10th ANKARA INTERNATIONAL AEROSPACE CONFERENCE **AIAC-2019-065**
18-20 September 2019 - METU, Ankara TURKEY

HEXACOPTER CONTROL WITH INPUT CONSTRAINTS : COMPARISON BETWEEN MODEL PREDICTIVE CONTROL AND LQR

Emre Can Suicmez* and Ali Turker Kutay†
METU
Ankara, Turkey

ABSTRACT

Control of a hexacopter platform with input constraints is achieved by using Model Predictive Control approach. Input constraints are directly introduced into on-line optimization instead of indirectly limiting the actuators via sturation, command generators or other approaches that is used in common PID or LQR controllers. A basic LQR controller with off-line calculated gain is also designed to test the effectiveness of MPC controller. Dynamic model of hexacopter and controller design is performed in MATLAB/Simulink environment. Results are analyzed by applying constraints on actuator limits and it is seen that MPC controller has advantageous characteristics compared to LQR for this specific control problem.

DEFINITIONS & ABBREVIATIONS

g : Gravitational acceleration
m : Mass of hexacopter
J : Inertia tensor of hexacopter
d : Motor centerline to cg. distance
k_{dh} : Drag constant for horizontal motion
k_{dv} : Drag constant for vertical motion
F_i : Force generated by i_{th} propeller
T_i : Torque generated by i_{th} propeller
ω_i : Rotational speed of i_{th} motor
k_f : Electric motor force constant(F_i vs ω_i)
k_t : Electric motor torque constant(T_i vs F_i)
$\sum F_{ext},\ \sum M_{ext}$: External net forces and moment action on hexacopter cg.
$F_g,\ F_p,\ F_d$: Gravity, Propulsion and Drag forces, respectively.
M_p : Propulsion moment acting on hexacopter cg.
$V = [u,\ v,\ w]^T$: Hexacopter(body) translational velocity expressed in body frame
$\omega = [p,\ q,\ r]^T$: Hexacopter(body) angular velocity expressed in body frame
$R = [x,\ y,\ z]$: Position of hexacopter relative to Earth frame
$\eta = [\phi,\ \theta,\ \psi]$: Euler angles(Roll, Pitch, Yaw) of hexacopter

*PhD Student in Aerospace Engineering, Email: e156123@metu.edu.tr
†Asst. Prof. Dr. in Aerospace Engineering, Email: kutay@metu.edu.tr

L_{BE} : Earth to body frame orthogonal transformation matrix
L_R : Transformation matrix from Euler angle rates to Body angular rates
U_v : Virtual control input vector
U_r : Physically realizable control input vector
X_0 : States at hover trim condition
U_0 : Inputs at hover trim condition
PWM : Pulse Width Modulation
ESC : Electronic Speed Controller
MPC : Model Predictive Control
$LQ-MPC$: Linear Quadratic Model Predictive Control
LQR : Linear Quadratic Control
$VTOL$: Vertical Takeoff and Landing
$c(\cdot)$: Cosine function
$s(\cdot)$: Sine function

INTRODUCTION

Quadcopter is the most popular VTOL configuration if payload is not a prior concern. Increasing the number of motors will provide more payload capacity; however, it also adds additional weight due to increased need for electric batteries and other components. Therefore; there is a trade-off between battery weight, total number of motors and complexity for a VTOL configuration. Hexacopter with six fixed pitch blade propellers is a very useful VTOL configuration considering this trade-off [Alamio A., Artale V. et al. , 2013]. Inherent fault tolerance is another advantage of hexacopter configuration. For these reasons, control study for a hexacopter platform is focused in this study.

In literature, several methods are used to control hexacopter configuration such as LQR, PID, adaptive control, faults tolerant control approaches, etc. In most of these approaches saturation of control inputs (motor rpm) is not embedded into controller design process and handled by using command limiters or other indirect methods.

