Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle

Matko Orsag, Christopher Korpela, and Paul Oh

***Abstract*— Compared to autonomous ground vehicles, UAVs (unmanned aerial vehicles) have significant mobility advantages and the potential to operate in otherwise unreachable locations. Micro UAVs still suffer from one major drawback: they do not have the necessary payload capabilities to support high performance arms. This paper, however, investigates the key challenges in controlling a mobile manipulating UAV using a commercially available aircraft and a light-weight prototype 3-arm manipulator. Because of the overall instability of rotorcraft, we use a motion capture system to build an efficient autopilot. Our results indicate that we can accurately model and control our prototype system given significant disturbances when both moving the manipulators and interacting with the ground.**

I. Introduction

Historically, UAV research has been focused on avoiding interaction with the environment. This fact is mostly due to poor payload capabilities available from micro UAVs in addition to the possibility of crashing and causing injury. So far, UAVs have been used mostly in surveillance and search and rescue missions. However, the ability for air vehicles to manipulate a target or carry objects they encounter could greatly expand the types of missions achievable by unmanned systems. Flying robots with dexterous arms could lead to transformative applications in near-Earth environments. Such applications could be infrastructure repair, vehicle inspections, or agricultural care. The authors envision a paradigm shift in the way UAVs are currently deployed.

While mobile manipulation continues to be a highly active field of study, much of the focus lies with ground vehicles. Wheeled and tracked vehicles maintain a stable base during manipulation and even robots that must dynamically balance (i.e. humanoids and Segways) can comparatively easily maintain stability. To date, the authors are unaware of any aerial mobile manipulators with one or more dexterous arms. There are many examples of sling-load implementations in addition to hook or snag style arms to interact with target objects [1], [2]. Some researchers have achieved aerial grasping through the use of 1-DOF (degree of freedom) grippers [3], [4]. While these results are beneficial, they simply implement beak or claw-style grasping and not true manipulation. The air vehicle provides the majority of the degrees of freedom. Taking inspiration from nature, the authors seek to implement true arm dexterity in aircraft similar to how an octopus would manipulate a seashell. One of the most important aspects of this research is to characterize the tightly coupled interaction between the manipulator arms and the aircraft. The reaction forces observed during arm movement and ground contact introduce destabilizing effects onto an already inherently unstable system. Recent advances in aircraft payloads and developments in light-weight arms signal a positive trend toward dexterous aerial manipulation. Payload limitations will most likely not represent the critical issue rather how arm trajectories and ground interactions produce forces and torques on the aircraft.

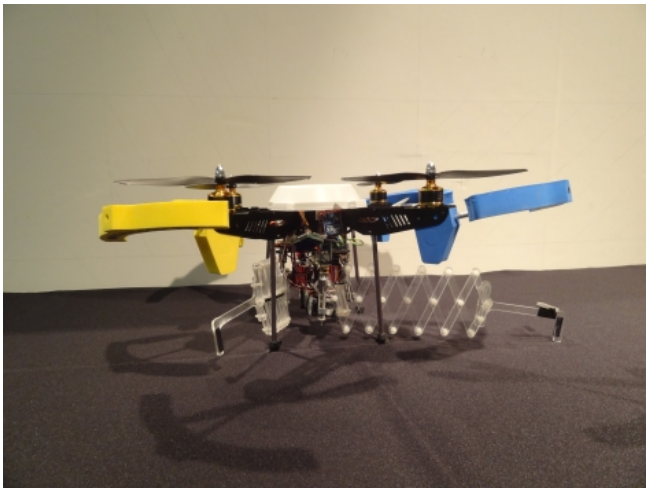


Fig. 1: MM-UAV as a serial chain manipulator

Building upon our previous work, we have garnered invaluable lessons from both our hybrid structure and our test and evaluation miniature gantry system [5], [6]. Even with perfect communication between two operators (one controlling the quadrotor and the other actuating the manipulators), the force feedback from the arms is still unclear. To facilitate testing and focus on manipulation, we recently introduced a motion capture system to provide hovering stability and eliminate the human pilot. For these experiments, our focus lies with the arm feedback and not as much with precision control of the aircraft. In a similar manner, our mini-gantry test rig emulates the quadrotor dynamics but again allows for the investigation of interactions with the environment during manipulation. These insights from the test rig are then applied to our MM-UAV prototype that was flown in a motion capture volume.

This paper introduces the model dynamics for the MM-UAV quadrotor prototype vehicle [7] as shown in Fig. 1.

Manuscript received May 15, 2012. This project was supported in part by a US NSF PIRE Grant 0730206 and US NSF CAREER Grant 0347430. The work of M. Orsag was supported by the U.S. Bureau of Educational and Cultural Affairs through the Fulbright Program.

M. Orsag, C. Korpela, and P. Oh are with the Drexel Autonomous Systems Lab, Drexel Univeristy, Philadelphia, PA 19104 USA {mo427, cmk325}@drexel.edu, paul@coe.drexel.edu

Fig. 2: GAUI 330 Quadrotor

We develop a tightly coupled dynamic and kinematic model to provide for coordinated control of the system. The end-effector pose for each arm is a combination of the six DOFs of the quadrotor and the two DOFs for the respective arm. We then implement this model on our prototype to analyze and perform aerial manipulation.

II. MM-UAV Hardware

The MM-UAV prototype is constructed using a low-cost quadrotor-type aircraft, a GAUI 330 (Fig. 2), and three manipulator arms (Fig. 3). The GAUI quadrotor uses four equally positioned brushless motors equipped with 8 inch propellers to provide lift and maneuverability. The total vehicle diameter is 330mm and the total weight is approximately 400g with a maximum flying weight of 1100g.

