Embedded Model Predictive Control of Unmanned Micro Aerial Vehicles

Tomas Baca
Faculty of Electrical Engineering
Czech Technical University in Prague
Email: tomas.baca@fel.cvut.cz

Giuseppe Loianno
GRASP Laboratory
University of Pennsylvania
Email: loiannog@seas.upenn.edu

Martin Saska
Faculty of Electrical Engineering
Czech Technical University in Prague
Email: martin.saska@fel.cvut.cz

Abstract—We propose a lightweight embedded system for stabilization and control of Unmanned Aerial Vehicles (UAVs) and particularly Micro Aerial Vehicles (MAVs). The system relies solely on onboard sensors to localize the MAV, which makes it suitable for experiments in GPS-denied environments. The system utilizes predictive controllers to find optimal control actions for the aircraft using only onboard computational resources. To show the practicality of the proposed system, we present several indoor and outdoor experiments with multiple quadrotor helicopters which demonstrate its capability of trajectory tracking and disturbance rejection.

MULTIMEDIA MATERIAL

A video attachment to this work is available at [23].

I. INTRODUCTION

Unmanned aircraft and MAVs in particular have undergone a fast development in the past few years. They have become an efficient tool for various applications, e.g., for inspecting buildings, aerial photography or package delivery, and some of them require the flight control to be automated. There are several platforms available that allow an outdoor autonomous flight (PixHawk, DJI) while relying on GPS or GLONASS system. Utilizing onboard sensors only is necessary in situations where GPS signal is not available or sufficient to control the MAV precisely. Automated flights of MAVs in indoor and outdoor GPS-denied environments bring new challenges and require different approaches than just modifying the current outdoor solutions. Our intention is to build a system capable of flying not only in laboratory conditions, but also in an uncontrolled indoor environment while enabling multi-robotic applications that require precise motion of several MAVs in small relative distances. Indoor multi-robotic flights have been successfully conducted in [1], [2] while relying on the Vicon system\(^1\), while we aim for a solution with a similar performance but with onboard sensors and computations only.

An embedded controller that will utilize the model predictive control (MPC) approach to drive MAVs and support trajectory following with sufficient precision for flying in compact groups is proposed in this paper. The design of the model predictive controller requires a formulation of a mathematical optimization problem, where the decision variables encode the desired control actions over a certain prediction horizon.

\(^1\)Vicon is a motion capture system that uses multiple cameras for capturing the object for its precise localization. See http://vicon.com.

A. Related work

In [6], a non-linear MPC formulation allowing to control several miniature, single-rotor helicopters in a collision-free way is presented. The MPC optimizes over a 2 s prediction horizon. However, an external camera localization system as well as a ground computer station are utilized.

ETH University in Zurich has a long tradition of research in the field of MAV control. Their last published MPC solution [7] also relies on an external localization system as well as an external PC ground station.

The current state-of-the-art in onboard MPC trajectory tracking was described in [3], [4]. In these works, an explicit formulation of MPC is formulated with integrated collision avoidance directly in the controller. Our solution allows to control the MAV with a much longer prediction horizon (200 steps, in comparison to 8 steps in [3], [4]) and therefore it supports trajectory tracking with significantly smaller position error and higher stability and robustness as shown in presented experimental comparison in Sec. VI-C. Our method is a first solution of onboard MPC designed for MAVs that provides sufficient length of prediction horizon for following arbitrary trajectories and for achieving control performance required by the compact multi-MAV systems.

Almost all of the presented works were implemented using a single MPC approach to control the vehicle together with creating the optimal trajectory. We aim to decompose these problems. This paper presents the MPC controller, while our other related work describes trajectory planning [8], [12].

From the available literature, a purely embedded implementation of MPC can be seen only in [3], [4], but our proposed solution surpasses their capabilities of trajectory tracking by having a significantly longer prediction horizon. In contrast with [3], [4], our system allows following complex trajectories with decreased position error and increased stability. The works [6], [7] utilize an external localization system or an external computational power which limits the usage of such systems to laboratory conditions.

