Online Planning for Energy-efficient and Disturbance-aware UAV Operations

Nicola Bezzo, Kartik Mohta, Cameron Nowzari, Insup Lee, Vijay Kumar, George Pappas

***Abstract*— In this paper we consider an online planning problem for unmanned aerial vehicle (UAV) operations. Specifically, a UAV has the task of reaching a goal from a set of possible goals while minimizing the amount of energy required. Due to unforeseen disturbances, it is possible that initially attractive goals might end up being very expensive during the execution. Thus, two main problems are investigated here: i) how to predict and plan the motion of the UAV at run time to minimize its energy consumption and ii) when to schedule next replanning time to avoid unnecessary periodic re-evaluation executions. Our approach considers a nonlinear model of the system for which a model predictive controller is used to determine the desired control inputs for each possible goal. These control inputs are then used to estimate the energy required to reach the different goals. Finally, a self-triggered scheduling policy determines how long to wait before replanning the goal to aim for. The proposed framework is validated through simulations and experiments in which a quadrotor must choose and reach some goal while being subject to external disturbances.**

I. INTRODUCTION

In recent years we have witnessed an increasing interest in using unmanned aerial vehicles (UAVs) for both military and civilian operations. UAVs have demonstrated great technological advantages and the myriad ways in which they can be applied are only increasing.

Even though the control performance and operation of these systems has become very sophisticated, a big challenge still remains on how to guarantee at run time that they can safely coordinate and achieve a desired goal (e.g., reaching a target and return to a base station). In fact, most control strategies usually assume one of the following limiting conditions: either i) there are no disturbances and the system is not compromised, operating in perfect conditions (e.g., battery fully charged, no environment disturbances) or ii) worst-case scenarios are considered (e.g., maximum energy consumption, strong environment disturbance), which in most cases is an unnecessary over-constrained condition that results in bad performance and underuse of resources. To overcome these limitations and leverage the full power and capabilities of these aerial vehicles, in this work we introduce a generalizable framework for online prediction and planning that takes into consideration the dynamics of these UAVs to guarantee both *safety* (i.e., something "bad" will never happen) and *liveness* (i.e., something "good" will eventually happen).

To this end, we use a model-predictive approach paired with self-triggered scheduling techniques to: i) estimate

N. Bezzo, C. Nowzari , G. Pappas, and I. Lee are with the PRECISE Center, University of Pennsylvania, Philadelphia, PA 19104, USA {nicbezzo, cnowzari, pappasg, lee}@seas.upenn.edu
K. Mohta and V. Kumar are with the GRASP Lab., University of Pennsylvania, Philadelphia, PA 19104, USA {kmohta, kumar}@seas.upenn.edu

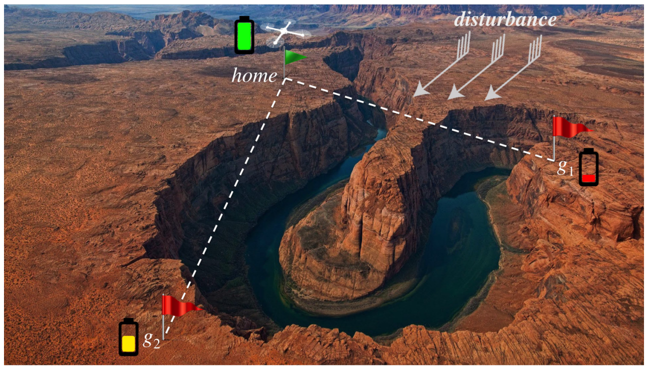


Fig. 1. Motivational cartoon representation of the case study used in this work. Due to the disturbance, a UAV might consume less energy to reach goal g_2 even though g_1 is closer.

safety critical states, ii) predict the future steps of the system, iii) plan and replan its motion, and iv) schedule next replanning time. We will refer to this procedure in short as *online planning*.

The specific case study considered in this work is a single UAV with a set of possible unprioritized goals. However, we are interested in seeking out the goal for which the total energy consumption is lowest, which depends on the unknown disturbances in the system. For instance, as shown in Fig. 1, wind might help the UAV reach a goal that is farther away while consuming less energy than a closer goal. The problem is that, since the disturbances are causal, we might incorrectly decide on the best goal to aim for. In general, this problem requires constant replanning to ensure that the goal the UAV is aimed at is the best one. Rather than making this decision periodically, we propose the self-triggered and self/event-triggered scheduling policies that determine, based on currently available information, at what time in the future the UAV will replan its motion.

A. Related Work

The study of unmanned aerial vehicles and in particular quadrotors has been gaining a lot of popularity among the robotics and control community due to their robustness and simplicity of control. For example, in [1] a quadrotor is used as a wireless relay node to maximize the communication throughput and quality between two sources; in [2] quadrotors are used to optimally carry loads, and in [3] the authors use vision-based control to coordinate and plan the motion for these UAVs. All of the above mentioned works are mostly concerned with motion and path planning problems and assume either ideal or worst case situations. In reality, these systems may be subject to disturbances and other unmodeled constraints that, if not handled properly at run time, could prevent the correct performance and realization of a given objective.

In the literature, we find some works dealing with a few aspects related to this problem of run-time monitoring and planning. [4] offers a comprehensive survey about the state of the art on run-time monitoring and checking and current issues. The authors point out that current challenges lie in how to: i) reduce the checking overhead, ii) select the right properties to monitor, iii) select the right feedback for an external observer, and iv) recover after detecting a violation of a monitored property. Authors in [5] propose a Learning-Based Model Predictive Control (MPC) to learn the ground effect and the unknown trajectory of a ball thrown to a quadrotor that needs to plan its trajectory to catch it. Similarly, in [6] the authors also use a MPC approach to generate state interception trajectories for a quadrotor in real time. Another similar work, [7], proposes a least-squares method to estimate the inertial parameters (mass and center of mass) of a load grasped by a quadrotor, to improve its flight performance. From a recovery point of view, in [8] the authors demonstrate how to stabilize a quadrotor even after a complete loss of up to three of the four propellers. Finally, similar to the work proposed in our paper, authors in [9] deal with the problem of path planning for coverage while finding the path that reduces energy consumption.

