

Autonomous Ship-board Landing using Monocular Vision

William K. Holmes *
wkh110@psu.edu

Jack W. Langelaan †
jlangelaan@psu.edu

Department of Aerospace Engineering, Penn State University
University Park, PA 16802

ABSTRACT

This paper describes a system for landing a vertical take-off and landing vehicle on a moving deck using a monocular vision system and inertial measurement unit. Data from the vision system and IMU are used to compute an estimate of relative deck position, and a touchdown trajectory generator based on tau-guidance is used to compute a path to landing. A trajectory-following controller follows the landing path. Computing relative deck state means that GPS is not required. To tests the system, landings to both a stationary deck and a moving deck were done in an indoor flight facility using a hexacopter as flight platform. Truth data was collected using a motion capture system. In all cases tested the vehicle landed successfully.

NOTATION

\mathbf{b}_i	position of i^{th} deck feature in deck frame
g	magnitude of gravitational acceleration
\mathbf{g}	gravity vector
k	tau-guidance parameter
\mathbf{Q}_{rot}	noise covariance for deck rotation
\mathbf{Q}_{trans}	noise covariance for deck translation
\mathbf{r}_d	relative deck position
\mathbf{s}_c	camera position in helicopter frame
\mathbf{s}_i	vector to i^{th} deck feature in camera frame
t	time
T	total thrust
\mathbf{T}_h	transformation from inertial frame to helicopter body frame
\mathbf{x}_d	deck position
\mathbf{x}_h	helicopter position
\mathbf{x}_r	deck relative state
\mathbf{z}_i^{cam}	bearing measurement to i^{th} deck feature
\mathbf{z}_{imu}	acceleration measurements from inertial measurement unit
ϕ	deck relative roll angle (in helicopter frame)
ψ	deck relative heading angle (in helicopter frame)
θ	deck relative pitch angle (in helicopter frame)
ω	helicopter body angular rate vector

INTRODUCTION

The ability to conduct autonomous resupply and casualty evacuation missions in all weather conditions will greatly improve utility of autonomous rotorcraft. The ability to take-off and land autonomously in high sea states is thus a critical technology.

Experimental work of this nature has largely been conducted in calm predictable environments. Some work has been done with articulated landing pads to counter the motion of the ship. However, this would add considerable cost to the ship and it would limit autonomous helicopters to ships equipped with the system during rough seas. The ideal solution would allow the helicopter to land autonomously on an uncooperative ship and would be wholly contained within the helicopter.

Vision systems provide information rich measurements with low weight and power consumption. They are capable of measuring bearings to features, rates of bearings, and depth (when using multiple cameras). Vision systems are also a passive sensor and therefore stealthy. Active sensors such as LIDAR and millimeter wave radar provide accurate depth information but at the cost of greater power consumption and often greater weight.

Vision-based landings have been made on stationary surfaces (Refs. 9, 12) and pads with a known constant velocity (Ref. 13). The general problem of landing a helicopter on a ship is considerably more difficult: the motion of the ship is driven by pseudo random forcing forces and the air wake of the ship can have significant impacts on the dynamics of the helicopter. A tether based approach was detailed in (Ref. 8) but this limits the system to ships with the cor-

*Graduate Research Assistant, Penn State University

†Associate Professor, Penn State University

Presented at the AHS 72th Annual Forum, West Palm Beach, Florida, May 17–19, 2016. Copyright © 2016 by the American Helicopter Society International, Inc. All rights reserved.

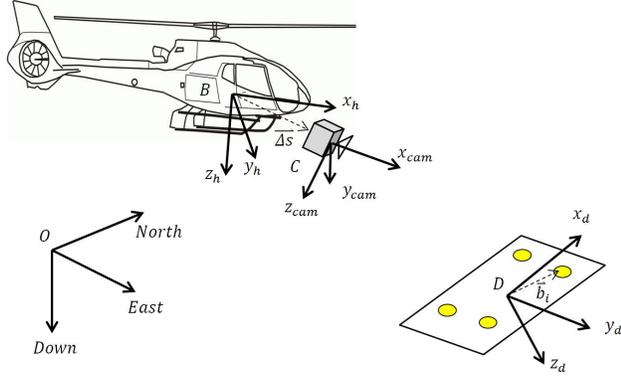


Fig. 1. Problem schematic.

rect hardware. Another vision-based landing approach is described in (Ref. 11) and (Ref. 10).

In earlier work we defined a ship-deck state estimator that computed the state of the ship in the inertial frame (Ref. 14). Here the estimator is re-cast so it estimates ship motion in the helicopter frame. In this formulation, landing can be defined as regulating the relative ship deck position and velocity to zero. Further, in the absence of GPS, the absolute position of the ship deck is unobservable, and estimators that attempt to compute an absolute ship deck position will be unable to do so. A landing system based on relative deck state estimation is thus independent of GPS.

