GPS – IMU based Autonomous Target Tracking Algorithm for use in UAS

Jaganathan Ranganathan #1 and William H. Semke #2
#1 University of North Dakota, Department of Earth System Science & Policy
Clifford Hall Room 333, 4149 University Ave Stop 9011, Grand Forks, ND 58202, USA
Jaganathan.ranganathan@gmail.com
#2 University of North Dakota, Department of Mechanical Engineering
Upson II Room 271, 243 Centennial Drive Stop 8359, Grand Forks, ND 58202, USA
william.semke@engr.und.edu

Abstract:

***Abstract*—Tracking a ground based target autonomously from an Unmanned Aircraft Systems (UAS) is a critical task for remote sensing or any Intelligence, Surveillance, and Reconnaissance (ISR) mission that requires precision pointing, and effective real-time data transmission/recording to the ground station. To achieve this, the authors introduce a novel non-linear closed form analytical algorithm derived based on coordinate transformation and vector algebra principles that was implemented onboard a small UAS. The Unmanned Aircraft Systems Engineering Laboratory (UASE) at University of North Dakota (UND), Grand Forks, ND developed a small UAV payload "SUNDOG" to demonstrate the autonomous tracking ability of the derived algorithm and its implementations. The major advantage of the algorithm is that, it allows the user to specify and maintains the camera/target orientation irrespective of the aircraft position and rotation, which helps the ground personnel to easily interpret the data effectively while tracking the target location in real-time. The equations provide an elegant closed form solution to a non-linear problem that can be easily and efficiently programmed. The algorithm was verified through several experimentations and demonstrated successfully in an UAS test flight. Actual flight data illustrating the effectiveness of the surveillance algorithm is presented.**

Keywords:Autonomous Tracking Algorithm, Precision Pointing, Multi-Axis Gimbal Tracking, Mini-UAV Payload.

I. INTRODUCTION

The objective of developing an autonomous tracking system is to allow small fixed wing UAS platforms to both station-keep on targets of interest and estimate accurate position information for those targets using onboard assets. Specific targets of interest vary from a stationary point on the ground (applications such as fire-fighting, surveillance, and atmospheric research) to cooperative and non-cooperative aircraft in the national airspace system for airborne sense and avoid ([1]-[2]). This concept is illustrated in Fig. 1.

An autonomous tracking algorithm for a three-axis gimbal system using Global Positioning System (GPS) and Inertial Measurement Unit (IMU) information is derived using coordinate transformation and vector algebra principles. The final outputs of the algorithm are the three rotational angles required to point/track at a known stationary/moving target with a prescribed orientation. Irrespective of the aircraft position and its rotation, the derived algorithm tracks any known stationary target on the ground maintaining the user defined orientation. A custom developed C++ programming in a Linux platform was implemented in the payload to provide a communication link with all the sensors such as the GPS, IMU, motion controllers, autopilot, etc., that sends, retrieves, and process information to track the target. It also establishes a wireless communication from the ground station to the aircraft for effective ground control operation. Several experiments have been carried out to verify the tracking ability and the results are presented.

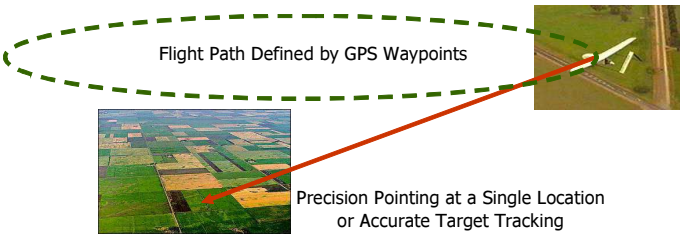

Fig. 1. Small UAS Autonomous Tracking System Concept

A small UAS imaging payload has been designed and constructed by the Unmanned Aircraft Systems Engineering (UASE) team at the University of North Dakota (UND) that allows for full 3- axis rotation of electro-optical (EO) and uncooled thermal infrared (IR) cameras ([3]-[5]). The first successful flight of this payload using the autonomous target tracking algorithm took place on October 18, 2008, in military restricted airspace over Camp Grafton Training Center, an Army National Guard Maneuver Training center near Devil's Lake, North Dakota. The payload was flown using the UND-Super Hauler UAV. Fig. 2 shows the electro-optical images captured using the autonomous target tracking algorithm during the actual flight.

Fig. 2. Electro-Optical images captured while using autonomous target tracking algorithm

Fig. 3 shows a CAD model and physical implementation of the three-axis gimbal system design, while Fig. 4 is a CAD model and physical implementation of an autonomous tracking payload known as the SUNDOG – Surveillance by University of North Dakota Observational Gimbal.

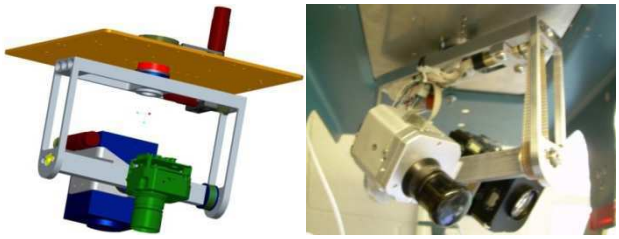

Fig. 3. Three-Axis Gimbal System CAD model (Left) and Physical implementation (Right)

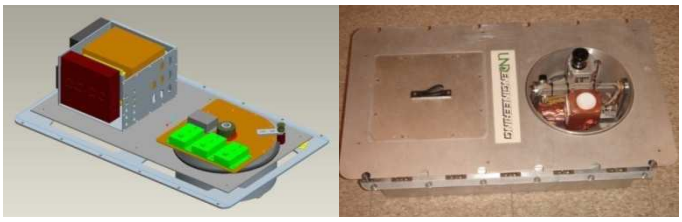

Fig. 4.CAD model (Left) and Physical Implementation (Right) of an Autonomous Tracking Payload "SUNDOG".

The SUNDOG payload has the capability to track any known stationary target location autonomously and also manually point the camera towards the target location using a joystick. For precision pointing, a non-linear closed form analytical algorithm was developed to determine the exact rotation angles for a three-axis gimbal system to point a digital imaging sensor at any known target with a prescribed orientation. For autonomous tracking, the calculated rotation angles must be provided to the gimbal system so that it can accurately locate the defined target and track it accordingly as the aircraft moves. The algorithm derived in this paper was successfully implemented and tested in the SUNDOG payload. Since the algorithm uses simple algebraic closed form expressions to calculate the pointing angles of the gimbal, it requires less computational time to track the target when compared to many other algorithms making it more easily implemented in a small UAS with limited resources ([6]-[10]). The algorithm also maintains the camera orientation to north (user-defined) in the inertial coordinate system. This allows ground personnel to compare the real time video directly to the maps without any orientation misalignment making the data much more easily understandable. The limitation of the tracking algorithm mentioned ([6]-[10]) compared to the algorithm presented

in this paper are, they do not maintain the camera orientation in the specified direction, there are discontinuities in the gimbal rotation, they are expensive, and are position and time dependent.