Each manipulator arm has 2 degrees of freedom, starting with a servo-driven revolute joint and a DC motor-driven translation joint. The arms were built out of light but sturdy acrylic material, having in mind the limited payload capacity of the quadrotor. The arms are rotated and placed so that the starting horizontal position allows for the landing gear to be shorter, saving weight and allowing the arms to be protected when landing. For additional manipulation capabilities, a hook was made, so that the arms are able to hook and push/pull objects from the floor.

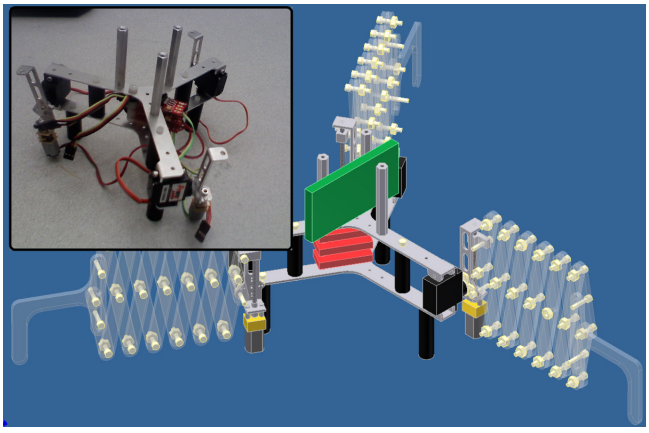


Fig. 3: Three 2-DOF Manipulators

MM-UAV			
J_{xx}	J_{yy}	J_{zz}	mass
$4.87gcm^2$	$4.97gcm^2$	$5.95gcm^2$	$721.8g$
Arms			
mass	MIN extension		MAX extension
$64.6g$	18 cm		54 cm
Measurements			
r	d_1		a_1
$3.7cm$	3.3 cm		3.1 cm

TABLE I: MM-UAV Construction Parameters

The final construction is shown in Fig. 4 and important construction points (i.e. Center of Assembly - COA, Center of Mass of manipulator arm - COM, and Center of Gravity of aircraft - COG) are appropriately marked. Additionally, measured distances between these points are shown. The exact aircraft parameters are listed in Table I. The table shows that the added manipulator did not change the basic quadrotor symmetry (i.e. $J_{xx} = J_{yy}$). This is proved to be vital for the actual controller design. The overall weight of the aircraft is kept bellow 950g, thus making the aircraft capable of lifting small objects bellow 200g.

III. Articulated-Body Dynamics

Mobile manipulators have been successfully deployed and researched on both autonomous underwater and ground vehicles as well as spacecrafts [8], [9], [10], [11]. Therefore, various mathematical formalisms have been used to model the dynamics of mobile manipulators. The dynamics of quadrotors are well understood [12] but there is a critical gap in applying mobile manipulation to air vehicles. To that end, this paper derives equations of motion for MM-UAV

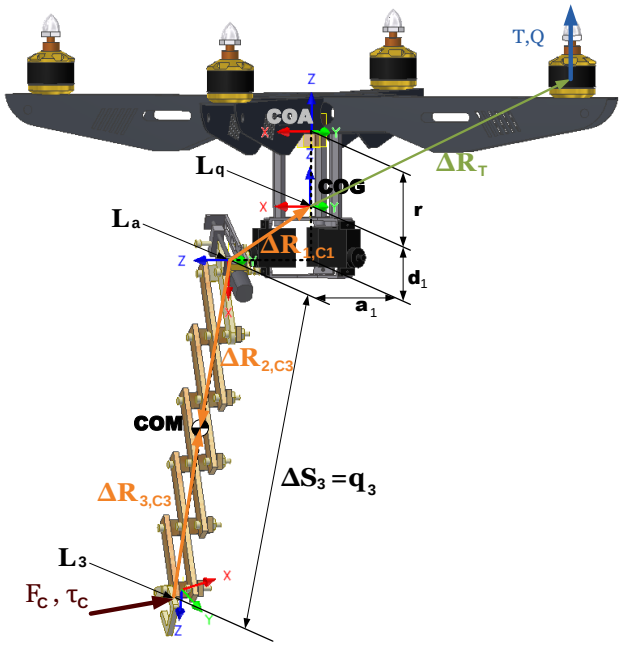


Fig. 4: CAD model design and final construction (showing only one arm for clarity)

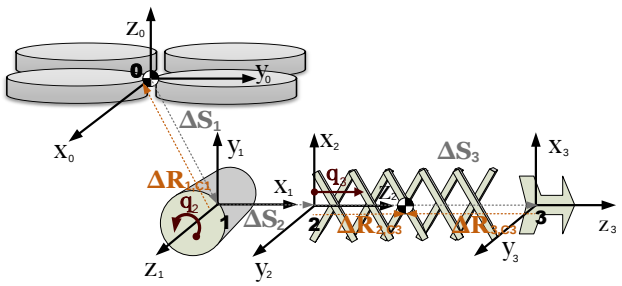


Fig. 5: MM-UAV Coordinate Frames

using recursive Newton-Euler dynamics equations. With this approach we can calculate the joint forces of each link, starting from the end-effector and end at the base of the mobile manipulator. For such an analysis, direct kinematics equations and Newton-Euler equations of motion need to be derived first.

A. Kinematics equations

Each MM-UAV arm is modeled as a serial chain RP (revolute-prismatic) manipulator shown in Fig. 5. The quadrotor body frame is translated and rotated relative to the world inertia frame, W. The body frame is first modeled as a revolute joint, fixed base frame, with a constant angular offset for each MM-UAV arm. The assumption of a fixed base frame is of course not valid for a flying quadrotor. Nevertheless, equations can easily be augmented to account for the mobile base by adding initial rotational and translational speeds and accelerations to the quadrotor's base frame. Viewing from the world frame W, however, requires attaching a six DOF joint in the serial chain [13]. In this paper, forces and torques are viewed from the quadrotor base frame in which the low level stabilization controller works.

