DESIGN AND DEVELOPMENT OF A NEXT GENERATION OMNI-DIRECTIONAL MULTIROTOR

Christopher Baird[1], Florentin von Frankenberg[1], and Scott B. Nokleby[1]
[1] *Mechatronic and Robotic Systems Laboratory, Univeristy of Ontario Institute of Technology, Oshawa, ON, Canada*
Email: christopher.baird@uoit.ca; florentin.vonfrankenberg@uoit.ca; scott.nokleby@uoit.ca

---

ABSTRACT

Design enhancements for a second generation omni-directional multirotor system known as the OmniRaptor are presented. The first major change over the original OmniCopter was the usage of a variable pitch propeller system for the side thrusters of the OmniRaptor instead of the fixed pitch propellers used in the OmniCopter. This was done to improve the responsiveness of the system to disturbances. The second change was to move the control system to onboard the OmniRaptor to make full use of the PX4 autopilot software. Preliminary flight test data from the OmniRaptor is shown to prove its functionality and future improvements for the OmniRaptor are discussed.

Keywords: UAV; Omni-Directional; PX4 Autopilot; ROS.

CONCEPTION ET DÉVELOPPEMENT D'UN POLYROTOR OMNIDIRECTIONNEL DE NOUVELLE GÉNÉRATION

RÉSUMÉ

Des améliorations de conception pour un système polyrotor omnidirectionnel de deuxième génération appelé OmniRaptor sont présentées. Le premier changement majeur par rapport à l'OmniCopter original était l'utilisation d'un système d'hélice à pas variable pour les propulseurs latéraux de l'OmniRaptor au lieu d'hélices à pas fixe utilisées dans l'OmniCopter. Le but de cette modification était d'améliorer la réponse du système aux perturbations. La deuxième modification consistait à installer le système de contrôle de l'OmniRaptor à bord afin d'utiliser pleinement le logiciel de pilote automatique PX4. Les données préliminaires des tests en vol de l'OmniRaptor prouvent sa fonctionnalité et les améliorations futures de l'OmniRaptor sont discutées.

Mots-clés : UAV ; omnidirectionnel ; PX4 pilote automatique ; ROS.

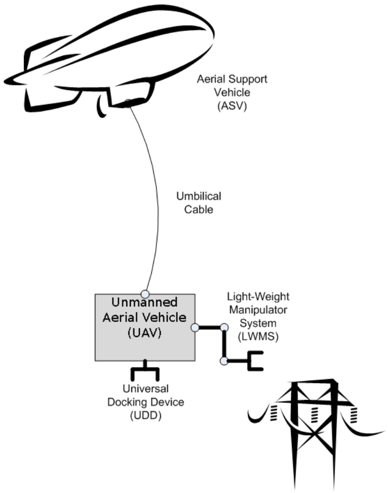

Fig. 1. A Diagram of the Aerial Manipulator System (AMS) [2].

1. INTRODUCTION

The field of Unmanned Aerial Vehicles (UAVs) and in particular Vertical Take-Off and Landing (VTOL) vehicles has been thoroughly studied and a wide variety of applications exist, from data collection, to parcel delivery, to physical interaction with the environment. The vast majority of VTOL UAVs use several fixed-pitch rotors where the direction of thrust of the rotors is collinear. These UAVs are known as quadcopters, quadrotors, or generally as multicopters or multirotors. The majority of these multirotors are underactuated and the vehicle orientation is coupled with its translational motion. This makes them unsuitable for tasks requiring physical interaction with the environment because they cannot instantaneously resist forces or torques in any direction. This is important for aerial mobile manipulator tasks during grasping or otherwise during contact with objects. In order to maintain a desired pose, the UAV must be able to resist reaction forces in arbitrary directions generated during contact with objects. Furthermore, even flying in close proximity to structures, other objects, or people requires prompt reaction to disturbances from turbulent airflow. Typical multirotors require time to adjust their orientation in order to counter such disturbances which has an adverse effect on position control, while omni-directional multirotors do not suffer from this drawback due to the ability to instantaneously generate forces and torques in any direction.

In [1] a novel omni-directional multirotor called the OmniCopter is presented which can resist arbitrary forces and torques with propellers mounted perpendicular to a basic quadcopter configuration. The OmniCopter can be seen in Figure 3, with the additional smaller propellers located along the X and Y axes of the multirotor. It is part of a larger project, described in [2] and [3]. For this larger project, a system is proposed to perform inspection and maintenance on remote industrial structures. The OmniCopter was developed to explore the effectiveness of decoupled multicopters for aerial mobile manipulation. In Figure 1 the architecture of the larger project, called the Aerial Manipulator System (AMS) is depicted. In the AMS, a larger support vehicle, the Aerial Support Vehicle (ASV), supplies power through a cable to a smaller and more dexterous UAV which performs inspection and maintenance tasks.