The SUNDOG payload has the capability for a ground-based payload operator to manually control the pointing of the cameras with a joystick based upon real time video data received from the EO camera. For this, an algorithm was developed to determine the target location based on the position, attitude information of the aircraft, and gimbal pointing angles. This paper focuses only on stationary targets, but the algorithm has the ability to be extended to ground and air borne moving targets.

II. ACCURATE GIMBAL POINTING ALGORITHMS

Kinematics Analysis of the System

A kinematic analysis is done on a three-axis gimbal system to get the appropriate model of gimbal rotations in order to point at a certain location on the ground from an aerial platform. The mathematical model includes an inertial system that has coordinates fixed to the Earth, a coordinate system that is body-fixed to the aircraft and a third coordinate system that is fixed to the gimbal. The end results of the analysis are the rotation angles, about each gimbal axis, that will result in the gimbal pointing at the correct spot with a desired orientation. Correct orientation will allow an image to have the same orientation to the ground coordinates independent of the direction of flight. The scenario investigated is when the inertial coordinates of the target, the aircraft location and the orientation angles are known that are required for accurate pointing.

As stated previously, this system includes three separate coordinate systems. These coordinates systems and their orientations are shown in Fig. 5. The orientation for the coordinate systems is arbitrary, but they do need to be defined. In this system the inertial coordinates were defined so that the x-axis is in the North direction, the y-axis is in the East direction and the z-axis is downward. The aircraft coordinates were defined with the x-axis being the same as the heading vector. The gimbal system is defined with the x-axis being the tilt axis of the gimbal and initially aligned with the x-axis of the aircraft system. The y-axis of the gimbal is the pan axis of the gimbal. The z-axis of the camera is also defined as the line-of-sight axis, which is required to help solve for the pointing and orientation parameters.

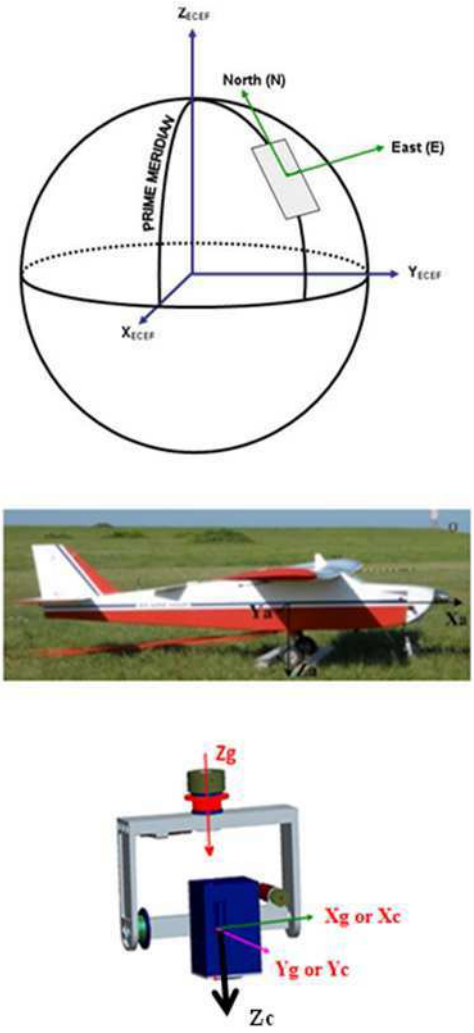

Fig. 5. Orientation of Inertial Coordinate (Top), Aircraft Coordinate (Middle), Gimbal & Camera Coordinate (Bottom)

The topic of coordinate transforms and the kinematics involved with flight have been well covered in the literature ([11]-[20]). Often it is advantageous to define a system using multiple coordinate systems. The problem with

multiple coordinate systems is that vector operations cannot be done if the vectors are defined by different coordinate systems without transformations. Therefore, the components of a vector in one coordinate can be transformed into a vector in a different coordinate system. The first transform that is needed for this analysis is a transform between the inertial system and the aircraft fixed system. For this transform, a 3-2-1 (NASA Standard Aircraft) rotation is used [13]. In this situation, ψ is the heading angle, or yaw, and corresponds to a rotation about the z-axis of the aircraft, θ is the pitch angle and is a rotation about the y-axis of the aircraft, and φ is the roll angle and is a rotation about the x-axis of the aircraft. These rotations are shown in Fig. 6. The order of rotation for this transform is ψ, θ, and then φ. The rotation matrix for this type of transform is well known and is shown in Eq. 1 as R_a.

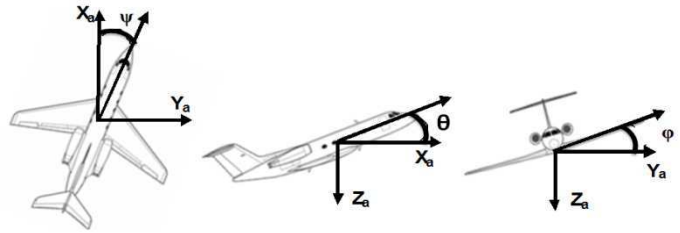

Fig. 6.Aircraft rotations of ψ, θ, and φ about the z, y, and x axis respectively.

$$[R_a]=\begin{bmatrix} c\psi c\theta & s\psi c\theta & s\theta \\ s\psi c\phi + c\psi s\theta s\phi & c\psi c\phi + s\psi s\theta s\phi & c\theta s\phi \\ s\psi s\phi + c\psi s\theta c\phi & c\psi s\phi + s\psi s\theta c\phi & c\theta c\phi \end{bmatrix} \quad (1)$$

The next transform needed is from the aircraft fixed coordinates to the gimbal's camera fixed coordinates. This is done by first rotating about the z-axis by γ, then rotating about the y-axis by β, and then rotating about the x-axis by α. The rotation matrix for each of these individual rotations is shown in Eqs. 2 – 4 as R_{1g}, R_{2g} and R_{3g}, respectively. To find the total rotation matrix needed, the rotation matrices from the individual rotations are multiplied together as shown in Eq. 5 with the total rotation matrix shown in Eq. 6 as R_g.

$$R_{1g}=\begin{bmatrix} c\gamma & s\gamma & 0 \\ -s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$R_{2g}=\begin{bmatrix} c\beta & 0 & -s\beta \\ 0 & 1 & 0 \\ s\beta & 0 & c\beta \end{bmatrix} \quad (3)$$

$$R_{3g}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & s\alpha \\ 0 & -s\alpha & c\alpha \end{bmatrix} \quad (4)$$

$$[R_g]=[R_{1g}][R_{2g}][R_{3g}] \quad (5)$$

$$[R_g]=\begin{bmatrix} c\gamma c\beta & s\gamma c\alpha + c\gamma s\beta s\alpha & s\gamma s\alpha - c\gamma s\beta c\alpha \\ -s\gamma c\beta & c\gamma c\alpha - s\gamma s\beta s\alpha & c\gamma s\alpha + s\gamma s\beta c\alpha \\ s\beta & -c\beta s\alpha & c\beta c\alpha \end{bmatrix} \quad (6)$$

The order of rotation used for the R_a matrix is arbitrary, but using a different order will result in a different rotation matrix. The order of rotation for the R_g matrix will depend on the system setup and other systems may have different locations for the drive motors of each axis or they may move in a different order than what was used here. Whatever the order, these rotation matrices define the crucial relationship for transforming between coordinate systems.