Using Denavit-Hartenberg (DH) parameterization, joint frames are set and direct kinematics equations for each serial chain are derived. This procedure is repeated for all three MM-UAV arms. Table II shows the resulting DH parameters [14] of each joint for all three arms where θ, d, a, and α are the standard DH parameters [15]; q_1^i, q_2^i, and q_3^i are joint variables of each manipulator arm i. Because the manipulator arms are identical, DH parameters for all joints are the same. For each base quadrotor joint, the arms are fixed and placed in an equilateral triangle pattern, with a 120° angle between them. This is why the base joints have a constant value θ^i.

Figure 5 shows the resulting displacement vectors $\Delta\vec{S}_j$, that show the distance between two adjacent frames (i.e. $j-1$ and j). Displacement vectors $\Delta\vec{S}_j$ are crucial for the derivation of Newton-Euler motion equations and therefore their values are shown in vector form in (III-A). Another important measurement of given kinematics are the distances between the joint j frame and link's ck center of mass, $\Delta R_{j,ck}$. They are shown in Fig. 5 and given with (III-A). The superscript i denotes the i^{th} arm and unit vectors are marked with the $\hat{}$ symbol.

	θ	d	a	α
Link 1				
Arm 1	$\frac{\pi}{2}$	d_1^1	a_1^1	$\frac{\pi}{2}$
Arm 2	$\frac{7\pi}{6}$	d_1^2	a_1^2	$\frac{\pi}{2}$
Arm 3	$\frac{11\pi}{6}$	d_1^3	a_1^3	$\frac{\pi}{2}$
Link 2				
Arms 1,2,3	$q_2^i + \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
Link 3				
Arms 1,2,3	0	q_3^i	0	0

TABLE II: Denavit-Hartenberg Parameters

$$\Delta\vec{S}_1^i = a_1^i cos\left(\theta_1^i\right)\hat{x}_0 + a_1^i sin\left(\theta_1^i\right)\hat{y}_0 + d_1^i\hat{z}_0$$
$$\Delta\vec{S}_2^i = 0, \Delta\vec{S}_3^i = q_3^i\hat{z}_3^i \tag{1}$$

$$\Delta\vec{R}_{1,C1}^i = -\Delta\vec{S}_1^i, \Delta\vec{R}_{0,C1}^i = 0$$
$$\Delta\vec{R}_{1,C2}^i = \Delta\vec{R}_{2,C2}^i = 0$$
$$\Delta\vec{R}_{2,C3}^i = -\Delta\vec{R}_{3,C3}^i = \frac{1}{2}q_3^i\hat{z}_2^i \tag{2}$$

B. Newton-Euler motion equations

In a mobile manipulating quadrotor, the aircraft's autopilot has to compensate for additional disturbance forces and torques produced from the manipulator. In order to fully understand the underlying aircraft dynamics, one needs to calculate these forces and torques. To derive the equations, all calculations are placed in the body coordinate frame (i.e. where the actual autopilot control takes place). This section derives the dynamic equations based on Fig. 5 which shows Arm 1 as a kinematic chain. The equations are given in vector form and can be easily applied to other manipulator arms.

Given the initial angular Ω_q and translational V_q velocities of the quadrotor, the angular $\vec{\omega}_j^i$ and translational $\vec{v}_j^i$ velocities of each observed joint frames j and arm i take the following vector forms:

$$\vec{\omega}_1^i = \vec{\omega}_0^i = \vec{\Omega}_q$$
$$\vec{\omega}_3^i = \vec{\omega}_2^i = \vec{\Omega}_q + \dot{q}_2\vec{z}_1 \tag{3a}$$

$$\vec{v}_0^i = \vec{V}_q$$
$$\vec{v}_2^i = \vec{v}_1^i = \vec{v}_0^i + \vec{\omega}_0^i \times \Delta\vec{S}_1^i$$
$$\vec{v}_3^i = \vec{v}_2^i + \vec{\omega}_2^i \times \Delta\vec{S}_3^i + \Delta\dot{\vec{S}}_3^i \tag{3b}$$

where $\dot{q}_j^i$ denotes the joint variable speed of each frame j and arm i, both for revolute and prismatic joints. Applying Newton's laws to obtain the necessary force and torque equations requires finding the derivatives of the velocity vectors (i.e. $\dot{v}_j^i$ and $\dot{\omega}_j^i$) expressed in the quadrotor base coordinate system. Using the theorem of Coriolis, the derivation yields the following expressions for the angular and translational accelerations of each observed joint frame:

$$\dot{\vec{\omega}}_1^i = \dot{\vec{\omega}}_0^i = \dot{\vec{\Omega}}_q$$
$$\dot{\vec{\omega}}_3^i = \dot{\vec{\omega}}_2^i = \dot{\vec{\Omega}}_q + \ddot{q}_2^i\hat{z}_1^i + \vec{\omega}_1^i \times \dot{q}_2^i\hat{z}_1^i \tag{4a}$$

$$\vec{v}_0^i = \vec{V}_q$$
$$\vec{v}_2^i = \vec{v}_1^i = \vec{v}_0^i + \vec{\omega}_0^i \times \Delta\vec{S}_1^i + \vec{\omega}_0^i \times \left(\vec{\omega}_0^i \times \Delta\vec{S}_1^i\right)$$
$$\vec{v}_3^i = \vec{v}_2^i + \vec{\omega}_2^i \times \Delta\vec{S}_3^i + \vec{\omega}_2^i \times \left(\vec{\omega}_2^i \times \Delta\vec{S}_3^i\right)$$
$$+2\vec{\omega}_2^i \times \Delta\dot{\vec{S}}_3^i + \Delta\ddot{\vec{S}}_3^i \tag{4b}$$