B. Contribution

To summarize the introduction, we present a control system for MAVs that allows the execution of quadratic MPC onboard the aircraft. It consists of a single electronic board equipped with two microcontrollers, a telemetry module and an SD-card data logger. The system has been tested while relying
on relative visual odometry and relative visual localization of neighbouring MAVs [9], [10], but it allows connecting various localization sensors. The proposed solution allows to control the MAV with 200 steps prediction horizon and to estimate and reject system disturbances using the Kalman filter.

The system is able to track the desired trajectories with errors in the order of centimeters when flying in indoor and outdoor environments. Various experiments have shown that the achieved results are comparable to state-of-the-art solutions that require external localization systems and external computational power. Our solution can be used independently with very small MAVs (5 inch propellers) to allow flying in compact formations in GPS-denied environments using only onboard sensors.

II. PRELIMINARIES

A. Notation

Vectors and matrices are denoted by bold lowercase and uppercase letters. Underlined vectors, e.g. $\vec{x}$, denote vectors created as $\{x_1^T, x_2^T, ..., x_n^T\}^T$. Furthermore, $x(t)$ denotes $x$ at the sample time $t$ and $x^{(W)}$ denotes $x$ expressed in coordinate system $W$.

B. System dynamics

Let us consider a small multirotor aircraft (MAV) actuated by varying speed of its fixed-pitch propellers. We simplify the model to a single rigid-body description, mostly because of small fixed-pitch propellers and a rigid airframe. Due to the presence of off-the-shelf attitude stabilization we model the MAV with the inner feedback loop already closed. By doing that, we move from a system actuated by thrust of its propellers to a system, where the inputs are the desired pitch ($\theta_D$), roll ($\phi_D$), yaw rate ($\psi_D$), and collective thrust ($U_D$). It is assumed that such system can be treated as a decoupled one [5].

Several coordinate systems are used to express states of the MAV (see Fig. 1). The first one is the world coordinate system $W$ with a fixed position in the world. Then there is a rotating world coordinate system $R$. It is rotated from $W$ around axis $W_z$ by angle $\phi$. The attitude angles $\theta$ and $\psi$ are measured in the inertial frame $I$, which is translated into the centre of gravity of the MAV. In the body frame $B$, axes are aligned with the mechanical frame of the MAV.

Assuming a complete decoupling of the system, the lateral dynamics can be modeled using Newton’s 2nd law of motion and expressed by the following equations:

$$\begin{align*}
\ddot{x}^{(W)} &= \frac{U}{m}(\sin \psi \cos \phi + \sin \theta \sin \phi), \\
\ddot{y}^{(W)} &= \frac{U}{m}(\sin \theta \cos \phi + \sin \psi \sin \phi),
\end{align*}$$

(1)

where $U$ is the total thrust force action on the center of gravity in the direction of the $B_z$ axis, $m$ is the mass of the MAV and $\phi$, $\theta$, $\psi$ are yaw, pitch and roll angles. It is assumed that the desired operating point is a hovering state, where $U$, $m$ are constants and the absolute values of $\theta$, $\psi$ are small. Further the control problem will be described using the coordinate system $R$, thus the position is not longer a function of $\phi$. We can then linearize Eq. 1 by substituting the first order term of their Taylor expansion:

$$\begin{align*}
\ddot{x}^{(R)} &= K_1 \dot{\psi}, \\
\ddot{y}^{(R)} &= K_1 \dot{\theta},
\end{align*}$$

(2)

where $K_1$ is a constant that needs to be identified experimentally. Using this model, the acceleration of the MAV is directly proportional to its attitude angle. The altitude and yaw dynamics are stabilized by the onboard stabilization and controlled separately by PID controllers. Their modelling and control design are out of the scope of this paper. Both decoupled attitude subsystems, also stabilized by the onboard stabilization, are modeled as a first order transfer function.

