Design and Development of a Bio-Inspired Airborne Tracking System With the Xbox Kinect

Anthony Dzaba, Evan Mucasey, Alex Bourreza, and Eugenio Schuster

***Abstract*—The ability to navigate in unfamiliar environments is a key issue in implementing any autonomous unmanned aerial vehicle (UAV). In an effort to move closer to real-world application capability, we envision a specialized, mobile vision system capable of tracking a group of low-cost UAVs outside of the lab setting. We present an omnidirectional array of Xbox Kinect units for image acquisition and a custom embedded computer cluster for image processing. In order to ease the computational burden of the proposed vision system, the array is mounted on a stabilized, holonomic sensor deck which keeps its frame of reference constant throughout flight maneuvers. A simple PID controller is tested in simulation and experimentation to demonstrate the ability to decouple the orientation of the vehicle from the orientation of the sensor deck. We conclude with a roadmap for implementing well-established machine vision techniques on the newly-developed hardware.**

I. Introduction

The growing list of applications for unmanned aerial vehicles (UAVs) includes general exploration, commercial surveillance, search & rescue, military operations, and light transportation. The potential of these applications has inspired considerable research interest in the development of UAVs in both standalone and swarm configurations. In order to realize practical applications in autonomous flight, we must first address issues with robust navigation and flight endurance. In recent years, a substantial body of research has been dedicated to the subfields of advance motion control and path planning as in [1], [2], [3], [4]. Problematically, these control schemes rely on the availability of high-precision pose measurements from non-portable global vision systems. As a consequence, the positive results demonstrated in the papers referenced above, cannot be duplicated outside of the lab environment.

This paper addresses the specific need for mobile position sensing. While attitude sensing inertial measurement units (IMUs) have become more and more prolific, portable solutions for robust position sensing remain elusive. The development of novel UAV sensor systems has seen limited research effort (see [5] for example). Challengingly, intuitive choices in vision-based solutions place high demands on UAV systems where resources are already limited. Desirable features include the ability to sense egomotion and external objects without incurring inordinate computational expense.

The tutorial [6] gives a thorough treatment of the two predominant vision-based solutions to the mobile navigation problem. These are visual simultaneous localization and mapping (VSLAM) and visual odometry (VO). The generally preferred solution, VSLAM, offers superior egomotion sensing and global mapping. Visual odometry, a subclass of structure from motion (SFM), offers good egomotion and environmental sensing at a substantially lower computational expense. This computational saving is afforded by sacrificing global map consistency. Notwithstanding, this may be an acceptable tradeoff as obtaining an iterative chain of local maps (as in VO) may be enough to satisfy the environment sensing requirements of the applications mentioned above.

One method of reducing the computational expense of visual odometry is the introduction of motion constraints . From a biomimetic perspective, we take note of the acute vision and vestibular gaze stabilization abilities of birds of prey, wherein the chaotic motion of the body (flight frame) is conveniently decoupled from the stabilized motion of the head (observer frame) which continuously tracks a target of interest [7]. In this work, we propose a novel mechatronic solution which emulates this behavior by actively stabilizing a sensor frame while leaving an independent flight frame free to articulate in controlling the aircraft position. The VO implementation can be further simplified by eliminating the need to identify 3D features via triangulation. When using a 3D vision sensor such as the Microsoft Kinect, we eliminate the need for advanced image acquisition techniques such as image correspondence through stereo vision or obtaining structure from motion over a sequence of image frames.

Since its inception, the Kinect unit has seen an immediate surge in research interest (see [8], [9] among many others). The findings of this paper suggest that we can use the Kinect to develop a solution for environment mapping, self-localization, and UAV tracking. We envision a mobile, vision system concept vehicle in the form of two nested quadrotors. The aptly named "Dual-quad Observer" features a repurposed holonomic sensor deck which houses an array of Kinect units on a mechanically stabilized platform. The Dual-quad is designed to "observe" a group of UAVs, not as an external lab fixture, but as an airborne member of a heterogeneous swarm.

This paper is organized as follows. In Section II, we evaluate the performance of the Kinect unit and present a general framework for developing a simple 3D vision system. In Section III, we describe the mechatronic platforms designed in support of the developing vision system. In Section IV, we derive a dynamical model for the mechatronic plant. In Section V, we present a simple attitude stabilization

A. Dzaba (anthony@lehigh.edu), E. Mucasey, A. Bourreza, and E. Schuster are with the Department of Mechanical Engineering & Mechanics, Lehigh University, Bethlehem, PA 18015 USA.

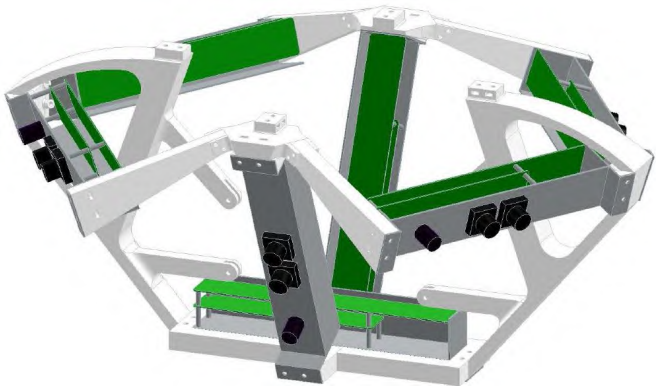


Figure 1: CAD model of the Kinect unit array: Seven Kinect units are arranged on the faces of an open, eight-sided polyhedron. The units are linked together with rapid prototyped, plastic finite element models. The unconnected arms in the center of the array hold the gigabit Ethernet switch (not pictured) through which the individual COMs in the computer cluster communicate.

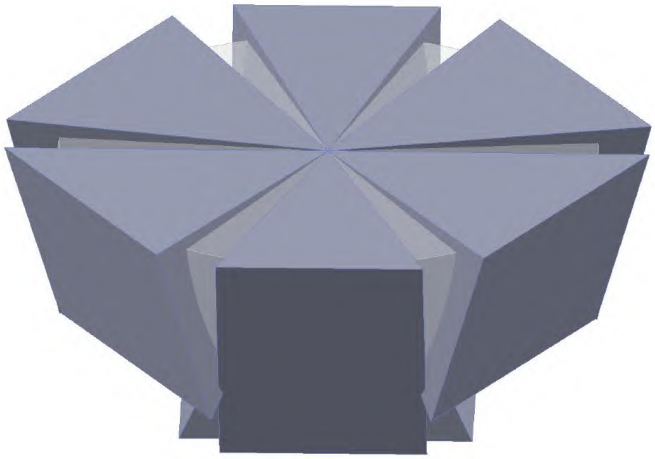


Figure 2: Sensor Geometry - A 3D model of the arrayed fields of view of the seven Kinect units. Notice the small area of overlap near the lower portion of the front facing unit. The seventh partially obscured unit is facing downward.

controller along with simulations and experimental results. Finally, in Section VI, we discuss findings and future work.

