

Robust Moving Target Handoff in GPS-Denied Environments

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-06-2021 / 29-06-2021

CITATION

Beard, Randal; Tolman, Skyler; Morris, Devon; K. Peterson, Cameron; Gupta, Riten (2021): Robust Moving Target Handoff in GPS-Denied Environments. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.14869476.v1>

DOI

[10.36227/techrxiv.14869476.v1](https://doi.org/10.36227/techrxiv.14869476.v1)

Robust Moving Target Handoff in GPS-Denied Environments

Skyler Tolman, Randal W. Beard *Fellow, IEEE*, T. Devon Morris, Cameron K. Peterson, Riten Gupta *Member, IEEE*

Abstract—Unmanned aerial systems (UAS) are effective for surveillance and monitoring, but struggle with persistent, long-term tracking due to limited flight time. Persistent tracking can be accomplished using multiple vehicles if one vehicle can effectively hand off the tracking information to another replacement vehicle. In this paper we propose a solution to the moving-target handoff problem in the absence of GPS. The proposed solution uses a nonlinear complimentary filter for self-pose estimation using only an IMU, a particle filter for relative pose estimation between UAS using a relative range measurement, visual target tracking using a gimbaled camera when the target is close to the handoff UAS, and track correlation logic using Procrustes analysis to perform the final target handoff between vehicles. We present extensive simulation results that demonstrates the effectiveness of our approach and perform Monte-Carlo simulations that indicate a 97% successful handoff rate using the proposed methods.

I. INTRODUCTION

Many surveillance and monitoring applications make use of small unmanned aerial systems (sUAS) which are relatively inexpensive, agile, and easy to deploy. These smaller vehicles, however, often have a limited fuel or battery capacity and cannot operate for extended periods of time. The limited flight time makes it difficult for a single sUAS to persistently track or monitor ground activity. This issue can be overcome by utilizing multiple vehicles to cooperatively monitor an area, while sharing global information between the vehicles or with a central station, such as in [1], [2]. While these types of multi-agent approaches are effective, they typically rely upon GPS to coordinate locations and information. GPS signals are not always reliable and can even be susceptible to jamming and spoofing [3]. Accordingly, solutions that are independent of GPS can allow for unhindered operation in a wider variety of situations. This paper focuses on providing a robust end-to-end solution for persistent tracking of moving ground targets from fixed-wing UAS without GPS.

The primary difficulty in tracking ground targets for extended periods of time without GPS is the transition or “hand-off” between two sUAS. When one vehicle that is tracking a

target becomes low on fuel, another vehicle must be deployed to replace the current sUAS without any loss of information. Enabling this handoff scenario is the primary motivation for this paper. The UAS that is currently tracking the target of interest is referred to as the “tracking UAS” and the oncoming replacement vehicle is referred to as the “handoff UAS.”

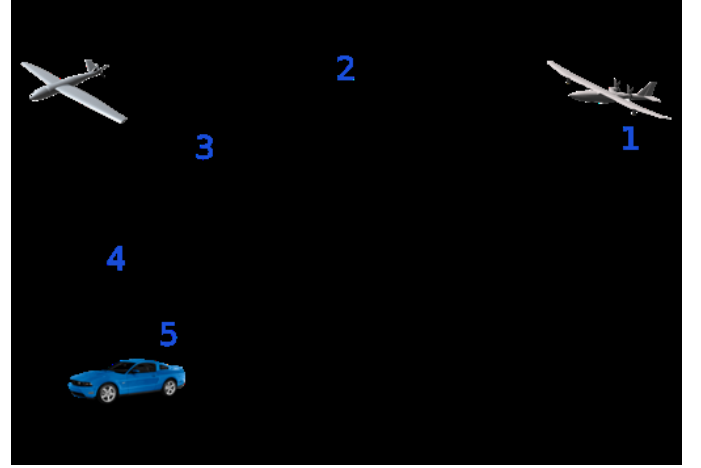


Fig. 1. Diagram of the target handoff problem. The numbers depict the five main components of the handoff problem, 1) self-pose estimation, 2) relative-pose estimation, 3) orbit insertion, 4) target tracking, and 5) the handoff logic. The equation and related symbols depict Equation (1)

Throughout the paper we will use the following notation. The position vector from a to b expressed in coordinate frame c will be denoted $\mathbf{p}_{b/a}^c$. The target will be denoted by t , the vehicle currently tracking the target will be denoted by r , and the handoff vehicle will be denoted by h . The body frame of the tracking and handoff vehicles will be denoted by r_b and h_b respectively, and the local level frame (unrolled and unpitched body frame) will be denoted by r_ℓ and h_ℓ . The inertial frame is denoted by i . The transformation from coordinates in frame a to coordinates in frame b will be denoted R_a^b . Using this notation, the geometry of Figure 1 shows that the fundamental geometric relationship for the handoff problem is given by

$$\mathbf{p}_{t/h}^{h_b} = R_{h_\ell}^{h_b} \left(\mathbf{p}_{t/r}^{r_\ell} + R_{r_\ell}^{h_\ell} R_{r_b}^{r_\ell} \mathbf{p}_{t/r}^{r_b} \right), \quad (1)$$

where the objective is to find $\mathbf{p}_{t/h}^{h_b}$, the position of the target relative to the handoff vehicle, in the body frame of the handoff UAS, given $\mathbf{p}_{t/r}^{r_b}$, the position of the target relative to the tracker, expressed in the body frame of the tracking UAS.

The handoff problem can be broken into the following five main parts, as depicted in Figure 1.

- 1) **Self-pose Estimation:** Self-pose estimation refers to the estimation of the rotation matrices $R_{r_\ell}^{r_b}$ and $R_{h_\ell}^{h_b}$,

This research was supported through the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation-sponsored industry/ university cooperative research center (I/UCRC) under NSF Award No. IIP-1161036 in conjunction with Air Force STTR contract number FA8650-15-M-1941.

Skyler Tolman was a Master’s student with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT, 84602 USA (e-mail: skyler237@gmail.com).

Randal Beard is with the Department of Electrical and Computer Engineering at Brigham Young University.

Devon Morris was a Doctorate student with the Department of Electrical and Computer Engineering at Brigham Young University.

Cameron Peterson is with the Department of Electrical and Computer Engineering at Brigham Young University (email: cammy.peterson@byu.edu).

Riten Gupta is with UtopiaCompression Corporation, Los Angeles, CA 90064 USA (email: riten@utopiacompression.com)

which represent the transformation between the local-level frames and the body frames of the tracker and handoff vehicles respectively. In essence, this requires estimating the roll and pitch angles of each vehicle using only the IMU and pressure sensors.

- 2) **Relative-pose Estimation:** Relative pose estimation is the problem of estimating $\mathbf{p}_{r/h}^{h_\ell}$, the position of the tracker relative to the handoff vehicle, expressed in the local-level frame of the handoff UAS, and $R_{r/h}^{h_\ell}$ the transformation from the local-level frame of the tracker to the local-level frame of the handoff UAS.
- 3) **GPS-denied Orbit Control:** Using the estimate of the target's position, the handoff vehicle then inserts itself into a similar orbit about the target. Initially the handoff UAS will use the relative line-of-sight (LOS) vector $\mathbf{p}_{t/h}^{h_b}$ to the target from Equation (1) to navigate into an orbit about the target. After the handoff process is complete, the handoff vehicle will orbit the target completely based on visual information, independent of measurements from the tracking UAS.
- 4) **Multiple-target Tracking:** Once the handoff UAS is in orbit about the target, it utilizes a gimballed camera to track the target. There is also the possibility that there are multiple moving objects on the ground, so the tracking algorithm must be capable of simultaneously tracking an arbitrary number of moving targets in real time.
- 5) **Handoff Logic:** When the handoff UAS is successfully tracking moving targets on the ground, it must use information from the tracking UAS to ensure it selects the correct target. Once the handoff UAS is sufficiently confident that it is tracking the correct target, it signals to the tracking UAS that the handoff is complete and transitions to using the visual LOS to orbit the target. At that point, the tracking UAS is safe to leave the area and the handoff is complete.

Figure 2 provides a system-level view of these five components and how information flows between them, where the final outward facing arrows for each UAS represent the roll command output of the GPS-denied orbit control.

