TOWARDS PERFORMING REMOTE MANIPULATION USING AN AUTONOMOUS AERIAL VEHICLE

Tony Baltovski[1], Scott Nokleby[1], Remon Pop-Iliev[1]
[1]*Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, ON, Canada*
Email: tony.baltovski@uoit.net; scott.nokleby@uoit.ca; remon.pop-iliev@uoit.ca

ABSTRACT

This work presents a system for performing remote manipulation using an Unmanned Aerial Vehicle (UAV) as part of an Aerial Manipulators System (AMS). The system utilizes a quad-rotor UAV with a manipulator mounted on top to perform the desired manipulation. Despite being a scaled down prototype, the system should easily be scaled up to a larger quad-rotor for practical use. The preliminary results show the successful addition of manipulation capabilities to a quad-rotor and an autonomous system to utilize the manipulator.

Keywords: Unmanned Aerial Vehicle (UAV); quad-rotor; manipulator; Aerial Manipulator System (AMS).

1. INTRODUCTION

1.1. Background

Performing maintenance in remote areas has proven to be no simple task, as is evident by the processes required for maintaining and servicing live, high-voltage power transmission lines. The power transmission towers span large distances in uneven terrain which places human workers in danger's way. The use of an autonomous robotic system can provide a more safe and efficient method of maintaining power transmission lines. There have been various solutions proposed such as boom trucks with robotic manipulators [1], hanging trolley robots [2] and autonomous helicopters for inspection [3]. The difficulty in servicing remote transmission towers are the large distances that must be covered and the ability to perform the necessary maintenance tasks.

1.2. Proposed Robotic System

The Aerial Manipulator System (AMS) is a complete system for servicing and maintaining power transmission towers which can be seen in Fig. 1 [4]. A dexterous manipulator is placed on an Unmanned Aerial Vehicle (UAV) platform to allow for aerial manipulation. Another part of the AMS, a stable docking solution is used to hold the UAV platform with manipulator to the tower. The UAV platform is tethered to a mother-ship which provides power that will allow for prolonged task execution. The mother-ship will be an efficient aerial vehicle capable of long distance flights and provide a storage for the UAV platform, as well as other necessary components. The AMS will allow for comprehensive aerial manipulation. The purpose of this work is to demonstrate the ability to fly a UAV with a robot manipulator to perform a task. This is the first stage in the long term development of the AMS.

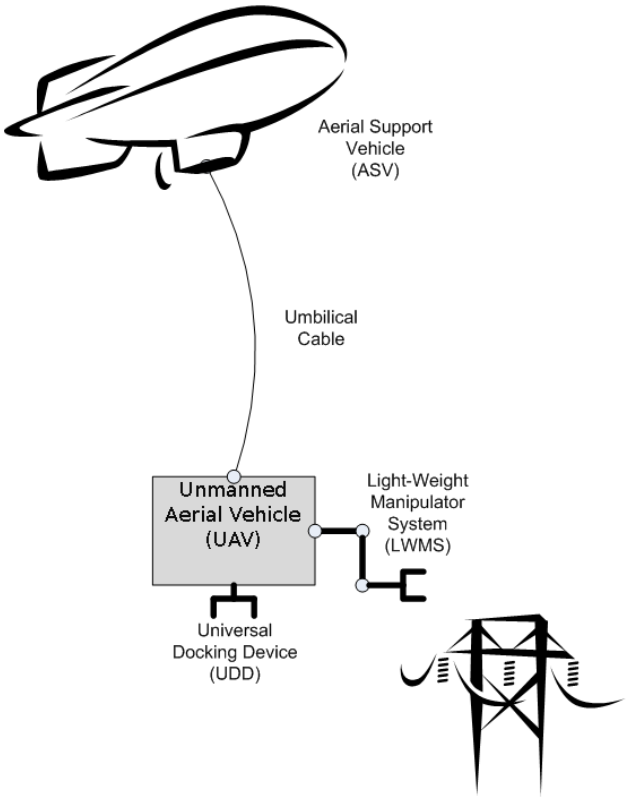

Fig. 1. Overview of AMS concepts

Fig. 2. The quad-rotor chassis for UAV platform

1.3. Existing Aerial Vehicles with Manipulation Capabilities

The recent advances in multi-rotors have provided a stable platform for adding manipulation capabilities. Multi-rotors are ideal for manipulation as they hover in place and perform Vertical Take-Off and Landing (VTOL). Early implementations of this concept comprised a simple gripper underneath the quad-rotor [5]. The quad-rotor was able to hover over an object and grip it in flight. Similarly, a simple serial chain manipulator was added to a quad-rotor to perform manipulation in-flight [6], which was later replaced by a four degree-of-freedom (DOF) manipulator [7]. Additionally, another 4-DOF manipulator was added to provide additional dexterity [8]. The focus of the previous works in the literature were on in-air manipulation, whereas the aim of the AMS is stationary manipulation. In addition, the goal of the AMS system's UAV is to have it completely autonomous. As such, the quad-rotor developed in this work is entirely autonomous.