In this study, it is aimed to design a controller that directly handles input constraints. Model Predictive Control (MPC) can solve this problem by performing on-line optimization. Due to significant computational effort of on-line optimization, it is hard to apply MPC controller in real-time for air platforms [Eren U., Prach A. et al. , 2017]. However, with increased computational power, there are some promising newly emergent works about real-time implementation of MPC controllers [Kamel M., Alonso-Mora J. et al. , 2017], [Lighart J.A.J., Poksawat P. , 2017]. Hexacopter or quadcopter VTOL platforms are very suitable to test these type of unconventional and risky control approaches due to low experimental costs.

To see the effects of MPC on input constraints and controller performance, a classical LQR controller is also designed for the same problem. Simulations are performed to compare the classical LQR controller with off-line calculated control gain and MPC approach with direct consideration of input constraints via on-line optimization.

Throughout the paper, first, dynamic model of hexacopter is generated in MATLAB/Simulink environment and details are given in the next section. Then, MPC and LQR controller design is given in "**Controller Design**" section. Finally, results are analyzed and MPC & LQR controllers are compared with each other.

HEXACOPTER DYNAMICS

Before diving into equations of motion, control inputs and strategy for a typical hexacopter is introduced.

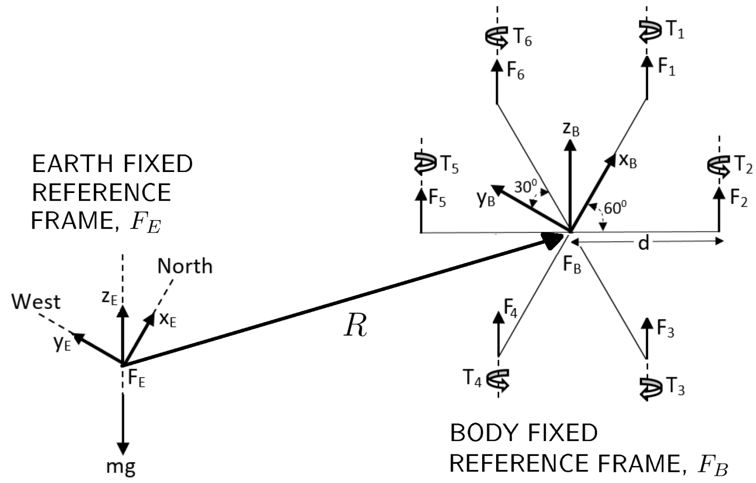

Figure 1: Reference frames, force & torque directions.

Control Inputs & Strategy

Hexacopter configuration has symmetrically distributed six propulsion units(ESC-motor-propeller) as can be seen in Figure 1.

In literature, force and torque generated by small diameter propulsion units are generally estimated as a function of rpm only. In this work, data of AscTech Hummingbird quadcopter [Achtelik M. , 2010] is used as propulsion units and force & torque generated by each propeller system is given as following:

$$F_i = k_f \omega_i^2 \;\;,\;\; T_i = k_t F_i \;\;,\;\; i = 1:6,\;\; k_f = 5.7 \cdot 10^{-8}\; N/rpm^2,\;\; k_t = 0.016\; m \tag{1}$$

In reality, control of hexacopter is achieved by adjusting the rotational speed (rpm) of each motor via ESCs (PWM signals). If there exist a difference in rpm for opposite motors, hexacopter tilts in that direction(θ or ϕ) and moves in horizontal plane(x or y) due to direction change in net thrust vector. To gain/lose altitude(z), rpm of each motor is increased/decreased by the same amount. To rotate hexacopter in its own z_b axes (yaw motion), nonzero net torque is required which can be obtained by small changes in rpm.

To summarize, by controlling Euler angles and altitude, it is possible to guide hexacopter on the surface of the Earth. To simplify the system, a virtual control input vector is defined to use in the design of optimal controllers and also a physically realizable control input is defined to simulate real actuator dynamics.

$$U_v = [Fz_c, Mx_c, My_c, Mz_c]^T \;\;,\;\; U_r = [F_1, F_2, F_3, F_4, F_5, F_6]^T \tag{2}$$

