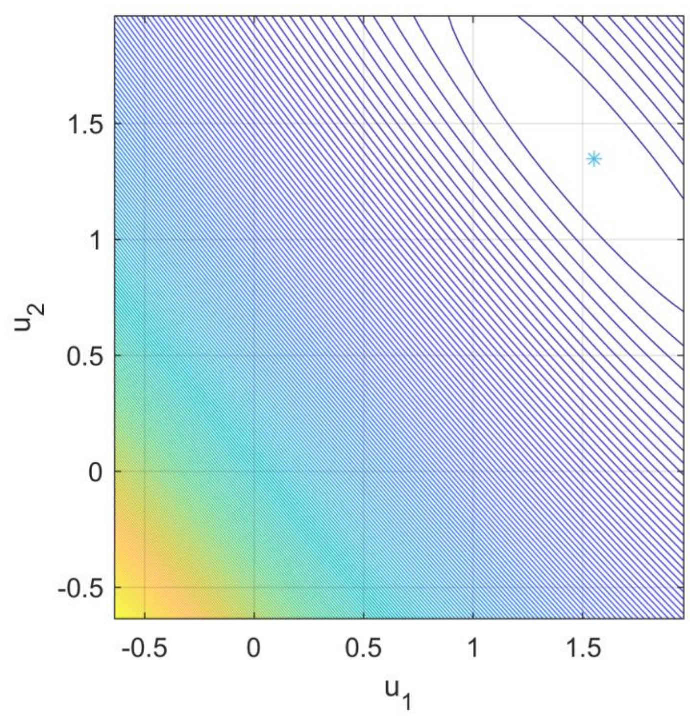
- **Cost Function evaluation:**
  - Vertical flight: $x_0 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ $x_{des} = [0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0]^T$



u_x



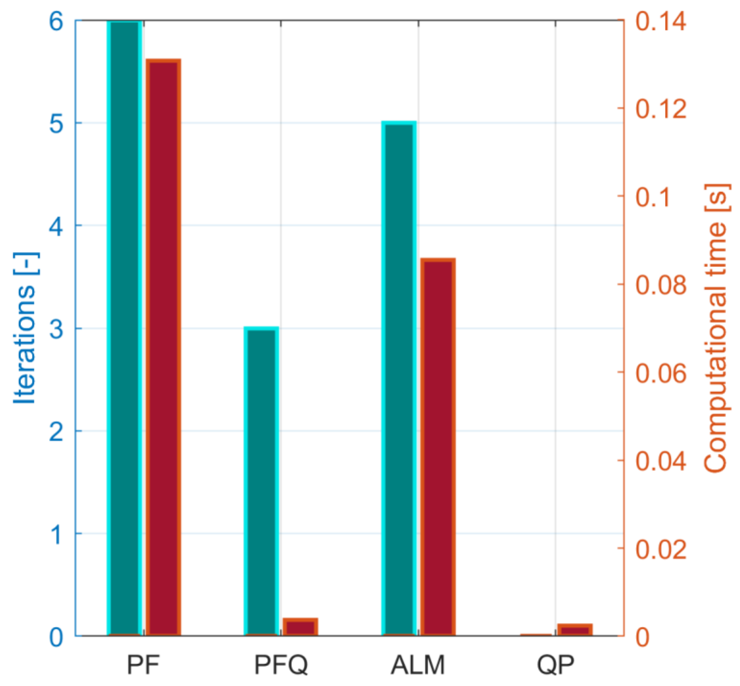
u_y



u_z

OPTIMIZATION METHODS EVALUATION [2]

- **Execution time:**
  - Vertical flight: $x_0 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ $x_{des} = [0\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0]^T$