The focus of this paper is the further exploration of omni-directional UAVs for aerial mobile manipulation. A UAV called the OmniRaptor shown in Figure 2 has been developed in order to overcome the various design limitations of the original OmniCopter. The OmniRaptor has also been designed to take further advantage of its base design as a augmented quadcopter platform by trying to fully utilize the features of the PX4 autopilot software.

Fig. 2. OmniRaptor Prototype

2. BACKGROUND

Significant work comparing variable pitch rotors and fixed pitch rotors on agile UAVs was conducted in [4] and [5]. It was found that there were considerable advantages to variable pitch rotors due to increased thrust rates of change and the ability to quickly and efficiently reverse thrust.

There has been a significant amount of recent work done in the field of omni-directional aerial vehicles. The majority of work in omni-directional aerial vehicles focuses on fixed pitch rotors which are either in a fixed orientation or have a number of rotors which are capable of tilting. Generally there is a focus on finding the optimal number and orientations of fixed pitch rotors which improve the capability of the UAV over traditional colinear thrust multicopters.

In [6] a theoretical study on omni-directional aerial vehicles with body-frame fixed unidirectional thrusters is conducted. Omni-directional aerial vehicles utilizing only uni-directional fixed-pitch thrusters are examined and a method for optimizing such designs is presented.

In [7] a hexarotor UAV is presented which utilizes fixed pitch rotors that are mounted in fixed orientations relative to the frame but are tilted such that their direction of thrust is not collinear as in traditional multicopter designs. A method for optimizing the tilt angle is presented, with the goal of resulting designs being the ability to resist arbitrary wrenches for the purpose of aerial manipulation.

In [8] an omni-directional overactuated quadcopter with four fixed-pitch, tilting rotors is presented. In [9] and [10] omni-directional, fixed-pitch, fixed orientation multirotors are presented.

Variable pitch quadcopters have also seen further study in [11]. In this work, the rotors are collinear and, therefore, the quadcopter is not omni-directional.

3. PHYSICAL DESIGN IMPROVEMENTS

In previous work developing the OmniCopter omni-directional aerial vehicle, it was discovered that while orthogonally directed thrusters could provide significant advantages for aerial manipulation, the low rate of change of thrust of fixed pitch rotors was a serious drawback [1]. The primary design improvement for aerial manipulators which is explored in this paper is the replacement of reversible fixed-pitch rotors with variable-pitch rotors on a new vehicle, the OmniRaptor.

Other improvements were also made in order to facilitate the development of the AMS for aerial manipulation. One of the goals for the larger scope of the project is for the OmniRaptor to be able to autonomously approach and grasp a beam such as those seen on power transmission towers using a conformal underac-

Fig. 3. The OmniCopter (left) and the OmniRaptor (right)

tuated gripper. Another goal is for the OmniRaptor to use a robotic arm to perform maintenance tasks on industrial structures. These goals meant that a much larger UAV was required with significantly more power in order to have enough thrust to carry the gripper and arm. Therefore, the OmniRaptor is approximately twice as large diagonally from motor to motor and weighs nearly four times the amount of the OmniCopter (see Figure 3).

One of the largest problems with the approach taken for the OmniCopter was that the fixed pitch side thrusters were too slow to respond to control inputs. For the vehicle to be effective at resisting arbitrary wrenches, it must be able to *instantaneously* respond to disturbance forces. Fixed pitch rotors rely on electronic speed controllers (ESCs) that can reverse the direction of the motors in order to reverse the direction of thrust. Most of the small brushless motors used do not provide position feedback to the ESCs, therefore the ESCs rely on small changes in the back-emf on the motor phases to detect the motor position with respect to the magnets. Since this method does not provide any information about the position of the motor when it is not moving, the ESC must initially guess which phases to fire in order to achieve the desired direction of rotation. This guess-and-check method can cause significant delay when reversing the direction of a brushless motor due to the necessity of completely stopping the motor and thus passing through the stopped-motor uncertainty condition before the direction of rotation can be reversed. When the required thrust is low or frequently changing direction, such as when maintaining a set-point position, this can cause a sluggish response. The motors also need to counteract the angular momentum caused by the propeller when changing directions quickly like when the OmniCopter is being acted on by a disturbance. These issues will just get worse as the system gets larger with more mass and more voltage needed to control the motors.