II. Development of the Vision System

In this section, we describe incremental progress in the development of a 3D vision system for the Dual-Quad Observer. Target features include the ability to sense indoor obstacles, self-localize, and track agents of UAV swarm. In Section II-A, we discuss the abilities and limitations of the Xbox Kinect unit. In Section II-B, we describe the geometry of an array of Kinect units arranged to emulate a compound eye. In Section II-C, we examine a set of Kinect images to evaluate their suitability for use in the proposed vision system. Later, in Section III, we will discuss the various mechatronic systems which will support the vision system upon its completion.

A. The Kinect Unit

The Xbox Kinect is a hybrid color imaging sensor and structured light-scanning depth sensor (among other capabilities not pertinent to this discussion). It consists of three main components: an infrared projector, an infrared image sensor, and a visible light image sensor. Applications include skeleton tracking, 3D reconstruction, human interface devising, and inertial motion detection. The unit operates at 30 Hz and is able to detect depth at distance greater than 4 meters. The image sensors have a relatively limited resolution of 480x640 pixels. Due to infrared noise issues, the depth sensor is not usable in direct sunlight where the projected structured light pattern is easily washed out.

It is important to note that this particular sensor technology is only one of several candidates suitable for use in the framework of this vision system. As such, the limitations of the Kinect are not necessarily the limitations of the Dual-Quad Observer. As advances in sensor technology continue to progress, it may be quite feasible to supplement or completely replace the Kinect with an alternate sensor system which is more appropriate for this application. That aside, the particular capabilities of the Kinect provide a unique opportunity to evaluate this vision system concept.

B. Compound Eye Design

At present, the Dual-quad Observer is designed to accomplish mapping, localization, and tracking using the Kinect as its sole sensor type. Fig. 1 illustrates a model assembly of seven Kinect units arranged in a convex array inspired by the compound eye of the drosophila fly. The geometry of the array is such that each unit looks in a different direction, divvying up the surface area of a roughly hemispherical bowl. Although it would be arguably simpler to simulate omnidirectional vision using a smaller number of Kinects on a continuously rotating frame, potential problems with the rolling shutter distortion, saccadic masking, and high-speed optic flow analysis make this alternative less attractive.

The Kinect orientations are optimized to minimize field of view (FOV) overlap while maximizing the cumulative solid angle coverage. In other words, we aim to reduce both the aggregate blind spot and overlapped areas between the Kinect units (where the later prevents inter-sensor interference that occurs when two or more infrared projectors cast structured light patterns on the same surface). The sensor geometry is illustrated in Fig 2. The six upper units are oriented in such a way that the generated maps will include portions of the 3D surroundings just above the array's horizon so that the Observer can "see" regions above the vehicle's altitude in the distance. An obvious improvement would be to include an eighth upward looking Kinect for monitoring overhead obstacles and head space clearance.

C. Image Analysis

In this subsection, we evaluate the ability of the Kinect unit to provide useful data for three tasks: mapping, localization,

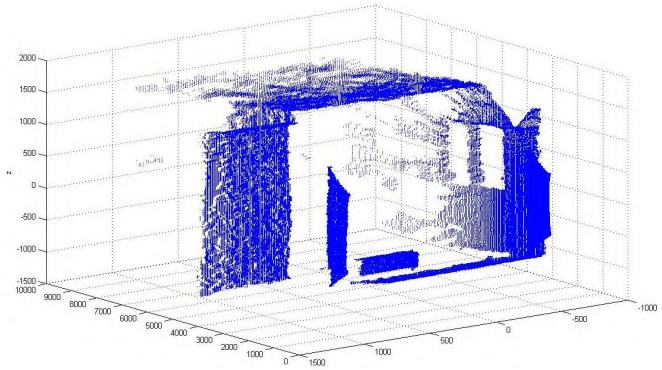
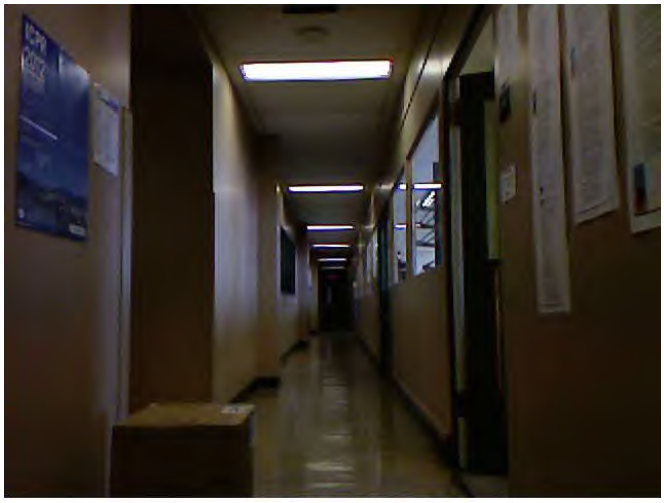


Figure 3: 3D point cloud - Note the correspondence between the rectangular voids in the point cloud and those in depth image of Fig. 4b. Both of these represent the windows on the right side of the hallway pictured in Fig. 4a. Notice the voids situated behind occluded areas: a characteristic of all line-of-sight depth sensors. This makes it difficult to make out the top of the box on the left side of the hallway. Despite image noise and complex scene structure, subtle details are still captured.

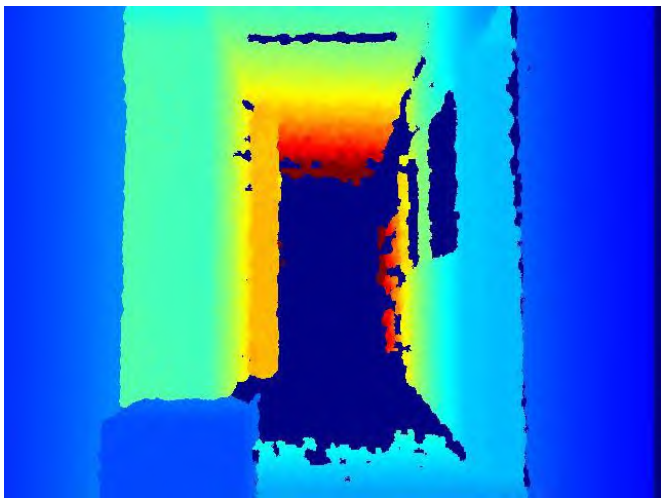
and tracking. The image analysis provided below is undertaken pursuant to the realization of a robust, high-accuracy vision system like those discussed in [10], [11].