Since the physically realizable control inputs are angular velocity of each motor (ω_i), a mapping is required between physically realizable and virtual control inputs. Following Equations are used to perform this mapping:

$$U_v = TU_r = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -0.87d & -0.87d & 0 & 0.87d & 0.87d \\ -d & -0.5d & 0.5d & d & 0.5d & -0.5d \\ -k_t & k_t & -k_t & k_t & -k_t & k_t \end{bmatrix} U_r \;,\; U_r = \bar{T} U_v = \begin{bmatrix} 0.167 & 0 & -0.333d & -0.167k_t \\ 0.167 & -1.443 & -0.167d & 0.167k_t \\ 0.167 & -1.443 & 0.167d & -0.167k_t \\ 0.167 & 0 & 0.333d & 0.167k_t \\ 0.167 & 1.443 & 0.167d & -0.167k_t \\ 0.167 & 1.443 & -0.167d & 0.167k_t \end{bmatrix} U_v \tag{3}$$

Since transformation matrix T is not square, pseudo-inverse is used to find $\bar{T}$ [Omari S. Hua M.D. et al. , 2013].

6-DOF Equations of Motion

Once control inputs are defined, dynamics of the hexacopter can be generated by considering the external forces and moments acting on the hexacopter cg. Equations of motion in six degrees of freedom is expressed in body fixed frame as following [Suicmez E.C. and Kutay A.T. , 2017]:

$$\left.\begin{aligned} \sum F_{ext} = F_g + F_p + F_d = m\dot{V} + \omega \times (mV) \\ \sum M_{ext} = M_p = J\dot{\omega} + \omega \times (J\omega) \\ F_g = -L_{BE}\begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \; F_p = \begin{bmatrix} 0 \\ 0 \\ Fz_c \end{bmatrix}, \; F_d = -diag(k_{dh},\ k_{dh},\ k_{dv})V \ , \; M_p = \begin{bmatrix} Mx_c \\ My_c \\ Mz_c \end{bmatrix}, \end{aligned}\right\} \quad (4)$$

Following kinematic relations are used to obtaion position and orientation of hexacopter wrt Earth surface. Transformation matrices can be obtained from [Suicmez E.C. and Kutay A.T. , 2017].

$$\left.\begin{aligned} \dot{R} = L_{EB}V \ , \ V = L_{BE}\dot{R} \ , \ L_{BE} = L_{EB}^{-1} = L_{EB}^{T} \\ \dot{\eta} = L_R^{-1}\omega \ , \ \omega = L_R\dot{\eta} \end{aligned}\right\} \quad (5)$$

By combining Equations (4) and (5), translational and rotational dynamics of hexacopter is obtained as following:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -qw + vr + gs(\theta) - k_{dh}u/m \\ pw - ur - gc(\theta)s(\phi) - k_{dh}v/m \\ -pv + uq - gc(\theta)c(\phi) - k_{dv}w/m + Fz_c/m \end{bmatrix} \ , \ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} (qr(Iy - Iz) + Mx_c)/Ix \\ (pr(Iz - Ix) + My_c)/Iy \\ (pq(Ix - Iy) + Mz_c)/Iz \end{bmatrix} \quad (6)$$

CONTROLLER DESIGN

LQR and MPC controllers are designed by using the same linearized hexacopter model. Before going into controller designs, a simplified linearized model of hexacopter is obtained in the following section.

Since the aim of the study is comparison between LQR and MPC controllers, the details of LQR controller and linearized hexacopter dynamics are not given and reader can refer to [Suicmez E.C. and Kutay A.T. , 2017] for details.

Linearized Hexacopter Model

Our aim is controlling altitude(z) and Euler angles(ϕ, θ, ψ). To reduce control tuning effort, derivative and integral of states are also added to the system dynamics. Linearization is performed at hover condition. By using the nonlinear dynamic model obtained in Equations (5) and (6), following linearized model is obtained:

$$\left.\begin{aligned} X = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, z, \dot{z}, \textstyle\int\phi, \int\theta, \int\psi, \int z]^T, \ \ X_0 = zeros(12,1), \ \ U_0 = [mg, 0, 0, 0]^T \\ \Delta\dot{X} = A\Delta X + B\Delta U, \ \ Y = C\Delta X \\ \Delta\dot{X} = \dot{X} - \dot{X}_0 = \dot{X}, \quad \Delta X = X - X_0 = X, \quad \Delta U = U - U_0 \end{aligned}\right\} \quad (7)$$