Three-Axis Gimbal Pointing Angles for Known Stationary Targets

In many remote sensing applications, a gimbal system is required to point a sensor at a given target location. To do this, the proper rotation angles (α, β, λ) must be given to the gimbal system so it can accurately locate the target. In this section an algorithm is developed to find these rotation angles. The inputs into the system are aircraft location in inertial coordinates (GPS information), aircraft attitude (roll, pitch, and yaw), target location in inertial coordinates, and the offset distance of the gimbal from the GPS receiver on the aircraft. If the GPS offset is not included or is not known, there will be errors in the angles of rotation given to the gimbal. The

analysis for finding the rotation angles is done as if looking through the camera and maintaining the z axis of the gimbal system to point along the vector going from the gimbal location to the target location.

The first step in finding the rotation angles is to define the known locations in the form of vectors. In Eqs. 7-9, the ***P*** vector is the position of the aircraft in inertial coordinates, the ***T*** vector is the position of the target in inertial coordinates and the $\boldsymbol{G_O}$ vector is the offset of the gimbal in aircraft fixed coordinates.

$$\vec{P} = \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \quad , \quad \vec{T} = \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \quad , \quad \vec{G}_O = \begin{Bmatrix} x_p \\ y_p \\ z_p \end{Bmatrix} \tag{7-9}$$

The location of the gimbal needs to be expressed in inertial coordinates to define the line of sight vector. To do this the inertial coordinates of the aircraft must first be transformed into the body-fixed coordinate system of the aircraft. This is done by multiplying the inertial coordinates by the rotation matrix, as shown in Eq. 10.

$$\{P_a\} = [R_a]\{P\} \tag{10}$$

Once this is done, the gimbal offset, which is already in the body-fixed coordinates, can be subtracted from the aircraft coordinates to give the actual location of the gimbal in body-fixed coordinates (Eq. 11).

$$\{G_a\} = \{P_a\} - \{G_O\} \tag{11}$$

The next step is to specify the required orientation of the camera image and this can be North, South, East or West. For example, to maintain the top of the camera image in North direction, the x unit vector in the gimbal system should always be aligned to the North direction, or in this case the inertial x-axis. Hence to maintain the camera orientation in Northern direction, Z axis rotation of the gimbal system (λ) should be equal and opposite to Z axis rotation of the aircrafts (ψ). Here, the gimbal location in aircraft coordinates needs to be transformed to the gimbal coordinate. This is done by multiplying the aircraft coordinate by the rotation matrix "R", as shown in Eq. 12.

$$\{G_{ag}\} = [R]\{G_a\} \tag{12}$$

Where, $R = \begin{bmatrix} c\lambda & s\lambda & 0 \\ -s\lambda & c\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and λ = - ψ.

Now the gimbal location needs to be transformed back into the inertial coordinates so that the vector between the gimbal and target can be defined. This is done by multiplying the gimbal location by the inverse of the systems rotation matrix, as shown in Eq. 13.

$$\{G\} = ([R][R_a])^{-1} - \{G_{ag}\} \tag{13}$$

The line of sight vector is then defined as the difference between the gimbal location and the target location, both in inertial coordinates, as shown in Eq. 14.

$$\{D\} = \{T\} - \{G\} = \begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{Bmatrix} \tag{14}$$

The line of sight vector then needs to be transformed into the body-fixed coordinates of the aircraft. This is done by using the aircraft rotation matrix, as shown in Eq. 15.

$$\{D_a\} = [R][R_a]\{D\} = \begin{Bmatrix} \Delta x_a \\ \Delta y_a \\ \Delta z_a \end{Bmatrix} \tag{15}$$

The final rotation is to transform the pointing vector into the gimbal fixed coordinate system. The angles needed for this rotation are the angles that will be the pointing angles for the gimbal. Therefore they are unknown, but, by using specific constraints, the equations from the rotation will provide the pointing angles. The transformation is shown in Eq. 16.

$$\{D_g\} = [R_g]\{D_a\} =$$
$$\begin{Bmatrix} (c\gamma c\beta)\Delta x_a + (s\gamma c\alpha + c\gamma s\beta s\alpha)\Delta y_a + (s\gamma s\alpha - c\gamma s\beta c\alpha)\Delta z_a \\ (s\gamma c\beta)\Delta x_a + (c\gamma c\alpha - s\gamma s\beta s\alpha)\Delta y_a + (c\gamma s\alpha + s\gamma s\beta c\alpha)\Delta z_a \\ (s\beta)\Delta x_a + (-c\beta s\alpha)\Delta y_a + (c\beta c\alpha)\Delta z_a \end{Bmatrix} = \begin{Bmatrix} \Delta x_g \\ \Delta y_g \\ \Delta z_g \end{Bmatrix} \tag{16}$$

In this system, the z-axis of the camera falls along the line of sight vector and this fact is used to solve the equations for the unknown variables α, β, and γ. Therefore, when the line of sight vector is expressed in terms of the gimbal fixed coordinate system, it will not have any components in the x or y direction of the gimbal system. This provides the following two equations (Eqs. 17-18).

$$\Delta x_g = 0 \; ; \; \Delta y_g = 0 \tag{17-18}$$

Therefore, by applying Eqs. 17-18 in Eq. 16, it reduces to the following equations, as expressed in Eqs. 19-20.

$$(c\gamma c\beta)\Delta x_a + (s\gamma c\alpha + c\gamma s\beta s\alpha)\Delta y_a + (s\gamma s\alpha - c\gamma s\beta c\alpha)\Delta z_a = 0 \tag{19}$$

$$(s\gamma c\beta)\Delta x_a + (c\gamma c\alpha - s\gamma s\beta s\alpha)\Delta y_a + (c\gamma s\alpha + s\gamma s\beta c\alpha)\Delta z_a = 0 \tag{20}$$

This results in two equations with three unknown variables. To solve Eqs.(19-20), the three unknown variables are reduced to two unknown variables by finding any one of the variables α, β, or γ. For this system, the variable γ is found by defining a constraint. This constraint comes from the fact that the x axis of both the gimbal/camera fixed coordinate and the inertial coordinate have been chosen to be aligned (parallel) to each other. Since, the x axis of the gimbal coordinate is already aligned to the x axis (North direction) of the inertial coordinate, which is obtained by Eq. 12, the rotation about the z axis of the camera or γ is equal to zero in the gimbal coordinate system. Applying this constraint in Eqs.19-20, Eqs. 21-22 are found.