2. EXPERIMENTAL SETUP

2.1. Vehicle Setup

The current test vehicle is based on a 3DRobotics Quad frame as a chassis as shown in Fig. 2. It utilizes four powerful Scorpion SII-2215-1127Kv brushless motors connected to two sets of opposing GEMFAN 10x4.5 inch propellers to produce enough thrust to carry the manipulator. The four Next Level Multi-rotor 30A Electronic Speed Controllers (ESCs) with SimonK firmware connect to a 3DRobotics APM 2.6 autopilot. The APM 2.6 is using the open-source ArduCopter firmware [9] for stabilization using an inertial measurement unit (IMU) consisting of a linear accelerometer, gyroscope and magnetometer. The vehicle is powered by two Turnigy 5000 *mAH* 3S Lithium polymer batteries, currently off-board. The autopilot traditionally accepts remote control (RC) signals from a transmitter, however, it is possible to send and receive data using a serial interface (USB). The MAVLink protocol [10] is used for communication over serial.

2.2. Manipulator Setup

The Lynxmotion AL5D was selected since it is a readily available small scale manipulator as can be seen in Fig. 3. The AL5D uses five servos to create a 4-DOF manipulator as well as a gripper. The servos are controlled by the Lynxmotion SSC32U servo controller which communicates over serial to accept joint angle commands. The modified Denavit-Hartenberg parameters for the AL5D (schematic shown in Fig. 4), which can be seen in Table 1, were used to derive the inverse kinematics [11]. Given a point $P = (X,Y,Z)$, gripper angle α and the link lengths l_i, $i = 1$ to 4 , the inverse kinematics are as follows for each joint:

$$\theta_1 = atan2(\pm Y, \pm X) \tag{1}$$

$$\theta_2 = atan2(b_2, a_2) \pm atan2(\sqrt{{a_2}^2 + {b_2}^2 - {k_2}^2}, k_2) \tag{2}$$

where

$$a_2 = X\cos(\theta_1) + Y\sin(\theta_1) - l_4\cos(\alpha) \tag{3}$$

$$b_2 = Z - l_1 - l_4\cos(\alpha) \tag{4}$$

$$k_2 = \frac{{a_2}^2 + {b_2}^2 + {l_2}^2 - {l_3}^2}{2l_2} \tag{5}$$

$$\theta_{23} = atan2(b3, a3) \tag{6}$$

where

$$a_3 = \frac{a_2 - l_2\cos\theta_2}{l_3} \tag{7}$$

$$b_3 = \frac{b_2 - l_2\sin\theta_2}{l_3} \tag{8}$$

$$\theta_3 = \theta_{23} - \theta_2 \tag{9}$$

$$\theta_4 = \alpha - \theta_2 - \theta_3 \tag{10}$$

where θ_i is the angle of the $i$ith joint. There are two solutions for joint 1 and joint 2, however, since the servos are only able to rotate 180 degrees, certain solutions will not be used. Note that atan2 denotes a quadrant corrected arctangent function.

Table 1. Modified Denavit-Hartenberg Parameters for the AL5D

F_{i-1}	a_{i-1}	α_{i-1}	d_i	θ_i	F_i
0	0	0	l_1	θ_1	1
1	0	$\pi/2$	0	θ_2	2
2	l_2	0	0	θ_3	3
3	l_3	0	0	θ_4	4
4	l_4	0	0	0	G