In this work we contribute to the current run-time monitoring and online planning literature by: i) developing a complete framework for prediction and planning of a UAV operation to minimize energy consumption and ii) building self-triggered scheduling techniques [10], [11] that reduce the otherwise computationally expensive periodic replanning employed by traditional motion planners. Note that the proposed framework is general for different types of predictive controllers. MPC is particularly attractive in this specific work because its solution is a sequence of future inputs, which are used to control and to calculate the UAV predicted energy dissipation. To the best of our knowledge the proposed scheme has never been investigated especially on non-linear systems like quadrotor UAVs and/or validated with extensive simulations and experiments as presented in this paper. From an implementation point of view, a final contribution is the realization of a fast prediction and replanning algorithm by using CVXGEN [12].

The rest of the paper is organized as follows: in Section II we outline the mathematical notation used in this work. In Section III we formally state the problems, followed by the UAV and disturbance models in Section IV. The online prediction and planning technique and guarantees are presented in Section V. Simulation results for a "quadrotor navigation subject to external disturbance" case study are presented in Section VI. Hardware validation on a real quadrotor is presented in Section VII and finally we draw conclusions in Section VIII.

II. Preliminaries

Through this paper we will denote vectors with bold lower case italic letters (e.g., $\boldsymbol{x}$) while matrices will be expressed with bold upper case italic letters (e.g, $\boldsymbol{A}$). Time varying variables will have the time index k within parenthesis. Since our formulation is in discrete time, $(k+1)$ will denote a sequential increment from k with sampling time t_s (i.e., $t_{k+1} - t_k = t_s$). At times, the use of (k) will be omitted for ease of notation but it will be clear from the context that the formulation is time dependent. A $\wedge$ symbol on top of a variable represents its estimated value (e.g., $\hat{x}$). $\|\cdot\|$ represents the Euclidean norm and finally $\text{unit}(b-a) = \frac{b-a}{\|b-a\|}$ (i.e., the unit vector in the direction of $(b-a)$). A state is denoted by $\boldsymbol{x}$ with generally a subscript to indicate the system it is referring to and we use $\boldsymbol{p} \in \mathbb{R}^3$ to refer to the position part of the state.

III. Problem Statement

In this work we are interested in finding an online planning policy for minimum energy path planning of UAVs. Formally, we consider the following problem:

Problem 3.1: **Online Prediction and Planning:** A UAV has the objective to navigate to one or multiple goals g_j where $j = 1, 2, \ldots, N_g$ with $\boldsymbol{x}_{g_j} \in \boldsymbol{X}_g = [\boldsymbol{x}_{g_1}, \boldsymbol{x}_{g_2}, \ldots, \boldsymbol{x}_{g_{N_g}}]$, while minimizing energy consumption. External disturbance may be present in unknown locations and at unknown times through the operating workspace.

Given the UAV dynamics as a function of its state $\boldsymbol{x}$, input $\boldsymbol{u}$, and disturbance $\boldsymbol{d}$,

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{d}(k))$$

we want to find a policy $\mathcal{P}$ at time t_k to:

1) predict its input and state evolution from the current time t_k up to a future horizon t_{k+h},
2) predict its energy consumption $E_{g_j}(k)$ from its current state $\boldsymbol{x}(k)$ to each goal $\boldsymbol{x}_{g_j} \in \boldsymbol{X}_g$, and
3) select goal $\boldsymbol{x}_{g^*} \in \boldsymbol{X}_g$ such that:

$$\begin{aligned} &E_{g^*}(k) \leq E_{\max}(k) \\ &E_{g^*}(k) \leq E_{g_j}(k) \quad \forall g_j \neq g^* \end{aligned} \tag{1}$$

where $E_{g_j}(k)$ is the predicted energy at time t_k needed by the UAV to reach g_j, and $E_{\max}(k)$ is the maximum energy available at t_k.

Solving Problem 3.1 will allow us to predict and replan the UAV target goal assignment at any time. However, running a prediction frequently and periodically, may be computationally expensive due to the very non-linear nature of the system. This bring us to the following second problem that we investigate in this work:

Problem 3.2: **Online Replan Scheduling:** While solving Problem 3.1, find a policy $\mathcal{T}$ to schedule next replanning time $t_{i+1}^{\text{plan}} = t_i^{\text{plan}} + \tau$, $i \in \mathbb{N}$, with $\tau \geq \alpha t_s$ and α a constant.

Thus, by solving Problems 3.1 and 3.2 it will be possible at run time to plan the operation of the UAV by selecting the goal with minimum predicted energy consumption and schedule next replanning execution time.

IV. System Models

In this section we outline the quadrotor dynamical model, linearization procedure, and disturbance and power models used by the MPC procedure later in Section V.

A. Quadrotor Model

A quadrotor has four rotors with two of them rotating clockwise while the other two rotating counter-clockwise. The thrust and torque produced by each propeller is proportional to the square of its rotational speed [13]. Denoting the rotational speed of each propeller by ω_i and the proportionality constants for thrust and moments by κ_f and κ_m respectively, we have

$$F_i = \kappa_f \omega_i^2, \quad T_i = \kappa_m \omega_i^2, \quad i = 1, \ldots, 4 \tag{2}$$

where F_i is the thrust produced by each propeller while T_i is the magnitude of the moment produced by it.

Assuming the length of the arm of the quadrotor is d, the net thrust and moments on the robot, which we consider as the inputs to the system, are given by,

$$\begin{bmatrix} F \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \kappa_f & \kappa_f & \kappa_f & \kappa_f \\ 0 & d\kappa_f & 0 & -d\kappa_f \\ -d\kappa_f & 0 & d\kappa_f & 0 \\ \kappa_m & -\kappa_m & \kappa_m & -\kappa_m \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

The dynamics of the quadrotor can be described as a non-linear system of the form $\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$ with the following state vector $\boldsymbol{x} = [\boldsymbol{p}^\mathsf{T}\ \phi\ \theta\ \psi\ v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^\mathsf{T}$ where $\boldsymbol{p} = [x\ y\ z]^\mathsf{T}$ is the world frame position, v_x, v_y and v_z are the world frame velocities, ϕ, θ and ψ are the roll, pitch and yaw Euler angles and ω_x, ω_y and ω_z are the body frame angular velocities. $\boldsymbol{u} = [u_1\ u_2\ u_3\ u_4]^\mathsf{T}$ contains the four motor inputs. For ease of discussion here we omit the full state space representation that can be found in [13], [7], [3].