$$(c\beta)\Delta x_a + (s\beta s\alpha)\Delta y_a - (c\gamma s\beta c\alpha)\Delta z_a = 0 \tag{21}$$

$$(c\alpha)\Delta y_a + (s\alpha)\Delta z_a = 0 \tag{22}$$

Solving Eq. 22 with respect to α gives Eq. 23

$$\alpha = -\tan^{-1}\left(\frac{\Delta y_a}{\Delta z_a}\right) \tag{23}$$

Similarly solving Eq. 21 with respect to β gives Eq. 24

$$\beta = -\tan^{-1}\left(\frac{\Delta x_a}{s\alpha\Delta y_a - c\alpha\Delta z_a}\right) \tag{24}$$

To maintain the camera orientation, the rotation about the Z axis of the gimbal system (λ) should be equal and opposite to ψ, as shown in Eq. 25.

$$\lambda = -\psi \tag{25}$$

With all three of these rotation angles defined (α, β, λ), a controller can give precise commands to the gimbal system to accurately locate a target and orient the camera's image, so that the top is always in the northern direction.

The algorithm derived here is a novel, nonlinear closed form analytical expression (Eqs. 23-25) which uses position and attitude information of the aircraft to calculate the exact rotation angles for a three-axis gimbal system to point a digital imaging sensor at a target. Since the algorithm uses simple algebraic expressions to determine the gimbal pointing angles, pointing the imaging sensors towards the target is computational efficient. The authors are not aware of any other published algorithms that result in closed form solutions with heading corrections for use onboard an UAS.

III. IMPLEMENTATION

A. SUNDOG

The Surveillance by University of North Dakota Observational Gimbal (SUNDOG) payload includes a three-axis precision pointing system for an EO camera and an uncooled thermal IR camera. This payload was developed to capture both the infrared and electro-optical video image of the ground which will be useful in agricultural and surveillance applications. Initially, a sensor operator on the ground was able to aim the two cameras in flight via a joystick control, useful for

applications in surveillance and target tracking. To eliminate the use of joystick and fully automate the target tracking, the novel nonlinear closed form algorithm presented was developed. This algorithm uses position and attitude information of the aircraft to calculate the pointing angles of the gimbal system. Most of the hardware and software utilized in the payload design is commercial-off-the-shelf or share-ware in nature. Fig. 7 provides a diagram of the commercial-off-the-shelf (COTS) hardware utilized in the payload and ground control station design.

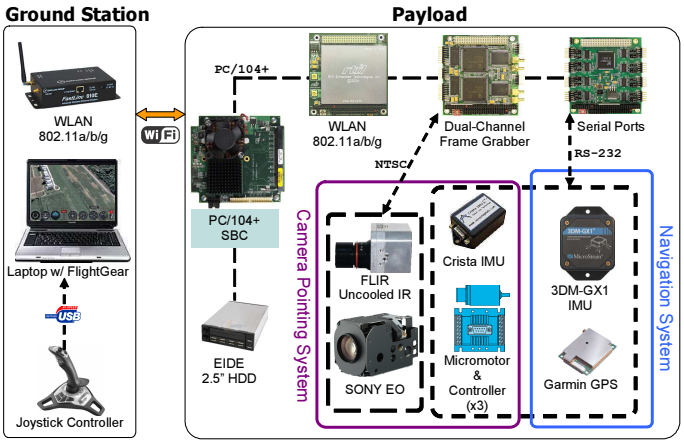

Fig. 7. COTS hardware and software diagram for the SUNDOG payload.

B. Software

The camera pointing system demands a complex coordination of software to accurately position the cameras based on ground station commands. The ground station runs Windows XP and includes applications such as Flight Gear, Putty, and VLC. Flight Gear is an open-source flight simulation application that allows wireless commands to be sent to the payload. The payload utilized a Linux operating system called OpenSuse 10.2 and includes OpenSSH and VLC applications along with a custom-written C++ program. Putty and OpenSSH permits secure system commands to be sent from the ground station to the payload. VLC wirelessly streams the video from the two cameras on the payload, which can then be received by the ground station and shown for near real-time feedback. The custom C++ program performs several tasks such as receiving the ground station's motor control commands and sending IMU and GPS data to the ground station. Wireless control of the payload motors from the ground station was achieved by using the PC/104+ stack. The motor control commands originate from a three-axis joystick connected to the ground station via a high-speed USB port. Flight Gear recognizes the joystick input and changes values within the program's property tree accordingly. Flight Gear's property tree contains a list of flight control variables that can be used to manipulate a remote aircraft or other system. There are other variables within the property tree that, when changed by incoming data, communicate to Flight Gear the navigation, orientation, and system status of the payload. In the case of motor control, the three-axis joystick inputs are interpreted and assigned to three of the Flight Gear flight control properties. While Flight Gear is running, it can constantly send these control property values to a designated IP address using the user datagram protocol (UDP) communication protocol.

On the payload side, custom C++ program was developed to set up a host socket to receive commands from the ground station IP address. The payload receives the UDP packets, which contain the control properties from Flight Gear. The programming code finds the desired properties, performs calculations on the property values, and sends the corresponding velocity commands to the three motion controllers through an RS-232 serial port. These parameters are translated into motor motion at the desired velocity in revolutions per minute. The C++ code compiled on the payload PC/104+ SBC utilizes POSIX threads or pthreads. These pthreads allow the C++ code continuously execute several operations, including scanning for incoming UDP packets, making calculations on the Flight Gear properties, and sending velocity commands to the motion controllers.

Last addition to the custom C++ program allows for position feedback from the motors. To achieve this, a method was added to the coding sequence, which sends a position query to each controller in the same manner that velocity commands are sent. The controller retrieves the position value from the incremental encoder and relays this value back to the PC/104+ SBC through the serial port as a string of ASCII characters. The C++

program parses the incoming serial data and outputs the motor position. This ability to retrieve motor position is important in automated object tracking.

The final feature added to the custom C++ program was implementing the autonomous tracking algorithm which uses the position and attitude information of the aircraft. This program receives the INS/GPS information from the Piccolo autopilot and calculates the gimbal rotation angles based on the procedure explained in Section II. After calculating the gimbal rotation in number of ticks, the program sends the appropriate commands to the motor controller to rotate the gimbal. The payloads transmit the INS/GPS and gimbal rotation values real-time to the ground station. The processing time to compute the rotational angles of the gimbal system is approximately the same as that of the input frequency of the system. For example, if the Piccolo sends the information to the program at a 20 Hz rate, then the processing rate of the program will be approximately 20 Hz. There is not a significant time delay in processing the algorithm because it computes the information within one millisecond.