Fixed pitch propellers work reasonably well for regular multicopters as they never need to reverse directions. This is possible by relying on gravity to generate any required downwards forces. Some aerobatic multirotors use reversible ESCs for 3D flying where the multicopter can fly while upside down or for generating downwards forces larger than those due to gravity. There is a significant time required for these systems to perform a flip manoeuvre, but because this only happens infrequently, it is not considered a major drawback. However, special motors made by IQ Motion Control are available to overcome this issue. These motors incorporate position sensing and integrate the electronic speed controller with the motor. This allows them to reverse directions very quickly. Unfortunately these motors are only made for small size multicopters and would not provide enough thrust for larger UAVs such as the OmniRaptor.

To address the above issue, the OmniRaptor was fitted with variable pitch propellers for its side thrusters

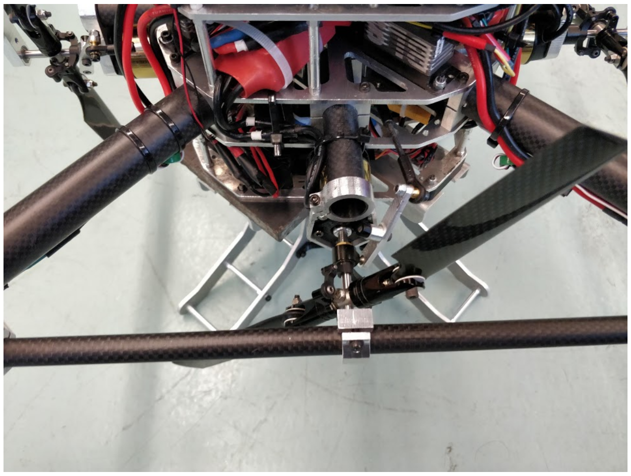

Fig. 4. The OmniRaptor's Variable Pitch Propeller Assembly

in place of the fixed pitch rotors used on the OmniCopter. This type of setup is typically used for the tail rotors of RC helicopters, as they also need to react quickly to disturbances to keep the helicopter balanced. The blade pitch of each propeller, and therefore the force provided by the propeller, is controlled by a servo that is connect through a linkage to the blade. A brace was added with a bearing to support the end of the motor shaft to prevent any wobbling and to add more rigidity to the system. The full assembly is shown in Figure 4.

This system adds a layer of complexity as there are now twice as many outputs to control the side thrusters as the OmniCopter had. This also adds some mechanical complexity due to the linkage from the servo to the propeller. To simplify the controls, all the motors are connected to the same output signal. The motors run at a specified PWM (Pulese Width Modulation) signal while the OmniRaptor is flying and the force is then controlled by servos changing the pitch of the blades. The side thrusters opposite each other are mirrored versions of each other, with the motors rotating in opposite directions to balance the torque generated. The pair of servos that act along the same body axis also share a single control input, as sets should provide the same force in the same direction. The linkage control arm next to the propeller in each set needs to be connected to the blade in the opposite direction of each other to ensure the force is going in the correct direction. Connecting the signals together simplifies the control of the side thrusters by reducing the signals required from eight to three, with the signal responsible for controlling the side thruster RPM remaining constant except during ramp-up. This approach is similar to the OmniCopter using only two outputs to control the side thrusters: one signal for each pair of coaxial thrusters. Methods to optimize power consumption by varying the motor speeds as well as the propeller pitch could be used in the future.

To compare the responsiveness of the variable pitch rotors used on the OmniRaptor with the fixed pitch rotors used on the OmniCopter, both systems were mounted on a thrust measurement stand and the thrust was measured to determine how quickly each system could reverse thrust from maximum thrust in one direction to maximum thrust in the opposite direction. It can be seen from Figure 5 that the fixed pitch propellers took approximately 0.75 seconds to complete the reversal, where as the OmniRaptor's variable pitch propellers reversed thrust in approximately 0.15 seconds. The comparison does not provide a true evaluation of the differences between variable pitch and fixed pitch rotors in general because the propellers on the OmniRaptor were significantly larger than those used on the OmniCopter. It is likely that reversing the direction of rotation on a larger propeller would take even more time than it would for smaller propellers, due to the added inertia. The side thrusters on the OmniCopter used 6 inch (15.24 cm) diameter propellers,