Newton-Euler dynamic equations state that the sum of forces and torques acting on each link cause the variations in linear and angular momentums. The proposed recursive algorithm takes into account the forces acting on the end and the beginning of each link as well as its mass m_j^i and moment of inertia $\mathbf{I}_j^i$. In addition, quadrotor mass and moment of inertia are marked m_q and $\mathbf{I}_q$ respectively. The sum of these forces, including the force of gravity, causes the change of linear momentum. The same rule can be further extended to calculate the relationship between the torques and angular momentums, yielding the following Newton-Euler equations of motion:

$$\vec{F}_3^i = \vec{F}_2^i = \vec{F}_c^i + m_3^i\vec{v}_3^i +$$
$$m_3\left(\vec{\omega}_3^i \times \left(\vec{\omega}_3^i \times \Delta\vec{R}_{3,C_3}^i\right) + \vec{\omega}_3^i \times \Delta\vec{R}_{3,C_3}^i\right)$$
$$\vec{F}_1^i = \vec{F}_2^i + m_q^i\vec{v}_1^i \tag{5a}$$

$$\vec{\tau}_3^i = \vec{\tau}_2^i = \vec{\tau}_c - \Delta\vec{R}_{2,C3}^i \times \vec{F}_3^i +$$
$$+\Delta\vec{R}_{3,C3}^i \times \vec{F}_c^i + \mathbf{I}_3^i\vec{\omega}_3^i + \vec{\omega}_3^i \times \left(\mathbf{I}_3^i\vec{\omega}_3^i\right)$$
$$\vec{\tau}_1^i = \vec{\tau}_2^i + \Delta\vec{R}_{1,C1}^i \times \vec{F}_2^i +$$
$$+\mathbf{I}_q\dot{\vec{\omega}}_q + \vec{\Omega}_q \times \left(\mathbf{I}_q\vec{\Omega}_q\right) \tag{5b}$$

C. MM-UAV dynamics in two realistic scenarios

Previous equations, given in vector form, show the full complexity of MM-UAV dynamics. Furthermore, having multiple arms adds to the degree of complexity to the overall problem. In order to simplify these equations, MM-UAV dynamics are reduced down to two realistic scenarios: *Manipulation stage*- where the mobile manipulator performs a certain task while the quadrotor remains hovering in place; *Flying stage*- where MM-UAV flies to a desired position, while the manipulator arms remain still in a stowed configuration.

Observing MM-UAV dynamics through these two scenarios allows the following simplifications to take place: while in *Manipulation stage*, the quadrotor remains hovering, and thus its motion and rotation can be neglected; *Flying stage* is exactly the opposite scenario, where the fact that the manipulator stays stowed means that it can be observed as a part of the MM-UAV fuselage. Solving equations (5) and (5) for $q_i^j, \ddot{q}_i^j; i \in (2,3), j \in (1,2,3)$ yields the following equation for MM-UAV dynamics:

$$\begin{bmatrix} F_q \\ \tau_q \end{bmatrix} = \begin{bmatrix} m_T\mathbf{I} & -\mathbf{A} \\ \mathbf{A} & \mathbf{J}_T - \mathbf{B} \end{bmatrix} \begin{bmatrix} \dot{\vec{v}} \\ \dot{\vec{\omega}} \end{bmatrix} + \begin{bmatrix} \mathbf{C} \\ \vec{\Omega}_q \times (\mathbf{J}_T - \mathbf{D})\,\vec{\Omega}_q \end{bmatrix} \tag{6}$$

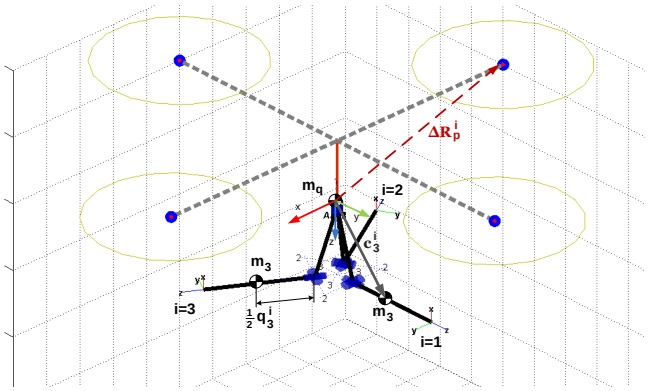


Fig. 6: MM-UAV Matlab model

where m_T is the total mass of MM-UAV, and $\mathbf{J}_T$ the total moment of inertia given in (7), observed from the center of mass, of each aircraft part (i.e. quadrotor and manipulator arms). Furthermore, expressions contained in symbols $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are given in (8). The equations show how the manipulator adds to the moment of inertia of the total aircraft. Additional terms in (8) have to be considered because the arms are displaced from the center of mass. The arm notating superscript i is assumed, but ommited for simplicity this paragraph.

$$\mathbf{J}_T = \mathbf{J}_q + m_3\sum_{i=1}^{3} q_3^i \tag{7}$$

$$\mathbf{A} = m_3\sum_{i=1}^{3} \vec{c}_3^{\,i}\times \tag{8a}$$

$$\mathbf{B} = m_3\sum_{i=1}^{3} \vec{c}_3^{\,i} \times \vec{c}_3^{\,i}\times \tag{8b}$$

$$\mathbf{C} = m_3\sum_{i=1}^{3} \vec{\Omega}_q \times \left(\vec{\Omega}_q \times \vec{c}_3^{\,i}\right) \tag{8c}$$

$$\mathbf{D} = m_3\sum_{i=1}^{3} \vec{\Omega}_q \times \left(\vec{\Omega}_q \times \vec{c}_3^{\,i}\right) \tag{8d}$$