In order to find the processing time of the gimbal system to attain the desired position from its current position, an analysis was carried out by using a timer function which compute the exact time taken by the gimbal system to reach the desired position. The three rotational angles (Alpha, Beta, Gamma) of the gimbal system were given as 5 degrees individually and the algorithm was run to determine the time taken to rotate the gimbal system. This experiment yielded an average of 0.01 seconds to manipulate and rotate the gimbal system accordingly. Based on the analysis, the gimbal system can operate at a maximum frequency of approximately 100 Hz. Since the algorithm does not require significant computational time to process and rotate the gimbal system, it was implemented and demonstrated successfully in real-time using the onboard SBC at a rate of 20 Hz.

IV. EXPERIMENTATION

In order to verify the tracking ability and the accuracy of the three-axis gimbal system, several experimentations including a laboratory experiment, mobile ground vehicle test, and actual flight test were carried out. Initially, a laboratory experiment was conducted to verify the pointing algorithm with change in attitude information by fixing the aircraft and the target location. Then an experiment using a mobile vehicle test was carried out by fixing the target location and changing the position of the payload. As the payload moves along with the vehicle, the Piccolo updates its GPS and attitude information which are then used to calculate the pointing angles of the gimbal system. While the pointing algorithm points the camera to the target location, a real time EO and IR video are simultaneously streamed and saved to determine whether the camera is pointing at a target or not. Similarly, flight tests with the payload was carried out by fixing the target location at the ground and specify the target location to point the gimbal based on the GPS and attitude information of the aircraft.

A. Actual Flight Test

The flight test was carried out at the Camp Grafton Training Center (CGTC) with custom developed UAV called “Super Hauler,” owned and operates by the UASE, to verify the SUNDOG payload with autonomous tracking algorithm. CGTC is a National Guard Maneuver Training Center operated by the North Dakota National Guard. CGTC is located 45 miles South of Devils Lake in northeast North Dakota. Based on UASE team requirement, the UAV “Super Hauler,” shown in Fig. 8, was built by Bruce Tharpe Engineering.

Fig. 8. UND's UAV "SuperHauler"

UAV "Super Hauler" is made of balsa wood and plywood with aluminum and carbon fiber elements. It has a wingspan of 3.7 meters, a wing area of 2.4 square meters, and 3 meters of length. The dry weight of the vehicle is 21.8 kilograms and has 13.6 kilograms of maximum payload carrying capacity. The main payload compartment has a volume of 0.054 cubic meters. Additionally, there is an open space in the rear part of the fuselage of about 0.035 cubic meters used for placement of avionics (including autopilot).The BTE Super Hauler is powered by a Desert Aircraft (DA) 100cc two stroke engine, whose specifications are shown in Table 1.

TABLE I
DA-100CC SPECIFICATION

Displacement	6.10 ci (100 cc)
Output	9.8 hp
Weight	5.8 lbs (2.63 kilos)
Bore	1.6771 in (42.6 mm)
Stroke Length	1.3779 in (35 mm)
RPM Range	1,000 to 6,700
Max. RPM	8,500
Fuel Consumption	1.172 gallons/hour @ 6000 RPM

Before the actual flight test, a preflight meeting was conducted with the internal and external pilot in command to discuss and decide the target location, actual flight path and the altitude required during the testing. A blue colored tarp of dimension 2 × 2 m was fixed on the ground and its GPS coordinate was used as the target location in the tracking algorithm. Fig. 9 shows the aircraft flight path obtained from the flight test in Google Earth. Apart from the preplanned flight path, different flight path patterns are carried out to test the target-locked and out of range region. The term "Target-Locked" refers to the aircraft position when the camera was pointed at the target and "Target-Out of Range" referred as the aircraft location when camera was not pointed to the target due to the angular limitation of the gimbal system. The NED coordinate information is shown in the Fig. 10 which illustrates that the aircraft was flown at a constant altitude during the autopilot stage. Fig. 11 shows the attitude information of the aircraft and Fig. 12 shows the rotation angles of a three-axis gimbal system in degrees. From Fig. 12, it can be noted that the gimbal rotational angles follows an identical pattern during the autopilot stage, as expected due to the repeated flight path.

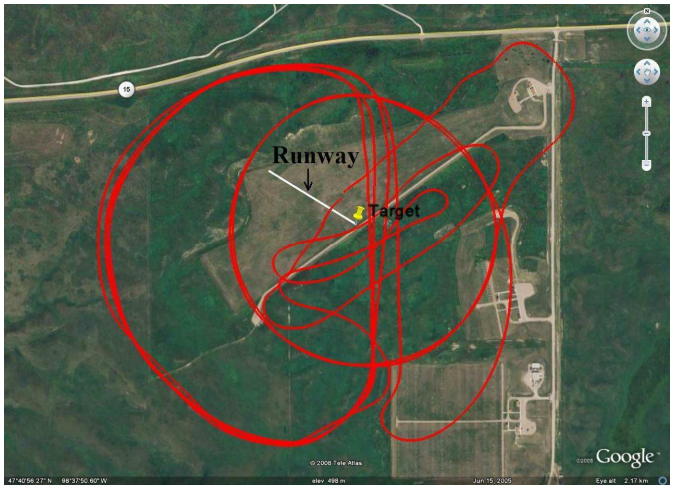

Fig. 9. Aircraft's flight path shown in Google Earth

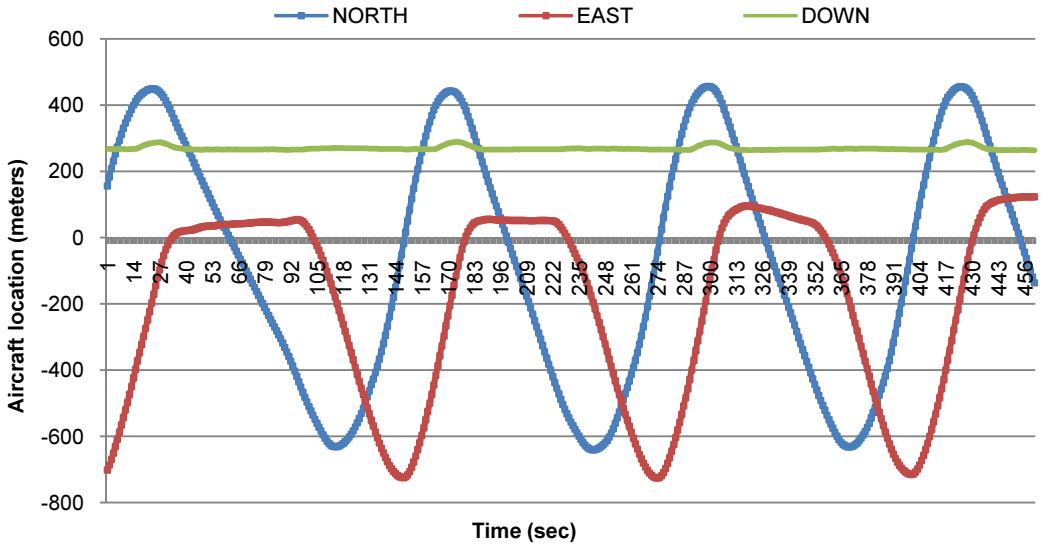

Fig. 10. Aircraft's Position – Flight Test

Fig.10-12 shows that as the North value decreases, the Y-axis rotation of the gimbal system (beta) increases. This similar pattern was obtained during the MGV test. Fig.13-15 shows the actual video snapshots obtained during the flight test with the EO camera. Since the orientation was specified to be in North direction, the algorithm maintains the camera orientation always in the North direction regardless of the aircraft position and rotation. This is seen in the images that the target remains in the center and the road maintains its orientation while the aircraft moves along its flight path. Note the changes in GPS location, IMU readings, and resulting gimbal rotations for the snapshots shown.