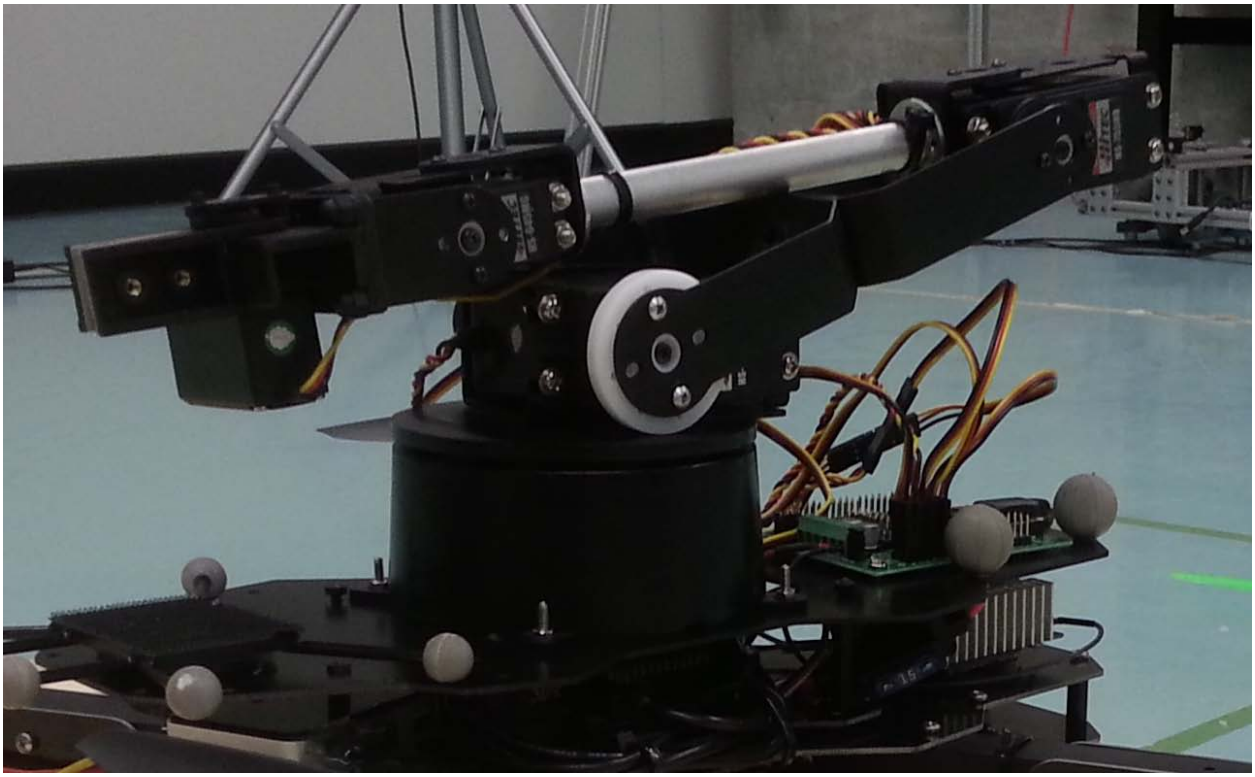

Fig. 3. Lynxmotion AL5D manipulator

2.3. Control Setup

The control system is implemented using the Robot Operating System (ROS) framework [12]. The `mavros` driver is used to translate the MAVLink protocol to a standard ROS message allowing for control by overriding the RC channels for roll, pitch and yaw rates, and throttle which is normalized thrust from 0 to 1. The APM autopilot will perform on-board stabilization while the off-board, position control is performed by a ROS based controller through the `mavros` driver as shown in Fig. 5. This ensures that quad-rotor will remain relatively level and stationary when there is no input being commanded. The RC values range from 1,000 to 2,000 where 1,500 is the middle for angular rates and 1,000 is the minimum for throttle. Since the autopilot performs the stabilization control, only a position controller with yaw was needed. A Proportional, Integral, and Derivative (PID) controller was created based on a set point pose which does not include roll and pitch since it is desired for the UAV to be level. The controller uses the error in the pose to determine the needed RC values. The RC channels are either an angular rate (roll, pitch, yaw) or an absolute (throttle). For the rate RC channels, the middle channel value is zero angular rate. To move in the X direction, a RC pitch rate must be sent to the controller which causes the motion in the X direction. A PID controller was created for error in the planar (XY) position and yaw using:

$$RC_{rate} = RC\_MID - K_{p,rate}error - K_{i,rate}\int_0^t error - K_{d,rate}\Delta t \tag{11}$$

where the RC rates where roll, pitch and yaw. Since the throttle was absolute, a slightly different controller was needed to use the error in height to convert to RC throttle values as :

$$RC_{throttle} = RC\_MIN + K_{p,throttle}error + K_{i,throttle}\int_0^t error + K_{d,throttle}\Delta t \tag{12}$$