In *Manipulation stage*, the quadrotor remains in hover, so that its translation and rotation speeds and accelerations, $\vec{V}_q$, $\dot{\vec{V}}_q$, $\vec{\Omega}_q$, $\dot{\vec{\Omega}}_q$ respectively, can be set to zero. The equations of motion are shown in 9. The contact force from the manipulation is marked with a vector $\vec{F}_c = \begin{bmatrix} f_{cx} & f_{cy} & f_{cz} \end{bmatrix}^T$.

$$\vec{F} = \frac{3m_3}{2}\begin{bmatrix} 0 & 0 & 0 & 0 \\ -q_3c_2 & -q_3s_2 & \frac{c_2}{3} & -\frac{2}{3} \\ -q_3s_2 & q_3c_2 & \frac{s_2}{3} & +\frac{2}{3} \end{bmatrix}\begin{bmatrix} \dot{q}_2^2 \\ \ddot{q}_2 \\ \ddot{q}_3 \\ \dot{q}_2\dot{q}_3 \end{bmatrix} + \begin{bmatrix} f_{cx} \\ f_{cy} \\ f_{cz} \end{bmatrix} \tag{9a}$$

$$\vec{\tau} = m_3\begin{bmatrix} \frac{2q_3{}^2}{3}\ddot{q}_2 + \frac{q_3}{2}\dot{q}_2\dot{q}_3 \\ 0 \\ 0 \end{bmatrix} + q_3\begin{bmatrix} 0 & s_2 & c_2 \\ s_2 & 0 & 0 \\ c_2 & 0 & 0 \end{bmatrix}\begin{bmatrix} f_{cx} \\ f_{cy} \\ f_{cz} \end{bmatrix} \tag{9b}$$

IV. Simulation

In order to fully verify the mathematical model derived in the previous section, MM-UAV is modeled and simulated in Matlab using the Robotics Toolbox [16]. Each arm is modeled as a separate serial chain of links using the parameters derived in Sec. III-A and shown in Table II. The Robotics Toolbox uses the recursive Newton-Euler method, similar to the method used in Sec. III-B.

For all three arms, base link 0 represents the body of the quadrotor. Therefore, its dynamics parameters (i.e. mass and moment of inertia) correspond to the quadrotor's dynamic parameters. Actual values, shown in table I are calculated using a CAD model of MM-UAV and all the simulations are run using these values.

Manipulator arms, on the other hand, are modeled as a simplified prismatic link. This means that we consider the link to be infinitesimally small in all directions, except in the actual direction of translation motion, where its length is equal to translation variable q_3^i [17]. This simplification allows us to write the following expression for the moment of inertia of arms with respect to their center of mass:

$$\mathbf{I}_3^i = \frac{m_3 {q_3^i}^2}{12} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

Simulation results shown in Fig. 7 were obtained using the Robotics Toolbox model of MM-UAV and under initial conditions of the aircraft set to zero. Under the assumption that the quadrotor remains hovering while performing manipulation tasks, Fig. 7 shows forces and torques induced from manipulator movement, acting on the body of the quadrotor. The two bottom images show the accelerations and the actual values of joint variables. It is assumed that the joint accelerations are ideally applied as shown in the last image.

In the first three seconds of the simulation, all three manipulator arms are simultaneously driven. It is at this time that the quadrotor experiences the least amount of disturbance from the manipulator motion. The only actual disturbance is a force in the z direction, as can be seen in Fig. 7. This is due to the fact that all the forces and torques in the x and y directions tend to cancel each other out because of the manipulator symmetric construction. Later in the simulation, each arm is moved separately. This causes dynamic disturbances and generates both forces and torques on the body. Moreover, because the manipulator loses its symmetry, there exists a constant, static torque even when the arms are still.

A second simulation is run, with initial conditions of the aircraft set so that it rotates with a constant yaw rotation speed. This was done to verify aircraft dynamics when it is simultaneously manipulating objects and rotating. In reality, roll and pitch angles of the aircraft should remain zero at all times when the quadrotor is hovering. Yaw angle, on the other hand, can vary, and thus be used as an additional manipulator degree of freedom. Equations (11) and (12) show additional terms in torques and forces produced from

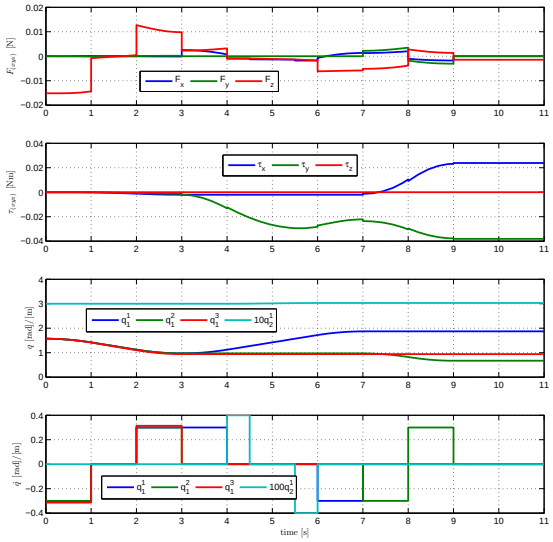


Fig. 7: Forces, torques, and joint movements in MM-UAV Matlab Simulation, with a stationary quadrotor: Rotating arms simultaneously for 3 seconds, after which each arm moves separately.