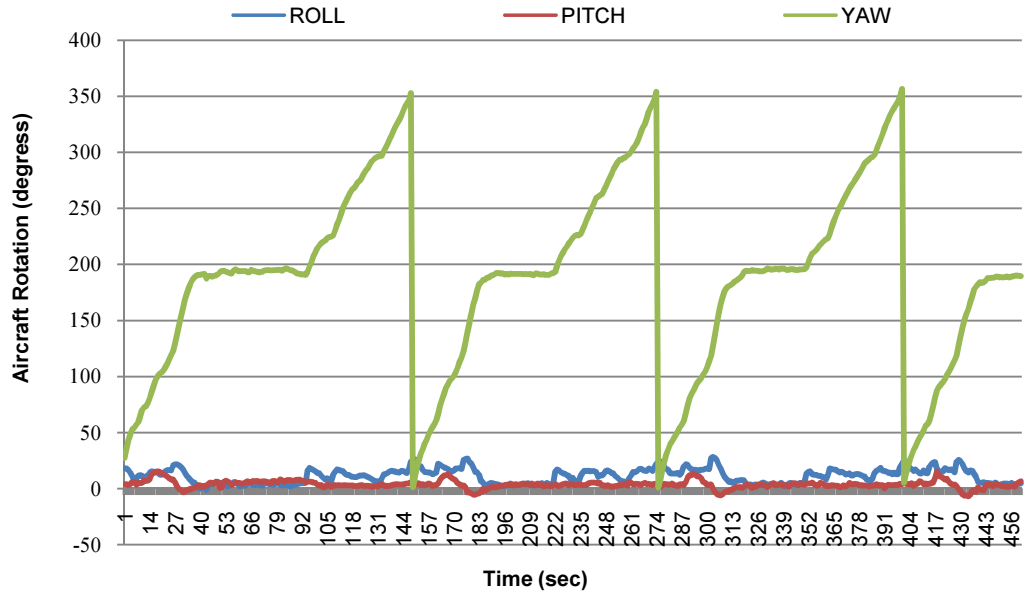

Fig. 11. Aircraft's Attitude information – Flight Test

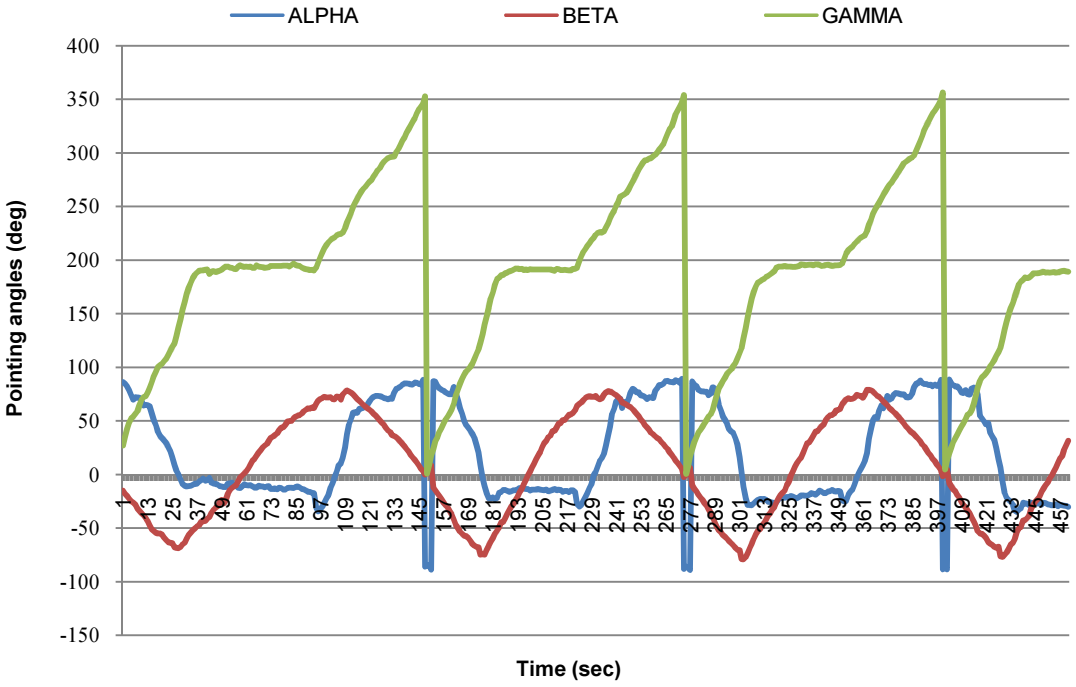

Fig. 12. Gimbal Rotation based on GPS & IMU

Fig. 13. EO video snapshot # 1 obtained from the flight test.