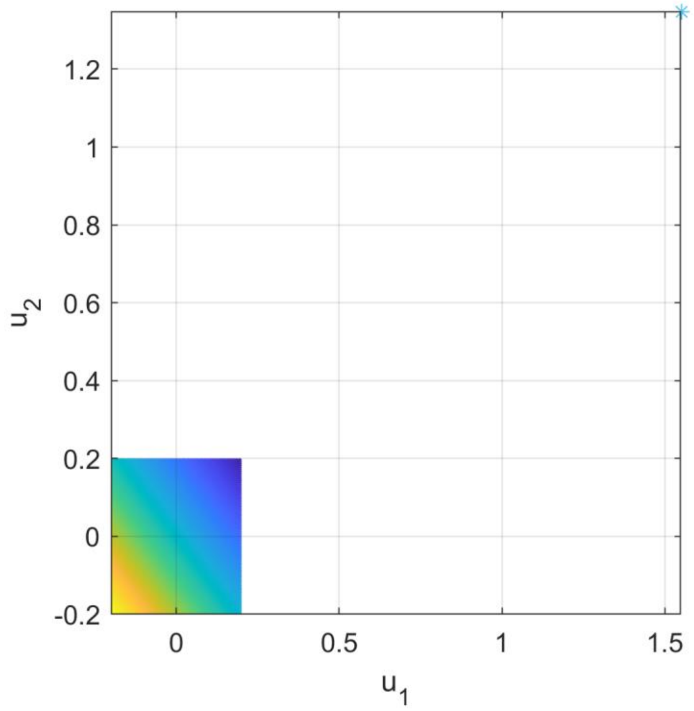
$$u = \begin{bmatrix} 0 \\ 0 \\ 1.5506 \\ 0 \\ 0 \\ 1.347 \end{bmatrix}$$

PF: minimized with *fminsearch()*

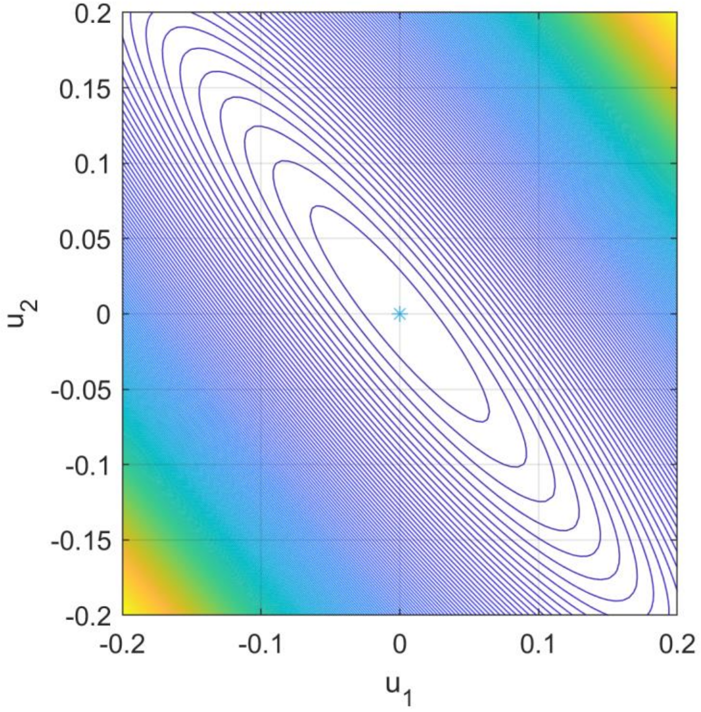
PFQ: minimized with analytical solution $\frac{dPF}{dx} = 0$

OPTIMIZATION METHODS EVALUATION [3]

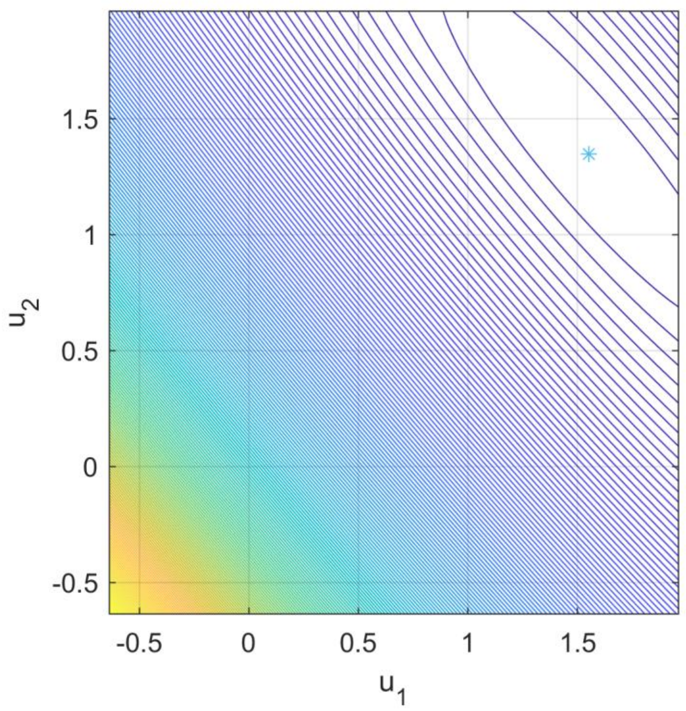
- **Cost Function evaluation:**
  - XZ flight: $x_0 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ $x_{des} = [10\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0]^T$