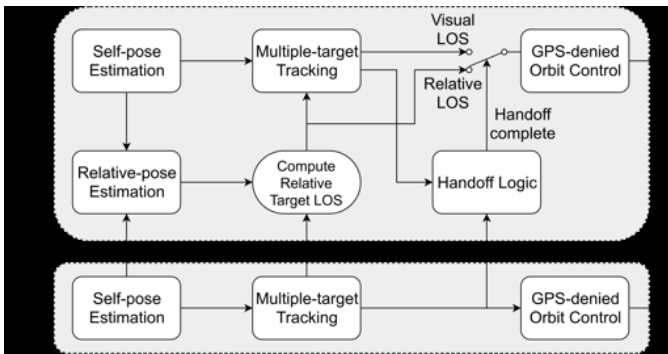


Fig. 2. Block diagram of the system components and information flow.

The primary contribution of this paper is an end-to-end solution to the target handoff problem that addresses each of these challenges using either an extension of previous work or a novel solution to the problem. Self-pose estimation is

addressed in Section II, and is performed using a complementary filter on $SO(3)$, and represents an extension of [4] that includes a better model resulting in better estimation results. The estimates from the complementary filter are inputs to a particle filter used to estimate the relative pose. A novel relative pose particle filter is derived in Section III representing a unique contribution of the paper. The challenge of inserting a UAS into an orbit without GPS is solved using a controller that produces appropriate roll commands based on an estimated line-of-sight vector to the target, and is described in Section IV, and represents a relatively minor extension of the orbit control algorithm described in [5]. The tracking and handoff UAS both utilize the Recursive-RANSAC (R-RANSAC) algorithm to visually track ground objects. While R-RANSAC was originally introduced in [6], Section V presents new results where the algorithm is used on a fixed-wing vehicle with a gimballed camera. Section VI describes a novel algorithm used to perform the handoff logic, completing the moving target handoff problem. Section VII gives simulation results and Section VIII offers some concluding remarks.

II. SELF-POSE ESTIMATION

There is extensive literature on estimating the attitude of a fixed-wing aircraft using both non-linear [4] and linear [7] methods. Our approach extends the complementary filter presented in [4] by including an improved velocity-dependent model for the angle-of-attack dynamics, and by limiting the estimator to the roll and pitch angles.

A. Complementary Pose Estimator

In this section, we will use the notation R_b^ℓ to denote the rotation from the body to the local level frame for either the tracking or handoff UAS.

Lemma 1. Let $\omega_{\alpha/\beta}^\gamma$ be the angular velocity of frame α with respect to frame β , expressed in frame γ , and let $\mathbf{e}_3 \triangleq (0, 0, 1)^\top$.

Assuming that

$$\omega_{\ell/i}^\ell = \mathbf{e}_3 \mathbf{e}_3^\top \omega_{b/i}^\ell, \quad (2)$$

the rotational kinematics of the body in the local level frame are given by

$$\dot{R}_b^\ell = R_b^\ell \left[\Pi_{R_b^\ell \mathbf{e}_3} \omega_{b/i}^\ell \right], \quad (3)$$

where $\Pi_x = I - xx^\top$ is the projection operator onto the plane orthogonal to x , and

$$\left[\begin{pmatrix} a \\ b \\ c \end{pmatrix} \right] = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix} \quad (4)$$

is the skew-symmetric matrix.

Proof. It is straightforward to show that for any two frames α and β , that $\dot{R}_\alpha^\beta = R_\alpha^\beta [\omega_{\alpha/\beta}^\alpha]$. Therefore $\dot{R}_b^\ell = R_b^\ell [\omega_{b/\ell}^b]$. However,

$$\omega_{b/\ell}^b = \omega_{b/i}^b - \omega_{\ell/i}^b \quad (5)$$

$$= \omega_{b/i}^b - R_b^\ell \omega_{\ell/i}^\ell \quad (6)$$

$$= \omega_{b/i}^b - R_b^\ell e_3 e_3^\top (R_b^\ell^\top \omega_{b/i}^b) \quad (7)$$

$$= \Pi_{R_b^\ell^\top e_3} \omega_{b/i}^b, \quad (8)$$

which completes the proof. $\square$

In the following discussion we will use ω^b to mean $\omega_{b/i}^b$. Accordingly, the mechanized velocity dynamics for the UAS are given by

$$\dot{v}^b = -\omega^b \times v^b + g R_b^{\ell\top} e_3 + a^b \quad (9)$$

where v^b is the inertial velocity of the aircraft resolved in the body frame, and a^b is the specific acceleration resolved in the body frame.

We will assume that the IMU measures the specific acceleration $a_m = a^b$ and a biased version of ω^b as $\omega_m = \omega^b + b_\omega$ where b_ω is a slowly varying bias. In unaccelerated flight, we have from Equation (9) that

$$g R_b^{\ell\top} e_3 = \omega^b \times v^b - a^b,$$

which implies that

$$\frac{\omega^b \times v^b - a^b}{\|\omega^b \times v^b - a^b\|} \times (R_b^{\ell\top} e_3) = 0. \quad (10)$$

In [4], Equation (10) is used as the innovation term in a complementary filter as

$$\dot{\hat{R}}_b^\ell = \hat{R}_b^\ell \left[\Pi_{R_b^{\ell\top} e_3} (\omega_m - \hat{b} + k_p \epsilon) \right] \quad (11)$$

$$\dot{\hat{b}} = -k_I \epsilon, \quad (12)$$

$$\epsilon = \frac{(\omega_m - \hat{b}) \times \hat{v} - a_m}{\|(\omega_m - \hat{b}) \times \hat{v} - a_m\|} \times (\hat{R}_b^{\ell\top} e_3) \quad (13)$$

where $k_p > 0$ and $k_I > 0$ are filter gains, and where $\hat{v}$ is an estimate of v^b given by

$$\hat{v} \approx v^b \approx v_a (\cos \hat{\alpha} \quad 0 \quad \sin \hat{\alpha})^\top, \quad (14)$$

where α is the angle-of-the attack and $\hat{\alpha}$ is the estimated angle-of-attack, and where we have assumed that the sideslip and flight path angles are zero.

B. Angle of attack dynamics

For a fixed-wing aircraft, when the sideslip and flight path angles are zero, the dynamic equations of motion for α are given by [8]

$$\dot{\alpha} = \frac{-F_T \alpha - F_L}{m v_a} + \frac{g}{v_a} + q, \quad (15)$$

where m is the mass, v_a the airspeed, F_L the lift force, F_T the thrust force, and q the pitch rate. Following [5], thrust and lift forces can be approximated as

$$F_T \approx k_M \delta_t^2 - k_T v_a^2$$

$$F_L \approx k_L v_a^2,$$

where k_M , k_T , and k_L are constants and $\delta_t \in [0, 1]$ is the throttle control setting. The resulting equation is

$$\dot{\alpha} = -\frac{c_0(\delta_t, v_a)}{v_a} \alpha + q + \alpha_0(v_a), \quad (16)$$

where

$$c_0(\delta_t, v_a) = \frac{k_M \delta_t^2 - k_T v_a^2}{m} \quad (17)$$

$$\alpha_0(v_a) = \frac{g}{v_a} - \frac{k_L}{m} v_a \quad (18)$$

which is similar to the angle-of-attack model used in [4] but where c_0 and α_0 are assumed to be constants. Replacing c_0 and α_0 with functions of v_a and δ_t result in an improvement in performance, especially during take-off and landing. If δ_t is not available to the observer, then it can be replaced with a nominal trim value. The estimate of the angle-of-attack is therefore given by

$$\dot{\hat{\alpha}} = -\frac{c_0(\delta_t, v_a)}{v_a} \hat{\alpha} + e_2^\top (\omega_m - \hat{b}) + \alpha_0(v_a). \quad (19)$$

C. Results

We obtained experimental results by estimating $\hat{R}_b^\ell$ and extracting roll and pitch estimates on a BAT-4 fixed-wing UAS. We commanded the UAS to loiter a moving ground target and used IMU and airspeed data to estimate the attitude of the vehicle. We used an on-board GPS-INS unit to estimate the true attitude for the purposes of comparison.