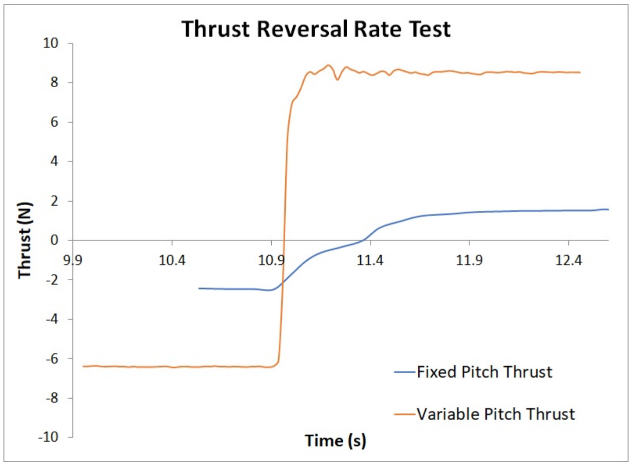

Fig. 5. Comparison of Responsiveness and Thrust Rate Between the Fixed-Pitch Thruster of the OmniCopter Versus the Variable Pitch Thruster of the OmniRaptor

while the rotors on the OmniRaptor are 11.25 inch (28.575 cm) in diameter. The larger variable pitch rotors used on the OmniRaptor produce approximately 4.5 times the thrust of the fixed pitch rotors on the OmniCopter. These tests demonstrate that the added complexity of using variable pitch propellers is offset by large improvements in thrust rate and reduced response time.

Thrust tests were also performed on the main motors and the side thrusters of the OmniRaptor to determine the relationship between force and PWM to be used in the control algorithm.

4. CONTROLS

4.1. Overview

The control calculations for the OmniCopter were performed off-board by a base station desktop computer and transmitted through an RC transmitter to the flight controller on the OmniCopter. The control signals were converted from serial over USB by using an Arduino microcontroller. The Arduino converted the USB signal to the PPM (Pulse Position Modulation) format used by the RC transmitter. The base station computer generated RC signals for thrust, roll, pitch, yaw, and two channels for the side thrusters. This approach worked but was not very elegant, as off-board control meant the OmniCopter could not be truly autonomous due to requiring a base station and the inelegance of having to bypass the RC receiver.

The OmniRaptor control system has been moved on-board the UAV. This was done to try and reduce any latency in the system and so that the system could eventually be used fully autonomously outdoors. The OmniRaptor has a Nvidia Jetson TX2 that runs the Robotic Operating System (ROS - www.ros.org [12]). The TX2 is then connected via UART to a Pixhawk 1 flight controller that runs PX4 autopilot firmware version 1.8.1. The Pixhawk is used to control the main four motors as a basic quadcopter configuration and the onboard sensors, where as the TX2 is used for the control calculations for the side thrusters and will eventually be used for SLAM (Simultaneous Localization and Mapping) calculations with an RGBD camera. To generate the control signals for the side thrusters, the TX2 is connected to an Arduino Nano through a rosserial link.

The PX4 autopilot firmware used on the flight controller is a very robust open-source firmware used by many consumer and industrial drones and simplifies much of the OmniRaptor's controls. PX4 was originally designed by a team at ETH Zurich, but is constantly getting updates and improvements to the software from a wide community of developers [13]. Being originally developed in a research lab, the PX4 firmware easily supports features that are only really used by researchers, for instance easily fusing motion capture position data with all the onboard sensors. This is the major reason for choosing PX4 over other autopilot software like Ardupilot.

To communicate between the flight controller and the TX2, a ROS node called MAVROS was used that converts the MAVLINK messages coming from the flight controller to ROS topics and will convert ROS topics back into MAVLINK messages. This allows the controller on the TX2 to get the exact data used by the PX4 software for calculations, like the position data, instead of having to estimate the data separately, which could results in a discrepancy between data on the different controllers. MAVROS can also optionally provide telemetry data to a base computer running the app QGround Control so that a user can directly monitor the vehicles status or send commands to the OmniRaptor. These values can also be controlled by the TX2.

4.2. Positioning

The initial OmniRaptor development and testing is taking place indoors in a relatively small area compared to the size of the UAV itself. The sensors that are typically used for outdoor flight do not perform well in small indoor spaces. For instance, GPS reception is weak indoors, magnetic fields measured by the compass are dominated by ferrous materials used in building construction, and the barometer detects air pressure changes due to the building's HVAC system. All these devices are therefore disabled through the PX4 software, but can be enabled when the OmniRaptor is eventually flown outside.

This leaves the IMU (Inertial Measurement Unit) still enabled. However, because of the size and the use of the variable pitch propellers, the OmniRaptor's flight controller is subjected to very high levels of vibration that negatively effect the IMU. The vibrations were greatly reduced by adding a vibration mount to the flight controller and 3D printed TPU (thermoplastic polyurethane) vibration dampers to all the motors. The vibrations are now in acceptable ranges for the IMU to be used, but not optimal.