To accomplish the mapping task, we seek to implement a simple 3D costmap which divides the environment space into an occupancy grid of voxels (volumetric pixels). The point cloud of Fig. 3 is a 3D representation of the hallway pictured in Fig. 4a. The point cloud is generated by transforming the corresponding 2D depth image of Fig. 4b. The three images can be used to populate a coarsely quantized costmap for obstacle detection. Note the clear feature correspondence where the windows and cardboard box are visible in all three representations.

Localization can be implemented via a combination of dead reckoning and image registration also known as loop closure. Static background features can be tracked and used as landmarks or reference points from which to measure fine displacements. Consider the red box pictured in the center of Fig. 5a. Notice the corresponding rectangular swatch in the depth image of Fig. 5b. These two images show how sufficiently salient features can be readily tracked in 3D. The active mechanical constraint of the sensor deck will greatly simplify the process of correlating features from one frame to the next as the motion between the Kinect array and the Earth-fixed frame can be approximated as a 3D translation with no rotation. A suitable localization algorithm can be implemented on any or all of the available Kinect units, providing a dynamic pool of features from which to draw. This would allow the algorithm to be selective in its choice of candidate features. The resulting lower incidence of spurious outliers should increase the overall accuracy of the visual odometry measurements.



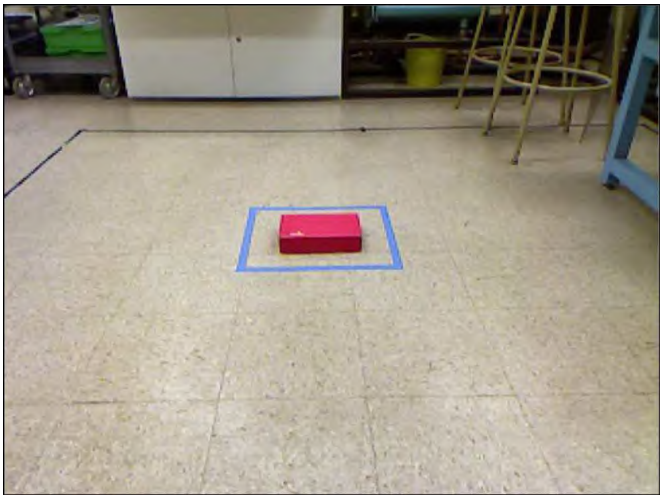
(a)



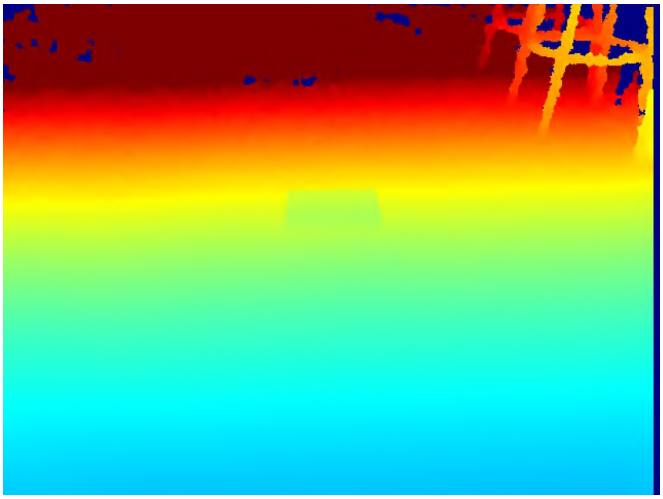
(b)

Figure 4: 3D Mapping - In the RGB image of Fig. 4a, the box on the left and windows on the right can be seen in the 2D depth map of Fig. 4b and the 3D point cloud of Fig. 3. In the depth map, colors on the blue end of the spectrum are closest. Dark blue regions are those for which the Kinect system could not determine a depth. This is typical with bright light sources which generally interfere with the projected infrared pattern.

Vehicle tracking can be accomplished by segmenting predetermined visual markers. In Fig. 6a, the elevation and azimuth of each marker can be readily obtained using ordinary thresholding. The third element of the marker positions can be obtained from a depth map like the one in Fig. 6b. Measurements of the 3D positions of the markers can be used to estimate the 6D pose of the vehicle upon which they are mounted. The markers should be chosen carefully to work with the limitations and disposition of the Kinect system. In Fig. 6b, note that the dark blue regions are those for which the Kinect could not determine a depth value. Issues with



(a)



(b)

Figure 5: Feature Segmentation - In Fig. 5a, the resolution of the color camera, although not very high, is sufficient to distinguish features in non-homogenous regions, such as the red box in the blue square. The corresponding depth contrast of the box can be seen in Fig. 5b where the light-blue patch stands in low relief against the uniformly flat floor. Also notice the detailed correspondence between the pair of stools at the top right of each image.

(a)

(b)

Figure 6: Marker Tracking - In Fig. 6a, the LED-powered markers and their reflections on the glossy tile floor are easily identifiable. The depth image of Fig. 6b can be used to segment out the reflective artifacts as they do not correspond to the physical markers. Note the noise in the depth perception of the quadrotor UAV. Range values are returned for only two of the four markers (the dark blue areas are unreadable). Marker geometry will have to be well-designed in order to provide consistent ranging performance.

reflectance, infrared light noise, and surface geometry all effect the Kinect unit's ability to accurately measure depth.

III. Mechatronic Synthesis

In this section, we describe the mechatronic systems designed to support the vision system functionality proposed in Section II. Features include onboard distributed computing, support for real-time sensor fusion, and a novel approach to holonomic actuation. In Section III-A, we describe the custom computer hardware which has been built to host the forthcoming vision system software. In Section III-B, we discuss a method for obtaining measurements of the attitude states and issues regarding sensor fusion. In Section III-C, we discuss the Dual-quad chassis which provides a holonomic platform for the Kinect array. Later, in Section IV, we will develop a model for the dynamics of this holonomic platform.

