Modular Robotic Propellers For Cooperative 3D Object Manipulation

Zijian Wang[1]

AA222 Project Report
Instructors: Prof. Mykel Kochenderfer & Jonathan Cox
Note: this paper has video attachments, accessible at `http://web.stanford.edu/~zjwang/aa222.html`

***Abstract*— In this paper, we present a novel modular robotic propeller system for cooperative 3D object transportation. The highlight of our approach is that it can account for arbitrary number of propellers, leading to a highly scalable and flexible solution for lifting heavy objects of various sizes. We use Model Predictive Control (MPC) as our core approach to compute the input force for every propeller by solving a Quadratic Programming (QP). Differential flatness is used to generate a 12-dimensional state, which can then be used by MPC to facilitate precise trajectory tracking in 3D. Simulation results demonstrate successful object hovering and trajectory tracking in 3D with 8 propellers using our proposed approach.**

I. Introduction

In this paper, we present a novel modular robotic propeller system in order to manipulate objects of any shape in a 3D environment. Traditionally, aerial robots such as quadrotors are used to either rigidly attach to the object[1], or to lift the the object with cables[2]. However, these approaches bring in more weight such as the airframes of the aerial robots. They also result in undesired aerodynamics and ill-conditioned system dynamics such as the swing of the object. Instead, we propose to attach modular propellers directly to the object in order to exploit the dynamics of the object and eliminate the need for airframes. A sketch of our approach is shwon in Figure 1. Each propeller module is supposed to have its own onboard computer, circuits and battery, such that the propellers can operate in a distributed fashion. The propeller is designed to be compact and has appropriate mechanism to attach to the object, so that a large manipulation force can be generated to transport bulky objects by putting propellers together. In this vein, our system is also highly reconfigurable and scalable in terms of the number of propellers and where they are mounted, which can be determined by the size and geometry of the object.

The main challenge of the proposed problem is twofold. First of all, the input is redundant since we allow arbitrary number of propellers (usually it should be greater than three). Secondly, the positions of the propellers may be very asymmetric, depending on the shape of the object. Both challenges pose difficulty for classic controller design such as PID and LQR. In this paper, we use Model Predictive Controller (MPC) to not only obtain the closed-loop control

[1]Z. Wang is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA, `zjwang@stanford.edu`

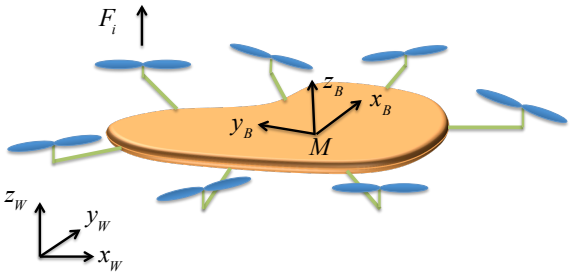

Fig. 1. An example of the 3D object manipulation using 7 propellers. We assume that the propeller are rigidly attached to the object. Our proposed approach can compute the optimal input force for every propeller, in order to lift the object in the air or track a specified trajecoty.

for the propellers, but also ensure optimality. The essence of the MPC is to solve an optimization problem with quadratic objective function and linear constraints, which is also referred to as quadratic programming (QP). The nonlinear dynamics of the object is linearized around the current state at every MPC iteration in order to obtain the linear constraints. Our goal is to be able to transport the object to anywhere in the 3D environment. In order to achieve this, we need to do *kino-dynamical motion planning* rather than purely using MPC, because the control generated by MPC using the linearized model is a local result. We therefore leverage the technique called *differential flatness*[3], [4] to plan a 12-state trajectory, and then use MPC as a closed-loop controller to track the planned trajectory.

Our approach can be used in a lot of real-life applications. For example, in a construction site, these modular propellers can be used to lift heavy materials quickly and economically without the hassle of setting up a tower crane. In a disaster relief scenario, our system can be used to remove large debris or even transport survivors as the first aid before large rescue vehicles and equipment can arrive.