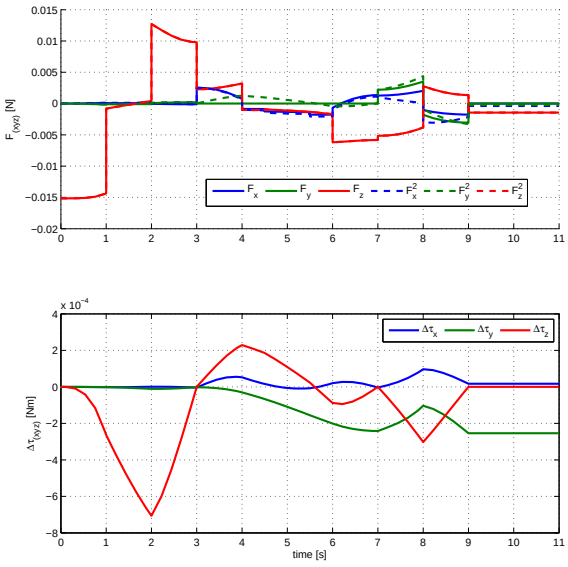


Fig. 8: MM-UAV Matlab Simulation, quadrotor with constant yaw rotation speed: Rotating arms simultaneously for 3 seconds, after which each arm moves separately. Showing absolute force and torque values relative to stationary experimental results.

MOCAP Data			
Number of Cameras	Refresh Rate	Pos. Resolution	Speed Resolution
18	10 Hz	1 mm	0.1 m/s

TABLE III: OptiTrack Motion Capture System Data

the angular velocity ω_z. The results of this simulation are shown side by side with previous results in Fig. 8. They show consistency with equations (11) and (12). The bottom plot actually shows the difference between the torques in hovering and rotating aircraft.

$$\vec{F}^i = \vec{F}^i|_{\omega_z=0} + \begin{bmatrix} m_3\omega_z\left(3\dot{q}_2 q_3 s_2 - \dot{q}_3 c_2\right) \\ -\frac{3}{2}m_3{\omega_z}^2 q_3 c_2 \\ 0 \end{bmatrix} \tag{11}$$

$$\vec{\tau}^i = \vec{\tau}^i|_{\omega_z=0} + \begin{bmatrix} \frac{1}{3}m_3{\omega_z}^2{q_3}^2 sin\left(2q_2\right) \\ \frac{1}{6}m_3\omega_z q_3 s_2\left(8\dot{q}_2 q_3 s_2 - 3\dot{q}_3 c_2\right) \\ \frac{1}{6}m_3\omega_z q_3 c_2\left(3\dot{q}_3 c_2 - 8\dot{q}_2 q_3 s_2\right) \end{bmatrix} \tag{12}$$

V. MM-UAV Controllers

For the low-level control, an off-the-shelf IMU (inertial measurement unit) is used to control the yaw, pitch, and roll of the quadrotor through gyroscopic sensor data. Integrated with the low-level controller, a high-level autopilot controller using a motion capture system that provides x, y, z position and velocity information is proposed. Motion capture is based on vision markers placed just above the center of mass of the vehicle. An operator controls the MM-UAV using a standard joystick and keyboard inputs. An autopilot program, implemented on a PC, uses available motion capture data (i.e. x, y, z position and speed) to navigate to the desired setpoint. Furthermore, the autopilot feeds forward the manipulator motion control signals sent via the joystick. Control schematics of this computer aided control system is depicted in Fig. 9.

The motion capture system used in this research is based on 18 V100:R2 OptiTrack cameras connected to a PC running Arena Software. The PC sends the data via the NatNet protocol. The controller PC implements a C++ class to read in the data being streamed from the motion capture computer. Fast Ethernet speed allows for a fast connection with practically no lag between sending and receiving data. Table III shows the parameters of the motion capture system.

A. UAV Controller

Manual control of the MM-UAV prototype has proven to be extremely difficult, even for the more experienced RC pilots. Therefore, a simple manipulation task would become unfeasible without computer aided control.

The top level position controller uses the motion capture position data to implement a two stage cascade position controller. Differentiating speed data, one is able to measure the speed of the aircraft. Obtaining speed data using a motion capture system is not a straight forward process because it is limited by the accuracy and sample time of the system. On the other hand, having separate speed and position loops enables separate tuning of the controller parameters. By disabling the position control loop, the speed control loop parameters can be tuned. When the speed loop is tuned, position control can be parametrized.

Taking into account the dynamics of the system a simplified Proportional controller for the corresponding speed loop, and a Proportional-Integral controller for the position loop is proposed. Effectively, this transforms the proposed two stage cascade controller structure into a single PI-Derivative position controller. The proposed PI-D controller equation (13) implies that the control difference $\vec{e}$ is taken through the proportional and integration channels, while the derivative channel is connected directly to speed measurement of the quadrotor $\vec{v}_0$. Equations are written in vector form because they are applied to x, y, z positions in 3D space. Controller structure is shown in Fig. 10.

$$\vec{u} = K_P\vec{e} + K_I\int\vec{e} + K_D\vec{v}_0 \tag{13}$$

K_P, K_I, and K_D are the PI-D control gains. Error vector $\vec{e}$ is the difference betwenn the x, y, z position data and the actual setpoint reference values, and $\vec{v}$ is the speed measurement. Due to the fact that the derivation channel (i.e. aircraft speed) is error sensitive, leading the control difference directly through can cause serious problems and possibly damage the aircraft. On the other hand, position data is much more reliable and can be used directly in the control loop. PI-D control is implemented as a C++ class, running at a 10 Hz refresh rate.