For linearization, we select a linearization point $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ and compute

$$\boldsymbol{A} = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}^*, \boldsymbol{u}^*)} \quad \text{and} \quad \boldsymbol{B} = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}^*, \boldsymbol{u}^*)}$$

Instead of linearizing the system just at the hover state (common procedure), here we linearize at its current operating point so that the linearization is valid even when the robot is not close to the hover state. This give us,

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}^*, \boldsymbol{u}^*) + \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*) + \boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u}^*)$$

B. Disturbance Model

When a quadrotor is flying, there are many factors which can cause it to deviate from the set point. In this work, we consider wind as the main external disturbance acting on the robot. Wind is an important factor to consider because when flying outdoors, wind can cause a big change in the robot's behavior. The effect of wind on small robots has not been studied very well but there are some empirical results [14].

The force on the robot due to wind is approximately proportional to the wind velocity. This leads to an additional term in the dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*) + [\boldsymbol{B} \mid \boldsymbol{I}] \begin{bmatrix} \boldsymbol{u} - \boldsymbol{u}^* \\ f(\boldsymbol{x}^*, \boldsymbol{u}^*) \end{bmatrix} + \boldsymbol{B}_d\, \boldsymbol{d} \qquad (3)$$

where $\boldsymbol{d} = [d_x\ d_y\ d_z]^\mathsf{T}$ is the wind velocity.

In this work we assume that we have an upper bound on the wind velocity magnitude, which gives us a bound for the disturbance. We also assume that the wind changes slowly and its velocity has a bounded rate of change.

The system model in (3) is discretized with a sampling time t_s, and with a slight abuse of notation, we write it as

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{B}_I f(\boldsymbol{x}^*, \boldsymbol{u}^*) + \boldsymbol{B}_d\, \boldsymbol{d} \quad (4)$$

C. Power Model

Since we are concerned with the energy consumption of the robot, we need a model of the power used by the platform. The two components using power on the robot are the motors and the onboard electronics. The power used by the motors when flying is more than an order of magnitude higher than the power used by the electronics, thus we only take into account the motor power in our calculations.

The power used by each motor is equal to

$$P_i = T_i\, \omega_i = \kappa_m \omega_i^3 \qquad i = 1, \ldots, 4$$

where T_i is as defined in (2). Thus, the net energy used by the motors is

$$P = \sum_{i=1}^{4} P_i = k_m \sum_{i=1}^{4} \omega_i^3 \qquad (5)$$

V. Online Prediction & Planning

In this section we describe the framework adopted for online prediction and planning for UAV missions to solve Problems 3.1 and 3.2. Specifically we follow the architecture depicted in the block diagram in Fig. 2.

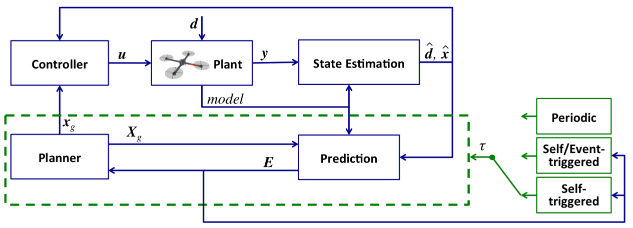


Fig. 2. Overall architecture for online prediction and planing used for UAV operations. Our contribution lies within the green colored sections of the block diagram.

The overall architecture is composed of: i) states and disturbance estimation, ii) energy prediction, and iii) path planning. Both estimation and prediction are model-based: the state estimate uses a model of the system to estimate the disturbance while the predictor makes predictions of states and inputs based on the current states and inputs and uses the worst and best disturbance cases to evaluate the total energy the system will consume over a finite planning horizon and a finite number of trajectories. Finally the planner will choose the trajectory with the minimum energy consumption.

Disturbance estimation can be performed using different techniques, both parametric and non parametric [5]. The literature offers several methods like Kalman filtering, least-square methods, expectation-maximization algorithms, and kernel regressions. In this work, for ease of discussion, we assume that a disturbance estimate is available to the quadrotor at every time.

A. Sequential MPC-based Energy Prediction

In order to solve the prediction Problem 3.1 we use a sequential linearization and model-predictive procedure to increase the accuracy of the prediction. We leverage the well known model predictive control (MPC) theory, [15], over a finite horizon h to predict the future evolution of the states and inputs of the UAV and then deduce the energy consumption based on the sequence of inputs obtained from the MPC calculation. Besides MPC, other methods like trajectory or simulation based predictions are also possible to use. MPC is particularly suitable for such energy prediction problem because its result is a series of inputs that are necessary for energy calculation, as shown in (5), thus giving a more precise prediction than other input independent techniques. Due to the UAV non-linearities, we cannot run the MPC for a long horizon and thus we execute a sequence of linearizations of the system every ℓ MPC steps (with $\ell \leq h$) for N_h times to increase the energy prediction accuracy. Therefore, the overall predictive horizon H becomes $H = N_h\ell$. To improve the execution timing, in the next section we will present an online self-triggered replan scheduling technique [10] that relaxes the periodic execution of the prediction presented in this section.