C. State space representation

The discrete formulation of the dynamical system is used. It allows to design a proper filtration method and the MPC controller itself. From now, all differential equations and state space formulation are written in a discrete form with a constant sampling rate $1/\Delta t$. Let us have an LTI system

$$x_{t+1} = Ax_t + Bu_t,$$

(3)

where $x_{t} \in \mathbb{R}^n$ is the state vector and $u_{t} \in \mathbb{R}^k$ is the input vector in the sample $t$. We assume that $C = I$, $D = 0$. Due to decoupling if the pitch and roll axes, the following $A$ and $B$ matrices describe both systems, respectively:

$$A = \begin{bmatrix}
1 & \Delta t & 0 & 0 & 0 \\
0 & 1 & \Delta t & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & P_1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix},
B = \begin{bmatrix}
0 \\
0 \\
0 \\
P_0 \\
0
\end{bmatrix}.$$

(4)

The modelling and control is further presented for the single axis only, since both are modelled and controlled identically. State vector is defined as $x_{t} = [x, \dot{x}, \ddot{x}, \dot{\psi}, \ddot{\psi}]^T$ and input $u_{t}$ is the desired attitude angle. States $x$, $\dot{x}$ and $\ddot{x}$ denote position, velocity and acceleration in $R_z$ axis, respectively. To allow disturbance estimation, we introduce two states $\ddot{x}_u$ and $\ddot{x}_d$. State $\ddot{x}_u$ is the acceleration caused by the control action and $\ddot{x}_d$ is the acceleration caused by an unknown force, which is considered as the disturbance. Parameters $P_0$ and $P_1$ are
the free parameters of the model which has been identified empirically. See Fig. 2 for the diagram of the LTI system.

All states of the model in Eq. 4 are being estimated by the standard linear Kalman filter, as described in [11].

III. MODEL PREDICTIVE CONTROL

Model predictive control is a control technique that solves finding of control signals as a mathematical optimization problem. Although we have already implemented the MPC for offline trajectory planning on an external PC in our previous work on multi-MAV control and group stabilization [8], [12], [13], [14], it’s realization for real-time control is a challenging task when aiming for onboard implementation. Our solution, which allows to realize the MPC onboard on a microcontroller, is built upon minimizing a quadratic function subject to box variable constraints. The function is constructed and minimized repeatedly while comprising the dynamic model of the MAV and the desired trajectory. The computational complexity of such Quadratic Programming (QP) depends on the used set of constraints and the convexity of the objective. Our approach allows to find the global optimum by a closed-form solution while providing optimal control with respect to the desired, feasible trajectory.

A. System prediction

Let us consider a discrete, linear, time-invariant system with $n$ states, $k$ inputs and a horizon of length $m$ in the form of Eq. 3. The prediction of future states $\tilde{x} = (x_{t+1}^T, x_{t+2}^T, ..., x_{t+m}^T)^T$ is expressed in matrix form as

$$\tilde{x} = \hat{A} x_{[0]} + \hat{B} u.$$ (5)

where $x_{[0]}$ denotes the initial state and control inputs are denoted as $u = (u_{[0]}^T, u_{[1]}^T, ..., u_{[m-1]}^T)^T$. Matrices $\hat{A}$ and $\hat{B}$ are basic building blocks of the MPC and are created as

$$\hat{A} = \begin{bmatrix} A & A^2 & \cdots & A^{m-1} \\ \vdots & & & \vdots \\ A & & & \end{bmatrix}, \hat{B} = \begin{bmatrix} B & 0 & 0 & 0 \\ AB & B & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{m-2}B & A^{m-3}B & \cdots & B \end{bmatrix}. \quad (6)$$

B. Objective function

The objective function for QMPC (Quadratic MPC) is formulated as the sum of squares of weighted control errors $\nabla$ combined with weighted control actions as:

$$V(\hat{x}, u) = \frac{1}{2} \sum_{i=1}^{m-1} \left( e_{[0]}^T Q e_{[i]} + u_{[i]}^T P u_{[i]} \right) + \frac{1}{2} e_{[m]}^T S e_{[m]}.$$

(7)

The control error is denoted by $e_{[i]} = x_{[i]} - \hat{x}_{[i]}$, where $\hat{x}_{[i]}$ is a point of the reference trajectory, $Q \in \mathbb{R}^{n \times n}$ is the state weighting matrix, $P \in \mathbb{R}^{k \times k}$ is the input weighting matrix, and $S \in \mathbb{R}^{m \times m}$ is the matrix weighting the final state values. Matrices $Q$, $S$ need to be positive-semidefinite ($Q, S \succeq 0$) and matrix $P$ needs to be positive-definite ($P > 0$) to ensure that the function $V(\hat{x}, u)$ is strictly convex. Moreover, elements of $\hat{x}$ and $u$ need to satisfy the system dynamics (Eq. 3).