The results shown in Figures 3 include a two-minute window of data from the flight test, where the estimated roll angle is extracted from $\hat{R}_b^\ell$. The altitude of the vehicle remained nearly constant but the roll angle of the vehicle ranged from about -10 to 25 degrees, which is fairly representative of the type of trajectories required to complete the target handoff. Similar results are obtained for the pitch angle [9], and are summarized in Table I.

Referring back to Equation (1), the tracking and handoff UAS use the complementary filters given in Equations (11)–(13) and (19) to estimate $R_{r_b}^{\ell}$ and $R_{h_b}^{\ell}$ respectively.

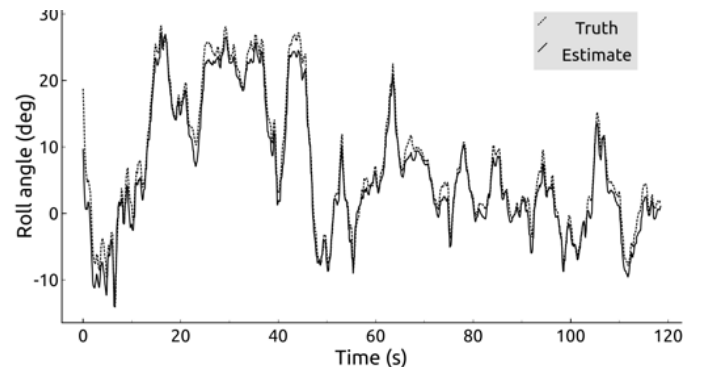


Fig. 3. Plot of roll estimate during a hardware flight test.

TABLE I
SELF-POSE ATTITUDE ESTIMATION RESULTS

	Roll	Pitch
Mean Squared Error (deg)	3.067	1.476
Average Error (deg)	1.277	-0.655

III. RELATIVE POSE ESTIMATION

The previous section showed how to estimate $R_{h_b}^{h_\ell}$ and $R_{r_\ell}^{r_b}$ for Equation (1) in GPS denied environments. This section addresses the problem of estimating the relative position $\mathbf{p}_{r/h}^{h_\ell}$ and the relative attitude $R_{r_\ell}^{h_\ell}$ in Equation (1).

Note that if the rotation matrix $R_{r_\ell}^{h_\ell}$ is parameterized in terms of Euler angles, then

$$\begin{aligned} R_{r_\ell}^{h_\ell} &= R_i^{h_\ell}(\psi_h) R_i^{r_\ell \top}(\psi_r) \\ &= \begin{pmatrix} \cos \psi_h & \sin \psi_h & 0 \\ -\sin \psi_h & \cos \psi_h & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \psi_r & -\sin \psi_r & 0 \\ \sin \psi_r & \cos \psi_r & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\psi_h - \psi_r) & \sin(\psi_h - \psi_r) & 0 \\ -\sin(\psi_h - \psi_r) & \cos(\psi_h - \psi_r) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= R_{r_\ell}^{h_\ell}(\psi_{h/r}), \end{aligned}$$

where ψ_h and ψ_r are the inertial heading of the handoff and tracking UAS respectively, and $\psi_{h/r} \triangleq \psi_h - \psi_r$ is the relative heading, which needs to be estimated.

Estimating the relative pose between vehicles is one of the fundamental challenges underlying the handoff problem. The relative pose between the two vehicles is the key link that allows the handoff vehicle to utilize information provided by the tracking UAS in order to locate and track the target. With two moving vehicles and no globally shared reference frame, the relative pose estimator is difficult to initialize and exhibits complicated dynamics. In this paper we assume the presence of a noisy measurement of the range between the two vehicles, as might be provided by a time-of-flight sensor. Time-of-flight sensors are much less complicated than a radar, for example, and might be implemented by commercial off-the-shelf hardware like a software defined radio. We will also assume that each vehicle has an on-board magnetometer to detect a noisy estimate of its inertial heading. The relative pose estimator developed in this section is one of the primary contributions of the paper.

A. Relative Pose Dynamics

Let $\mathbf{p}_{r/h}^{h_\ell}$ represent the LOS vector from the handoff vehicle to the tracking vehicle in the handoff vehicle's local level frame, given by

$$\mathbf{p}_{r/h}^{h_\ell} = R_i^{h_\ell}(\mathbf{p}_{r/i}^i - \mathbf{p}_{h/i}^i). \quad (20)$$

Differentiating with respect to time gives

$$\dot{\mathbf{p}}_{r/h}^{h_\ell} = -\left[\boldsymbol{\omega}_{h_\ell/i}^{h_\ell}\right] \mathbf{p}_{r/h}^{h_\ell} + \mathbf{v}_{r/h}^{h_\ell}. \quad (21)$$

Note that because the handoff local level frame only differs from the inertial frame by a rotation about the inertial z -axis, that

$$\boldsymbol{\omega}_{h_\ell/i}^{h_\ell} = \mathbf{e}_3 \mathbf{e}_3^\top R_{h_b}^{h_\ell} \boldsymbol{\omega}_{h_b/i}^{h_b} = \mathbf{e}_3 \dot{\psi}_{h/r}$$

which implies that

$$\dot{\psi}_{h/r} = \mathbf{e}_3^\top R_{h_b}^{h_\ell} \boldsymbol{\omega}_{h_b/i}^{h_b}, \quad (22)$$

which can be estimated from the IMU on the handoff UAS and its estimate of $R_{h_b}^{h_\ell}$ provided by the algorithm in the previous section.

B. Filter State

Because we receive measurements of the range directly, it is useful to write the x and y components of $\mathbf{p}_{r/h}^{h_\ell}$ in terms of magnitude and angle as

$$\mathbf{p}_{r/h}^{h_\ell} \triangleq (\rho \cos \vartheta, \rho \sin \vartheta, z_{h/r})^\top, \quad (23)$$

where ρ is the relative range, ϑ is the relative bearing, and $z_{h/r}$ is the relative altitude. Accordingly, we will define the state of the particle filter as

$$\hat{\mathbf{x}} = (\hat{\rho}, \hat{\vartheta}, \hat{z}_{r/h}^{h_\ell}, \hat{\psi}_{h/r})^\top. \quad (24)$$

Differentiating Equation (23) gives

$$\dot{\mathbf{p}}_{r/h}^{h_\ell} = \begin{pmatrix} \cos \vartheta & -\rho \sin \vartheta & 0 \\ \sin \vartheta & \rho \cos \vartheta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\rho} \\ \dot{\vartheta} \\ \dot{z}_{h/r} \end{pmatrix}. \quad (25)$$

Setting Equation (25) equal to Equation (21) gives, after some algebra

$$\begin{pmatrix} \dot{\rho} \\ \dot{\vartheta} \\ \dot{z}_{h/r} \end{pmatrix} = \begin{pmatrix} 0 \\ -\dot{\psi}_{h/r} \\ 0 \end{pmatrix} + \begin{pmatrix} \cos \vartheta & \sin \vartheta & 0 \\ -\frac{1}{\rho} \sin \vartheta & \frac{1}{\rho} \cos \vartheta & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{v}_{r/h}^{h_\ell}.$$

Accordingly, the dynamics of the particle filter are given by

$$\dot{\hat{\mathbf{x}}} = \begin{pmatrix} [\cos \hat{\vartheta} \sin \hat{\vartheta} 0] \hat{\mathbf{v}}_{r/h}^{h_\ell} \\ \frac{1}{\hat{\rho}} [-\sin \hat{\vartheta} \cos \hat{\vartheta} 0] \hat{\mathbf{v}}_{r/h}^{h_\ell} - \mathbf{e}_3^\top \hat{R}_{h_b}^{h_\ell} \boldsymbol{\omega}_{h_b/i}^{h_b} \\ \mathbf{e}_3^\top \hat{\mathbf{v}}_{r/h}^{h_\ell} \\ \mathbf{e}_3^\top \hat{R}_{h_b}^{h_\ell} \boldsymbol{\omega}_{h_b/i}^{h_b} \end{pmatrix}, \quad (26)$$

where we have used Equation (22). Noting that

$$\mathbf{v}_{r/h}^{h_\ell} = \mathbf{v}_{r/i}^{h_\ell} - \mathbf{v}_{h/i}^{h_\ell} = R_{r_\ell}^{h_\ell}(\psi_{h/r}) R_{r_b}^{r_\ell} \mathbf{v}_{r/i}^{r_b} - R_{h_b}^{h_\ell} \mathbf{v}_{h/i}^{h_b}, \quad (27)$$

Equation (26) can be expressed as

$$\dot{\hat{\mathbf{x}}} = f(\hat{\mathbf{x}}, \hat{R}_{h_b}^{h_\ell}, \hat{R}_{r_b}^{r_\ell}, \hat{\mathbf{v}}_{r/i}^{r_b}, \hat{\mathbf{v}}_{h/i}^{h_b}, \boldsymbol{\omega}_{h_b/i}^{h_b}), \quad (28)$$

where $\hat{R}_{h_b}^{h_\ell}$ and $\hat{R}_{r_b}^{r_\ell}$ are estimated using the self-pose estimator in Section II, $\hat{\mathbf{v}}_{r/i}^{r_b}$ and $\hat{\mathbf{v}}_{h/i}^{h_b}$ are estimated using Equation (14), and the airspeed measurement is from an on-board pitot tube, and $\boldsymbol{\omega}_{h_b/i}^{h_b}$ comes from the rate gyros of the handoff vehicle.