To properly position the OmniRaptor while inside the lab, an OptiTrack motion capture system with 14 cameras running at 100 Hz was used. Two BlackTrax trackers are mounted on the OmniRaptor, each with 3 LED stringers that emit infrared light that can be tracked with the motion capture cameras. The cameras do not have to emit their own infrared light. This eliminates much of the noise while tracking the OmniRaptor especially as the lab has multiple windows. A rigid body was created in the OptiTrack motion capture software following the instructions in the PX4 documentation. The rigid body pose is then passed from the motion capture computer to the TX2 over WiFi, which gets sent to a ROS node to rename the topic to the one used by the PX4 software. MAVROS then sends the data to the flight controller.

PX4 then automatically fuses the data with the IMU data and any other sensor data if it is enabled, in an Extended Kalman Filter (EKF). The offset between the mocap (motion capture) frame to the flight controllers frame was entered into the PX4 firmware, along with the delay time between the calculation of the pose from the IMU and the calculation of the mocap pose. The delay was experimentally determined to be around 75 ms. The delay time appeared to have a large effect on how well the motion capture's position estimate and the flight controller's position estimate tracked each other. A benefit of using this system over just using motion capture, like in the OmniCopter, is that if the motion capture system loses track of the markers for a couple of seconds, or the WiFi momentarily goes out, the OmniRaptor will not lose position information as it can estimate its position from the IMU alone. Losing position information was a large concern for the OmniCopter. Another benefit is that other sensors can easily be added to the system to improve the positioning with minimum effort to reprogram the EKF, as PX4 enables it to automatically change when new sensors are enabled.

4.3. Controller

PX4 offers a special flight mode called the offboard mode, which allows the flight controller to be controlled by a separate computer through MAVROS commands. The connected computer must consistently send set-points to MAVROS before the flight controller will change modes. The set-points can be either a position, velocity, or acceleration in XYZ axes, or the attitude, angular velocity, or angular acceleration.

The OmniRaptor is at its core a quadcopter with extra motors added to it. The four main motors are controlled by the Pixhawk flight controller while the extra side motors are controlled by the TX2. These controllers are totally separate systems. Although this may not by the most optimal control method, it does allow for the OmniRaptor to make full use of the quadcopter configuration in PX4, which has already had thousands of hours worth of testing. With that, the OmniRaptor uses offboard mode by sending the flight controller attitude set-points and an overall thrust value between zero and one. The flight controller will then attempt to achieve that attitude. The side thrusters are used in combination with the total thrust of the main lift rotors in order to control the position of the OmniRaptor.

A velocity based PID (Proportional-Integral-Derivative) controller was designed to move the OmniRaptor to any position in 3D space, by following a trapezoidal velocity trajectory between the OmniRaptor's current position and the goal position set by the user or calculated from the RGBD camera. The outputs of the controller are the PWM values going to the servo pairs which control rotor pitch in the X and Y axes and the thrust value that is transferred to the flight controller. An offset is added to these values based on the OmniRaptor's attitude to offset the force of gravity. The servo PWMs are then transferred to the Arduino with rosserial.

The Arduino takes the value and checks to make sure it is within an acceptable range and then generates the appropriate signal. The PX4 currently does not allow for auxiliary outputs to be controlled while in offboard mode which is why the Arduino is used. This should be added to version 1.9 of PX4, at which time the Arduino will be removed from the system to cut down on complexity and reduce latency.

5. PRELIMINARY FLIGHT TESTS

The PX4 firmware uses a front, left, down co-ordinate system. That system is used for the plots in this section.

In order to reduce the risk of crashes during initial tests, a test stand was designed to hold the UAV in place about the pitch axis approximately through its centre of gravity. This allows the UAV to roll, but holds it fixed in the pitch and yaw axes. A 2.5 kg steel plate was added to the bottom of the OmniRaptor to simulate the weight and approximate centre of mass of the OmniRaptor with the gripper and robotic arm attached. In total the OmniRaptor weighs aproximately 10.46 kg. Figure 6 shows the OmniRaptor on its test stand.