Note that this paper merges two course projects, i.e., AA222 and AA203. On the AA222 side, the main focus is on solving the MPC as a constrained optimization problem, while for AA203 side, the use of differential flatness for robot trajectory planning is the main focus.

II. Related Work

Our work is closely related to a family of research called multi-robot manipulation, where the idea is to coordinate the efforts from a group of robots in order to move objects that cannot be handled by an individual robot. Our work belongs to this scope in the sense of manipulating objects in 3D. A minimalist multi-robot manipulation approach was presented in [5] to move furniture and the tradeoff between different amount of sensing, communication and computation was discussed. A popular solution for multi-robot manipulation is called caging, where robots move in a formation that keeps the object always inside the formation [6], [7]. Forces from the robotic group can be coordinated given the desired acceleration and a communication network [8]. Recently, experiments were conducted using real robots, including massive uniform manipulation with 100 Kilobot robots using one global control input [9], [10], a kinematic manipulation scheme based on decentralized centroid estimation [11], [12], four omni-directional robots that can lift and move a passenger vehicle [13]. There are also attempts that uses no communication, such as moving a tall object towards destination based on visual occlusion [14], or using force sensing or motion measurement as an implicit way for coordination [15], [16], [17].

A few other work also provide us much insight in developing our work. [18] discussed about the modeling and control of a quadrotor. A distributed flight array was developed in [19] which consists of a number of hexagon-shaped modular propellers. A learning-based MPC was proposed in [20] to control a quadrotor with uncertain dynamics using an onboard computer.

III. Problem Formulation

The motion of the object can be described by Newton-Euler equations. The state variable contains 12 states, i.e., $\xi = (x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r)$, which corresponds 3D position, three linear velocities, three Euler angles (roll, pitch, yaw) and three corresponding angular velocities. The rotation matrix can be defined in the form of ZYX Euler angles:

$$R = \begin{bmatrix} C\theta C\phi & S\phi S\theta C\psi - C\phi S\psi & C\phi S\theta C\psi + S\phi S\psi \\ C\theta S\phi & S\phi S\theta S\psi + C\phi C\psi & C\phi S\theta S\psi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix}, \tag{1}$$

where C stands for $cos(\cdot)$, and S stands for $sin(\cdot)$. Also denote the mass of object as m and the moment of inertial as I_x, I_y, I_z along the x, y, z axis in the body reference frame.

We have a group of N propellers, each of which is indexed as $P_i, i \in \{1, 2, \cdots, N\}$. Denote the force generated by the propeller by F_i, which points up vertically in the body reference frame. The force can be controlled by varying the speed of the rotor. There will also be an associated torque, which is proportional to the force, denoted as $T_i = c_i F_i$, where c_i can be positive or negative depending on whether the propeller rotates clockwise(CW) or counterclockwise(CCW). We also assume that we know the positions of the propellers in the object's body reference frame, represented by the coordinate (x_i, y_i) for every propeller P_i. Then the 12-state dynamics can be written down using Newton Euler equation,

$$\dot{\mathbf{v}} = -g\mathbf{e}_3 + \frac{1}{m} R \sum_{i=1}^{N} F_i \mathbf{e}_3, \tag{2}$$

$$\dot{\mathbf{h}} = \mathbf{v}, \tag{3}$$

$$\dot{\boldsymbol{\omega}} = J^{-1} \left(\sum_{i=1}^{N} [F_i x_i, F_i y_i, F_i c_i]^T \right) - J^{-1} \Omega J \boldsymbol{\omega}, \tag{4}$$

$$\dot{R} = R\Omega, \tag{5}$$

where $\mathbf{v} = [v_x, v_y, v_z]$ is the linear velocity in the world reference frame, g is the acceleration of gravity, $\mathbf{e}_3 = [0, 0, 1]^T$, $\mathbf{h} = [x, y, z]^T$ is the position of the object in the world reference frame, $\boldsymbol{\omega} = [p, q, r]$ is the angular velocity in the body reference frame, $J = diag(I_x, I_y, I_z)$, and $\Omega = [0, -r, q; r, 0, -p; -q, p, 0]$ is the tensor of $\boldsymbol{\omega}$ that is used for converting cross product into matrix multiplication.