Given direct measurements of the relative heading, relative altitude, and range between the two vehicles, the primary challenge of the relative pose estimator is to determine the correct value for $\hat{\vartheta}$, the relative bearing between the two vehicles. The value of $\hat{\vartheta}$ is not directly observable with a single range measurement because we only receive the magnitude of

the relative position vector. However, with two measurements the change in range can be used to narrow the possible values of $\hat{\vartheta}$ to two options. As seen in Equation (26),

$$\dot{\hat{\rho}} = [\cos \hat{\vartheta} \quad \sin \hat{\vartheta} \quad 0] \hat{\mathbf{v}}_{r/h}^{h_\ell} = \|\hat{\mathbf{v}}_{r/h}^{h_\ell}\| \cos \hat{\beta}, \quad (29)$$

where $\hat{\beta}$ is the estimated angle between the relative velocity vector and the relative position vector.

Solving for $\hat{\beta}$ gives $\hat{\beta} = \pm \cos^{-1} \left(\frac{\dot{\hat{\rho}}}{\|\hat{\mathbf{v}}_{r/h}^{h_\ell}\|} \right)$, which implies that there are two possible values for $\hat{\vartheta}$ based on each measurement. Due to this ambiguity, unimodal filters like the EKF or UKF are not well suited to this problem. Alternatively, particle filters can initialize and propagate a bi-modal distribution, and are therefore a more suitable alternative.

C. Particle filter implementation

The particle filter proposed in this paper consists of N particles, each with a state and dynamics given by Equations (24) and (26). We assume measurements of relative range using a time-of-flight sensor, the relative altitude using barometric sensors on-board each aircraft, and relative heading using magnetometers on-board each aircraft, and that the altitude and heading of the tracking vehicle can be transmitted to the handoff vehicle with zero delay. The measurement model is therefore given by

$$\mathbf{y} = h(\mathbf{x}) \triangleq \begin{pmatrix} \sqrt{\rho^2 + (z_{r/h}^{h_\ell})^2} \\ z_{r/h}^{h_\ell} \\ \psi_{h/r} \end{pmatrix}.$$

Let $\hat{\mathbf{x}}_k^{(i)}$ denote the i^{th} particle at time k , let $\tilde{\mathbf{y}}_k^{(i)}$ denote the residual between the actual and predicted measurements of the i^{th} particle given by

$$\tilde{\mathbf{y}}_k^{(i)} = \mathbf{y}_k - h(\hat{\mathbf{x}}_k^{(i)}),$$

and let Q be the covariance of the measurement noise. Then the weight for the i^{th} particle at time k is given by

$$w_k^{(i)} = \frac{\exp \left(\tilde{\mathbf{y}}_k^{(i)\top} Q^{-1} \tilde{\mathbf{y}}_k^{(i)} - \max_l \left\{ \tilde{\mathbf{y}}_k^{(l)\top} Q^{-1} \tilde{\mathbf{y}}_k^{(l)} \right\} \right)}{\sum_{j=1}^N \exp \left(\tilde{\mathbf{y}}_k^{(j)\top} Q^{-1} \tilde{\mathbf{y}}_k^{(j)} - \max_l \left\{ \tilde{\mathbf{y}}_k^{(l)\top} Q^{-1} \tilde{\mathbf{y}}_k^{(l)} \right\} \right)}, \quad (30)$$

where the subtraction term enhances the numerical conditioning of the algorithm.

The particles are then resampled with probability proportional to their weights. However, to reduce particle deprivation and minimize the chance of throwing away good particles, we employ low-variance resampling [10] and selective resampling based on the number of effective particles [11], where the number of effective particles is estimated by

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w^{(i)})^2}. \quad (31)$$

In order to perform selective resampling, it is important to update the weights of the particles at each successive time step between resamples. This is accomplished using

$$w_{k+1}^{(i)} = \eta w_k^{(i)} \tilde{\mathbf{y}}_{k+1}^{(i)\top} Q^{-1} \tilde{\mathbf{y}}_{k+1}^{(i)}, \quad (32)$$

where η is a normalization constant. After the particles are resampled, all weights are reset to $\frac{1}{N}$.

D. Results

The relative pose particle filter was tested in simulation using simulated IMU, airspeed, altitude, magnetometer, and range measurements, each with added Gaussian noise. In general, the relative pose estimate converged within 30 seconds and provided a sufficiently accurate estimate for the handoff UAS to locate the target and insert into a similar orbit. As expected, the particle filter displayed a bimodal distribution before converging onto the proper bearing angle as shown in Figure 4.

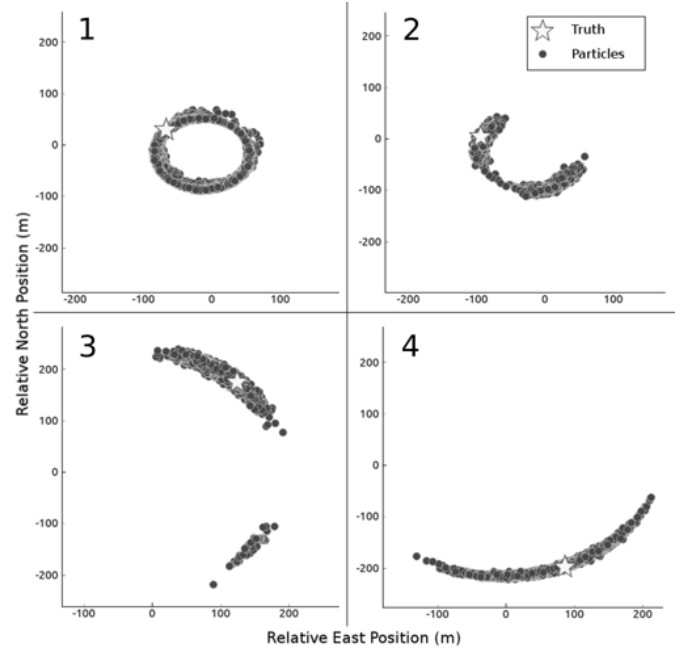


Fig. 4. *Top left*: particles are initialized uniformly across all bearing angles; *Top right*: particles begin to split into a bimodal distribution; *Bottom left*: particles are grouped in two distinct clusters, but with more particle density around the correct bearing; *Bottom right*: the particles consolidate and center around the true value

Figure 5 shows the estimate of the bearing diverging slightly, then converging sharply as the particles consolidate around the correct value. We observed that because the estimated line of sight is parameterized using range and bearing, and because the range between the two vehicles is often large, small errors in the bearing angle can lead to large errors in the xy -plane. Accordingly, the relative pose estimate is not ideal for obtaining a fine-tuned estimate of the target's position. However, it does serve the intended purpose of providing a reasonable estimate of the relative pose, facilitating orbit insertion as described in Section IV. As seen in Figure 6, after the relative pose estimate converges the relative position

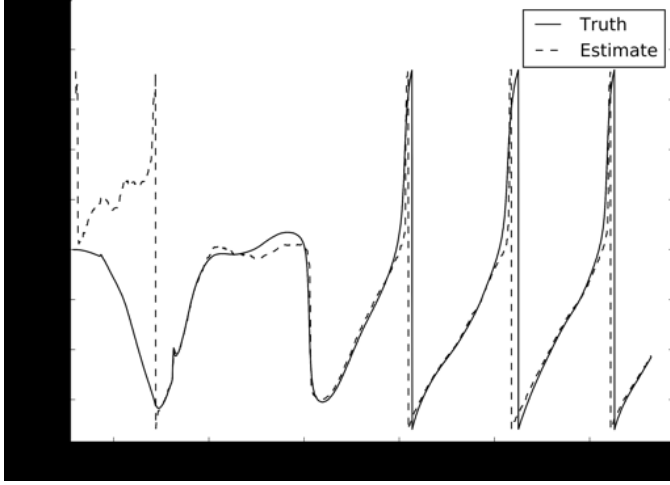


Fig. 5. Plot of $\hat{p}_{t/h}$ estimate; diverges slightly as particles split into bimodal distribution, then converges sharply as particles consolidate around the true value.