To predict the energy consumption of the UAV reaching a specific goal g_j, we consider the following optimization

$$\mathcal{J}_j(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{D}(k)) = \min_{\boldsymbol{z}} \sum_{k=0}^{h-1} \boldsymbol{e}_{\boldsymbol{x}}^{\mathsf{T}}(k)\boldsymbol{Q}\,\boldsymbol{e}_{\boldsymbol{x}}(k) + \boldsymbol{e}_{\boldsymbol{u}}^{\mathsf{T}}(k)\boldsymbol{R}\,\boldsymbol{e}_{\boldsymbol{u}}(k) + \Delta\boldsymbol{u}^{\mathsf{T}}(k)\boldsymbol{R}_{\Delta\boldsymbol{u}}\,\Delta\boldsymbol{u}(k) \tag{6}$$

$$\text{subject to} \quad \boldsymbol{u}_{\min,i} \leq \boldsymbol{u}_i \leq \boldsymbol{u}_{\max,i}, \quad i = 1,\ldots,4$$
$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{B}_d\,\boldsymbol{d}(k) + \boldsymbol{B}_I f(\boldsymbol{x}^*, \boldsymbol{u}^*)$$

where the result of (6) is an array of inputs from the current time to the h horizon time, as follows: $\boldsymbol{z}^{\mathsf{T}}|_k^{k+h-1} = \left[\boldsymbol{u}(k|k)^{\mathsf{T}}, \ldots, \boldsymbol{u}(k+\ell-1|k)^{\mathsf{T}}, \ldots, \boldsymbol{u}(k+h-1|k)^{\mathsf{T}}\right]$. $\boldsymbol{e}_{\boldsymbol{x}}(k)$ and $\boldsymbol{e}_{\boldsymbol{u}}(k)$ are the errors between the current and the desired state and input respectively. $\Delta\boldsymbol{u}$ is introduced to limit the input rate of change: in other words the last term in (6) allows for a smooth input transition without sudden undesired jumps. $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{R}_{\Delta\boldsymbol{u}}$ are weight matrices [15] and $\boldsymbol{D}(k)^{\mathsf{T}} = [\boldsymbol{d}(k|k)^{\mathsf{T}}, \boldsymbol{d}(k+1|k)^{T}, \ldots, \boldsymbol{d}(k+h-1|k)^{\mathsf{T}}]$. The current disturbance $\boldsymbol{d}(k|k)$ is given or estimated at t_k as described above. Future estimates of the disturbance $\boldsymbol{d}(k+i|k)$ for $i \in \{1, 2, \ldots, h-1\}$ depend on appropriate assumptions on the disturbance. We assume that the disturbance $\boldsymbol{d}(k|k) \in \mathcal{D} = [-\boldsymbol{d}_{\max}, \boldsymbol{d}_{\max}]$ for some known $\boldsymbol{d}_{\max} > 0$. This will be explained in more detail in Section V-B; for now we can assume $\boldsymbol{d}(k+i|k) = \boldsymbol{d}(k|k)\ \forall i$.

As previously mentioned, only the first ℓ elements of $\boldsymbol{z}$ are used for the energy calculation, followed by the linearization of the system and the operation is repeated for N_h consecutive times before the planner makes a decision. We will refer to this procedure as $\mathcal{P}_j^H(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{D}(k))$ or in short $\mathcal{P}_j^H$. The output of $\mathcal{P}_j^H$ is the following prediction: $\boldsymbol{z}^{\mathsf{T}}|_k^{k+H-1} = \left[\boldsymbol{u}(k|k)^{\mathsf{T}}, \boldsymbol{u}(k+1|k)^{\mathsf{T}}, \ldots, \boldsymbol{u}(k+H-1|k)^{\mathsf{T}}\right]$.

Given that $\boldsymbol{u} = f(\omega_1, \omega_2, \omega_3, \omega_4)$ (see Section IV-A), we can calculate the total energy consumed over the horizon H while going to goal g_j by

$$E_{g_j}\Big|_{t_k}^{t_{k+H-1}} = t_s \sum_{i=k}^{k+H-1} P_{g_j}(i) \tag{7}$$

where $P_{g_j}(i)$ is the power consumed at time t_i computed as given by (5).

This prediction will be limited only to the H horizon which may not be sufficient to cover the entire trajectory to the goal. Extending the prediction to an horizon sufficiently large to cover the entire trajectory to a goal is generally computationally expensive if we have long distances to cover. Thus we use the following suboptimal approximation. Let's first consider the typical flight phases of a quadrotor.

A quadrotor has three main transition phases during a flight: ① an initial acceleration, ② a steady motion, and ③ a final deceleration, as depicted in Fig. 3.

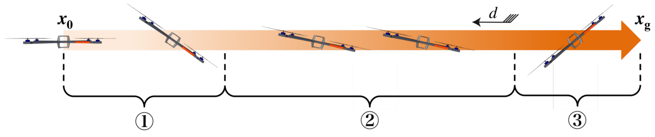


Fig. 3. The three phases during a UAV flight.

During long flights where energy prediction is of foremost importance, both phases ① and ③ are usually short in comparison to phase ② but can use high power. We also notice that phase ② doesn't change (if the disturbance is constant), that is, after the initial transition and stabilization, the UAV keeps the same attitude until it starts to brake to stop to its goal location. With these considerations in mind we are thus interested in estimating the state and input evolution at least until the beginning of phase ②. We then use the following interpolation to calculate the predicted energy consumption

$$E_{g_j}\Big|_{t_k}^{t_{g_j}} = E_{g_j}\Big|_{t_k}^{t_{k+H-1}} + P_{g_j}(k+H-1)(t_{g_j} - t_{k+H-1}) + \Delta E_{g_j,3} \tag{8}$$

where $t_{g_j} = \|\boldsymbol{p}(t) - \boldsymbol{p}_{g_j}\|/v_{max}$ with v_{max} being the maximum speed of the UAV.

The energy required to decelerate and stop ($\Delta E_{g_j3} \geq 0$), during phase ③) can be calculate assuming a shorter phase ② (since in phase ② the system is in steady-state) which will allow the MPC to cover all three phases. Alternatively ΔE_{g_j3} can be neglected during long flights or taken into account with a worst case analysis. Choosing N_h is critical in (8) to have a more accurate estimation. A too short prediction horizon will leave the system inside phase ①, and therefore (8) would lead to a wrong estimation. In general and based on our experiments, phase ① takes less than 1s on a typical research quadrotor.