The objective of our problem can be stated as follows. Given a desired position or trajectory of the object in 3D, design the input forces of all propellers such that the object can be transported to the destination or follow the given trajectory.

IV. Model Predictive Control

In the previous section, we introduced the dynamics of the system in the form of $\dot{\xi} = f(\xi, u)$, where $u = [F_1, F_2, \cdots, F_N]$ is the control input. f is a nonlinear function, as revealed by Eq. (2)-(5). The objective is to use model predictive control (MPC) to design the input variable u in order to track a given reference trajectory $\xi_r(t)$, generated by some trajectory planning algorithm.

We first linearize the nonlinear dynamics around the current state ξ_t, u_t as follows,

$$\dot{\xi} \approx f(\xi_t, u_t) + \frac{\partial f}{\partial \xi}\Big|_{(\xi_t, u_t)} (\xi - \xi_t) + \frac{\partial f}{\partial u}\Big|_{(\xi_t, u_t)} (u - u_t). \tag{6}$$

Let

$$\frac{\partial f}{\partial \xi}\Big|_{(\xi_t, u_t)} = A_t, \tag{7}$$

$$\frac{\partial f}{\partial u}\Big|_{(\xi_t, u_t)} = B_t, \tag{8}$$

then the linearized moldel around (ξ_t, u_t) can be written as

$$\dot{\xi} = A_t \xi + B_t u + a(\xi_t, u_t), \tag{9}$$

where

$$a(\xi_t, u_t) = f(\xi_t, u_t) - A_t \xi_t - B_t u_t \tag{10}$$

is an affine term, which may not necessarily be zero since ξ_t, u_t is not necessarily the equilibrium.

The linearized dynamics (9) can be further discretized using Euler's method as follows

$$\xi_{k+1} = \xi_k + (A_t \xi_k + B_t u_k + a(\xi_t, u_t))h, \tag{11}$$

where $k \in \mathbb{N}$, and $\xi_0 = \xi_t$. The objective of MPC is to track a given trajectory $\xi_r(t)$ using the discrete linearized prediction model (11). To clarify notation, we define the predicted state at time $t + kh$ based on the linearization at time t as $\xi(t + kh|t)$. Then the predicted trajectory tracking error can be defined as

$$\Delta\xi(t + kh|t) = \xi(t + kh|t) - \xi_r(t + kh). \tag{12}$$

Finally, the MPC aiming to minimize the trajectory tracking error can be formulated as follows,

$$\begin{aligned}
\text{minimize} \quad & p(\Delta\xi(t + Nh|t)) + \sum_{k=0}^{N-1} q(\Delta\xi(t + kh|t), u(k)) \\
\text{subject to} \quad & \xi(t + (k+1)h|t) = \xi(t + kh|t) + \\
& \qquad h\big[A_t\xi(t + kh|t) + B_t u(k) + a(\xi_t, u_t)\big] \\
& u_{min} \le u_k \le u_{max} \\
& \xi(t|t) = \xi_t \\
& k = 0, 1, \cdots, N-1
\end{aligned} \tag{13}$$

where h is the time step and N is the planning horizon. $p(\Delta\xi(t + Nh|t)) = \Delta\xi(t + Nh|t)^T P \Delta\xi(t + Nh|t),\ P \ge 0$ is a quadratic terminal penalty function, and $q(\Delta\xi(t + kh|t), u(k)) = \Delta\xi(t + kh|t)^T Q \Delta\xi(t + kh|t) + u(k)^T R u(k),\ Q \ge 0,\ R \ge 0$ is the stage cost.