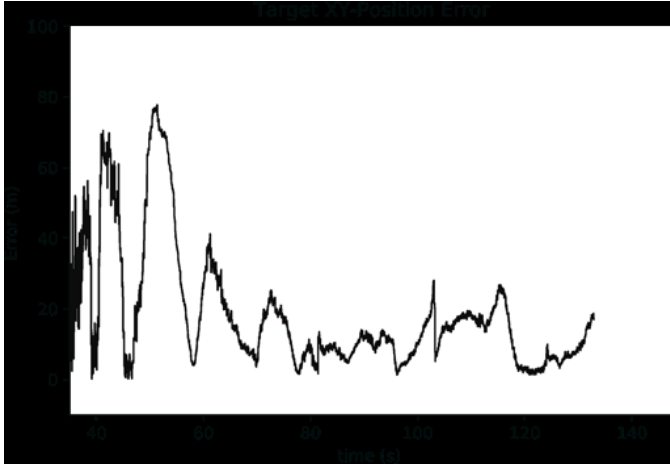


Fig. 6. Plot of estimated target xy -position error.

error remains near or below 20 meters. This level of accuracy is sufficient to locate the target and retain it in the field of view, allowing the handoff UAS to utilize visual information to perform the handoff task.

IV. GPS DENIED ORBIT INSERTION

Once the handoff vehicle has obtained a reasonable estimate of the relative transformation between itself and the tracking UAS, it is ready to insert itself into a similar orbit about the target. Because there is no GPS data available, the orbit control must only use estimates of the target's relative position and velocity. Initially, the handoff UAS will use the relative pose estimate to compute the LOS between itself and the target according to Equation (1), but by the time the target handoff is complete, it will need to orbit the target based entirely on visual data.

Another challenge of GPS-denied orbiting is that the vehicle cannot use global waypoints or orbit centers to loiter about the target, but must instead give heading rate or roll commands. The implementation here assumes the ability to command

roll directly, but the derivation for commanding heading rate would be similar. This section describes the technique used to compute appropriate roll commands to orbit the target using only the target's relative position and velocity. We also discuss in Section IV-B how the target is orbited using only visual information after the handoff is complete.

A. Orbit Insertion

1) *Target relative state:* As previously noted, without GPS, the state of the target must be represented relative to the UAS. We represent the position of the target relative to the handoff UAS as $\mathbf{p}_{t/h}^{h_\ell}$. This is initially given by Equation (1) as

$$\mathbf{p}_{t/h}^{h_\ell} = \mathbf{p}_{r/h}^{h_\ell} + R_{r_\ell}^{h_\ell} R_{r_b}^{r_\ell} \mathbf{p}_{t/r}^{r_b}, \quad (33)$$

where $\mathbf{p}_{r/h}^{h_\ell}$ and $R_{r_\ell}^{h_\ell}$ are estimates from the particle filter, $R_{r_b}^{r_\ell}$ comes from the tracker's complimentary attitude filter, and $\mathbf{p}_{t/r}^{r_b}$ is measured by the tracker.

We assume that the handoff UAS is moving much faster than the target and accordingly use the velocity of the handoff vehicle to approximate the target's relative velocity according to

$$\mathbf{v}_{t/h}^{h_\ell} = \mathbf{v}_{t/i}^{h_\ell} - \mathbf{v}_{h/i}^{h_\ell} \approx -\mathbf{v}_{h/i}^{h_\ell}. \quad (34)$$

Given the relative position and velocity of the target from Equations (33) and (34), estimates of the range to the target, $\hat{\rho}_{t/h}$, and the angle between the relative position and relative velocity vector, $\hat{\chi}_{t/h}$, are given by

$$\begin{aligned} \hat{\rho}_{t/h} &= \|(I - e_3 e_3^\top) \hat{\mathbf{p}}_{t/h}^{h_\ell}\| \\ \hat{\chi}_{t/h} &= -\text{atan2}(\|\hat{\mathbf{v}}_{t/h} \times \hat{\mathbf{p}}_{t/h}^{h_\ell}\|, \hat{\mathbf{v}}_{t/h} \cdot \hat{\mathbf{p}}_{t/h}^{h_\ell}). \end{aligned}$$

2) *Control implementation:* Define ρ^d to be the desired orbit radius and λ to be the desired direction of the orbit with $\lambda = +1$ for a clockwise orbit and $\lambda = -1$ for a counter-clockwise orbit. Following [5] the desired angle between the relative position and velocity vectors is given by

$$\chi^d = \lambda \left[\tan^{-1}(-k_o(\hat{\rho}_{t/h} - \rho^d)) + \frac{\pi}{2} \right], \quad (35)$$

where k_o is a positive gain.

If the handoff UAS is following a radius of ρ^d at a speed of v then simple kinematics implies that $\dot{\chi} = \lambda v / \rho^d$. Similarly, for a fixed wing UAS, the coordinated turn condition is given by [5]

$$\dot{\chi} = \frac{g}{v} \tan \phi,$$

where ϕ is the roll angle. Equating these two expressions and letting $v = \|\hat{\mathbf{v}}_{t/h}\|$ gives the feedforward roll angle in an orbit of radius ρ^d as

$$\phi_{ff} = \lambda \tan^{-1} \left(\frac{\|\hat{\mathbf{v}}_{t/h}\|}{g \rho^d} \right).$$

Therefore, the GPS denied orbit insertion control strategy is given by

$$\begin{aligned} e_\chi &= \hat{\chi}_{t/h} - \chi^d \\ \phi^c &= \phi_{ff} + k_p e_\chi + k_d \dot{e}_\chi, \end{aligned}$$

where ϕ^c is the commanded roll angle of the handoff UAS, where k_p and k_d are positive gains, and where $\dot{e}_\chi$ is computed numerically. Convergence analysis for the kinematic equations of motion are similar to the arguments provided in [5].

B. Vision-based Orbiting

As the handoff UAS begins tracking the proper target, it can improve its orbit control by transitioning from using the relative position estimate based on Equation (33) to using direct visual tracking. As long as the handoff UAS is tracking the target, the visual track provides a reliable measurement of the relative position of the target. In order to estimate the target's position, the UAS must transform the visual line of sight into the proper frame and recover the scale of the LOS vector. The handoff UAS's target tracking algorithm will produce the target's position in normalized image plane pixel coordinates of the handoff UAS's camera.

Let

$$\epsilon_{t/h}^c = \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ 1 \end{pmatrix} \quad (36)$$

represent the normalized image coordinates of the target received from the tracking algorithm. The normalized image coordinates can be expressed in the body-level frame of the handoff vehicle as

$$\epsilon_{t/h}^{h_\ell} = R_{h_b}^{h_\ell} R_g^{h_b} R_c^g \epsilon_{t/h}^c, \quad (37)$$

where

$$R_c^g = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad (38)$$

is the fixed rotation from the camera to the gimbal frame,

$$R_g^{h_b} = \begin{pmatrix} \cos \alpha_{el} \cos \alpha_{az} & -\sin \alpha_{az} & \sin \alpha_{el} \cos \alpha_{az} \\ \cos \alpha_{el} \sin \alpha_{az} & \cos \alpha_{az} & \sin \alpha_{el} \sin \alpha_{az} \\ -\sin \alpha_{el} & 0 & \cos \alpha_{el} \end{pmatrix}, \quad (39)$$

is the rotation from the gimbal to the body frame, where α_{az} and α_{el} are the gimbal azimuth and elevation angles respectively, and $R_{h_b}^{h_\ell}$ is determined by the method described in Section II.