The procedure presented in this section (summarized in Algorithm 1) is used at replanning time to evaluate the predicted energy to any goal point the UAV can reach. Thus the overall output of the online planner will be a list of reachable goal points with the respective energy predictions $\boldsymbol{E} = [E_{g_1}, \ldots, E_{g_{N_g}}]$. The planner will then choose the goal location $\boldsymbol{x}_{g^*}$ that satisfies the following constraint: $\boldsymbol{x}_{g^*} = \{\boldsymbol{x}_{g_j} \in \boldsymbol{X}_g | E_{g_j} = \min \boldsymbol{E}\}$.

In Algorithm 1, steps 5 to 9 are the procedure $\mathcal{P}_j^H$ described above, while $B(k)$ is a monotonically decreasing function that represents the maximum energy available at t_k. Furthermore, we assume that for any goal g_j, there exists a disturbance $\boldsymbol{d}^+(k) \in \mathcal{D}$ such that the system will consume the maximum possible energy $E_{g_j,\max}$ as defined later in (10) (more consideration will be given in the next section). Step 14 of Algorithm 1 is added to improve the algorithm by eliminating goals as they become infeasible and by ensuring that the quadrotor can reach a destination before its battery is exhausted (i.e., liveness guarantee).

B. Self-triggered & Self/Event-triggered Scheduling

The online prediction and planning procedure described in Section V-A evaluates the energy consumption at every planning time and assign the goal to go that minimizes the energy consumption of the UAV. Predicting periodically is computationally expensive and generally unnecessary especially in situations where the goals are far apart

Algorithm 1 Online Prediction & Planning

```
1:  Estimate disturbance $\boldsymbol{d}(k|k)$
2:  Linearize system around the current state $\boldsymbol{x}(k)$
3:  **for all** $\boldsymbol{x}_{g_j} \in \boldsymbol{X}_g$ **do**
4:      $m = 0$
5:      **while** $m < N_h$ **do**
6:          Linearize system around the $(\ell \cdot m)^{\text{th}}$ state
7:          Run MPC (6) over the interval $[\ell \cdot m, \ell \cdot m+h-1]$
8:          $m \leftarrow m + 1$
9:      **end while**
10:     Calculate $E_{g_j}$ using (7) and (8)
11:     Calculate $E_{g_j,\text{max}}$ for $\boldsymbol{d}^+(k)$
12: **end for**
13: **if** $\exists\ \boldsymbol{x}_{g_j} \in \boldsymbol{X}_g$ s.t. $E_{g_j,\text{max}} \geq B(k)$ **and** $\boldsymbol{X}_g$ is not a singleton **then**
14:     $\boldsymbol{X}_g = \boldsymbol{X}_g \setminus \{\boldsymbol{x}_{g_j}\}$
15: **end if**
16: $\boldsymbol{E} = [E_{g_j}]_{g_j \in \boldsymbol{X}_g}$
17: Plan the trajectory to $\boldsymbol{x}_{g^*}$ where $g^* = \operatorname{argmin} \boldsymbol{E}$
```

and the UAV is already close to its destination. To avoid running the prediction and planning periodically here we propose a self-triggered replanning scheduling approach to address Problem 3.2. The self-triggered scheduler computes the next replanning time based on the predicted maximum and minimum energy consumption. Thus, in order to find the self-triggering policy we need to predict the maximum and minimum energy consumption which depends on the disturbance and dynamics of the system. To do this, for a given time t_k, we let

$$\mathcal{Z}_j = \{\boldsymbol{z}' \mid \exists \boldsymbol{D}' \in \mathcal{D}^H \text{ s.t. } \mathcal{P}_j^H(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{D}') \Rightarrow \boldsymbol{z}'\} \tag{9}$$

then we define the maximum and minimum required energy to reach goal g_j as

$$E_{g_j,\text{max}} = \max_{\boldsymbol{z}' \in \mathcal{Z}_j} E_{g_j}\Big|_{t_k}^{t_{g_j}}, \quad E_{g_j,\text{min}} = \min_{\boldsymbol{z}' \in \mathcal{Z}_j} E_{g_j}\Big|_{t_k}^{t_{g_j}} \tag{10}$$

respectively, where $E_{g_j}\Big|_{t_k}^{t_{g_j}}$ is as given in (8).

It is easy to show that $E_{g_j,\text{max}}$ is when $\boldsymbol{d}^+(k) = \boldsymbol{d}_{\text{max}}\, \text{unit}(\boldsymbol{x}_0 - \boldsymbol{x}_{g^*})$, i.e., the disturbance is as large as possible in the direction away from goal g_j. On the other hand, it is not trivial to compute $E_{g_j,\text{min}}$ since the transient phase also affects this. Instead, we compute $\hat{E}_{g_j,\text{min}} \geq E_{g_j,\text{min}}$ by assuming $\boldsymbol{d}^-(k) = \min\{\boldsymbol{v}_{\text{max}}, \boldsymbol{d}_{\text{max}}\}\, \text{unit}(\boldsymbol{x}_{g_j} - \boldsymbol{x}_0)$, i.e., the disturbance exactly matches the velocity constraint of the UAV when possible. This last consideration is because during the second phase of a quadrotor flight (see section V-A) its roll and pitch angles $\phi = \theta = 0$ meaning that the only energy consumed by the UAV is for hovering ($u_2 = u_3 = u_4 = 0$). Although we cannot compute $E_{g_j,\text{min}}$ exactly, in general, the difference $|E_{g_j,\text{min}} - \hat{E}_{g_j,\text{min}}|$ is negligible because most of the energy is used during phase ② with a low consumption of energy during the transient from its current velocity to a steady state velocity.