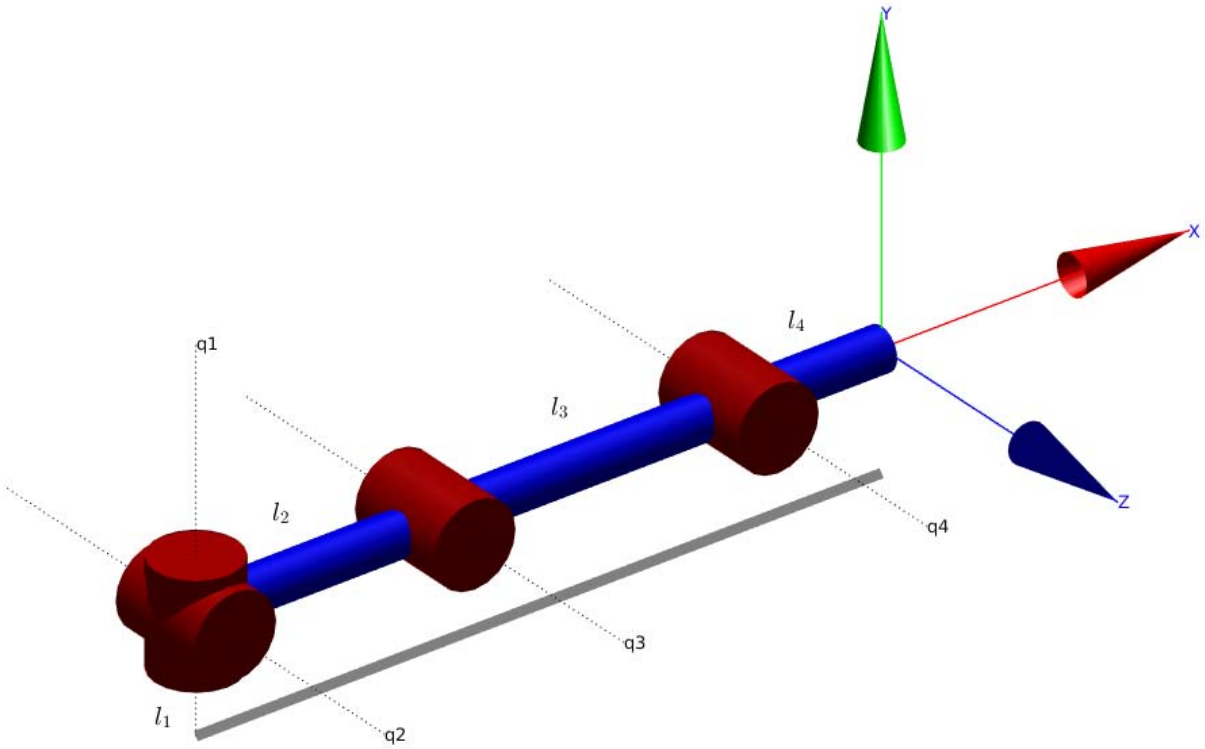

Fig. 4. Lynxmotion AL5D schematic in zero displacement configuration where q_i is the given joint

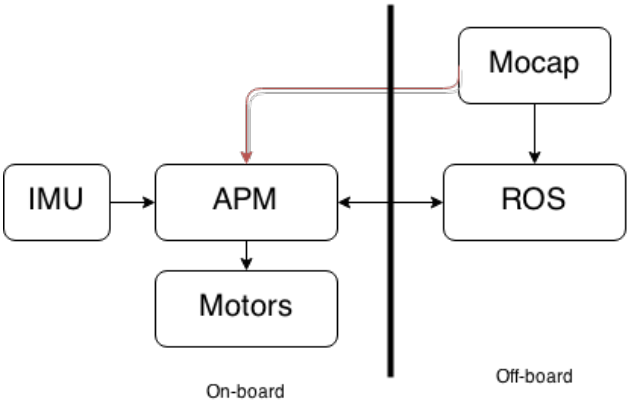

Fig. 5. A general overview of the control system

During mode changes, the RC_MIN is reset to the throttle at the current height to keep the quad-rotor from descending.

The UAV's autonomy planner uses a target pose to generate a simple direct trajectory which consists of a take-off to a certain clearance height, then a direct path over to the desired pose and finally landing on the desired pose. Once the take-off is completed, a new set point is generated along the trajectory but at a smaller step. The set-point is relative to the quad-rotor's current pose and is a small step in the trajectory. This allows for accounting for errors in the position at the current step. This process is autonomous with the use of a joystick to arm, initialize task and an emergency stop for safety. Pressing the trajectory button on the joystick starts the autonomous sequence of getting to the desired pose. To perform launch, the UAV enters take-off mode. Take-off mode blocks all set points until a clearance height is achieved from the ground. The desired planar position and yaw is determined from the initial pose of the vehicle. Once the vehicle has achieved the clearance height, the mode automatically switches to hover mode. In hover mode, it is possible to change the set point and this is done to move along points on the trajectory. Once the desired end point of the trajectory is reached, the mode automatically changes to landing mode to perform the landing. Landing mode is like take-off mode where it blocks set points and uses the last desired planar position and yaw.

Once the UAV has landed at the desired pose, the manipulator is given the pose of the target for manipulation. The planning for the manipulation starts with the manipulator rising from its tucked configuration to a configuration where no joint has a chance of colliding with the quad-rotor's chassis. The first joint is moved prior to moving the rest of the joints, this process ensures there is no possibility of collisions with the quad-rotor's chassis. Following this, the remaining joints are moved to the desired pose and the manipulation task is executed. Upon returning to the tucked configuration, the previous steps are completed in reverse.