Using a flat-earth approximation, we can recover the appropriate scale of the LOS vector using the measured altitude above ground level h of the vehicle as

$$\frac{\epsilon_{t/h}^{h_\ell}}{\mathbf{e}_3^\top \epsilon_{t/h}^{h_\ell}} = \frac{\mathbf{p}_{t/h}^{h_\ell}}{h},$$

which implies that

$$\mathbf{p}_{t/h}^{h_\ell} = h \frac{\epsilon_{t/h}^{h_\ell}}{\mathbf{e}_3^\top \epsilon_{t/h}^{h_\ell}}. \quad (40)$$

This visually-derived estimate of the target's position can then be used to improve or replace the estimate given by Equation (33).

C. Results

We implemented this control scheme in the full handoff simulation with noise on all sensors and using the estimated target LOS as the input to the control. The orbit control converges to the desired orbit and allows the handoff vehicle to keep the target in the field of view to begin tracking and perform the handoff. Initially the handoff UAS uses the outputs of the relative- and self-pose filters to estimate the target's position and, despite the noise in both estimators, the orbit control accomplishes the desired task of following a moving target without GPS. Figures 7 shows a top-down view of the target and handoff vehicle positions over time.

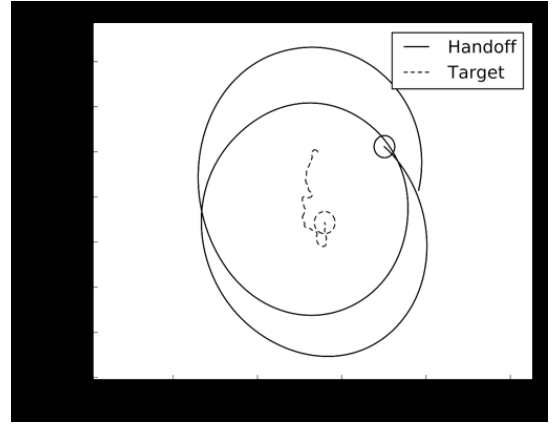


Fig. 7. Plot of the handoff UAS orbiting the target.

V. MULTIPLE TARGET TRACKING

In our implementation, the tracking UAS and the handoff UAS both visually track the ground targets using the R-RANSAC-based visual multiple target tracking (VMTT) algorithm, originally presented in [12], and extended in [13], [14], [15], [16] to tracking from multi-rotor aircraft. In this paper we extend the algorithm to fixed-wing aircraft. Fixed-wing aircraft operate under different conditions and constraints than multirotor aircraft. They must maintain forward velocity with nonholonomic coordinated turn constraints, and they often fly at higher altitudes and faster speeds. These differences require some unique integration and adaptation of the original algorithm into our particular system. This section will provide a brief overview of the visual front-end and the R-RANSAC algorithm, along with an explanation of both our implementation and results of using R-RANSAC on a fixed-wing vehicle.

A. Visual Front-end

The visual front-end pipeline consists of three main steps. First, the video frame at the previous time step is processed to find good features to track. Our implementation uses the *OpenCV* function *goodFeaturesToTrack* (<https://opencv.org/>). The features that are found using this method are then propagated to the current video frame using optical flow. Our implementation uses the pyramidal Lucas-Kanade method in the *OpenCV* function *calcOpticalFlowPyrLK*. The second step in the visual front end uses the features in the previous and

current image frames to find the homography transformation between frames. Our implementation uses the *OpenCV* function *findHomography*. The homography transformation is then used to warp the previous image to the current image. Our implementation uses the *OpenCV* function *warpPerspective*. Assuming that most of the features in the image are not moving, the homography transformation will correspond to the motion in the scene that is due to the motion of the UAS. Therefore, the third step in the visual front end is to find all features in the previous image that do not warp correctly to the current image through the homography transformation. Assuming a relatively flat scene, these features will correspond to moving objects in the environment. These features are then passed as measurements to the R-RANSAC tracking algorithm described below.

B. R-RANSAC Tracker

The Recursive RANSAC tracking algorithm is based upon the original Random Sample Consensus Algorithm (RANSAC) algorithm, which was first introduced in [17] as an efficient method to reject outliers and estimate model parameters. R-RANSAC uses RANSAC to find and initialize good model tracks that are “recursively” propagated through time using a linear Kalman filter. The R-RANSAC algorithm maintains a bank of the best M tracks, and scores each track based on consistency with the visual measurements to prune low probability tracks and add new potential tracks. Tracks with persistently high scores are marked as moving objects in the environment.

C. Adaptations for Fixed-wing vehicles

1) *Two-axis gimbal*: One of the main differences between tracking from a fixed-wing vehicle as opposed to a multirotor aircraft, is that the vehicle must remain in constant motion. Accordingly, a gimbal is necessary to maximize the aircraft’s ability to keep the target in the field of view. In this project we use a two-axis gimbal to control both the azimuth and elevation angles of the camera. The appropriate azimuth and elevation angles are $\alpha_{az} = \text{atan2}(\ell_y, \ell_x)$, and $\alpha_{el} = -\sin^{-1}(\ell_z)$, where $\ell = (\ell_x, \ell_y, \ell_z)^\top$ is the normalized line of sight vector from the UAS to the target in the vehicle body frame.

2) *Parameter Tuning*: Some of the other challenges of implementing the R-RANSAC tracker on a fixed-wing vehicle include higher operating altitudes, faster velocities, and continuously varying viewpoints of the target. These conditions can make it more difficult for the vehicle to pick up on good, consistent features of the target and can also lead to high variation in the apparent motion of the target. The apparent motion of the target in the camera frame is minimized when the line of sight to the target and the target’s velocity are closely aligned. In a constant orbit about a ground target with nearly constant velocity, this alignment happens twice per revolution. R-RANSAC assumes that targets of interest are moving and therefore uses motion in the image plane to track targets, which makes it more difficult for the algorithm to continue tracking the target when the apparent motion is low.

To overcome these challenges, we tuned the algorithm parameters to help the tracker be better suited to having fewer good features and periodically low apparent motion. See [9] for a summary of parameters used and [14] for a detailed description and analysis of each parameter.

D. Results

Using the R-RANSAC tracker and a two-axis gimbal setup, we were able to achieve reasonable tracking results for moving ground targets. We tested the tracking algorithm on a fixed-wing aircraft flying approximately 300 meters above a moving vehicle on the ground. With proper tuning, the UAS was able to track the ground target despite significant jitter in the image. Figure 8 shows a snapshot of tracking a moving ground target.



Fig. 8. Snapshot of tracking a moving ground target from a fixed-wing aircraft

VI. HANDOFF LOGIC

After the handoff vehicle points its gimbal in the direction of the target and visually tracks all moving objects in its field of view, it must then determine which object correlates with the target being tracked by the tracking UAS. In order to estimate the correlation between two tracks, the handoff UAS must first align the tracks. For tracks that have a low residual after alignment, the UAS can compare the resulting transformation with the estimated relative pose to see if the two estimates coincide. If a track both aligns well with the information from the tracking UAS and also coincides with the relative pose estimate, it is considered a good match. This section introduces a novel method used for track alignment and also the logic used to determine if the track is a sufficiently good match to complete the handoff.

A. Track Alignment

The result of the previous two sections is an estimate of potential target tracks from the vision systems of both the tracking and handoff UAS. At time step n , the visual tracking system on the tracking and handoff UAS produces estimates of $\mathbf{p}_{t[n]/r[n]}^{r_\ell[n]}$ $\mathbf{p}_{t[n]/h[n]}^{h_\ell[n]}$, respectively, where we have explicitly noted the target $t[n]$ at time n and the frames $r_\ell[n]$ and $h_\ell[n]$ at

time n . In order to compare the two tracks, we must transform the past N estimates of the target into the current time k .