A. Embedded System

Once fully developed, the image processing and motion control algorithms will be implemented on an embedded computer cluster. The computer cluster is assembled as an array of Gumstix computer on modules (COMs) and expansion boards. The system features seven slave COMs and one master COM which coordinates high-level control and data

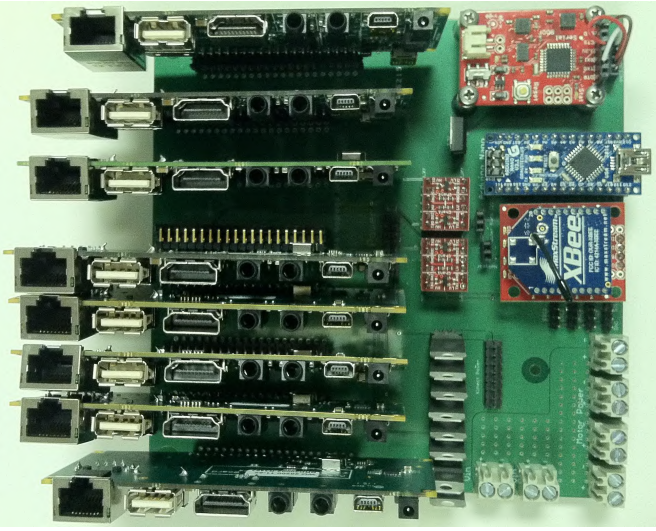


Figure 7: Computer cluster motherboard: Eight COMs (left) communicate with the onboard IMU, microcontroller, and wireless radio through a custom designed printed circuit board. The Ethernet ports (left) are used to network the COMs over an onboard switch. This physical architecture compares directly with the logical architecture of Fig. 8.

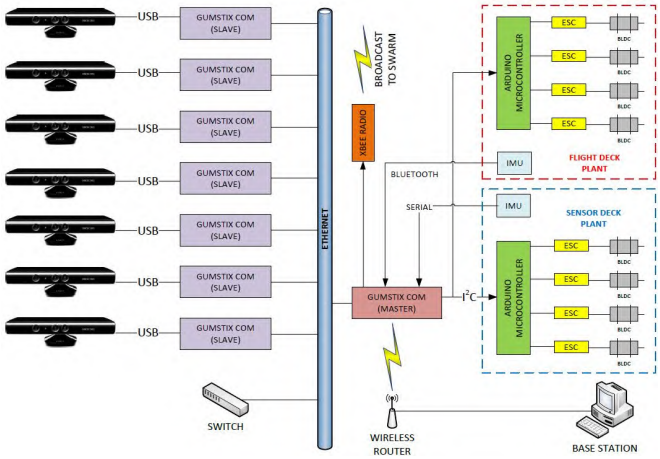


Figure 8: System architecture map: The seven Kinect units (left) connect to seven slave COMs which communicate with a single master COM over a high-speed network. The master COM combines the local maps generated by the slave COMs into a global map. State measurements are received from the flight and sensor deck IMUs to compute the control inputs sent to the eight actuators. User commands are received over Wi-fi from a remote base station. This architecture is physically implemented on the circuit board in Fig. 7.

aggregation. Each slave COM runs embedded Linux while interfacing with a single Kinect unit. The cluster interfaces with a custom designed motherboard, viewable in Fig. 7. The motherboard routes power and communications to and from the vehicle's various sensors, actuators, and microcontrollers. The embedded system also facilitates communication with a local wireless network and a base station for centralized control and data logging.

The system architecture is illustrated in Fig. 8. There are two actuator subsystems, each composed of an Arduino microcontroller, four electronics speed controllers (ESCs), and four brushless DC motors. The communications suite includes an XBee wireless radio for broadcasting tracked positions to a group of UAVs. The flight and sensor decks each have their own IMUs for measuring their respective attitude states. The control algorithm is designed in Simulink. Source files are downloaded to the master COM where they are used to generate code to implement the control algorithm outside of the MATLAB programming environment (which would require substantial overhead to run on the embedded system).

B. Inertial Sensing

The Dual-quad is outfitted with two inertial measurement units. Each IMU contains triple-axis accelerometers, gyroscopes, and magnetometers. These are used to measure 3D angular position and 3D angular velocity. One IMU is attached to the flight deck while the other is attached to the sensor deck. Because the integrated MEMS accelerometers are sensitive to both static and dynamic accelerations, it may be difficult to distinguish between a change in inclination and a sudden change in translational motion. This is not a major issue when the translational accelerations are small and brief. On the other hand, should the flight deck assume a steep angle to impose a large translational acceleration (on the order of 1 g), the accelerometers on the sensor deck could easily become saturated, fooling the IMU into "thinking" that the orientation of the sensor deck has drastically changed when in fact, the sensor deck still remains level. Since the Dual-quad position remains fixed during the experimentation, this issue will be addressed more thoroughly in a future work.

C. Mechanics

The Dual-quad Observer is a concept vehicle composed of two connected quadrotors. See Fig. 9. The outer quadrotor, referred to as the flight deck, connects to the inner quadrotor, referred to as the sensor deck, via a ball joint. The novelty of the ball joint connection is that while translational forces are transmitted from one deck to the other (so that both translate together), rotational moments are not. This also allows the attitude of the the sensor deck to be decoupled from the attitude of the flight deck. As a result, the attitude of the flight deck can be modulated to control the position of the entire vehicle while the attitude of the sensor deck can be stabilized at its nominal equilibrium in order to provide a fixed-orientation frame of reference for the vision system. The combined system can be modeled as two discrete quadrotors coupled only through their externalized forces.

Fixing the rotation between the Earth-fixed frame and the sensor deck-fixed frame reduces the number of relative degrees of freedom from six to three. Limiting the relative motion between these frames to translation alone greatly