Linearization is performed by using "linearize" command of MATLAB and time-invariant A, B, C matrices are found as following:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.375 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 555.56 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 555.56 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 312.5 \\ 0 & 0 & 0 & 0 \\ 1.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad C = I_{12\times12} \tag{8}$$

LQR Controller

First, conventional LQR controller is designed by using the linearized model obtained in Equations (7) and (8). LQR gain is calculated off-line by solving Algebraic Riccati Equation which does not include input constraints. It is possible to use a low gain to avoid actuator saturation but this will degrade the performance of the controller for specific cases.

To achieve desired transient characteristics a systematic iterative approach is used to tune LQR weights (Q and R matrices). Details of the tuning process are given in [Suicmez E.C. and Kutay A.T. , 2017]. The same strategy is used and following LQR gain (K_{lqr}) is found by using MATLAB "lqr" command.

$$\left.\begin{array}{r} Q = diag[5, 0.12, 5, 0.12, 10, 0.384, 4000, 100, 0.5, 0.5, 1, 500], \quad R = diag[100, 1000, 1000, 1000] \\ \\ K_{lqr} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 7.4950 & 3.3169 & 0 & 0 & 0 & 2.2361 \\ 0.0767 & 0.0199 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0224 & 0 & 0 & 0 \\ 0 & 0 & 0.0767 & 0.0199 & 0 & 0 & 0 & 0 & 0 & 0.0224 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1099 & 0.0330 & 0 & 0 & 0 & 0 & 0.0316 & 0 \end{bmatrix} \\ \\ with \;\; \Delta U = -K_{lqr}(X - X_d), \;\; X_d : Desired\ state\ vector \end{array}\right\} \tag{9}$$

LQR controller is designed off-line and does not handle actuator constraints directly. Instead, it is common to use command generators to smooth reference inputs to avoid actuator limits and testing the controller in nonlinear simulations with actuator dynamics. A similar approach is used and details of simulation environment is given in Section **"Comparison between LQR and MPC"**.

At this point, MPC controller is expected to produce actuator outputs that are in predefined limits by introducing input constraints and minimizing the cost function on-line with predicted states and inputs. Design of MPC controller is given in the following section.

MPC Controller

In MPC, finite time optimization problem is solved for a predetermined horizon by using predicted states and inputs for each time step. Once the optimal control input vector is found for entire horizon, only the first input is applied at that time step[Wang L. , 2009]. In MPC, state and input constraints are introduced into the cost function directly and optimization is performed considering the constraints over the prediction horizon. Since MPC is an on-line optimization approach, convexity is very important to obtain solution easily [Eren U., Prach A. et al. , 2017].

In this study, to compare results of LQR reasonably, a linear MPC structure is used. Linear Quadratic-MPC(LQ-MPC) formulation defined in [Suzuki Y., Dunham W. et al. , 2019] is used by using only Q and R matrices in the cost function. MPC predicts the future states by using the linearized system dynamics given in Equation (7). In nonlinear simulations, current state measurements are used to estimate the future states by using linear dynamics. It is also noted that cost function of both LQR and MPC have the same Q and R matrices.