u_x



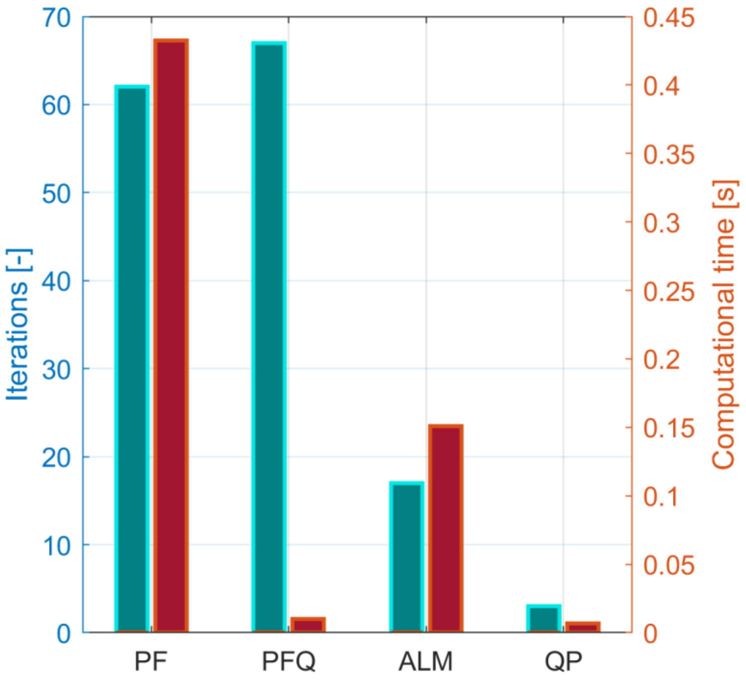
u_y



u_z

OPTIMIZATION METHODS EVALUATION [4]

- **Execution time:**
  - XZ flight: $x_0 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ $\quad x_{des} = [10\ \ 0\ \ 10\ \ 0\ \ 0\ \ 0]^T$



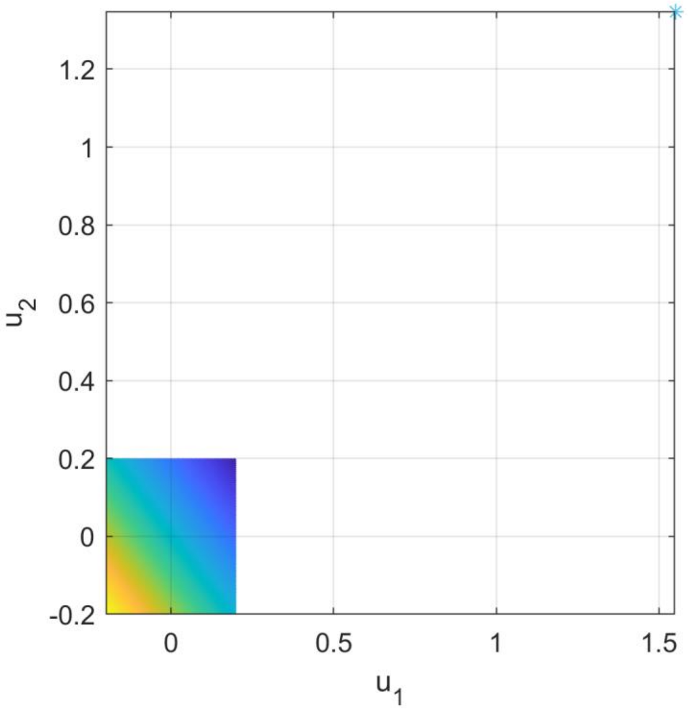
$$u = \begin{bmatrix} 0.2 \\ 0 \\ 1.5506 \\ 0.2 \\ 0 \\ 1.347 \end{bmatrix}$$