Thus, associated with each goal there will be a current disturbance energy prediction, used to plan and replan the trajectory of the UAV and a minimum and maximum energy consumption prediction used to determine next replanning time. After running the MPC procedure in Section V-A the planner chooses one of the goals $\boldsymbol{x}_{g^*} = \boldsymbol{x}_{g_j} \in \boldsymbol{X}_g$ where to send the UAV. The self-triggered policy will then evaluate the following risk between the selected goal $\boldsymbol{x}_{g^*}$ and all other available goals $\boldsymbol{x}_{g_j}$:

$$r_j = \frac{\max\{E^*_{\text{max}} - E_{j,\text{min}}, 0\}}{E_{g_j,\text{max}} - E^*_{\text{min}}} \quad \forall \boldsymbol{x}_{g_j} \in \boldsymbol{X}_g \tag{11}$$

The numerator in (11) should be perceived as the *worst-case risk*, i.e., the maximum amount of energy that we might waste by choosing a suboptimal goal $g_j \neq g^*$. Note that $E^*_{\text{max}} - E_{g_j,\text{min}} \leq 0$ implies there is no risk at all in sticking with the chosen goal g^* since goal g_j is guaranteed to be more expensive in terms of energy. This is reflected in the definition of (12) below. The denominator is needed to capture the *chance* of realizing this worst-case risk. Intuitively, the larger the denominator is, the less chance there is that the worst-case risk will be realized.

We then select the maximum risk among all the possible goals $r_{\text{max}} = \{r' | r' \geq r_j \ \ \forall \boldsymbol{x}_{g_j} \in \boldsymbol{X}_g\}$.

Thus, given a planning time t_i^{plan}, the next time we will replan the goal t_{i+1}^{plan} is given by $t_{i+1}^{\text{plan}} = t_i^{\text{plan}} + \tau$, where

$$\tau = \alpha \left(1 + \tfrac{\beta}{r_{\text{max}}}\right) \tag{12}$$

with α a constant that determines the minimum replanning scheduling time and the scaling for the next replanning time and β a design parameter. Note that setting $\beta = 0$ recovers an algorithm in which the target goal is evaluated periodically. As mentioned above, note that $\tau = \infty$ when $r_{\text{max}} = 0$. Simulations illustrate the efficacy of this framework in Section VI.

A drawback of the self-triggered framework is that we are always assuming worst-case conditions for the wind $\boldsymbol{d}(k) \in \mathcal{D}$ at all times. Instead, we provide here an alternate framework that combines ideas from event-triggered and self-triggered control. The main idea is as follows.

Rather than using worst-case conditions of the disturbance in $\mathcal{D}$ at all time to compute the self-triggered decision time (12), we instead try to estimate the future disturbance based on available information to provide less conservative triggering times (i.e., longer periods of time without active replanning). In between planning times we continue to monitor the disturbance, and if at any time the disturbance violates our estimate, we will then re-evaluate which goal to aim for. This idea is reminiscent of team-triggered control ideas for networked systems [11].

More specifically, at planning time $t_{k^*} = t_i^{\text{plan}}$, rather than computing $E_{g_j,\text{max}}$ and $E_{g_j,\text{min}}$ for each goal assuming worst case conditions in $\mathcal{D}$, we compute an estimate set $\hat{\mathcal{D}}(k) \subset \mathcal{D}$ for $k \geq k^*$. Note that this subset can be a time-varying set. Then, we define a subset $\hat{\mathcal{Z}}_j$ of $\mathcal{Z}_j$ given in (9) using

$$\begin{aligned}\hat{\mathcal{Z}}_j = \{\boldsymbol{z}' \mid \exists \boldsymbol{d}'(k) \in \hat{\mathcal{D}}(k) \text{ for } k \in [k^*, \ldots, k^* + H - 1],\\ \text{s.t. } \mathcal{P}_j^H(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{D}'(k)) \Rightarrow \boldsymbol{z}'\}.\end{aligned}$$

where $\boldsymbol{d}'(k)$ is an element of the sequence $\boldsymbol{D}'(k)$.

We then compute $\hat{E}_{g_j,\text{max}}$ and $\hat{E}_{g_j,\text{min}}$ as in (10) where we replace $\mathcal{Z}_j$ with $\hat{\mathcal{Z}}_j$. It is then guaranteed that $E_{g_j,\text{max}} \geq$

$\hat{E}_{g_j,\max}$ and $E_{g_j,\min} \leq \hat{E}_{g_j,\max}$. This means the next replanning time $t_{i+1}^{\text{plan}} = t_i^{\text{plan}} + \tau$ given by (12) will in general be longer, allowing more time in between replanning the goals.

However, the problem now is that it may be the case that the actual disturbance $\boldsymbol{d}(k)$ does not lie in our estimated set $\hat{\mathcal{D}}(k)$ for some time $t_k > t_i^{\text{plan}}$. To alleviate this issue, we also implement an event-triggered framework that tells the quadrotor it needs to re-evaluate its goal ahead of schedule in case this occurs. Since $\boldsymbol{d}(k|k)$ is estimated at all times $t_k \in [t_i^{\text{plan}}, t_{i+1}^{\text{plan}}]$, if there exists some time $t_{\bar{k}}$ at which $\boldsymbol{d}(\bar{k}|\bar{k}) \notin \hat{\mathcal{D}}(\bar{k})$, we set $t_{i+1}^{\text{plan}} = \max\{t_i^{\text{plan}} + \alpha, t_{\bar{k}}\}$. In other words, if at least α steps have passed since the last planning time, we will immediate re-evaluate the goal, otherwise we will wait until time $t_i^{\text{plan}} + \alpha$.