B. Manipulator control

Each MM-UAV arm consists of a servo powered rotation joint, and a DC powered linear actuator controlling the extension of the arms. Controls are implemented on a Robostix

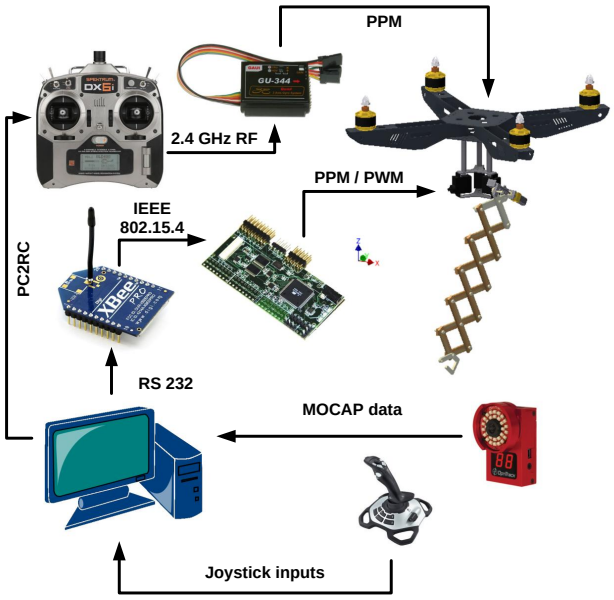


Fig. 9: MM-UAV Control Scheme

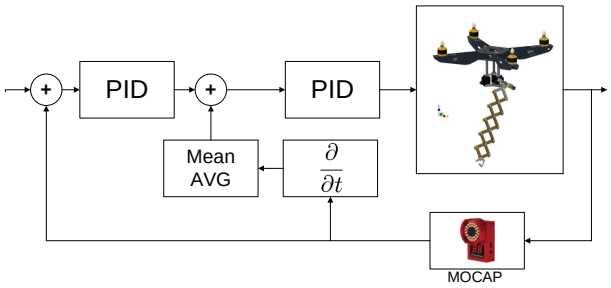


Fig. 10: MM-UAV Control Structure

board programmed with 3 PPM (pulse-pulse modulation) and 3 PWM (pulse-width modulation) outputs. PPM outputs are connected to the rotation servos which implement position control. Therefore, Robostix generates the desired rotation setpoints for each servo motor. On the other hand, DC motors are controlled through the applied PWM signals that varies their speed. Actual position control is not implemented. The operator controls the extension of the arms using visual contact with the aircraft.

VI. Experimental Results

In this section, the results of one positioning and grabbing experiment are shown (Fig. 14). The MM-UAV is first lifted off the ground using manual control. After liftoff, the autopilot control is turned on. It first navigates the quadrotor to position $(500, -300)$, followed by a change of set point to $(0, 100)$. The results are shown in Figs. 11, 12, and 13.

As the quadrotor takes off under manual control, a transition between manual and autopilot control takes place. It can be seen on Fig. 13 as a peak in yaw, pitch, and roll data. The effect of this transition is plotted in Fig. 11, where initial speed and orientation of the aircraft causes it to move in the direction different from the desired direction to the first set point value.

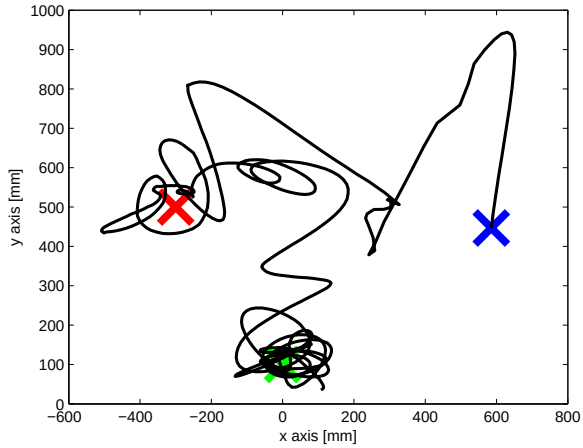


Fig. 11: MM-UAV Trajectory

Simulation results from Sec. IV show that the manipulator produces the least amount of disturbance when all three

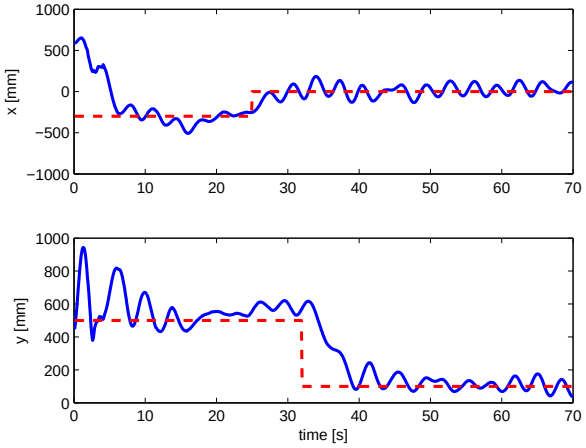


Fig. 12: MM-UAV Changes in Setpoint

arms are moving simultaneously. Using these results, the actual grabbing manipulation in this experiment is achieved by simultaneous arm movement. Nevertheless, experimental results show that the quadrotor exhibits oscillatory behavior. This is mainly due to the inherent limitations of the proposed controller structure.

VII. Conclusions

We have presented a model and control methodology to perform mobile manipulation using an unmanned aerial vehicle and multiple manipulator arms. By characterizing the forces and torques experienced by the quadrotor body during both flight and manipulation, we have developed a controller to compensate for reactionary forces. Our simulation and results indicate that was can significantly reduce the oscillatory effects in order to successfully manipulate a target object. Motion capture facilitates aircraft stability but our prototype is not constrained by a vision-based system.