The position of the target relative to the tracking vehicle at time n is defined as

$$\mathbf{p}_{t[n]/r[n]}^{r_\ell[n]} = R_i^{r_\ell[n]} \left(\mathbf{p}_{t[n]/i}^i - \mathbf{p}_{r[n]/i}^i \right). \quad (41)$$

Transforming to the current time gives

$$\begin{aligned} \mathbf{p}_{t[n]/r[k]}^{r_\ell[k]} &= R_{r_\ell[n]}^{r_\ell[k]} R_i^{r_\ell[n]} \left(\mathbf{p}_{t[n]/i}^i - \mathbf{p}_{r[k]/r[n]}^i - \mathbf{p}_{r[n]/i}^i \right) \\ &= R_{r_\ell[n]}^{r_\ell[k]} \left(\mathbf{p}_{t[n]/r[n]}^{r_\ell[n]} - \mathbf{p}_{r[k]/r[n]}^{r_\ell[n]} \right). \end{aligned}$$

This equation is used to propagate the LOS from time n into time k , where $R_{r_\ell[n]}^{r_\ell[k]}$ and $\mathbf{p}_{r[k]/r[n]}^{r_\ell[n]}$ are determined by integrating the vehicle's estimated angular and linear velocities, respectively. This same propagation process applies to the handoff vehicle's tracks as well.

The tracking and handoff estimates of the target at time n are related by the expression

$$\mathbf{p}_{t[n]/h[n]}^{h_\ell[k]} = \bar{R}_{r_\ell[k]}^{h_\ell[k]} \mathbf{p}_{t[n]/r[n]}^{r_\ell[k]} + \boldsymbol{\xi}_{r_\ell[k]/h_\ell[k]}^{h_\ell[k]}.$$

Summing over the past N estimates and solving for the translation vector between frames gives

$$\boldsymbol{\xi}_{r_\ell[k]/h_\ell[k]}^{h_\ell[k]} = \bar{\mathbf{p}}_h - \bar{R}_{r_\ell[k]}^{h_\ell[k]} \bar{\mathbf{p}}_r,$$

where

$$\bar{\mathbf{p}}_* = \frac{1}{N} \sum_{n=k-N+1}^k \mathbf{p}_{t[n]/*[k]}^{*[k]}, * = h, r.$$

Defining the matrices

$$P_* = \left(\mathbf{p}_{t[k]/*[k]}^{*[k]} - \bar{\mathbf{p}}_*, \dots, \mathbf{p}_{t[k-N+1]/*[k]}^{*[k]} - \bar{\mathbf{p}}_* \right), * = h, r,$$

gives $P_h = \bar{R}_{r_\ell[k]}^{h_\ell[k]} P_r$. Since $\bar{R}_{r_\ell[k]}^{h_\ell[k]}$ corresponds to a rotation about the inertial z -axis, it has the form

$$\bar{R}_{r_\ell[k]}^{h_\ell[k]} = \begin{pmatrix} R_\psi & 0_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{pmatrix},$$

where $R_\psi \in SO(2)$. We can estimate R_ψ by solving the least squares problem

$$R_\psi = \arg \min_{R \in SO(2)} \|P_{h,1:2} - R P_{r,1:2}\|_F^2, \quad (42)$$

where $\|\cdot\|_F$ represents the matrix Frobenius norm, and where $P_{*,1:2}$ is the first two rows of P_* .

The solution to Equation (42) can be found in closed form using singular value decomposition (SVD), according to the result from the orthogonal Procrustes problem [18]. In our case, we constrain the problem to only include rotation matrices ($\det(R) = 1$) about the z -axis, which is a modification of the Kabsch algorithm [19].

First, we compute the cross covariance between $P_{h,1:2}$ and $P_{r,1:2}$, given by $M = P_{h,1:2} P_{r,1:2}^\top$. Using SVD, M can be decomposed as $M = U \Sigma V^\top$. The optimal rotation is given by

$$R_\psi = U \Sigma' V^\top, \quad (43)$$

where $\Sigma' = \text{diag}(1, d)$ and d is the sign (± 1) of $\det(UV^\top)$, ensuring that R_ψ is a valid rotation matrix.

B. Track Comparison

After computing the estimate for R_ψ that aligns the tracks, the residual error from Equation (42) can be used as a measure of how well the two tracks are aligned. We choose a threshold T_r for the residual and say that any track which produces a residual less than the threshold is a potential match with the target.

Equation (43) gives the rotation that minimizes the Frobenius norm of the error between the two sets of points, but it is possible that two similarly shaped tracks could produce a low residual, but not be true matches. To increase our confidence that an object corresponds with the target, we also compare the result of the Procrustes analysis with the relative pose estimate between the two vehicles.

For any object that has a residual below the threshold, we also compare the computed rotation, $\bar{R}_{r_\ell}^{h_\ell}$, with the rotation estimated by the relative pose filter denoted as $\hat{R}_{r_\ell}^{h_\ell}$. The error between the two rotations is given by

$$e_\theta = \cos^{-1} \left(\mathbf{e}_1^\top \bar{R}_{r_\ell}^{h_\ell} \hat{R}_{r_\ell}^{h_\ell \top} \mathbf{e}_1 \right), \quad (44)$$

where $\mathbf{e}_1 = (1 \ 0 \ 0)^\top$.

Using the two tracks, we can also estimate the relative translation between the two UAS that can be compared with the relative LOS estimate to further verify that the tracks match.

If both the angle and translation errors between the Procrustes result and the relative pose estimate are below their respective thresholds, then we declare the two tracks to be a match.

C. Handoff Transition

To complete the handoff process, the handoff UAS will switch from using the relative pose estimate for determining the target's position to using the visual LOS. To avoid a discontinuous jump in the LOS input to the orbit control, we introduce a blending parameter, γ_b , used to transition from one source of the LOS to the other according to

$$\ell_b = \gamma_b \ell_v + (1 - \gamma_b) \ell_r, \quad (45)$$

where ℓ_b is the blended LOS and ℓ_v and ℓ_r are the visual and relative-estimate-based LOS vectors respectively. The blending parameter is initialized to zero ($\ell_b = \ell_r$) and evolves according to

$$\gamma_b^+ = \zeta m + (1 - \zeta) \gamma_b, \quad (46)$$

where ζ is a tunable parameter that determines the blending transition rate and m is a binary value representing whether or not the tracks match, given by

$$m = \begin{cases} 1, & \|P_h - \bar{R}_{r_\ell}^{h_\ell} P_r\|_F < T_r, \ e_\theta < T_\theta, \text{ and } e_t < T_t \\ 0, & \text{otherwise} \end{cases}. \quad (47)$$

The blending parameter γ_b can also help to ensure that the handoff only occurs after the errors remain below the desired thresholds for multiple consecutive time steps. We consider the handoff to be officially complete when γ_b rises above the threshold T_γ which we set to 0.95.

VII. SIMULATION STUDY

A. Simulation Setup

To test the full system, we simulated the handoff scenario using Gazebo 7 and ROS. While we have hardware results for several components individually, an end-to-end hardware demonstration was not feasible given the scope of the project. However, the simulation environment, which includes hardware emulation in the loop, demonstrates a fully working system with simulated noise in software, suggesting that the methods described here could provide a strong basis for a successful hardware implementation.

The aircraft dynamics are simulated according to the framework presented in [5], with parameters for a small fixed-wing UAS. The ground targets are represented by pedestrian objects that wander randomly within a 140m by 270m area. In order to provide visual features, a satellite image of a rural area is used as the ground plane for the simulated world.

Each UAS is equipped with IMU and magnetometer sensor plugins and the range sensor used in Section III is simulated by computing the norm of the distance between the two UAS, with zero-mean Gaussian noise. Each vehicle also has a simulated camera with a two-axis gimbal, where the simulated camera returns a color pixel array at 30 Hz.

For simplicity, both UAS are deployed simultaneously, but the tracking UAS uses absolute position to navigate directly to the target, while the handoff UAS begins estimating the relative pose between aircraft to determine the target's location. The handoff UAS is able to successfully determine the relative pose between the aircraft and insert into a similar orbit, but opposite in direction to help maximize the observability of the relative pose between the aircraft. As the targets enter the handoff vehicle's field of view, the handoff UAS begins tracking each target and, over time, gains sufficient confidence to complete the handoff. The handoff vehicle continues to track and orbit the target using only visual information.

B. Results

We ran the simulation 500 times with different initial conditions and noise sequences and measured the time it took for the handoff vehicle to complete the handoff and whether or not it selected the correct target among 5 different randomly moving targets. The standard deviations for the simulated range and magnetometer measurements was 3 meters and 0.01 radians, respectively. A failure is when the handoff UAS could not determine the correct target within 15 minutes. The UAS was able to accurately determine the correct target in 97.2% of the simulation runs, with an average handoff time of 5 minutes and 14 seconds.