3. PRELIMINARY RESULTS

3.1. Test Setup

For this experiment, the NaturalPoint OptiTrack motion capture system shown in Fig. 6 with 12 V100:R2 cameras was used to determine the pose of the UAV. The cameras track infra-red markers on the chassis and this can be seen in Fig. 7. The data is processed on a Windows-based PC using NaturalPoint Motive software and then broadcasts the data in the NatNet protocol at 100 Hz. Using a gigabit Ethernet connection, the NatNet data is sent to a Ubuntu PC where the `mocap_optitrack` ROS node decodes the data. The data from the motion capture system used is at 25 Hz to avoid overfilling the serial communication to the UAV platform. Once the pose data is obtained from the motion capture system, the `ams_manager` and `ams_quad_controller` are constantly subscribing to the data. Also, the `ams_trajectory_planner`

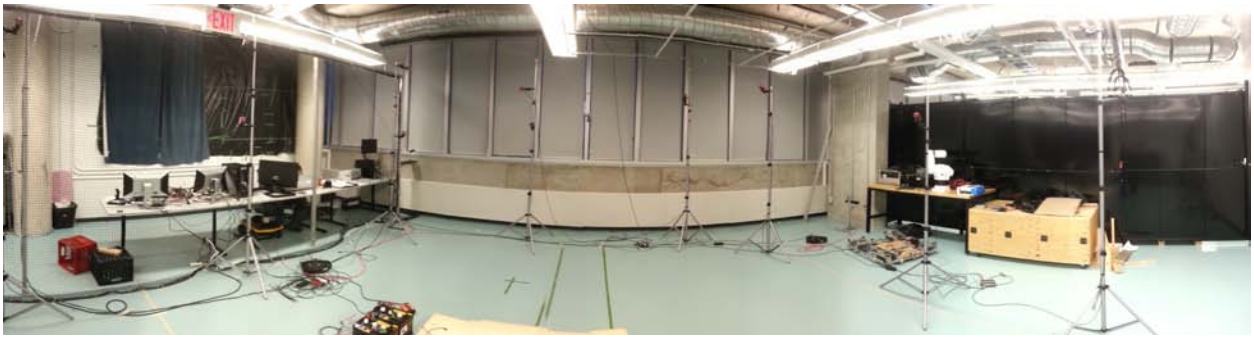

Fig. 6. Motion capture test cell

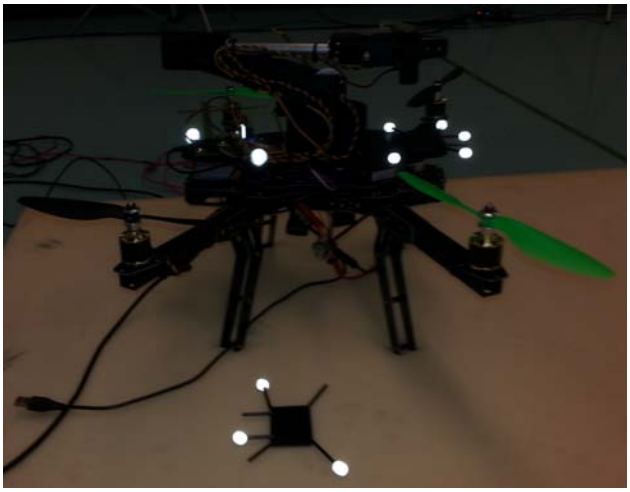

Fig. 7. Motion capture markers on quad-rotor

computes a trajectory using the initial pose of the UAV platform to the target and passes the data to the `ams_manager` which is responsible for automatically changing modes and the set points and controlling the RC values as previously outlined. Set points are changed using a service call to the `ams_quad_controller`. Finally, the RC data is then sent to the driver `mavros` which is renamed `ams` for clarity. The overall control system for this test experiment can be seen in Fig. 8.

3.2. Flight Test

A simple test was constructed that consisted of the quad-rotor dropping off a payload, which was comprised of a traditional nine volt battery, to a target. The payload was placed in the gripper of the manipulator initially since the gripper is rather small and the servos are not able to perform precise pick-and-place tasks. The quad-rotor would start away from the desired target with a different yaw orientation which was marked using a triangular set of infra-red markers shown in Fig. 7. The ground station operator would then arm the quad-rotor and enable the action. The entire flight test can be seen in Fig. 9 which performs the actions as described in the previous section. A video of this test can be found at `https://www.youtube.com/watch?v=Fp6eQ-Q4tfA`. As shown, the quad-rotor was successfully able to fly to the target and the manipulator was able to release the payload over a marked target. The operator is able to visualize the current pose of the UAV and its path along the trajectory as shown in Fig. 10.