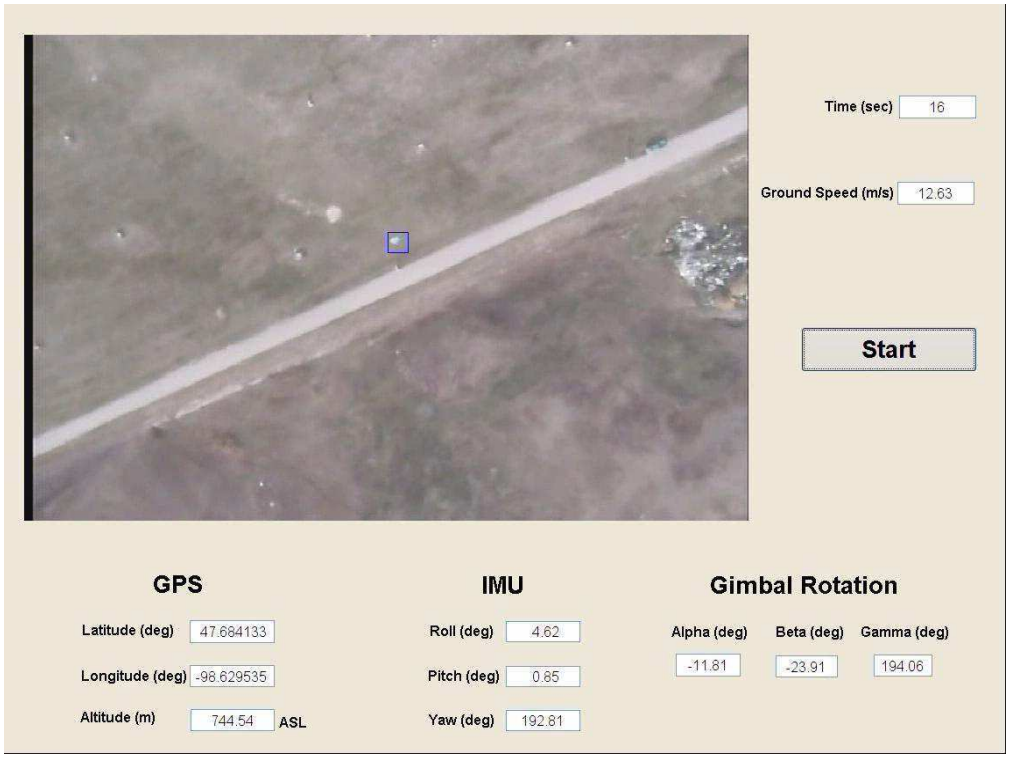

Fig. 14. EO video snapshot # 2 obtained from the flight test.

Fig. 15. EO video snapshot # 3 obtained from the flight test.

Based on the data obtained, an error analysis was carried out to estimate the tracking accuracy. The actual distance between the aircraft and the target location are measured in terms of NED coordinates, and then compared with the distance moved by the pointing algorithm to calculate the accuracy of the tracking algorithm. Fig. 16 shows the graphical form of target offset obtained in the flight test experimentation.

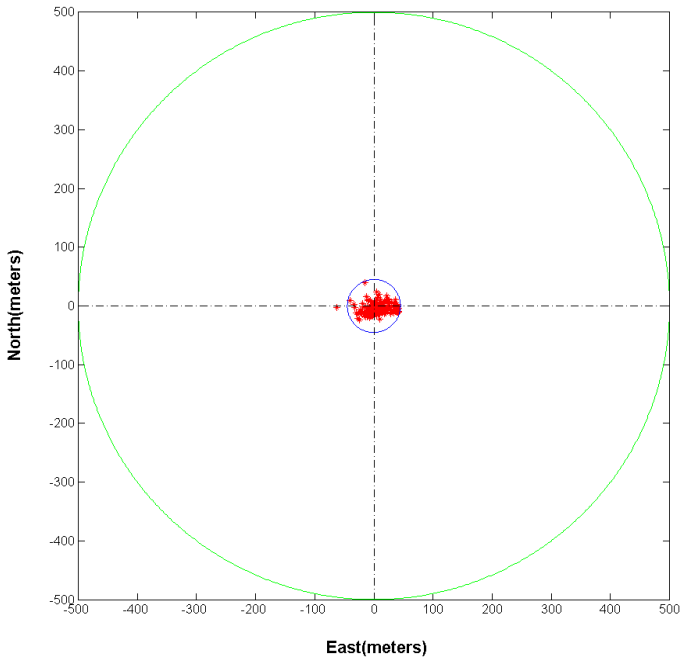

Fig. 16. Target Offset – Flight Test

In Fig. 16, green circle represents the maximum field of view of the camera (500 m) at an altitude of 300 m and the blue circle represents the error obtained from the flight test including the IMU error (±2 deg), GPS error (3-6 m) and gimbal manual setup error (±2 deg). The results obtained from the actual flight test are more significant and promising because the target offsets are well within the maximum error possible range, the target was always in the field of view of the camera when the target was locked and the camera orientation was always aligned in the specified direction (North). The accuracy of the algorithm obtained from the flight test is 95 % with an average of 22 meters and a standard deviation of 8 meters. Another error analysis was carried out to find out the angular error occurred due to the target offset. The angular error is calculated based on target offset obtained and the normalize distance between the aircraft and the target location. Based on the angular error analysis calculation, an average error of 3.45 degrees and -1.77 degrees are obtained in the East and North direction, respectively. The results obtained from the angular error analysis shows a significant and better understanding of the data because the angular error obtained in terms of North and East direction are well within the maximum possible error ±6 degrees, which includes the IMU error (±2 degrees), gimbal manual setup error (±2 degrees), and GPS error (3 - 6 meters).

Fig. 17-18 shows the aircraft location when the target was locked and out of range, respectively. In Fig. 17, the green pattern represents that the target was locked and the red pattern represents the target was out of range due to the angular restriction in the gimbal system. The gimbal rotation angles are restricted (Alpha = ± 25 deg& Beta = ± 75 deg) due to the physical structure and wiring connection of the gimbal. Since the roll limitation of the gimbal system (± 25 deg), the target tracking was restricted to a minimum distance in the East-West direction. The distance varies depending upon the altitude of the aircraft. This is the current condition, but by increasing the roll limit to 80 degrees (future), the aircraft can have broader aerial field of view to track the target, as shown in Fig. 18.