V. Differential Flatness

Obtaining a feasible trajectory, e.g. $\xi_r(t)$ in (12), for a differentially constrained dynamical system is often nontrivial, especially for nonholonomic dynamics like ours in (2)-(5). Furthermore, in most cases we only have a desired trajectory in lower dimension, such as 3D position plus the yaw angle. The motivation in this section is to leverage the differential flatness theory to solve for 12-dimension state trajectory from a lower dimension pre-defined trajectory containing only $x(t), y(t), z(t), \psi(t)$. Then we can use the MPC controller in the previous section to track the designed 12-dimensional trajectory.

The dynamics (2)-(5) is proven to be differentially flat. Concretely, the differential flatness in our case means that given a "flat" trajectory

$$\sigma(t) = [\sigma_1, \sigma_2, \sigma_3, \sigma_4]^T = [x, y, z, \psi]^T, \tag{14}$$

which is differentiable up to third order (i.e., we can explicitly write down $(\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma)$), we can uniquely determine the 12-dimensional trajectory

$$\xi(t) = \beta(\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma), \tag{15}$$

which ensures that the given reference trajectory $\sigma(t)$ is feasible and does not violate the dynamics. β is a one-on-one mapping from the flat output $\sigma(t)$ to the higher-dimensional full state trajectory.

Now we discuss the derivation of $\beta(\cdot)$, which is summarized from [4]. Recall that $\xi = (x, y, z, v_x, v_y, v_z, \phi, \theta, \psi, p, q, r)$. There are several obvious relationships. For example, (x, y, z) are directly given by $(\sigma_1, \sigma_2, \sigma_3)$, (v_x, v_y, v_z) are directly given by $(\dot\sigma_1, \dot\sigma_2, \dot\sigma_3)$, and ψ is given by σ_4. To solve for ϕ and θ, consider (2),

$$\dot{\mathbf{v}} + g\mathbf{e}_3 = \frac{\sum_{i=1}^N F_i}{m}(R\mathbf{e}_3), \tag{16}$$

where both sides of the equal sign are vectors. We also observe that $\dot{\mathbf{v}} = [\ddot\sigma_{1:3}]$, and $R\mathbf{e}_3$ is indeed $\mathbf{z}_b$ (the z axis in the body frame), which is a unit vector. Using the fact that these two vectors should have the same direction, we have

$$\frac{[\ddot\sigma_{1:3}] + g\mathbf{e}_3}{\|[\ddot\sigma_{1:3}] + g\mathbf{e}_3\|} = \mathbf{z}_b. \tag{17}$$

We know the terms on the left side of (17), and $\mathbf{z}_b$ is actually the third column in the rotation matrix R in (1). Then by (17) we have three equations and and two unknowns ϕ and θ, which we can solve for. After this, we know all three angles, meaning that we also know the entire rotation matrix R. Then by (5), we have

$$\boldsymbol{\omega} = (R^T\dot R)^\vee, \tag{18}$$

where $\vee$ is an operation that transforms a skew symmetric matrix back to a column vector. $\dot R$ can be computed by directly taking the derivative of R, which can be written in terms of $(\sigma, \dot\sigma, \ddot\sigma)$.

In summary, by going through all the computations described above, one can explicitly find that mapping from the low dimensional flat output $(\sigma, \dot\sigma, \ddot\sigma, \dddot\sigma)$ to the 12-dimensional state trajectory ξ, as follows,

$$\begin{aligned}
&[x, y, z, v_x, v_y, v_z, \psi] = [\sigma_1, \sigma_2, \sigma_3, \dot\sigma_1, \dot\sigma_2, \dot\sigma_3, \sigma_4] \\
&\theta = \text{atan2}(\beta_a, \beta_b) \\
&\phi = \text{atan2}(\beta_c, \sqrt{\beta_a^2 + \beta_b^2}) \\
&[p, q, r]^T = (R^T\dot R)^\vee
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
\beta_a &= -\cos\sigma_4\ddot\sigma_1 - \sin\sigma_4\ddot\sigma_2 \\
\beta_b &= -\ddot\sigma_3 + g \\
\beta_c &= -\sin\sigma_4\ddot\sigma_1 + \cos\sigma_4\ddot\sigma_2
\end{aligned} \tag{20}$$

VI. Simulation Results

In this section, we demonstrate several simulation results with closed loop control using MPC. In all simulations, eight propellers are attached to the disc-shaped object with a mass $2kg$. Eight propellers are placed by some random choices of positions, in order to show the generality of the proposed approach. All the simulations were done in Matlab and used CVX[1] as the solve for the QP problem.