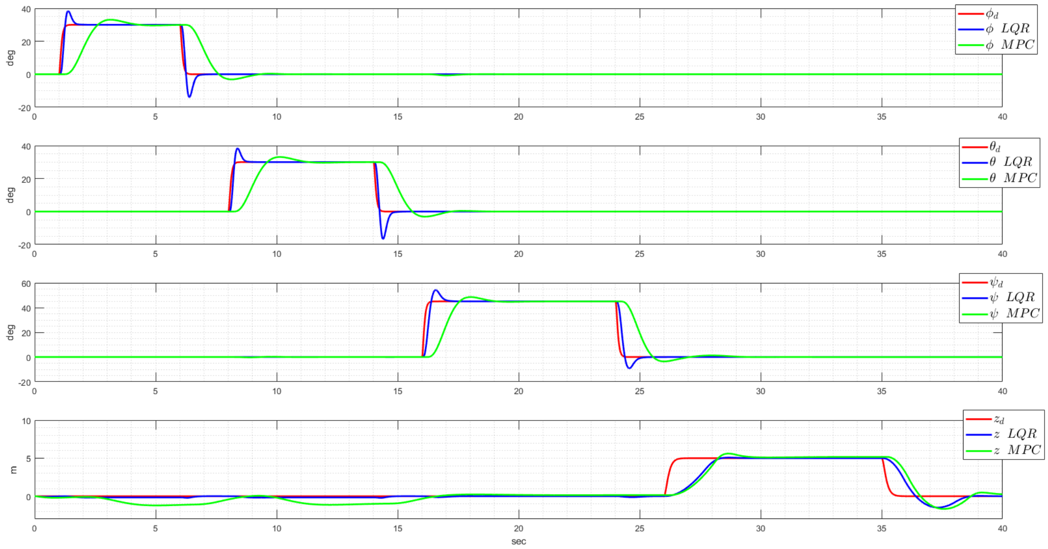

Figure 2: LQR vs MPC controller step input responses for states ϕ, θ, ψ and z.

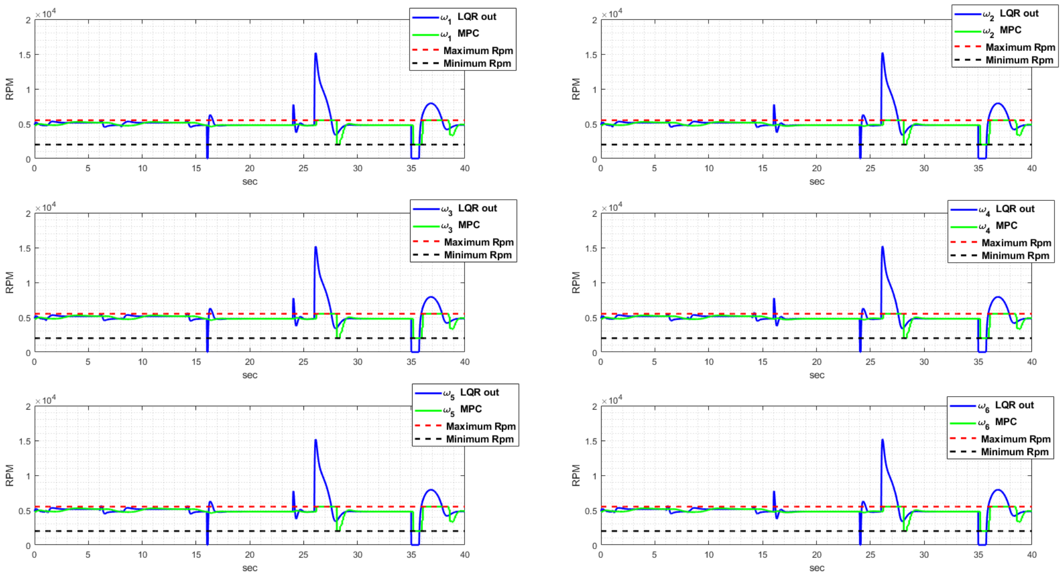

Figure 3: LQR vs MPC controller rpm outputs and actuator rpm limits.

MATLAB has "MPC Designer" toolbox ("mpcDesigner" command) for linear constraint MPC design problems. One can obtain the MPC controller in script or simulink model form by supplying the linearized system dynamics, weights of the LQ cost function (Q and R matrices), input and output constraints, prediction & control horizons and sample time. To have fast MPC solution, sample time is selected as 0.1, prediction horizon is chosen as 12 and control horizon is used as 3. State estimation and LQR controller works at 100 hz for our simulation, so that MPC controller has slower sampling rate for faster on-line optimization. Although MPC algorithm runs slower than the main simulation, results show that MPC could control the hexacopter successfully.

Actuator dynamics are limited at maximum of 5500 rpm and there is also a minimum limit of 2000 rpm that is desirable not to go below. Since our control variables are net force and moments defined in Equation (2), rpm limits are expressed as a net force (Fz_c) by using the relations given in Equation (1). These maximum and minimum net force limits are defined as input constraints in optimization.