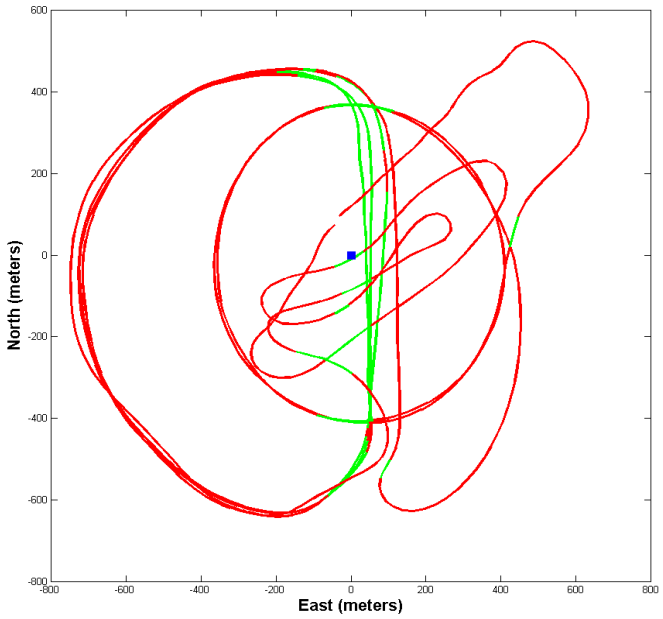

Fig. 17. Target Locked and Out of Range when Alpha = ± 25 deg& Beta = ± 75 deg

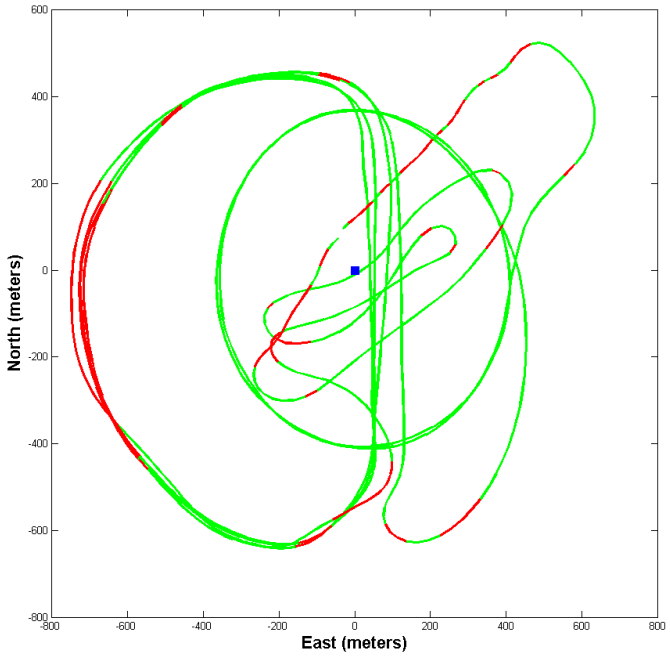

Fig. 18. Target Locked and Out of Range when Alpha = ± 80 deg & Beta = ± 75 deg

Based on the error analysis and the EO video images obtained from the flight test, we can conclude that the target specified was always centered in the field of view of the camera when the target was locked. The results obtained from the flight test are very significant and promising because the target was well within the maximum error possible range and the camera was always oriented in the specified direction (North).

V. CONCLUSION

A novel non-linear, closed form analytical expression for the three-axis gimbal system was derived to calculate the exact pointing angles with specified orientation based on GPS and IMU information of the aircraft. The orientation of the camera can be specified in any direction (North, East, West, and South). The autonomous target tracking (ATT) algorithm was successfully implementation in a payload called "SUNDOG"- Surveillance by University of North Dakota Observational Gimbal installed in a small UAS. ATT was tested through several experimentation including a laboratory, a MGV test, and actual flight tests. An average accuracy of 95 %, or 5 % inaccuracy, with a standard deviation of 8 meters and an average of 22 meters was obtained in the flight experiments. The results obtained from the experiments are very significant and promising since the target was always in the field of view of the camera and within the maximum error possible range including the IMU, GPS and gimbal manual setup error. Also, the target was always maintained in the specified orientation (North). Without any human interface, the ATT algorithm was demonstrated successfully in real-time using a custom developed C++ program in an UAS flight.

ACKNOWLEDGMENTS

University of North Dakota Department of Mechanical Engineering and the UASE Laboratory, North Dakota Department of Commerce, United States Air Force UAV Battle Lab contract number FA4861-06-C-C006, NASA Grant NNG05WC01A, Intercollegiate Academic Funding (UND), Research Development and Compliance (RD&C).

REFERENCES

[1] W. Semke, J. Ranganathan, and M. Buisker, "Active Gimbal Control for Surveillance using Small Unmanned Aircraft Systems," International Modal Analysis Conference XXVI (2008).

[2] J. Ranganathan, and W. Semke, "Three-Axis Gimbal Surveillance Algorithms for Use in Small UAS," ASME - IMECE, **1,** 31-40 (2008).

[3] Lendway, M., Berseth, B., Trandem, S., Schultz, R., and Semke, W., "Integration and Flight of a University-Designed UAV Payload in an Industry-Designed Airframe," Proceeding of the Association Unmanned Vehicle Systems International (2007).

[4] W. Semke, R. Schultz, D. Dvorak, S. Trandem, B. Berseth, and M. Lendway, "Utilizing UAV Payload Design by Undergraduate Researchers for Educational and Research Development," ASME IMECE, **7**, 113-120 (2007).

[5] M. Lendway, B. Berseth, F. Martel, S. Trandem, and K. Anderson, "A University-Designed Thermal-Optical Imaging Payload for Demonstration in a Small Experimental UAS," AIAA (2007).

[6] Bal, A., and Alam, M.S., "Automatic Target Tracking in FLIR Image Sequences Using Intensity Variation Function and Template Modeling", IEEE Transactions on Instrumentation and Measurement, **54** (5), 1846-1852 (2005).

[7] M. Kirchhof, and U. Stilla, "Detection of moving objects in airborne thermal videos",ISPRS Journal of Photogrammetry and Remote Sensing, **61** (3-4), 187-196 (2006).

[8] J. P. Le Cadre, Olivier Trimois, "Bearings-only tracking for maneuvering sources", IEEE Transactions on Aerospace and Electronic Systems, **34** (1), 179-199 (1998).

[9] A.J. Lipton, H. Fujiyoshi, R.S. Patil, "Moving Target Classification and Tracking from Real-time Video," Proc. IEEE Workshop Application of Computer Vision, 8-14 (1998).

[10] Carl C. Liebe, Kenneth A. Brown, SurapholUdomkesmalee, Curtis W. Padgett, Michael P. Brenner, Ayanna M. Howard, Terry R. Wysocky, David I. Brown, and Steven C. Suddarth "VIGIL - a GPS based target-tracking system", Proc. SPIE, **3365,** 10 (1998).

[11] J. Ranganathan, "Closed Form Analytical Multi-Axis Gimbal Tracking Algorithms for use in Small Unmanned Aircraft Vehicles", M.S. Mechanical Engineering, University of North Dakota (2008).

[12] Buisker, M., "Statistically Significant Factors that Affect the Pointing Accuracy of Airborne Remote Sensing Payloads," M.S. Mechanical Engineering, University of North Dakota, (2007).

[13] H. Baruh, "Analytical Dynamics," WCB/McGraw-Hill (1999)

[14] W. F. Phillips, C. E. Hailey, and G. A. Gebert, "Review of Attitude Representations Used for Aircraft Kinematics," Journal of Aircraft, **38** (4), 718-737 (2001).

[15] P.C. Hughes, "Spacecraft Attitude Dynamics," Dover Publications, Inc., Mineola, New York (2004).

[16] J. M. Hilkert, "Kinematic Algorithms for Line-of-Sight Pointing and Scanning using INS/GPS Position and Velocity Information," Proceedings of SPIE, **5810**, 11-22 (2005).

[17] Weiss, H., "Quaternion-Based Rate/Attitude Tracking System with Application to Gimbal Attitude Control," Journal of Guidance, Control, and Dynamics, **16** (4), 609-616 (1993).

[18] M. Quigley, M.A. Goodrich, S. Griffiths, A. Eldredge, and R.W. Beard, "Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera," IEEE - ICRA, 2600-2605 (2005).

[19] Yoon Sugpil, and J.B. Lundberg, "Equations of Motion for a Two-Axes Gimbal System", Aerospace and Electronic Systems, IEEE Transactions, **37** (3), 1083–1091 (2001).

[20] P. Goodell, M. Lendway, D. Dvorak, A. Rhoads, and S. Trandem, "UAV payload and Ground Station Development: Camera Pointing System", Senior Design Project Report, University of North Dakota (2007).