A. Hovering

We first test our MPC controller in a hovering scenario, where the goal is a given point in 3D and the reference trajectory in the MPC formulation is just a static point. Note that differential flatness is not used here because we only care about the goal point and have no requirement on the specific

[1]CVX: a Matlab-based solver for convex optimization problem. http://cvxr.com

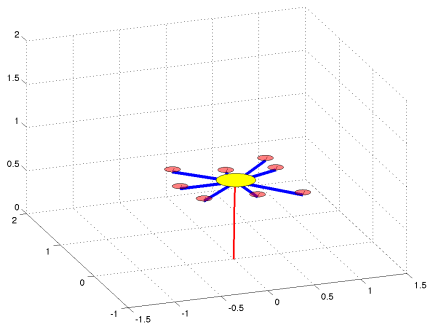

Fig. 2. Hovering simulation. The propeller system successfully lift the object starting at [0,0,0] to the goal position [0,0,1]. The red line shows the elapsed trajectory of the object.

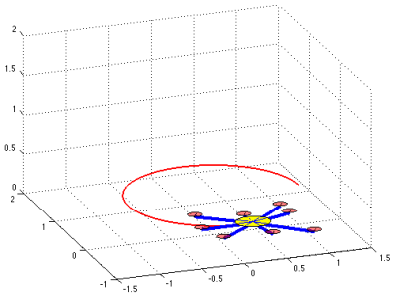

Fig. 3. A snapshot of transporting the object along a circular trajectory. The red solid line shows the elapsed trajectory traveled by the object.

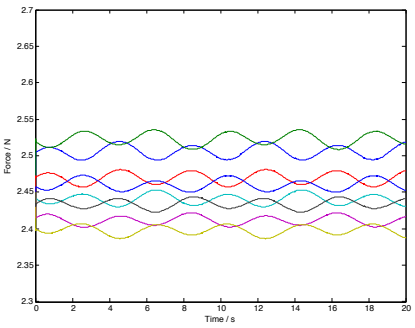

Fig. 4. The force inputs from eight propellers from $t = 0$ to $t = 20s$ in the circular trajectory tracking simulation.

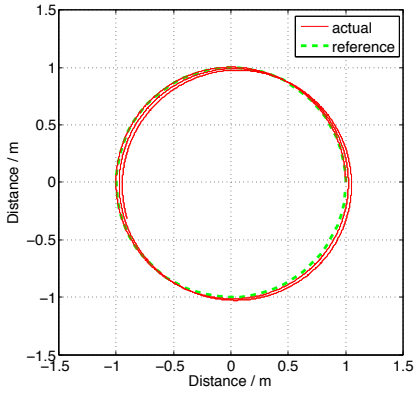

Fig. 5. Trajectory tracking performance. Green dashed line is the desired trajectory while the red line is the actual trajectory.

trajectory. The simulation shows that the MPC works well in lifting the object in the air even if the propellers are highly asymmetric. In addition to a level initial pose, we also try the case where the initial Euler angles are not zero. The result shows that MPC can also stabilize the object in the air. Both simulation videos can be found in the video attachment.