We also conducted Monte Carlo simulations to test the limitations of our approach and to evaluate some of the trade-offs of certain parameters. The main trade-off we identified was the time it took the handoff vehicle to make a decision versus the accuracy of that decision. The handoff logic thresholds and the track comparison window size seemed to be the primary determining factors for this trade-off. As the requirements for the handoff logic become more difficult to meet, namely lower thresholds and larger comparison window

sizes, the accuracy of the handoff increases, but it also takes increasingly long to make a decision. To characterize this trade-off between the speed of the decision and the accuracy, we varied both the window size and the residual threshold. We ran 100 iterations of each parameter configuration and averaged the results. Figure 9 shows the trade-off between time and accuracy observed for various window size and threshold parameters.

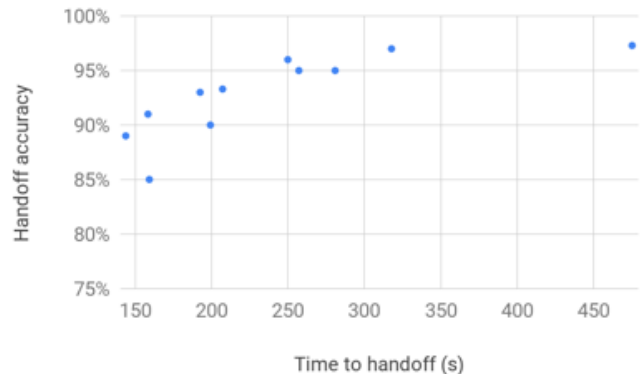


Fig. 9. Relationship between the time to handoff and handoff accuracy for various window size and threshold values.

If the window size was too small or the threshold too high, then the accuracy suffered. The handoff vehicle would make a decision sooner, but it was less likely to make the correct decision. As the window size increased and the threshold lowered, the conditions for handoff were more stringent and accordingly, accuracy increased. However, as the accuracy exceeded 97%, the time it took the handoff vehicle to make a decision increased significantly. Using a window size of 100 samples and residual threshold of 15 meters seemed to provide a reasonable balance, giving 97.2% accuracy and a handoff time of 314 seconds, as described above.

VIII. CONCLUSION

This paper has described a complete architecture for visual handoff between two fixed-wing UAS of a moving ground target in a GPS-denied environment. The complete solution requires that each UAS estimates its own roll and pitch angles, as well as the relative position and heading between the two vehicles. Self pose for each vehicle was estimated using a complementary filter, and the relative pose was estimated using a novel particle filter. After estimating the relative pose, the handoff vehicle is inserted in an orbit that is similar, but in the opposite direction, to the tracking vehicle. The paper then described novel handoff logic that enables the target handoff.

The solution presented in this paper is shown to facilitate target handoff with high reliability, even with significant sensor noise. While there is a trade-off between the time it takes the system to make a decision and the accuracy of that decision, we found that the handoff vehicle was able to locate the correct target with 97% accuracy within a reasonable time frame.

REFERENCES

- [1] W. D. Watson, D. Couto, and J. Sussman-Fort, "Persistent, robust, effective surveillance using small unmanned aerial systems (press)," in *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, Oct 2018, pp. 1–5.
- [2] M. Valenti, D. Dale, J. How, D. P. de Farias, and J. Vian, "Mission health management for 24/7 persistent surveillance operations," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2012. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2007-6508>
- [3] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21513>
- [4] R. Mahony, M. Euston, J. Kim, P. Coote, and T. Hamel, "A non-linear observer for attitude estimation of a fixed-wing unmanned aerial vehicle without gps measurements," *Transactions of the Institute of Measurement and Control*, vol. 33, no. 6, pp. 699–717, 2011. [Online]. Available: <http://dx.doi.org/10.1177/0142331209343660>
- [5] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.
- [6] P. C. Niefeldt and R. W. Beard, "Multiple target tracking using recursive RANSAC," in *Proceedings of the American Control Conference*, Portland, OR, June 2014, pp. 3393–3398.
- [7] J. K. Hall, N. B. Knoebel, and T. W. McLain, "Quaternion attitude estimation for miniature air vehicles using a multiplicative extended Kalman filter," in *2008 IEEE/ION Position, Location and Navigation Symposium*, May 2008, pp. 1230–1237.
- [8] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2003.
- [9] S. Tolman, "Multiple agent target tracking in gps-denied environments," Master's thesis, Brigham Young University, Provo, Utah, 2019.
- [10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.
- [11] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear Constraint Network Optimization for Efficient Map Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, sep 2009.
- [12] P. C. Niefeldt and R. W. Beard, "Recursive RANSAC: Multiple signal estimation with outliers," in *9th IFAC Symposium on Nonlinear Control Systems*, Toulouse, France, September 2013, pp. 430–435.
- [13] K. Ingersoll, P. C. Niefeldt, and R. W. Beard, "Multiple target tracking and stationary object detection using recursive-RANSAC and tracker-sensor feedback," in *Proceedings of the International Conference on Unmanned Air Vehicles*, Denver, CO, May 2015.
- [14] P. C. Niefeldt, K. Ingersoll, and R. W. Beard, "Comparison and analysis of recursive-RANSAC for multiple target tracking," *IEEE Transaction on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 461–476, feb 2017.
- [15] P. C. Lusk and R. W. Beard, "Visual multiple target tracking from a descending aerial platform," in *Proceedings of the American Control Conference*, Milwaukee, June 2018, pp. 5088–5093.
- [16] J. H. White, K. T. Salva, and R. W. Beard, "Extending motion detection to track stopped objects in visual multi-target tracking," in *Proceedings of the American Control Conference*, Milwaukee, June 2018, pp. 5825–5830.
- [17] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [18] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar 1966. [Online]. Available: <https://doi.org/10.1007/BF02289451>
- [19] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 34, no. 5, pp. 827–828, Sep 1978. [Online]. Available: <https://doi.org/10.1107/S0567739478001680>



Skyler Tolman received the B.S. degree in Computer Engineering from Brigham Young University in 2017, and the M.S. degree in Electrical and Computer Engineering from Brigham Young University in 2019. His interests are in autonomous systems and robotic software architecture and design. He currently works as a robotic software engineer at the Bastian Solutions R&D group in Boise, ID.



Randal W. Beard received the B.S. degree in electrical engineering from the University of Utah, Salt Lake City in 1991, the M.S. degree in electrical engineering in 1993, the M.S. degree in mathematics in 1994, and the Ph.D. degree in electrical engineering in 1995, all from Rensselaer Polytechnic Institute, Troy, NY. Since 1996, he has been with the Electrical and Computer Engineering Department at Brigham Young University, Provo, UT, where he is currently a professor. His primary research focus is autonomous control of unmanned air vehicles, multiple target tracking, and multivehicle coordination and control. He is a past associate editor for the IEEE Control Systems Magazine, the Journal of Intelligent and Robotic Systems, and the IEEE Transactions on Automatic Control. He is a fellow of the IEEE and AIAA.



Timothy Devon Morris received the B.S. degree in applied mathematics from Brigham Young University in 2017, M.S. degree in electrical engineering from Brigham Young University in 2019 and is currently pursuing an M.S. in computer science from Georgia Institute of Technology. His interests are in autonomous systems, applied differential geometry and sensor fusion. He currently works as a perception engineer and scrum master at Torc Robotics in Blacksburg, Virginia.



distributed estimation, and target tracking.

Cameron K. Peterson received her B.S. and M.S. degrees in applied physics from Brigham Young University and Johns Hopkins University respectively. In 2012 she received a Ph.D. degree in aerospace engineering from the University of Maryland, College Park. She worked for the Johns Hopkins University Applied Physics Laboratory until 2016 where she has been with the Electrical and Computer Engineering Department at Brigham Young University, Provo, UT. Her research interests include cooperative control of autonomous vehicles,

PLACE
PHOTO
HERE

Riten Gupta Riten Gupta received the BS degree in 1996 from UC Berkeley and the MS and Ph.D. degrees in 1999 and 2001 from the University of Michigan, all in Electrical Engineering. He is a Research and Development Scientist at Utopia-Compression Corporation, Los Angeles, CA, where he leads government programs in the areas of communications and networking, computer vision, and unmanned system state estimation, and control.