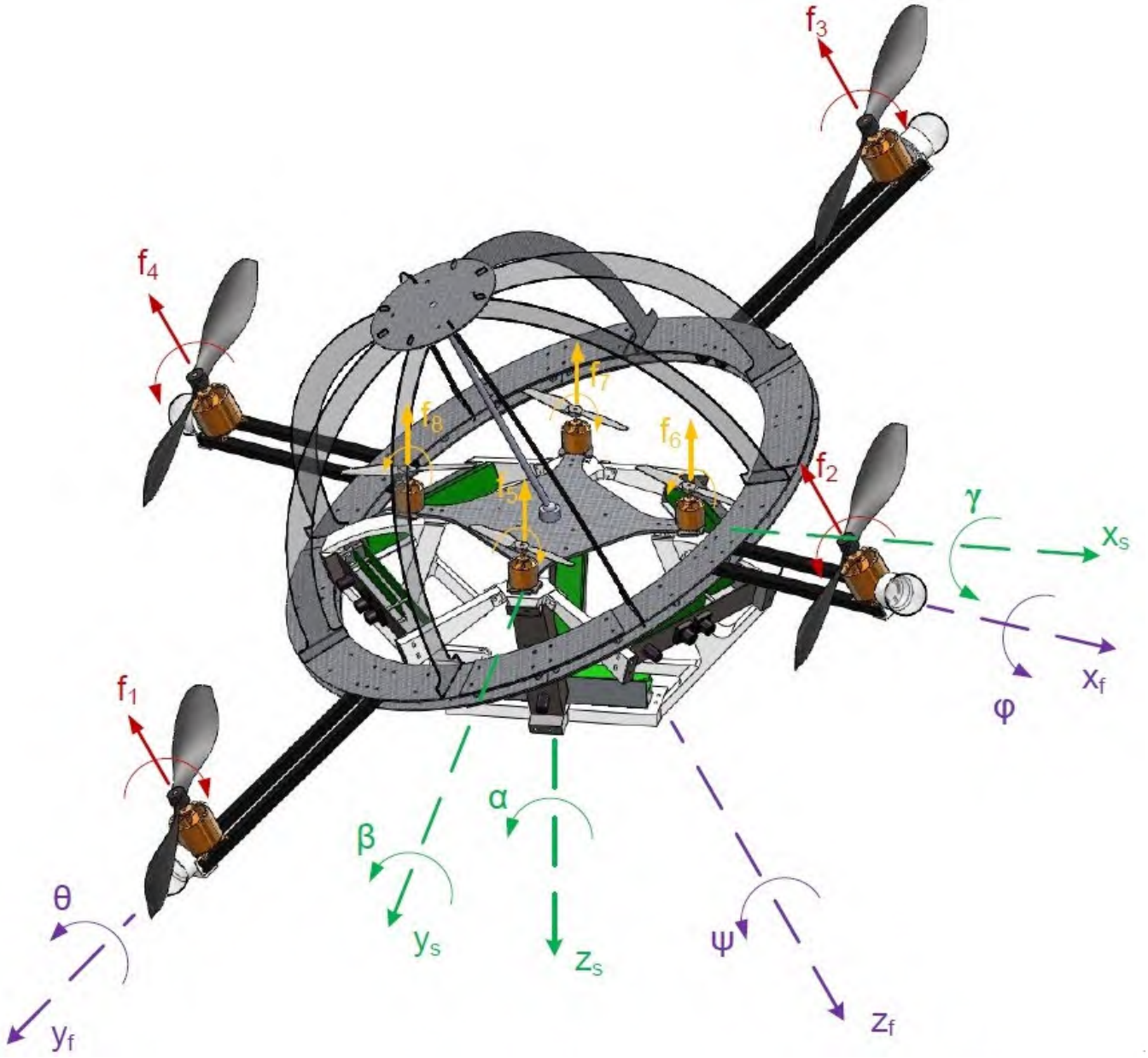


Figure 10: Free-body diagram of external forces and moments. Each actuator set (motor & propeller) generates a single force and moment. These 8 inputs are used to control the Dual-quad's 9 outputs which are the 3D flight deck attitude, the 3D sensor deck attitude, and the 3D vehicle position. Note how the orientation of the flight deck axes (purple) is independent of the orientation of the sensor deck axes (green).

simplifies the computational burden of implementing visual odometry [6]. The ball joint is a critical component of the physical plant as it defines the maximum articulation angles between the flight deck and the sensor deck. By extension, this determines how much translational force the flight deck can apply to the vehicle without affecting the stability of the sensor deck. It is especially important that the friction in the joint be as low as possible so that the moment transmissibility between the sensor and flight decks can be considered negligible.

The larger, outer quad is outfitted with 500 Watt brushless DC motors (BLDCs) and 14 inch propellers, each capable of producing about 20 Newtons of thrust. The smaller, inner quad is outfitted with 300 Watt motors and 5 inch propellers, each capable of producing about 3 Newtons of thrust. The gross mass of the Dual-quad is about 4 Kg giving the system a thrust to weight ratio of about 2:1. The larger, more efficient props are designed to do most of the payload lifting and all of the translational force generation. The smaller, less efficient props are designated for sensor deck attitude control.

IV. Dynamical Model

In this section, we derive the equations of motion associated with the mechatronic hardware described in the

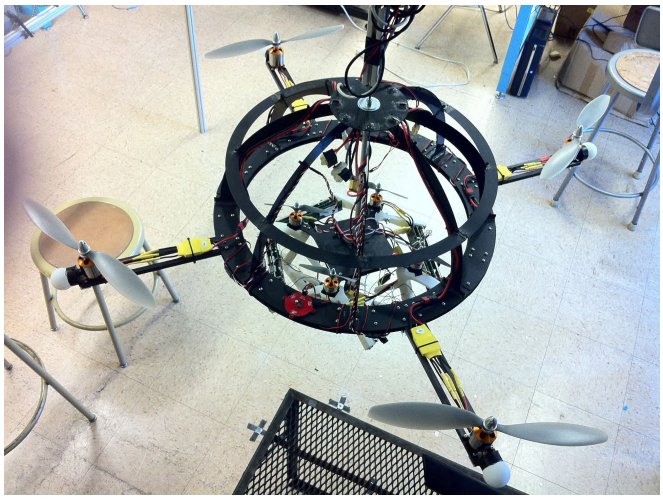


Figure 9: The Dual-quad Observer prototype: A small inner quadrotor is seen nesting inside a large outer quadrotor. The Kinect unit array (partially occluded) is mounted to the bottom of the inner quadrotor so that the compound eye may "observe" vehicles below.

Section III. In Section V, we will use the obtained model to design and evaluate a suitable control strategy. The Dual-quad Observer is actuated by eight brushless DC motors outfitted with orthogonal sets of counter-rotating propellers, as shown in Fig. 10. The equations of motion are derived in the inertial frame using the approach proposed in [12]. The inertial axes are separated from the Earth-fixed axes by a 3D translation vector and a scalar yaw rotation denoted ψ. The Dual-Quad position is given by the 3D vector $\xi = [x, y, z]^T$. The attitude of the flight deck is given by the Tait-Bryan angles $\eta = [\psi, \theta, \phi]^T$ while the attitude of the sensor deck is given by the analogous angles $\zeta = [\alpha, \beta, \gamma]^T$. We define the Lagrangian

$$L(q, \dot{q}) = T_{trans} + T_{rot} + U, \tag{1}$$

$$q = [\xi, \eta, \zeta]^T, \tag{2}$$

$$T_{trans} = \frac{m}{2}\dot{\xi}^T\dot{\xi}, \tag{3}$$

$$T_{rot} = \frac{1}{2}\Omega^T\hat{\mathbf{I}}_{\xi\xi}\Omega + \frac{1}{2}\Upsilon^T\check{\mathbf{I}}_{\xi\xi}\Upsilon, \tag{4}$$