During tests flights, the quad-rotor struggled to produce enough thrust to maintain an altitude. This also

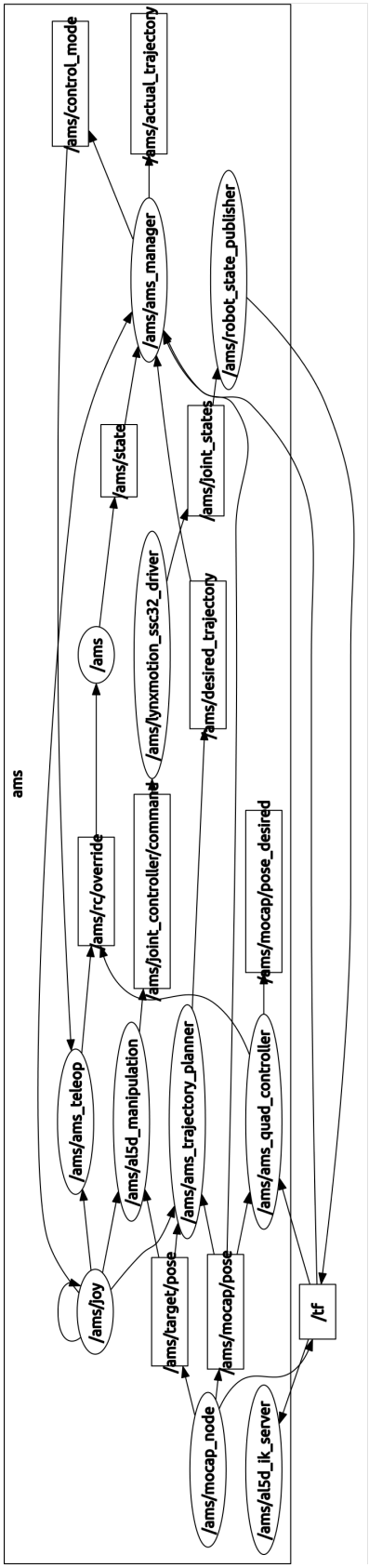

Fig. 8. Overview of Control system in ROS

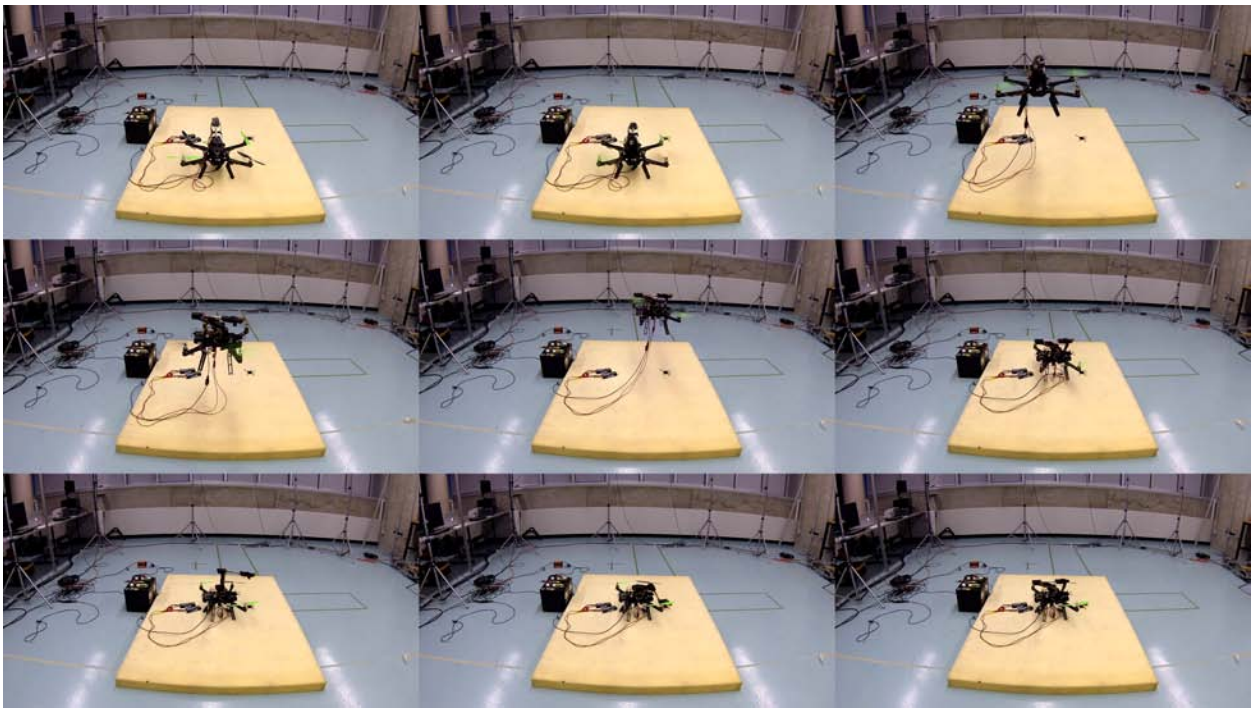

Fig. 9. The test flight of the UAV platform with the manipulator

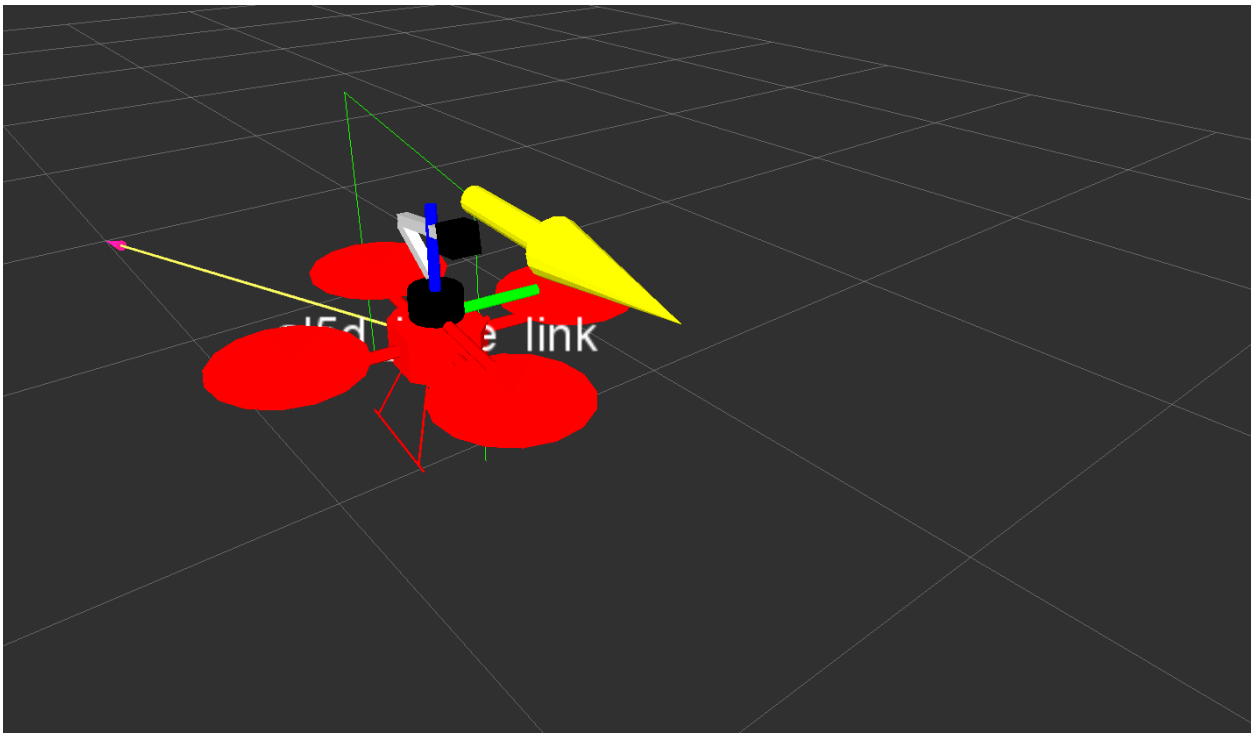

Fig. 10. The ground station display

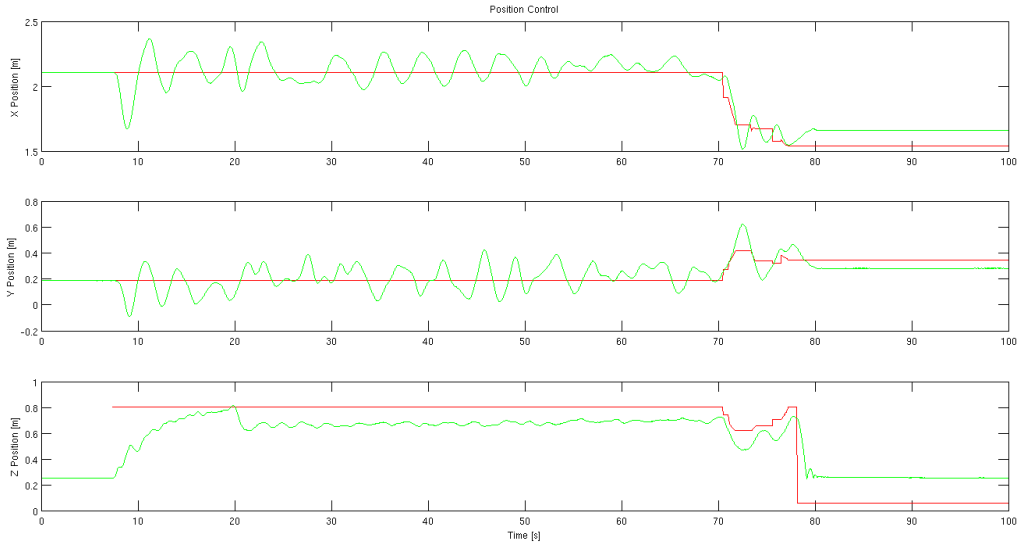

Fig. 11. The actual position versus the desired position for the quad-rotor

caused poor pose control since the motors were constantly operating at maximum power to try to maintain the desired height. Corrections in planar position and yaw caused the quad-rotor to lose altitude because in order to perform the manoeuvres, the motors had to be slowed down. The position tracking can be seen in Fig. 11 and it can be seen that the UAV does not achieve the desired height while following the trajectory. It should be noted that these results are preliminary, but the results show that the proof-of-concept prototype was able to perform the desired manipulation task.