Furthermore by inducing Eq. 5 into Eq. 7, the objective function can be rewritten into the matrix form

$$J(u) = \frac{1}{2} u^T \left( \hat{B}^T \hat{Q} \hat{B} + \hat{P} \right) u + \hat{Q} \hat{B}^T (\hat{A} x_{[0]} - \hat{x}) u.$$ (8)

where $\hat{x} = (x_{[1]}^T, x_{[2]}^T, ..., x_{[m]}^T)^T$ is the reference trajectory for all states consisting of a state vector for each step of the prediction horizon. $Q \in \mathbb{R}^{m \times m}$ and $P \in \mathbb{R}^{km \times km}$ are weighting matrices, created from $Q$, $P$ and $S$. Matrices $\hat{A} \in \mathbb{R}^{km \times km}$ and $\hat{H} \in \mathbb{R}^{km \times km}$ then define the quadratic form.

Finally, the optimization task can be formulated as minimizing the objective function $J(u)$ subject to constraints on $u$

$$\min_{u \in \mathbb{R}^{km}} J(u) = \frac{1}{2} u^T \hat{H} u + \tilde{c}^T u$$

s.t. constraints on $u$.

This problem is solved repeatedly for new $x_{[0]}$ in each control step while only the first component of the input vector is used.

C. Solving unconstrained MPC

The control signals are obtained by finding the optimum $u^*$ of the objective function. Assuming a convex function in our task, we can find a global minimum by translating the function into a quadratic form. Since quadratic forms with semidefinite matrices $\hat{H}$ have minimum in the origin, the vector $u^*$ of the translation between the original function and the quadratic form is the optimum. The translation can be found in a closed-form as

$$u^* = -\hat{H}^{-1} \tilde{c}.$$ (10)

There is an important phenomenon that needs to be taken into account. A condition number of $\hat{H}$ determines whether the optimum of the objective is sharply defined or whether it lies within a wide plateau. Optimizing it using a closed-form can be affected by numerical instabilities when calculating the matrix inversion $\hat{H}^{-1}$. We can regularize the matrix $\hat{H}$ by increasing penalization $P$ since it directly increases values on its diagonal and thus increases its eigenvalues (supposing $P > 0$).
D. QMPC with input constraints

Input box constraints are a special case of general linear inequalities of QP. They can be modeled by a set of inequalities taking the form

\[ \mathbf{b} \leq \mathbf{u} \leq \mathbf{g}, \]  

(11)

where \( \mathbf{b} \in \mathbb{R}^{km} \) and \( \mathbf{g} \in \mathbb{R}^{km} \) are constraint vectors denoting the lower and upper bound on inputs, respectively. They create a convex set of feasible solutions. When the global optimum of a convex quadratic form is not a feasible solution, the actual optimum satisfying the constraints is located on the facet of the convex polytope of the feasible set [17]. One can find it by projecting the unconstrained optimum orthogonally on the feasible set and then taking steps along the facet in the direction of \( -\nabla J \). In the case of box constraints imposed on decision variables, the projection can be done by using the median function as follows:

\[ \hat{\mathbf{u}}^* \leftarrow \text{median}(\mathbf{b}, \mathbf{u}^*, \mathbf{g}), \forall i = 0, ..., km, \]  

(12)

where \( \mathbf{u}^* \) is the global unconstrained optimum and \( \hat{\mathbf{u}}^* \) is the projected constrained optimum. Optimizing the quadratic form in Eq. 9 with input constraints by applying Eq. 12 leads to a feasible solution. For getting the global constrained optimum, one could utilize a gradient descend algorithm to walk along the facet on which \( \hat{\mathbf{u}}^* \) resides. Nevertheless, the controller can be easily tuned to produce outputs that are not saturated and therefore the tool for correcting the infeasible results of QMPC is implemented as a safety mechanism only.