PF: minimized with *fminsearch()*

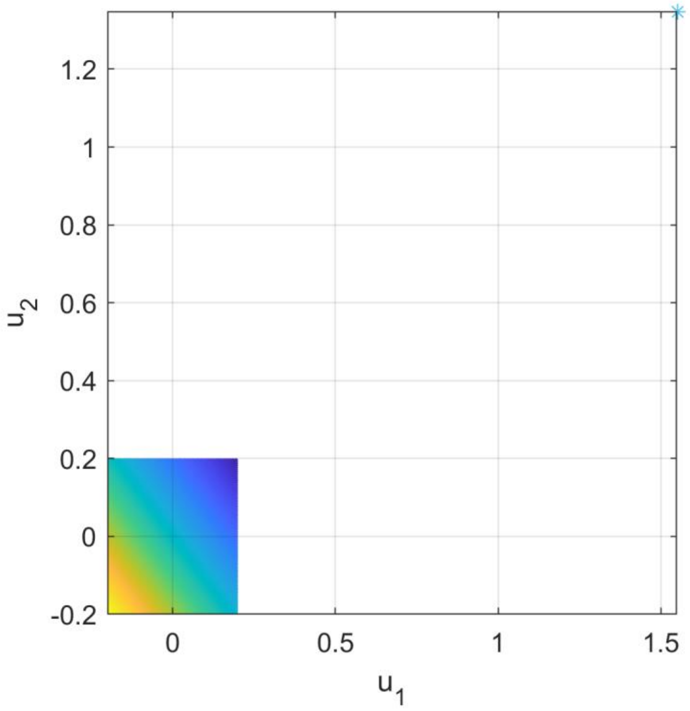
PFQ: minimized with analytical solution $\frac{dPF}{dx} = 0$

OPTIMIZATION METHODS EVALUATION [5]

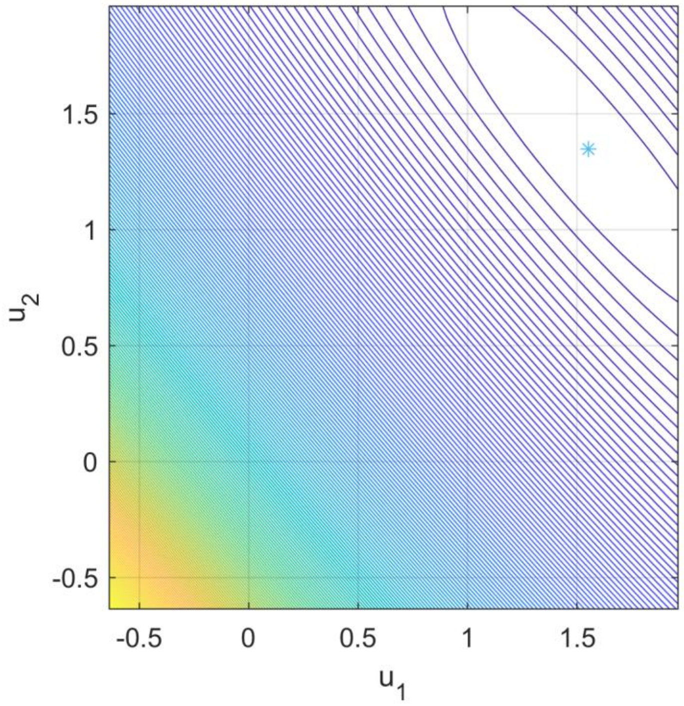
- **Cost Function evaluation:**
  - XYZ flight: $x_0 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ $\quad x_{des} = [10\ \ 10\ \ 10\ \ 0\ \ 0\ \ 0]^T$