B. Trajectory Tracking

In this part we show the complete simulation combining both MPC and differential flatness for a trajectory tracking task. As a benchmark test, first we test the overall closed-loop tracking with a circle with radius $1m$ at the fixed height $0.5m$. Note that we arbitrarily choose $\phi = 0$. The desired trajectory (or flat output) in this case is $\sigma(t) = [\cos(0.8t), \sin(0.8t), 0.5, 0]$. A snapshot of the tracking process is shown in Figure 3. The control inputs computed by MPC are very smooth, as shown in Figure 4. The performance of the trajectory tracking is demonstrated in Figure 12. As it can be seen, during the $20s$ simulation time, the precision of the trajectory tracking is very high.

In addition the the circular trajectory, we also tested some other trajectories such as logarithm spiral. All the simulation videos can be found in the video attachment.

VII. Conclusion and Future Work

In this paper, we propose a novel modular propeller system for 3D cooperative object transportation. The dynamics of the system is analyzed and formulated. We use differential flatness to generate the 12-dimensional state trajectory and use model predictive control to follow the trajectory generated by differential flatness. Simulation results demonstrate the generality and versatility of the proposed approach, including performing both hovering and trajectory tracking under different scenarios.

In terms of future work, there are many directions we can take. First of all, we intend to implement a distributed MPC framework, where we can leverage the parallel computation power of each propeller in order to accelerate the optimization solving process and make the system more scalable. Secondly, we are interested in developing our own solver for the constrained optimization problem in more efficient languages such as C++ or Julia, which can be readily deployable to real-time embedded system. Thirdly, automated trajectory generation will be useful for our system, since right know the desired trajectory $\sigma(t)$ is given by hand. Lastly, we also intend to study how to optimize the location of the propellers in order to maximize efficiency, stability or agility.

References

[1] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 545–558.

[2] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[3] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2520–2525.

[4] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 6567–6572.

[5] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," in *Proc. of the International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 1995, pp. 235–242.

[6] G. A. Pereira, M. F. Campos, and V. Kumar, "Decentralized algorithms for multi-robot manipulation via caging," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 783–795, 2004.

[7] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-robot manipulation via caging in environments with obstacles," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1471–1476.

[8] J. M. Esposito, "Decentralized cooperative manipulation with a swarm of mobile robots," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, 2009, pp. 5333–5338.

[9] A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin, "Massive uniform manipulation: Controlling large populations of simple robots with a common input signal," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 520–527.

[10] M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal, "Collective transport of complex objects by simple robots: theory and experiments," in *Proceedings of International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 2013, pp. 47–54.

[11] G. Habibi, K. Zachary, W. Xie, M. Jellins, and J. McLurkin, "Distributed centroid estimation and motion controllers for collective transport by multi-robot systems," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2015, pp. 1282–1288.

[12] G. Habibi, Z. Kingston, Z. Wang, M. Schwager, and J. McLurkin, "Pipelined consensus for global state estimation in multi-agent systems," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015, pp. 1315–1323.

[13] A. Amanatiadis, C. Henschel, B. Birkicht, B. Andel, K. Charalampous, I. Kostavelis, R. May, and A. Gasteratos, "Avert: An autonomous multi-robot system for vehicle extraction and transportation," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 1662–1669.

[14] J. Chen, M. Gauci, and R. Gross, "A strategy for transporting tall objects with a swarm of miniature mobile robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 863–869.

[15] G. Baldassarre, V. Trianni, M. Bonani, F. Mondada, M. Dorigo, and S. Nolfi, "Self-organized coordinated motion in groups of physically connected robots," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 1, pp. 224–239, 2007.

[16] Z. Wang and M. Schwager, "Multi-robot manipulation without communication," in *The 12th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2014, pp. 135–149.

[17] Z. Wang and M. Schwager, "Multi-robot manipulation with no communication using only local measurements," in *Decision and Control (CDC), 2015 IEEE Conference on*, 2015, pp. 380–385.

[18] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2004.

[19] R. Oung, A. Ramezani, and R. DAndrea, "Feasibility of a distributed flight array," in *Decision and Control (CDC), IEEE Conference on*, 2009, pp. 3038–3044.

[20] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2012.