E. Reducing the complexity by move blocking

Until now, we have considered that each decision variable of the optimization task directly corresponds to a value of input signal in a particular time of the prediction horizon. The move blocking technique [19] allows us to project a smaller number of variables to cover a longer prediction horizon. The transformation is denoted as follows:

\[ \mathbf{u} = \mathbf{Uu}_*, \]  

(13)

where \( \mathbf{U} \in \{0,1\}^{km \times kl} \), \( l \in \mathbb{N} \) is the number of decision variables and \( \mathbf{u}_* \in \mathbb{R}^{kl} \) is the reduced input vector. The MPC is then modified by creating new matrix \( \mathbf{B}_* = \mathbf{BU} \) to solve the optimization with, and then simply project the variables on the whole horizon. An example matrix \( \mathbf{U} \) for \( l = 6 \) decision variables, \( k = 1 \) inputs and \( n = 12 \) horizon length is

\[ \mathbf{U}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \]  

(14)

where the decision variables are unevenly distributed to control the actual control inputs. By distributing the variables exponentially we benefit from densely packed control signals in the beginning of the horizon while still having some control ability near the end of the horizon. Thus we impose the control criterion for a much longer time in the future, which improves the stability of the system.

IV. EXPERIMENTAL PLATFORM

The experimental platforms are multirotor aircraft equipped with off-the-shelf attitude stabilization (KK2.1.5), a custom control board, and the px4flow sensor [16] combining an optical flow camera and an ultrasonic rangefinder. The custom control board (Fig. 3) was designed in our laboratory specifically for the purposes of MAV control. It supports a variety of communication interfaces, wireless XBee communication, an SD-card data-logger and separate microcontrollers (ATXmega128a3u and STM32F415) for communication and the MPC controller. The platform is suited for further expansion by adding additional sensors, as is for example the vision relative localization [9], [10] being used for multi-MAV experiments.

V. IMPLEMENTATION

The system setup comprises of two separate MPC controllers driving both lateral axes. Because both lateral axes are controlled identically (due to decoupled system description), the following chapters contain figures for a single axis only without loss of generality.

When implementing the input constrained MPC into embedded hardware, the structure of the problem can be exploited in the following way. Since matrix \( \mathbf{H} \) does not depend on the desired trajectory nor the initial condition, it can be precomputed offline. This also holds for its inversion \( \mathbf{H}^{-1} \) and matrices \( \mathbf{B} \) and \( \mathbf{A} \). Thanks to that we can store them in a ROM (read-only memory, designated for the program), which supports execution on microcontrollers with a small amount of RAM. The workflow of testing and tuning different parameters of the controller was built upon a simulation in Matlab. One part of the simulation is a script that can generate C code with all matrices precomputed, that can be easily inserted into sources for the microcontroller. An open-source implementation together with the board schematics can be found on http://mrs.felk.cvut.cz/software to enable easy reproduction of our solution with an arbitrary platform.
In order to provide a trajectory tracking mechanism that does not violate state constraints, two requirements have to be considered. Firstly, the trajectory itself should satisfy the system dynamics and imposed constraints. Secondly, the trajectory should start in the current state of the MAV. The input governor is a system that modifies the desired trajectory in a way that satisfies these requirements. The governor produces a feasible trajectory from the current state of the MAV. Our input governor limits the rate of change of the desired position trajectory by that value while ensuring the trajectory initiates in the current state of the MAV.

The MPC formulation as stated in section III does not provide offset-free tracking. In the case of external disturbances or uncertainty in the model, this would lead to a steady state control error. Several techniques exist to solve the problem. The classical one is a delta input formulation of MPC [15]. Moreover, an additional integral feedback loop could be created around the MPC controller. But since our estimator is able to observe the disturbance within the system, the classical formulation of MPC is in fact able to control the system without the steady state error [18]. Moreover, it has some other useful properties, e.g. no windup issues (like with classical integral feedback) and that the controller (considering its separation from the estimator) can be completely stateless.