4. CONCLUSIONS AND FUTURE WORK

The flight test successfully demonstrated the ability to perform remote manipulation using a UAV as a mobile platform. Despite designing the system to accommodate the additional mass of the manipulator, due to certain aerodynamic factors that were not considered, the resulting system was slightly under-powered, resulting in slightly poor pose control. The next iteration of the quad-rotor platform will be designed to carry a larger payload and have sufficient power to perform all necessary movements. In addition, additional functionalities will be added to the system such as re-attempting landing if the desired target is not within range of the arm. Nonetheless, the current system was able to complete the manipulation task.

The presented system shows the possibilities of having an autonomous UAV with manipulation capabilities for the AMS. The motion capture system can be removed for a more practical method of estimating the position of the quad-rotor and target using GPS and visual estimation. The current AMS UAV is only a scaled proof-of-concept that shows the concept of the AMS UAV is possible. The modular nature of the system design allows for easy scaling to a larger system. This work is a pivotal progression towards implementing the AMS for remote manipulation.

ACKNOWLEDGEMENTS

The authors would like to thank the Natural Sciences and Engineering Research Council (NSREC) of Canada for their financial support of this work.

REFERENCES

1. Aracil, R. and Ferre, M. "Telerobotics for aerial live power line maintenance." In "Advances in Telerobotics," pp. 459–469. Springer, 2007.
   URL http://www.springerlink.com/index/w0j7u5k701365h78.pdf
2. Montambault, S. and Pouliot, N. "Field experience with LineScout Technology for live-line robotic inspection and maintenance of overhead transmission networks." *2010 1st International Conference on Applied Robotics for the Power Industry (CARPI 2010)*, pp. 1–2. doi:10.1109/CARPI.2010.5624454, October 2010.
   URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5624454
3. Hrabar, S., Merz, T. and Frousheger, D. "Development of an autonomous helicopter for aerial powerline inspections." *2010 1st International Conference on Applied Robotics for the Power Industry (CARPI 2010)*, pp. 1–6. doi:10.1109/CARPI.2010.5624432, October 2010.
   URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5624432
4. Baltovski, T., Nokleby, S. and Pop-Iliev, R. "Design and Development of a Swinging Arm Mechanism for an Aerial Manipulator System." In "Mechanical Engineering and Mechatronics, 2013 ICMEM International Conference on," Vol. 1, pp. 1–7. International ASET, Toronto, Canada, 2013.
5. Ghadiok, V., Goldin, J. and Ren, W. "On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor." *Autonomous Robots*, pp. 41–68. ISSN 0929-5593. doi:10.1007/s10514-012-9286-z, March 2012.
   URL http://www.springerlink.com/index/10.1007/s10514-012-9286-z
6. Orsag, M., Korpela, C. and Oh, P. "Modeling and control of MM-UAV: Mobile manipulating unmanned aerial vehicle." *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 69, pp. 227–240. ISSN 09210296. doi:10.1007/s10846-012-9723-4, 2013.
7. Orsag, M., Korpela, C., Pekala, M. and Oh, P. "Stability control in aerial manipulation." In "2013 American Control Conference," pp. 5581–5586. ISBN 978-1-4799-0178-4. ISSN 07431619. doi:10.1109/ACC.2013.6580711, 2013.
   URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6580711
8. Korpela, C., Orsag, M., Pekala, M. and Oh, P. "Dynamic stability of a mobile manipulating unmanned aerial vehicle." *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4922–4927. ISSN 10504729. doi:10.1109/ICRA.2013.6631280, 2013.
9. "APM open source autopilot." http://ardupilot.com/.
10. Meier, L., Camacho, J., Godbolt, B., Goppert, J., Heng, L., Lizarraga, M. et al.. "Mavlink: Micro air vehicle communication protocol." *http://qgroundcontrol.org/mavlink/start*, 2013.
11. Craig, J.J. *Introduction to Robotics: Mechanics and Control*, Vol. 3. Pearson Prentice Hall Upper Saddle River, 2005.
12. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A.Y. "ROS: an open-source robot operating system." In "ICRA workshop on open source software," Vol. 3, p. 5, 2009.