Fig. 6. The OmniRaptor on the Test Stand

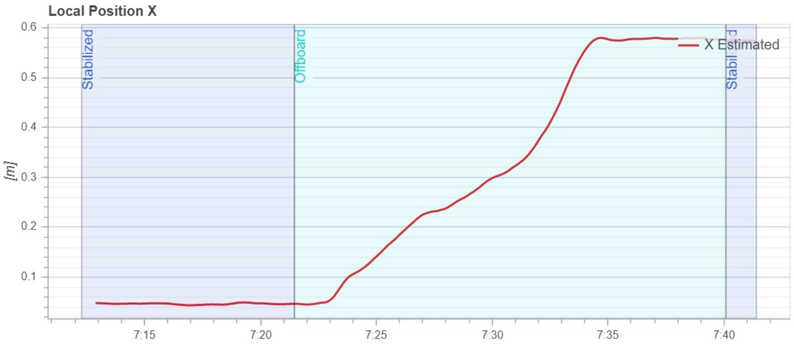

Fig. 7. X Axis Position of the OmniRaptor While Being Commanded to Move 0.5 m From the Start Point in the Test Stand with Wheels

A rod was used to hold the OmniRaptor fixed about one axis instead of cables like other quadcopter test stands. This was done to allow testing of the side thrusters, enabling them to exert a horizontal force on the test frame. This cannot be done with a cable suspension system. This setup will change the tuned values for the roll axis slightly, as there is friction from the rod and it is harder to ensure the OmniRaptor is hovering, but the values will be tuned enough to get it air born without crashing and the PID values can be fine tuned while in the air.

The roll axis in the OmniRaptor's flight controller was then tuned in stabilize mode according to the standard process for traditional multirotors by giving it step inputs and checking the flight logs for the step response and then changing the PID gains inside the flight controller. The values obtained by this testing can also be used for the pitch axis because the OmniRaptor is close to symmetrical about the two axes. These PID values are also used for attitude control when operating in offboard mode.

The OmniRaptor could then be tested in offboard mode to ensure the flight controller could keep the drone stable while it is at an attitude and if the OmniRaptor can move to achieve a required attitude. The OmniRaptor could easily and accurately control its attitude while on the test stand. The test stand worked very well for this, as the side thrusters could either be turned on or off without the threat of the OmniRaptor flying into things. The custom controller running on the TX2 could also be safely tested to ensure the side thrusters were functioning correctly.

When the side thrusters and the control system were deemed to be functioning correctly, caster wheels were mounted onto the bottom of the test stand so that the OmniRaptor could roll while still in the test stand. The OmniRaptor was then tested fully autonomously with everything including the side thrusters working. The OmniRaptor was then commanded to move half a meter in the X axis from its starting position while remaining at the attitude that the OmniRaptor was at when the offboard switch was flipped. During the test, the OmniRaptor remained at the correct attitude and rolled across the floor to approximately the desired set-point. Figure 7 shows the X axis position during this test. The controller is only using proportional control, so the OmniRaptor overshoots its goal, but the wheels prevent the OmniRaptor from correcting the overshoot. The performance of this test could be improved by some tuning and incorporating full PID control. However, rolling on the ground adds too much resistance to the system, as compared to when it is flying, so if this was done, the PID parameters would be incorrect during actual flight operations.

The OmniRaptor was then removed from the test stand and flown in the regular stabilize mode with the side thrusters turned off to ensure the tuning of the roll and pitch axis that was done while on the test stand was correct. Figure 8 is from the first tests off the test stand and show that the tuning done on the test stand leads to a reasonably well tuned attitude controller. The thrust value during this test was also found at a stable hover and used to compare it to the equation that was acquired during the thrust testing. The actual