We have set the horizon length to 200 steps (2.2 s) with 20 decision variables in the objective. They are distributed exponentially over the prediction horizon. First 10 variables directly correspond to the first 10 control actions. Other 9 variables cover control actions evenly up to the 100th control action. The last variable sets the control action for the last 100 steps. The system accepts desired trajectories as sequences of regularly sampled positions in 2D plane. Therefore, we only penalize the position states by setting the respective elements of matrices Q, S and P (consequently merged to \( \bar{Q}, \bar{P} \)). This left us three parameters to set in order to tune the performance of the controller. They were found empirically as

\[
Q_{11} = 1.07, S_{11} = 10, P_{11} = 0.0000025
\]

(15)

for both lateral axes. Nevertheless, the proposed system is robust to parameter setting. The same values can be used across different platforms, as shown in experiments with various MAVs in our fleet (Fig. 4). Only the dynamical model needs to be updated for each of them.

VI. Experimental Evaluation

Presented experiments focus on verification of properties of the system such as stability, trajectory tracking and state estimation (position drift and disturbance rejection).

A. Onboard localization

The onboard state estimation and disturbance rejection was tested in the GRASP laboratory at the University of Pennsylvania using the Qualisys localization system [22] as a ground truth. Fig. 5b shows how the onboard estimate drifts during the flight, where the MAV is tracking a trajectory using the proposed system. In general, under bad lighting conditions or if the optical flow sensor is saturated, the drift of the position estimate is larger. Our observations correspond with the results presented by the authors of the px4flow system [16].

B. Trajectory tracking

The system was put into use on various MAV platforms of MRS lab. at CTU in Prague [21]. Its trajectory tracking capabilities have been verified in a variety of conditions including indoor and outdoor environments, natural and artificial light and with different payloads. Statistically, the MAVs were tracking trajectories with a standard deviation of 13 cm. The statistical set consists of 61 minutes of flight data gathered over 12 months of development, testing and usage of the system. The dataset was selected with aim to provide sufficiently diverse environment types and scenarios. The system surpasses performance of our prior solution that utilized PID controllers. Although PID controllers performed well by means of way-point following, they are outperformed by the MPC at outdoor trajectory tracking as depicted in Fig. 4d and 5d. The system performance can be seen in a video compilation (Fig. 4c) [23], where one of the scenarios is tracking a trajectory as shown in Fig. 5a.

C. Stability and horizon length

The horizon length \( m \) has a key role in making the MPC controller asymptotically stabilizing. As described in [20], such property can be verified by solving linear matrix inequalities. It can be shown that too short prediction horizon can lead to a destabilization of the system. Fig. 5c shows how the system responded to external disturbance (the MAV was pushed) while having only 0.8 s prediction horizon\(^2\), although its trajectory tracking capabilities were comparable to 2.2 s horizon. Although the disturbance estimator and the MPC could be tuned to dampen such oscillations, it would significantly sacrifice the tracking performance.

VII. Conclusion

We have proposed a software and hardware solution that allows the execution of the model predictive controller onboard of micro aerial vehicles. We have designed a system with a disturbance estimator and a quadratic MPC, which optimizes control actions over the prediction horizon of 2.2 s which has crucial impact to system stability. The controller has been tested in various indoor and outdoor scenarios.

Acknowledgments

The work has been supported by contract with Khalifa University of Science via HS no. 13167/830/8301616C.

References


\(^2\)Note that the state-of-the-art works on embedded MPC control of MAVs [3], [4] enable only \( m = 8 \), which results to the same horizon length, even with longer sampling time (0.1 s) in comparison to our system (0.011 s).
Fig. 4: Photos from experiments: (a) formation of multiple MAVs flying indoors, (b) three MAVs maintaining a formation while localizing a ground robot, (c) and (d) examples of verification experiments shown in videos at [23].

![MAV tracking sine trajectory](image)

(a) MAV tracking trajectory. Data obtained via onboard estimator.

![Localization drift](image)

(b) Demonstration of the drift of the onboard localization if using only the optical flow. Ground truth supplied by Qualisys [22].

![Inducing oscillations](image)

(c) Oscillations induced into the system with 0.8 s prediction horizon.

![MPC and PID comparison](image)

(d) Comparison of the MPC and previously used PID controller.

Fig. 5: Data obtained in real experiments in indoor and outdoor environments.