VI. Simulation Results

The case study investigated in this work is an autonomous navigation of a quadrotor UAV under disturbance (wind in our case) and limited energy constraints. Specifically, we consider that a quadrotor has the task of flying from a starting position $\boldsymbol{p}_{home} = (0, 0, 2)\,\text{m}$ to a final goal $g^* \in [g_1, g_2]$. The two available goal points are located at equal and opposite directions from the starting point of the UAV, at $\boldsymbol{p}_{g_1} = (100, 0, 2)\,\text{m}$ and $\boldsymbol{p}_{g_2} = (-100, 0, 2)\,\text{m}$. In the simulations below we assume that the quadrotor has a maximum speed $v_{max} = \pm 1\,\text{m/s}$. There is a constant disturbance $d_x = 0.5\,\text{m/s}$ except in the marked regions (see Fig. 4). A disturbance of velocity $d_{x,1} = -2\,\text{m/s}$ is introduced at time $t_k = 0.9\,\text{s}$ for a duration of $2\,\text{s}$, while between $6\,\text{s}$ and $8\,\text{s}$ we introduce a stronger disturbance $d_{x,2} = 3\,\text{m/s}$ in the opposite direction. Outside these regions, there is a constant disturbance $d_x = 0.5\,\text{m/s}$. Following the framework introduced in the previous sections, at run time the UAV will predict the future inputs, states, and energy consumptions based on the current disturbance, plan the motion to one of the two goals, and finally compute the next replanning time.

Initial simulations were performed using the Matlab MPC Toolbox [15] however to optimize the execution timing, we eventually opted for a faster implementation using CVXGEN [12] which is able to run 30 times faster than the Matlab MPC toolbox and with same results.

Fig. 4 shows the current ($\circ$), min ($\times$), and max ($+$) predicted energies for both trajectories (to g_1 (blue) and g_2 (red)) during a self-triggered replanning simulation with $\mathcal{D} = [-4, +4]\,\text{m/s}$, as described in Section V-B. For ease of presentation here we show just the first 11s of the simulation since after the last planning execution at 7.5 s the scheduled next replanning time $\tau = \infty$. This can be easily noticed from (12): whenever the risk r is high, like in the beginning, the next replanning time τ is low, whereas as the UAV moves toward one of the goals, the risk becomes lower and τ higher. In fact, as the UAV gets closer to g_2, turning around toward g_1 would imply consuming an higher energy to travel a longer distance even with strong wind $d_{x,2}$.

Fig. 5 shows a comparison between the three replanning scheduling techniques presented in this paper: i) periodic, ii) self-triggered, and iii) self/event-triggered scheduling, all for the same case study described above. In all three cases, the quadrotor switches goal during $d_{x,1}$ and stays on that goal till the end of the simulation. For ease of presentation and to avoid overcrowded plots, here we have omitted the min and max energy predictions data points, shown in Fig. 4. Both self-tricgegred (Figs. 4 and 5(b)) and self/event-triggered (Fig. 5(c)) implementations increase τ as the simulation evolves until a point in which $\tau = \infty$, with a clear improvement from a periodic implementation in Fig. 5(a). Even more, the self/event-triggered case decreases the frequency of replanning because it considers a lower $\hat{\mathcal{D}} = [-2.8, +2.8]\,\text{m/s}$ during the execution. Note that the event-triggered part of the scheduler intervenes as soon as a disturbance $d_x > 2.8\,\text{m/s}$ is detected (Fig. 5(c) at 6 s).

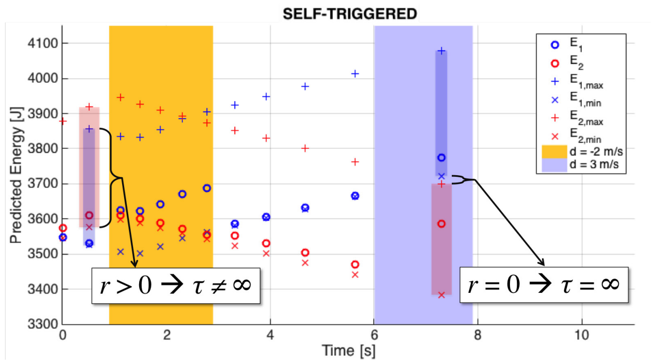


Fig. 4. Self-triggered implementation with min and max energy prediction for both goal trajectories.

VII. Cross Wind Hardware Implementation

To validate the proposed strategy we implemented a series of indoor experiments with a KMel Nano+ quadrotor UAV [3]. We followed the architecture shown in Fig. 6 to run the experiments. Specifically since our online implementation is designed in Matlab/CVXGEN, we decided to use the newly released Robotics System Toolbox that allows us to connect Matlab directly to ROS. The goal output of our online planner is sent, through ROS, to a linux-based machine running the controller for the quadrotor. The pose of the quadrotor is estimated using a Vicon motion capture system.

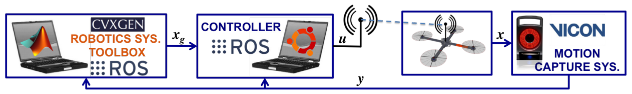
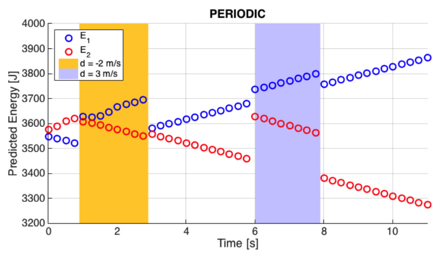


Fig. 6. Architecture diagram of the setup used during the experiments.

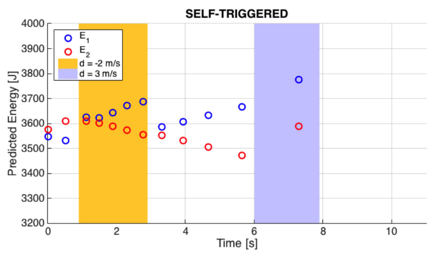
Wind perpendicular to the motion of the UAV was created through a 24” industrial heavy duty drum fan (orange device in Fig. 7) capable of generating wind speeds d_{low} = 3m/s and d_{high} = 6m/s.

The UAV starts at position $\boldsymbol{p}_{home} = (-0.5, -0.85, 0.7)\,\text{m}$ and has a $v_{\max} = 0.3$m/s. Two goals are located at $\boldsymbol{p}_{g_1} = (2.0, -0.7, 0.7)\,\text{m}$ and $\boldsymbol{p}_{g_2} = (0, 1.7, 0.7)\,\text{m}$ (i.e., $\|\boldsymbol{p}_{g_2} - \boldsymbol{p}_{home}\| > \|\boldsymbol{p}_{g_1} - \boldsymbol{p}_{home}\|$). The fan is centered at $\boldsymbol{p}_{fan} = (0, -1.0, 0)\,\text{m}$ blowing air in the $+y$ direction (similar to Fig. 1).