u_x



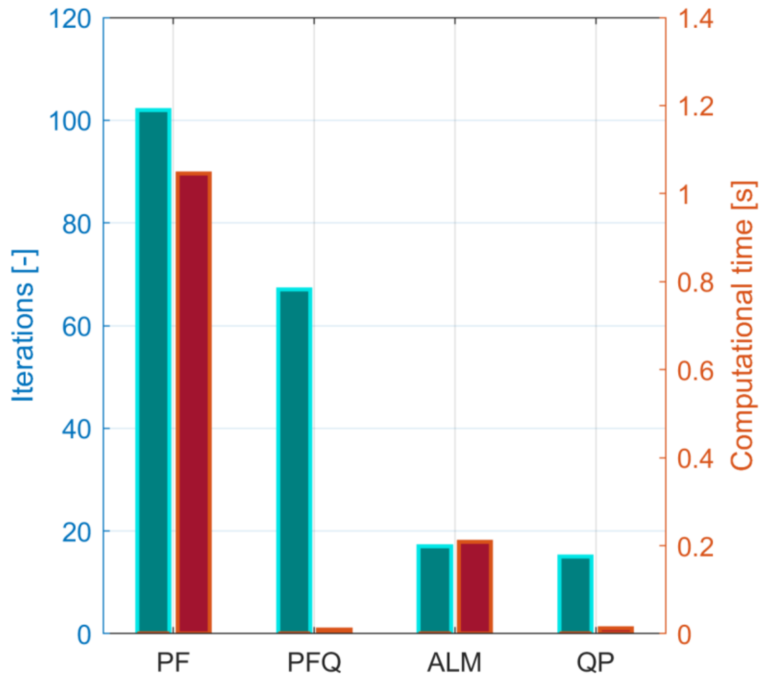
u_y



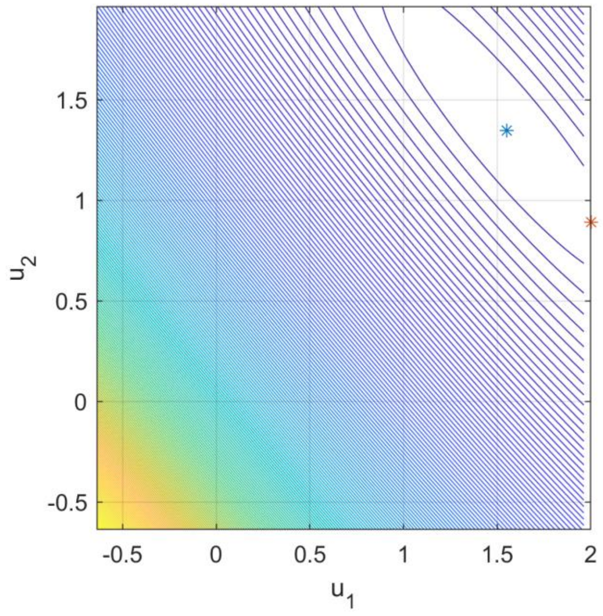
u_z

OPTIMIZATION METHODS EVALUATION [6]

- **Execution time:**
  - XYZ flight: $x_0 = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ $x_{des} = [10\ \ 10\ \ 10\ \ 0\ \ 0\ \ 0]^T$



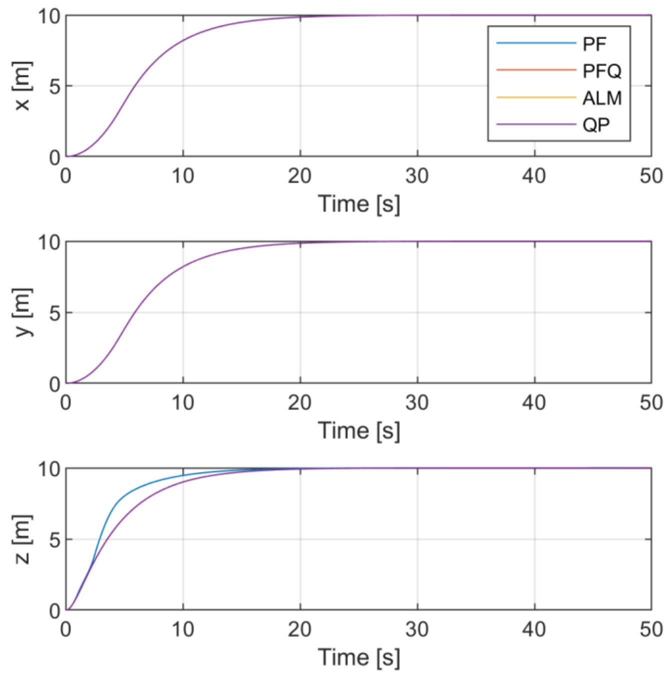
$$u = \begin{bmatrix} 0.2 \\ 0.2 \\ 1.5506 \\ 0.2 \\ 0.2 \\ 1.347 \end{bmatrix} \qquad u_{PF} = \begin{bmatrix} 0.2 \\ 0.2 \\ 2.0 \\ 0.2 \\ 0.2 \\ 0.8905 \end{bmatrix}$$