Comparison between LQR and MPC

To test and compare LQR and linear MPC controllers, a nonlinear simulation model built in Simulink is used. The details of the simulation model are given in [Suicmez E.C. and Kutay A.T. , 2017]. In simulations, aim is tracking 30 degree commands for roll and pitch angles, 50 degree for yaw angle and 5 m. altitude command. Figure 2 shows the results of LQR and MPC controllers. For the same scenario, rpm values generated by LQR and MPC controllers are also given in Figure 3. It is seen that LQR controller generates commands that are out of actuator limits, whereas MPC controller outputs are within the limits. It is noted that in nonlinear simulations LQR outputs are saturated to actuator limits;however, due to the predictive nature of MPC, MPC controller has a more smooth behaviour according to Figure 2. The advantage of handling the actuator constraints on-line can be seen while analyzing roll and pitch dynamics. In Figure 2, the overshoot for roll and pitch dynamics are not symmetric for LQR controller since actuator limits are not symmetric according to trim values. On the other hand, MPC controller handles actuator constraints on-line in a predictive manner so that the control inputs are generated to avoid these type of undesired behaviours.

It can be concluded that LQ-MPC approach could provide feasible solution around linear region if the optimization problem is convex. Apart from input constraints, output constrains such as maximum angle deviations could be also included into MPC design for specific problems.

CONCLUSION

In this paper, LQR and a linear MPC approach are compared with each other. To have reasonable comparisons, cost functions are defined the same with identical Q and R matrices. LQR and MPC controllers are generated by using MATLAB environment. Both controllers are tested by using the same nonlinear simulation environment also built in MATLAB/Simulink. In MPC design, actuator constraints are defined as total maximum force and added to on-line optimization problem. Results show that, as expected, MPC controller has better characteristics compared to LQR due to predictive nature of handling actuator constraints. Since MPC have different tuning parameters compared to LQR such as prediction or control horizon, comparisons are not exactly reasonable. However, it can be seen that MPC have advantages compared to LQR for convex optimization problems that can be solved on-line easily with increased technology in computer speed and algorithms.

References

Achtelik M. (2010) *Nonlinear and adaptive control of a quadcopter*, Dipl.-Ing Dissertation, Lehrstuhl fr Flugsystemdynamik, Technische Universitat Munchen , Garching, Germany, 2010

Alamio A., Artale V., Milazzo C., Ricciardello A. and Trefiletti L.(2013) *PID Controller Applied to Hexacopter Flight*, Journal of Intelligent and Robotic Systems, vol.73:261-270, November 2013

Eren U., Prach A., Kocer B.B., Rakovic S.V., Kayacan E., Acikmese B. (2017) *Model predictive control in aerospace systems: Current state and opportunities*, Journal of Guidance, Control, and Dynamics vol.40, no.7, pp.1541-1566, 2017

Suzuki L. (2009) *Failure Detection and Control of Distributed Electric Propulsion Aircraft Engines*, AIAA SciTech Forum, San Diego, California, January 2019

Kamel M., Alonso-Mora J., Siegwart R. and Nieto J. (2017) *Nonlinear Model Predictive Control for Multi-Micro Aerial Vehicle Robust Collision Avoidance*, Autonomous Systems Lab, ETH Zurich, Delft Center for Systems and Control, TU Delft, arXiv, March 2017

Lighart J.A.J., Poksawat P. (2017) *Experimentally Validated Model Predictive Controller for a Hexacopter*, IFAC PapersOnline, 50-1 2017

Omari S., Hua M.D., Ducard G. and Hamel T. (2013) *Hardware and Software Architecture for Nonlinear Control of Multirotor Helicopters*, IEEE/ASME Transactions on Mechatronics, vol.18, no.6,pp.1724-1736, December 2013

Suicmez E.C. and Kutay A.T. (2017) *Attitude and Altitude Tracking of Hexacopter via LQR with Integral Action*, International Conference on Unmanned Aircraft Systems(ICUAS), Miami, FL, USA, June 2017

Wang L. (2009) *Model Predictive Control System Design and Implementation Using MATLAB*, Springer, London, 2009