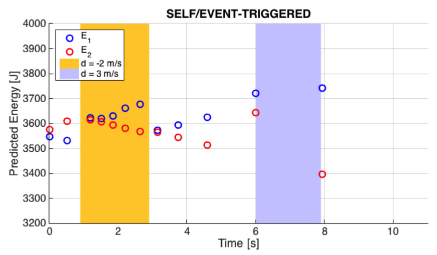
The objective is to navigate from *home* to one of the goals while minimizing energy. Fig. 7 shows a sequence of snapshots of the self-triggered experiment for low disturbance (Fig. 7(a)) and high disturbance (Fig. 7(b)). In both cases the UAV initially chooses g_1 ($E_{g_1} < E_{g_2}$) since there is no disturbance at the starting position, g_1 is closer than g_2, and there is no prior knowledge about the disturbance at $\boldsymbol{p}_{fan}$. With $d_{\text{low}} \rightarrow E_{g_1} < E_{g_2}$ during the entire trajectory while with d_{high} as soon as wind is detected, at replanning time,



(a)



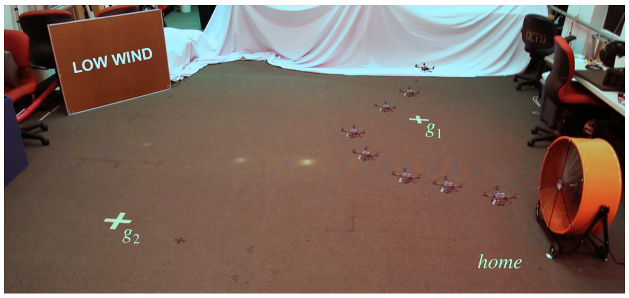
(b)



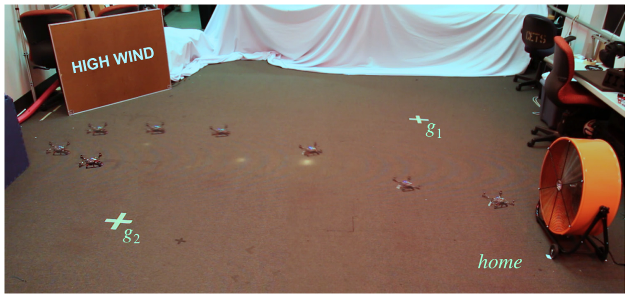
(c)

Fig. 5. First 11s of energy prediction for the (a) periodic, (b) self-triggered, and (c) self/event-triggered replanning simulations.

the UAV decides to switch to g_2 since $E_{g_2} < E_{g_1}$. Fig. 8 shows the predicted energy (similar to Fig. 4) at every step during the experiments.

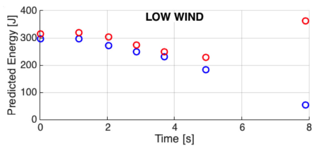


(a)

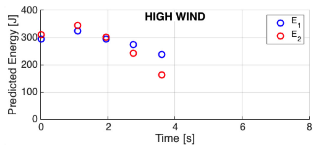


(b)

Fig. 7. Self-triggered experimental results with a) low wind and b) high wind disturbances.



(a)



(b)

Fig. 8. Predicted energy for the a) low wind and b) high wind self-triggered experiments presented in Fig. 7.

VIII. Conclusions & Future Work

In this paper we have presented an online planning scheme for UAV operations under environmental disturbances. Through the use of MPC, self-triggered and self/event-triggered scheduling techniques we were able to demonstrate that it is possible to predict the energy consumption of a complex system like a quadrotor to better plan its motion and reduce unnecessary periodic replanning executions.

Current and future work is centered on extending the proposed framework to multi-agent heterogeneous robotic systems missions and to investigate the scalability of the proposed framework when planning the motion to multiple goals. We are also exploring different prediction techniques and how to adapt the UAV speed based on the disturbance. A better study of the wind distribution for real outdoor experiments is in our agenda, too.

We believe that the developed online planning framework presented in this work could be running in the background of any robotic system operation to guarantee safety and liveness properties.

Acknowledgments

This work is based on research sponsored by DARPA under agreement number FA8750-12-2-0247.

References

[1] N. Bezzo, B. Griffin, P. Cruz, J. Donahue, R. Fierro, and J. Wood, "A cooperative heterogeneous mobile wireless mechatronic system," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 1, pp. 20–31, 2014.

[2] R. Ritz and R. D'Andrea, "Carrying a flexible payload with multiple flying vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 3465–3471.

[3] K. Mohta, V. Kumar, and K. Daniilidis, "Vision-based control of a quadrotor for perching on lines," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3130–3136.

[4] M. Clark, X. Koutsoukos, J. Porter, R. Kumar, G. Pappas, O. Sokolsky, I. Lee, and L. Pike, "A study on run time assurance for complex cyber physical systems," DTIC Document, Tech. Rep., 2013.

[5] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control," *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[6] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *European Control Conference (ECC)*. IEEE, 2013, pp. 1383–1389.

[7] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 2668–2673.

[8] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *IEEE International Conference onRobotics and Automation (ICRA)*. IEEE, 2014, pp. 45–52.

[9] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of uavs," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. Villa Real, Portugal: IEEE, April 8-10 2015, pp. 111–117.

[10] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[11] C. Nowzari and J. Cortés, "Team-triggered coordination for real-time control of networked cyberphysical systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, 2016, to appear.

[12] J. Mattingley and S. Boyd, "Cvxgen: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[13] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[14] R. Leishman, J. Macdonald, R. Beard, and T. McLain, "Quadrotors and Accelerometers: State Estimation with an Improved Dynamic Model," *Control Systems, IEEE*, vol. 34, no. 1, pp. 28–41, Feb 2014.

[15] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2012.