$$U = -mgz, \tag{5}$$

$$\Omega = \hat{W}_v\dot{\eta}, \tag{6}$$

$$\Upsilon = \check{W}_v\dot{\zeta}, \tag{7}$$

$$\hat{W}_v(\eta) = \begin{bmatrix} -\sin\theta & 0 & 1 \\ \cos\theta\sin\psi & \cos\psi & 0 \\ \cos\theta\cos\psi & -\sin\psi & 0 \end{bmatrix}, \tag{8}$$

$$\check{W}_v(\zeta) = \begin{bmatrix} -\sin\beta & 0 & 1 \\ \cos\beta\sin\alpha & \cos\alpha & 0 \\ \cos\beta\cos\alpha & -\sin\alpha & 0 \end{bmatrix}, \tag{9}$$

$$\hat{\mathbf{I}}_{\xi\xi} = \begin{bmatrix} \hat{I}_{xx} & & \\ & \hat{I}_{yy} & \\ & & \hat{I}_{zz} \end{bmatrix},$$

$$\check{\mathbf{I}}_{\xi\xi} = \begin{bmatrix} \check{I}_{xx} & & \\ & \check{I}_{yy} & \\ & & \check{I}_{zz} \end{bmatrix},$$

where $\hat{\mathbf{I}}_{\xi\xi}$ is the moment of inertia tensor for the flight deck, $\check{\mathbf{I}}_{\xi\xi}$ is the moment of inertia tensor for the sensor deck, m is the vehicle mass, and g is the acceleration due to gravity. The vector Ω is the 3D angular velocity of the flight deck, resolved in its own frame, and Υ is the 3D angular velocity of the sensor deck, resolved in its own frame.

In (5), note that negative values of z represent positive altitude (and therefore increasing potential energy) because the z-axis of the inertial frame points toward the ground, in the direction of the Earth's gravitational force. The external forces, as resolved in the vehicle-fixed frame, are

$$\hat{F} = \begin{bmatrix} 0 \\ 0 \\ -\hat{F}_z \end{bmatrix} \triangleq \begin{bmatrix} 0 \\ 0 \\ (f_3 + f_1) + (f_2 + f_4) \end{bmatrix}, \tag{10}$$

$$\check{F} = \begin{bmatrix} 0 \\ 0 \\ -\check{F}_z \end{bmatrix} \triangleq \begin{bmatrix} 0 \\ 0 \\ (f_7 + f_5) + (f_6 + f_8) \end{bmatrix}, \tag{11}$$

$$\hat{\tau} = \begin{bmatrix} \hat{\tau}_\psi \\ \hat{\tau}_\theta \\ \hat{\tau}_\phi \end{bmatrix} \triangleq \begin{bmatrix} (f_3 + f_1)b_\eta - (f_2 + f_4)b_\eta \\ (f_2 - f_4)l_\eta \\ (f_3 - f_1)l_\eta \end{bmatrix}, \tag{12}$$

$$\check{\tau} = \begin{bmatrix} \check{\tau}_\alpha \\ \check{\tau}_\beta \\ \check{\tau}_\gamma \end{bmatrix} \triangleq \begin{bmatrix} (f_7 + f_5)b_\zeta - (f_6 + f_8)b_\zeta \\ (f_6 - f_8)l_\zeta \\ (f_7 - f_5)l_\zeta \end{bmatrix}, \tag{13}$$

where f_1-f_4 are the four outer rotor thrusts and f_5-f_8 are the four inner rotor thrusts. The parameters b_η and b_ζ are electromechanical constants corresponding to outer and inner motor sets, respectively. The parameters l_η and l_ζ are the moment arms between opposing motors on the outer and inner decks, respectively. In (10) and (11), note that the z-components of the external force vectors are negative semi-definite because the positive semi-definite cumulative thrust, $\hat{F}_z$ and $\check{F}_z$, are opposite in direction to the flight-fixed and sensor-fixed z-axes (which both point toward the ground when the flight deck and sensor deck attitudes are at their nominal horizontal equlibria). The flight-fixed axes, sensor-fixed axes, and external force vectors are illustrated in Fig. 10. Note that the sums and differences on the right-hand side of (10) and (13) are eight independent equations in the eight scalar control inputs which compose the input vector $\begin{bmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 \end{bmatrix}^T$. The elements of this vector combine to exert a 3D force and two 3D moments on the vehicle.

We map $\hat{F}$ and $\check{F}$ from their native frames to the inertial

frame using the relation

$$F_\xi = R_\eta \hat{F} + R_\zeta \check{F}, \tag{14}$$

$$R_\eta \hat{F} = \begin{bmatrix} -\hat{F}_z \sin\theta \\ \hat{F}_z \cos\theta \sin\phi \\ -\hat{F}_z \cos\theta \cos\phi \end{bmatrix}, \tag{15}$$

$$R_\zeta \check{F} = \begin{bmatrix} -\check{F}_z \sin\beta \\ \check{F}_z \cos\beta \sin\gamma \\ -\check{F}_z \cos\beta \cos\gamma \end{bmatrix}, \tag{16}$$

$$R_\eta = \begin{bmatrix} c_\theta & 0 & s_\theta \\ s_\theta s_\phi & c_\phi & -c_\theta s_\phi \\ -c_\phi s_\theta & s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{17}$$

$$R_\zeta = \begin{bmatrix} c_\beta & 0 & s_\beta \\ s_\beta s_\gamma & c_\gamma & -c_\beta s_\gamma \\ -c_\gamma s_\beta & s_\gamma & c_\beta c_\gamma \end{bmatrix}, \tag{18}$$

where R_η and R_ζ are the rotation matrices from the flight and sensor frames (respectively) to the inertial frame. The functions $\sin\theta$ and $\cos\phi$ have been abbreviated as s_θ and c_ϕ. Evaluating the Lagrangian

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \begin{bmatrix} F_\xi \\ \tau \end{bmatrix}, \tag{19}$$

we obtain the equations of motion

$$m\ddot{\xi} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = F_\xi, \tag{20}$$

$$\hat{\mathbb{J}}\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \hat{\tau}, \tag{21}$$

$$\check{\mathbb{J}}\ddot{\zeta} + C(\zeta, \dot{\zeta})\dot{\zeta} = \check{\tau}, \tag{22}$$

$$\hat{\mathbb{J}}(\eta) = \hat{W}_v^T \hat{\mathbf{I}}_{\xi\xi} \hat{W}_v, \tag{23}$$

$$\check{\mathbb{J}}(\zeta) = \check{W}_v^T \check{\mathbf{I}}_{\xi\xi} \check{W}_v, \tag{24}$$

$$C(\eta, \dot{\eta}) = \dot{\hat{\mathbb{J}}} - \frac{1}{2}\frac{\partial}{\partial \eta}(\dot{\eta}^T \hat{\mathbb{J}}), \tag{25}$$

$$C(\zeta, \dot{\zeta}) = \dot{\check{\mathbb{J}}} - \frac{1}{2}\frac{\partial}{\partial \zeta}(\dot{\zeta}^T \check{\mathbb{J}}). \tag{26}$$