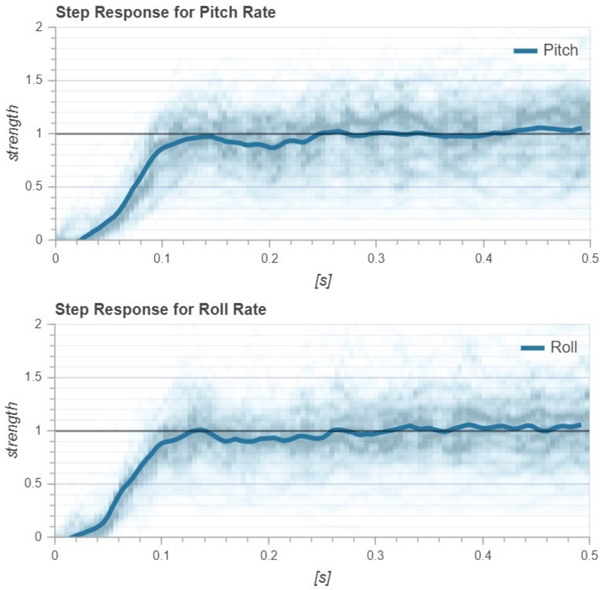

Fig. 8. Step Input of the OmniRaptor's Pitch and Roll Rates While in the Air

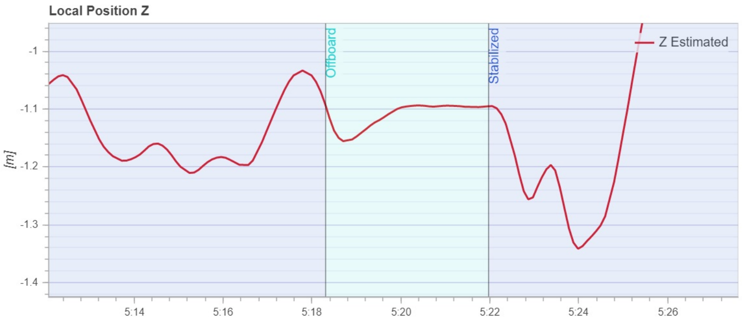

Fig. 9. Z Axis Position During Offboard Testing with the Side Thrusters Turned Off While in the Air

throttle value was found to be very close to the estimated.

The OmniRaptor was then tested in offboard mode with the side thrusters turned off, to ensure that the OmniRaptor can remain at a stable altitude and attitude. Figure 9 shows the OmniRaptor trying to maintain the altitude it was at when it switches into offboard mode. The different coloured sections on Figure 9 shows what flight mode the flight controller is in at that time. There is an initial upwards velocity when the OmniRaptor goes into offboard mode which causes the Omniraptor to gain altitude, but the PID controller corrects for this and then maintains the correct altitude until the flight controller switches back to stabilize mode.

The next test was to run the OmniRaptor in stabilize mode with the side thrusters turned on to determine if the OmniRaptor can remain stable at a specified location in the XY plane. During this test, the thrust, roll, pitch, and yaw were controlled manually via an RC transmitter by a human pilot. The control system used only the side thrusters to attempt to maintain position at a specified location in the XY plane. Figure 10 shows the position data during this test with the side thrusters turning on at 8:20 and then off again at 8:28. The OmniRaptor oscillates around the set-point as the controller is only running with proportional control. The results show that the system is functioning properly, but needs to be tuned.

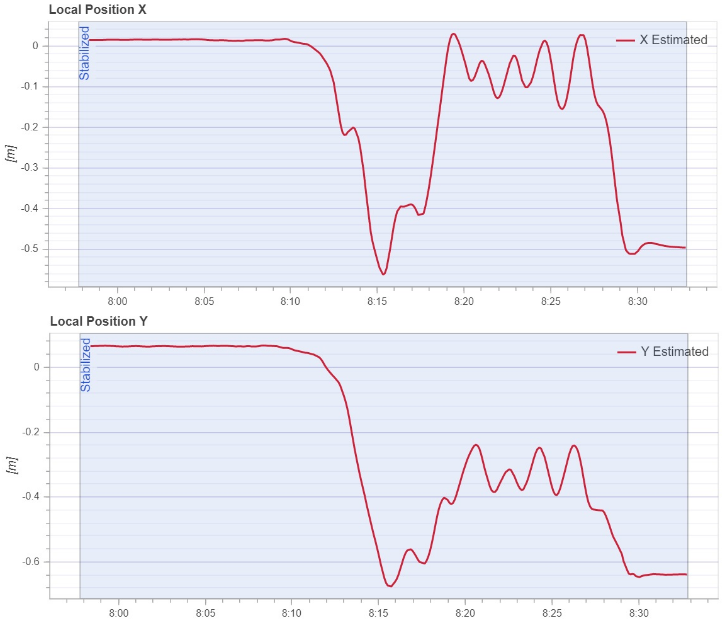

Fig. 10. X and Y Axes Positions During the Test of the Side Thrusters While in the Air

From these preliminary flight tests it is shown that the OmniRaptor is functioning correctly and with some further tuning will have acceptable performance.

6. FUTURE IMPROVEMENTS

There are a number of improvements that can be made to the OmniRaptor. First, the OmniRaptor control parameters need further tuning, especially the gains for the side thrusters, to ensure that the system can remain stable enough to avoid crashing in the confined indoor test space. Once tuning is complete, the system will be tested on how well it can perform other tasks that are not possible for traditional multirotor UAVs, such as moving to a set-point while remaining level or remaining stationary while pitched at an angle.

The PX4 flight controller will be tested to see what other autonomous features the flight controller can do. This includes autonomous take off and landing from a flat surface and commanding the OmniRaptor to follow pre-programmed missions.

The OmniRaptor will have a RGBD camera mounted onto it so that it is able to detect a landing point on a structural beam and determine if it is a suitable landing area. With the camera, it is also possible to do local planning and obstacle avoidance, as this is newly implemented into the PX4 software and just needs to be tested for reliability, especially in an indoor environment.