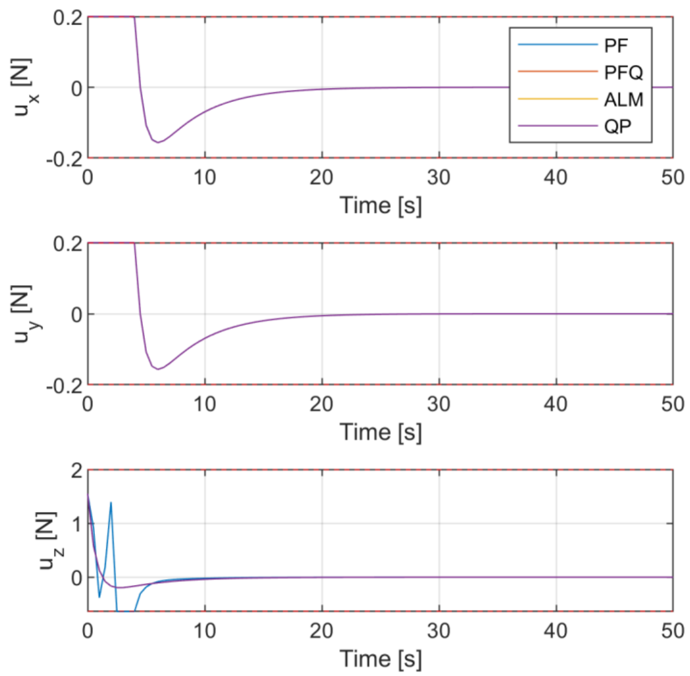


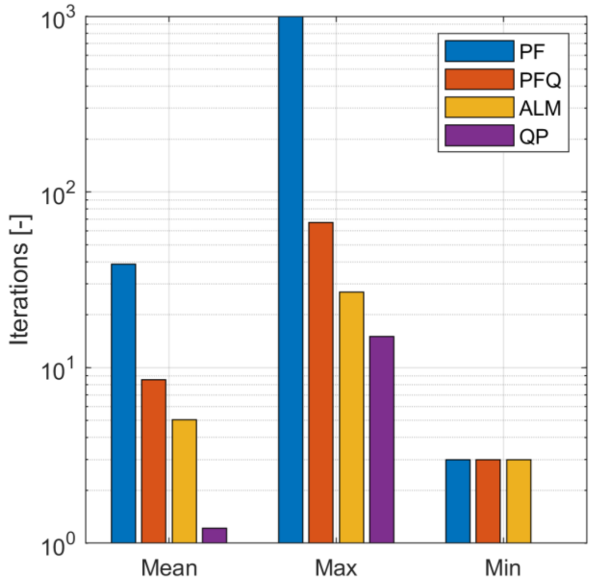
u_z

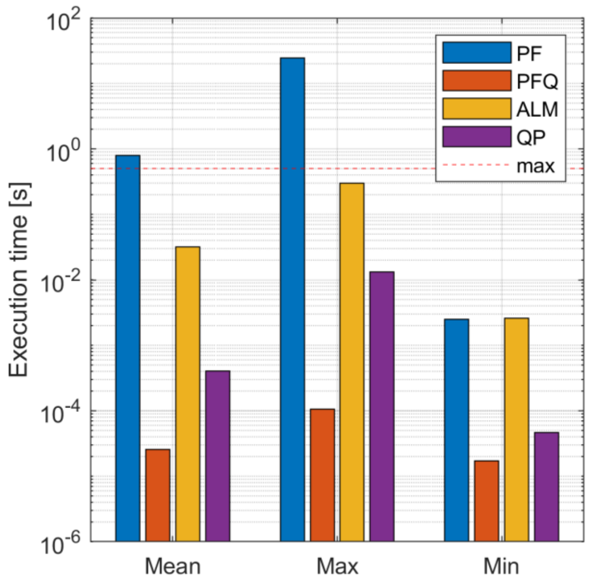
FULL SIMULATION

- **Methods comparison in real time performance:**
  - Simulation results: $x_0 = [0 \quad 0 \quad 0] \rightarrow x_{ref} = [10 \quad 10 \quad 10]$









CONCLUSIONS

- **Conclusions:**
  - Different optimization methods are analyzed to build a Model Predictive Control strategy for the quadrotor motion.
  - A quadratic cost function is built in order to minimize the prediction of the quadrotor position and the forces acting on the system.
  - The constrains acting on the problem are lower and upper bounds for the forces.
  - First, the performance of the methods are evaluated for the first step of simulation.
  - Then, a full simulation scenario is considered.
  - PF **does not** respect the requirements in terms of execution time.
  - The other three considered methods provide good results, but QP and PFQ are only available when quadratic objective functions and linear constraints are considered, while ALM can work also with non-quadratic objective functions and nonlinear constraints.

THANK YOU FOR YOUR ATTENTION