Rearranging (20) and (21), we have

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} -F_z s_\theta - F_z s_\beta \\ F_z c_\theta s_\phi + F_z c_\beta s_\gamma \\ -F_z c_\theta c_\phi - F_z c_\beta c_\gamma + g \end{bmatrix}, \tag{27}$$

$$\begin{bmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} = \hat{\mathbb{J}}^{-1}\left(\begin{bmatrix} \hat{\tau}_\psi \\ \hat{\tau}_\theta \\ \hat{\tau}_\phi \end{bmatrix} - C(\eta, \dot{\eta})\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix}\right), \tag{28}$$

$$\begin{bmatrix} \ddot{\alpha} \\ \ddot{\beta} \\ \ddot{\gamma} \end{bmatrix} = \check{\mathbb{J}}^{-1}\left(\begin{bmatrix} \check{\tau}_\alpha \\ \check{\tau}_\beta \\ \check{\tau}_\gamma \end{bmatrix} - C(\zeta, \dot{\zeta})\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}\right), \tag{29}$$

where the eight inputs are $\hat{F}_z \in R$, $\check{F}_z \in R$, $\hat{\tau} \in R^3$, and $\check{\tau} \in R^3$, and the nine outputs are $\xi \in R^3$, $\eta \in R^3$, and $\zeta \in R^3$.

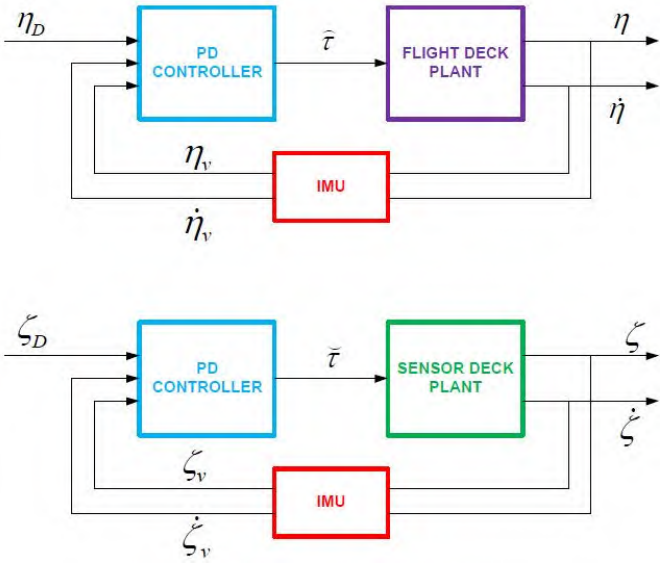


Figure 11: Block diagram of the closed-loop control system: A simple PD controller is used to implement feedback control on the Dual-quad plant. The v subscripts on the state variables represent voltage measurements from the IMU. The variables η_D and ζ_D represent the desired reference attitudes for the flight deck and sensor deck, respectively.

V. Stabilizing Controller

In this section, we develop and analyze a model-based control strategy using the equations of motion derived in the Section V. In Section V-A, we design a simple PID controller for stabilizing the attitude of both the flight deck and the sensor deck. In Section V-B, we present simulation results. In Section V-C, we present experimental results for the physical prototype.

A. Control Design

We close the loop using a simple PD controller. The controller would be more aptly termed a PP controller as no derivatives are calculated. Instead a gain is attached to the gyro measurements producing less noise than taking the derivative of the angular position measurements from the accelerometers. A diagram of the closed loop is shown in Fig. 11. The cumulative thrust inputs, $\hat{F}_z$ and $\check{F}_z$, are ignored as these can be set to arbitrary values when controlling attitude alone.

B. Simulation

The linear PD controller is simulated on the nonlinear dynamical model (27)-(29) using a Simulink block diagram. The results are given in Fig. 12. The controller is able to independently stabilize each of the six attitude states. Notice the quick rise time, absence of overshoot, and zero steady-state error. The simulated model assumes that the measurements of all of the states are available without noise corruption.

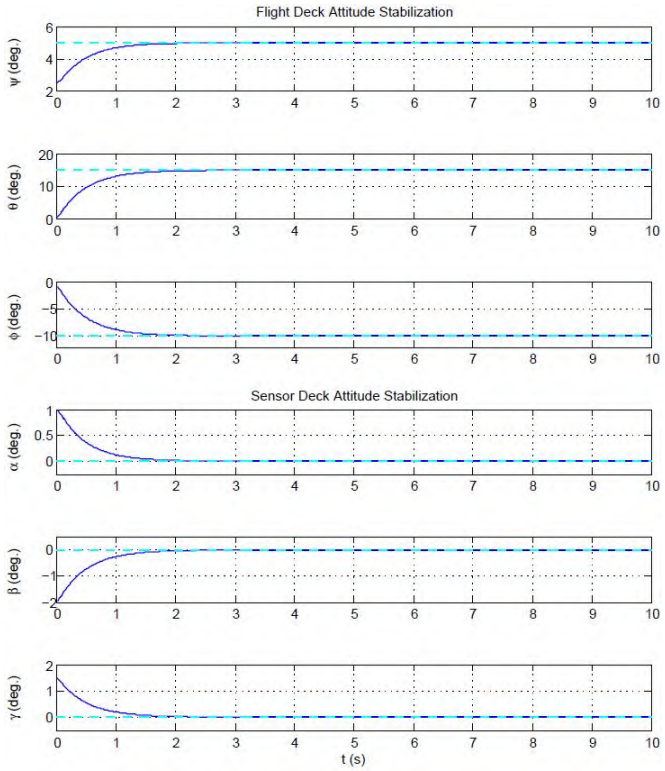


Figure 12: Simulation showing the response of the flight deck (top three axes) and the sensor deck (bottom three axes) to step changes in the flight deck's attitude reference values (dashed lines). Nonzero values are chosen for the flight deck's attitude references to simulate commanded changes in the Dual-Quad's position. The values of the sensor deck's attitude references are always set to zero by default in order to maintain a level orientation for the Kinect array.