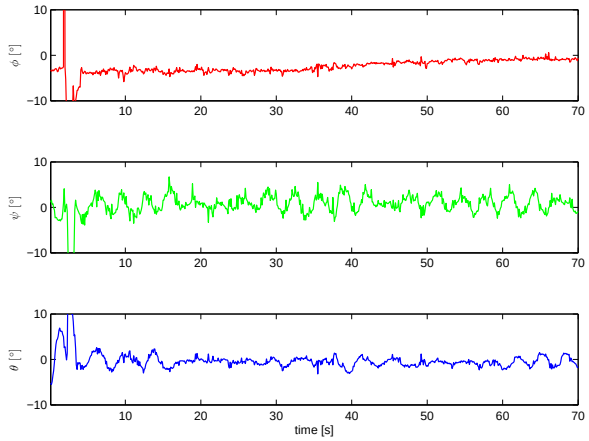
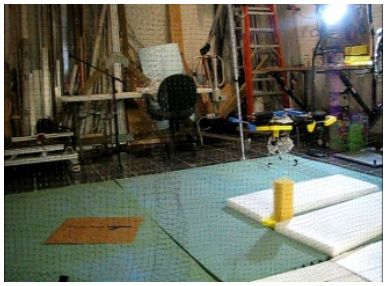
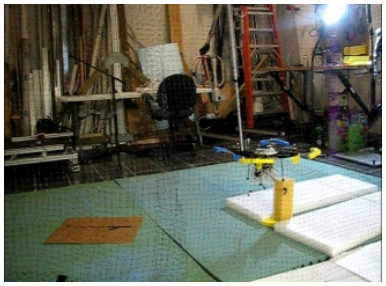


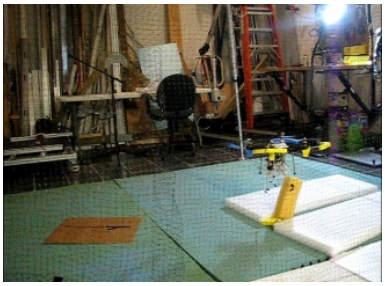
Fig. 13: MM-UAV Yaw, Pitch, and Roll Data



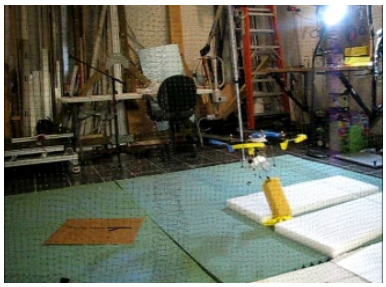
(a) Approach



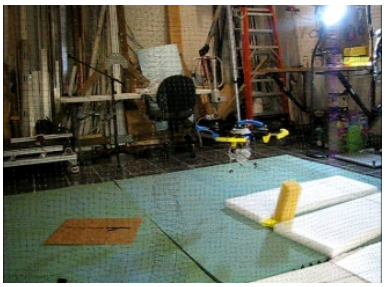
(b) Grab



(c) Move



(d) Release



(e) Hover

Fig. 14: Images from MM-UAV manipulation test experiments

REFERENCES

[1] M. Bernard and K. Kondak, "Generic slung load transportation system using small size helicopters," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '09*, 2009, pp. 3258–3264.

[2] N. R. Kuntz and P. Y. Oh, "Towards autonomous cargo deployment and retrieval by an unmanned aerial vehicle using visual servoing," *ASME Conference Proceedings*, vol. 2008, no. 43260, pp. 841–849, 2008.

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

[4] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2011, pp. 2491–2498.

[5] C. M. Korpela, T. W. Danko, and P. Y. Oh, "MM-UAV: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent and Robotic Systems*, vol. 65, no. 1-4, pp. 93–101, 2012.

[6] C. Korpela, M. Orsag, T. Danko, B. Kobe, C. McNeil, R. Pisch, and P. Oh, "Flight satbility in aerial redundant manipulators," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2012, to Appear.

[7] C. M. Korpela, T. W. Danko, and P. Y. Oh, "Designing a system for mobile manipulation from an unmanned aerial vehicle," in *Proc. IEEE Conf. Technologies for Practical Robot Applications (TePRA)*, 2011, pp. 109–114.

[8] F. Aghili, "Optimal control of a space manipulator for detumbling of a target satellite," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '09*, 2009, pp. 3019–3024.

[9] D. N. Dimitrov and K. Yoshida, "Momentum distribution in a space manipulator for facilitating the post-impact control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2004)*, vol. 4, 2004, pp. 3345–3350.

[10] M. Ishitsuka and K. Ishii, "Modularity development and control of an underwater manipulator for auv," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems IROS 2007*, 2007, pp. 3648–3653.

[11] D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, and M. Uchiyama, "Reaction null-space control of flexible structure mounted manipulator systems," *IEEE Trans. Robot. Autom.*, vol. 15, no. 6, pp. 1011–1023, 1999.

[12] G. M. Hoffmann, H. Huang, S. L. Wasl, and E. C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *In Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2007.

[13] S. McMillan, D. E. Orin, and R. B. McGhee, "Efficient dynamic simulation of an underwater vehicle with a robotic manipulator," *IEEE Transactions On Systems, Man, and Cybernetics*, vol. 25, pp. 1194–1206, 1995.

[14] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[15] R. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics, and Control (2nd Edition)*. Springer, 2010.

[16] P. Corke, "A robotics toolbox for MATLAB," *IEEE Robotics and Automation Magazine*, vol. 3, no. 1, pp. 24–32, Mar. 1996.

[17] J. Craig, *Introduction to robotics: mechanics & control*, ser. Introduction to robotics : Mechanics & control / John J ... Craig. Addison-Wesley Pub. Co., 1986.