The OmniRaptor will be fitted with a grasping mechanism that is currently being designed. Once installed, the system will be tested landing on beams at multiple different angles. A robotic arm will then be designed and fitted to the OmniRaptor to do different manipulation tasks while it is landed and attached to a beam or while it is flying.

7. CONCLUSIONS

A larger version of an omni-directional aerial vehicle was developed which features variable pitch rotors orthogonal to the main lift rotors. Tests were conducted to demonstrate that higher rates of change of thrust can be achieved with variable pitch rotors as opposed to fixed pitch rotors. A more advanced control scheme with all control calculations done on-board the UAV was implemented. The new set-up utilizes the full capabilities of the PX4 flight controller.

Overall, it was demonstrated that the OmniRaptor has had several design upgrades when compared to the original OmniCopter, specifically as it pertains to the responsiveness of the side thrusters and the use of a more advanced control scheme. The OmniRaptor has been shown to be operational in some preliminary flight testing and future work will be done to ensure the OmniRaptor is robust enough to be able to land on a power transmission tower and repair it with a robotic arm.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Professor Roger Boudreau, Université de Moncton, is thanked for doing the French translation of the paper title and abstract.

REFERENCES

1. von Frankenberg, F. and Nokleby, S.B. "Inclined Landing Testing of an Omni-directional Unmanned Aerial Vehicle." *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 42, No. 1, pp. 61–70, mar 2018.
2. Baltovski, T., Nokleby, S.B. and Pop-Iliev, R. "Design and Development of an Swinging Arm Mechanism for an Aerial Manipulator System." In "Proceedings of the International Conference on Mechanical Engineering and Mechatronics," Toronto, Canada, August 2013.
3. Baltovski, T., Nokleby, S.B. and Pop-Iliev, R. "Towards Performing Remote Manipulation Using an Autonomous Aerial Vehicle." In "Proceedings of the 2015 CCToMM Symposium on Mechanisms, Machines, and Mechatronics," Ottawa, Canada, May 2015.
4. Cutler, M., Ure, N.K., Michini, B. and How, J. "Comparison of Fixed and Variable Pitch Actuators for Agile Quadrotors." In "AIAA Guidance, Navigation, and Control Conference," American Institute of Aeronautics and Astronautics, Reston, Virigina, August 2011.
5. Cutler, M. and How, J.P. "Analysis and Control of a Variable-Pitch Quadrotor for Agile Flight." *Journal of Dynamic Systems, Measurement, and Control*, Vol. 137, No. 10, p. 101002, July 2015.
6. Tognon, M. and Franchi, A. "Omnidirectional Aerial Vehicles With Unidirectional Thrusters: Theory, Optimal Design, and Control." *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, pp. 2277–2282, jul 2018.
7. Rajappa, S., Ryll, M., Bulthoff, H.H. and Franchi, A. "Modeling, Control and Design Optimization for a Fully-Actuated Hexarotor Aerial Vehicle with Tilted Propellers." In "2015 IEEE International Conference on Robotics and Automation (ICRA)," pp. 4006–4013. IEEE, may 2015.
8. Ryll, M., Bulthoff, H.H. and Giordano, P.R. "A Novel Overactuated Quadrotor Unmanned Aerial Vehicle: Modeling, Control, and Experimental Validation." *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 2, pp. 540–556, mar 2015.
9. Park, S., Her, J., Kim, J. and Lee, D. "Design, Modeling and Control of Omni-Directional Aerial Robot." In "2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)," pp. 1570–1575. IEEE, October 2016.
10. Brescianini, D. and D'Andrea, R. "Design, Modeling and Control of an Omni-Directional Aerial Vehicle." In "2016 IEEE International Conference on Robotics and Automation (ICRA)," pp. 3261–3266. IEEE, may 2016.
11. Sheng, S., Sun, C., Sheng, S. and Sun, C. "Control and Optimization of a Variable-Pitch Quadrotor with Minimum Power Consumption." *Energies*, Vol. 9, No. 4, p. 232, mar 2016.
12. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Ng, A. "ROS: an Open-Source Robot Operating System." In "Proceedings of the ICRA Workshop Open Source Software," , 2009.
13. Meier, L., Honegger, D. and Pollefeys, M. "PX4: A Node-Based Multithreaded Open Source Robotics Framework for Deeply Embedded Platforms." In "2015 IEEE International Conference on Robotics and Automation (ICRA)," pp. 6235–6240, May 2015.