Figure 13: Experimental Setup: The Dual-quad Observer mounted on a vertical bar suspended from the top of a safety cage. Both the flight deck and sensor deck are free to rotate in 3DOF while the vehicle position remains fixed.

C. *Experimental Results*

The purpose of this experiment is to demonstrate the independent controllability of the flight and sensor decks. The experiment is conducted inside an aluminum safety cage outfitted with nylon netting (see Fig. 13). An external ball joint is used to attach the flight deck to a rigid vertical bar suspended from the top of the cage. The bar is designed to keep the position of the Dual-quad fixed, although in practice, some motion does occur because the bar acts as a cantilevered beam. A second internal ball joint connects the sensor deck to a rod suspended from the anterior apex of flight deck's ribbed dome. Both the flight deck and the sensor deck are free to rotate in three degrees of freedom.

Angular position references are sent to the closed-loop plants (one for each deck) over USB. PID control is implemented on the embedded system at 100 Hz. The system response is viewable in Fig. 14. The stability performance of both decks is good in all three degrees of freedom. The flight deck appears to have lower steady-state error than the sensor deck although both magnitudes are acceptably small. The difference in performance may be ascribed to the difference in actuator size on the respective decks. The vibrational noise can be attributed to both experimental and intrinsic sources. The cage and the vertical bar introduce disturbances to the flight deck plant while the flight deck and the internal rod introduce disturbances to the sensor deck plant (and vice versa). When the static thrust of both decks is kept small, the translation forces vanish and the large-scale mechanical vibrations decay to zero due to natural dampening.

VI. Conclusions

As intended, we were able to demonstrate a preliminary proof of concept for mapping the environment, self-localization, and tracking UAVs from a moving platform. The bio-inspired, mechanically stabilized sensor deck provided an excellent physical base for the Kinect array. The stability response proved to be robust in the presence of vibrational noise and electromagnetic interference. In our next steps, we will continue to develop the vision system, optimizing the image processing algorithms to handle the peculiar noise issues of the Kinect data streams. We will devise a strategy for robust feature selection so that the visual odometry process can produce the highest quality measurements possible given the limitations of the hardware. Particular attention

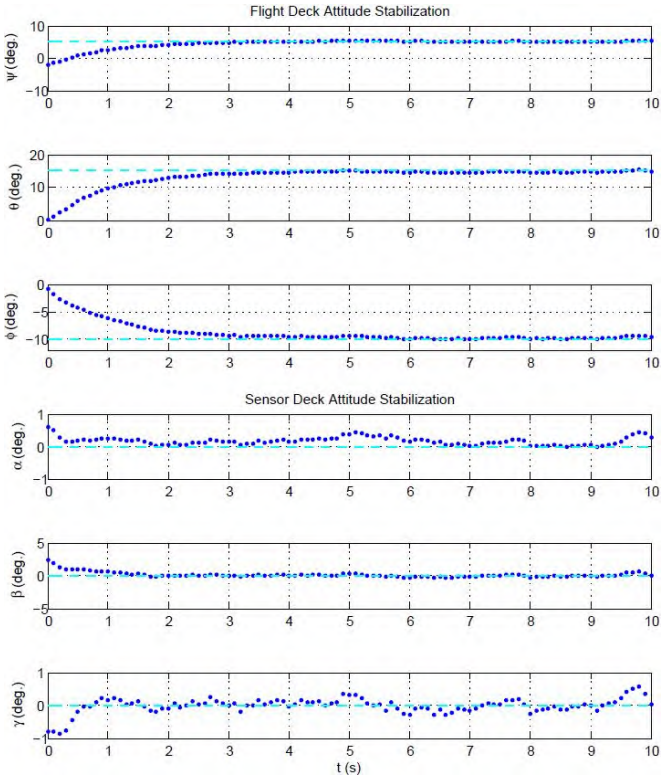


Figure 14: Experimental stabilization results for the flight deck and sensor deck: The flight deck (top three axes) responds to step changes in its attitude reference values (dashed lines) while the sensor deck (bottom three axes) remains stabilized at the nominal equilibrium of the state space.

will be paid to finding a method for increasing the depth salience of the visual markers used to track UAVs. Once the vision strategy is complete, computational performance will need to be improved before the software can be ported to the embedded system. Future work will also include designing a more complex control law for the sensor deck in order to improve disturbance rejection. Similarly, we plan to implement a position feedback controller on the flight deck. This should allow us to evaluate the performance of the vision system while the Dual-quad Observer is translating in 3D space.

REFERENCES

[1] A. Ahmadzadeh, N. Motee, A. Jadbabaie, and G. Pappas, "Multi-vehicle path planning in dynamically changing environments," in *Robotics and Automation (ICRA), Proc. 2009 IEEE Int. Conf.*, May 2009, pp. 2449 –2454.

[2] V. Desaraju and J. How, "Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees," in *Robotics and Automation (ICRA), Proc. 2011 IEEE Int. Conf.*, May 2011, pp. 4956 –4961.

[3] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," in *Experimental Robotics, Proc. 2010 Int. Symp.*, Dec 2010.

[4] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-UAV testbed," *IEEE Robot. Automat. Mag.*, vol. 17, no. 3, pp. 56–65, Sep. 2010.

[5] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *Robotics and Automation (ICRA), Proc. 2011 IEEE Int. Conf.*, May 2011, pp. 2992 –2997.

[6] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Automat. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[7] A. Haque and J. D. Dickman, "Vestibular gaze stabilization: different behavioral strategies for arboreal and terrestrial avians," *J. Neurophysiology*, vol. 93, no. 3, pp. 1165–1173, 2005.

[8] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor," in *Mechatronics (ICM), 2011 IEEE Int. Conf.*, Apr. 2011, pp. 358–362.

[9] N. Ganganath and H. Leung, "Mobile robot localization using odometry and kinect sensor," in *Emerging Signal Processing Applications (ESPA), 2012 IEEE Int. Conf.*, Jan. 2012, pp. 91–94.

[10] R. Newcombe, Izadi, S., and Hilliges, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and Augmented Reality (ISMAR), 2011 IEEE Int. Symp.*, Oct. 2011, pp. 127–136.

[11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *User Interface Software and Technology, Proc. 24th ACM Symp.*, ser. UIST '11, 2011, pp. 559–568.

[12] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control. Syst. Mag.*, vol. 25, no. 6, pp. 45–55, Dec. 2005.

[13] J. Smisek, M. Jancosek, and T. Pajdla, "3d with kinect," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE Int. Conf.*, Nov. 2011, pp. 1154–1160.