PROBLEM STATEMENT

Given a helicopter equipped with a monocular camera and an inertial measurement unit (IMU), the problem at hand is to enable landing on a ship deck (Figure 1).

The ship deck is undergoing unknown motion in the inertial frame O . A camera fixed in a known position and orientation in the helicopter's frame tracks features located at known positions \mathbf{b}_i in the deck-fixed frame D . An inertial measurement unit measures the helicopter's acceleration (and the gravity vector projected into the helicopter frame) and angular rate. It is assumed that the helicopter is equipped with an autopilot module that is capable of following moderately high level commands (in the implementation discussed here, which uses a Y6 multi-rotor as demonstration vehicle, this means orientation and total thrust)

The approach taken here is to fuse vision and inertial data to compute an estimate of relative deck state and to use this estimate of deck relative state to compute a safe path to touchdown. A major goal of this research was to develop a system whose computational overhead is low enough that all major components can run on a small single-board computer such as an ODroid XU-4. A high level block diagram is shown in Figure 2.

Given data from a monocular camera and IMU, the key task is to compute an estimate of the relative ship deck state,

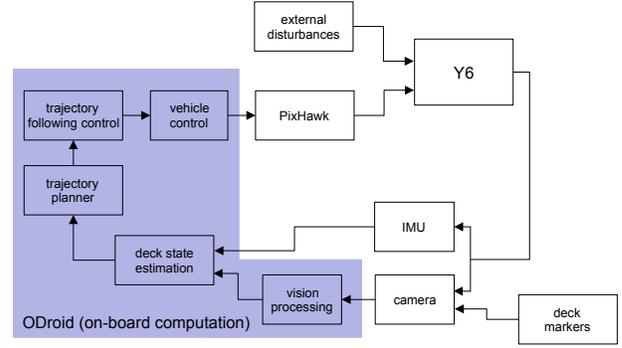


Fig. 2. Block diagram of vision-based landing system.

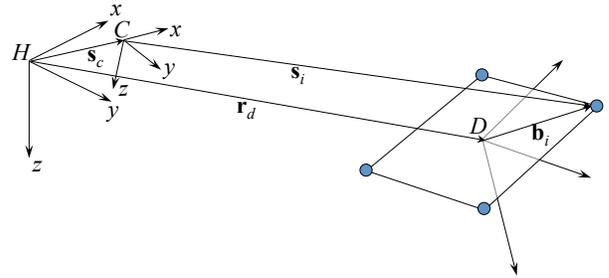


Fig. 3. Relevant coordinate frames.

i.e. to estimate

$$\mathbf{x}_r = [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^T \quad (1)$$

COORDINATE FRAMES AND SYSTEM MODELS

Coordinate frames

Referring to Figure 3, the camera is located with known orientation and position \mathbf{s}_c in the helicopter frame H . A vision processing system computes bearings \mathbf{s}_i to each marker located at known positions \mathbf{b}_i in the deck frame D . The position of the deck in the helicopter frame is denoted \mathbf{r}_d .

Deck Motion Model

This system described here tracks the position of the ship deck relative to the helicopter, expressed in the helicopter body frame. It uses measurements of helicopter acceleration, angular rates, and attitude from an on-board IMU. Defining the relative position of the deck as $\mathbf{r}_d = \mathbf{x}_d - \mathbf{x}_h$, the velocity and acceleration of the deck relative to the helicopter (expressed in the inertial frame) are

$$\frac{d\mathbf{r}_d}{dt} = \frac{d\mathbf{x}_d}{dt} - \frac{d\mathbf{x}_h}{dt} \quad (2)$$

$$\frac{d^2\mathbf{r}_d}{dt^2} = \frac{d^2\mathbf{x}_d}{dt^2} - \frac{d^2\mathbf{x}_h}{dt^2} \quad (3)$$

Expressed in the helicopter frame, relative deck motion is

$$\frac{d\mathbf{r}_d}{dt} = \dot{\mathbf{r}}_d + \boldsymbol{\omega} \times \mathbf{r}_d \quad (4)$$

where $\mathbf{r}_d = [x \ y \ z]^T$ (the position of the deck in the helicopter frame) and $\boldsymbol{\omega}$ is the angular rate of the helicopter body frame. Acceleration of the deck is

$$\frac{d^2\mathbf{r}_d}{dt^2} = \ddot{\mathbf{r}}_d + \dot{\boldsymbol{\omega}} \times \mathbf{r}_d + 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_d + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_d \quad (5)$$

We are interested in the relative velocity and relative acceleration of the deck:

$$\dot{\mathbf{r}}_d = \frac{d\mathbf{r}_d}{dt} - \boldsymbol{\omega} \times \mathbf{r}_d \quad (6)$$

$$\ddot{\mathbf{r}}_d = \frac{d^2\mathbf{r}_d}{dt^2} - \dot{\boldsymbol{\omega}} \times \mathbf{r}_d - 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_d - \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_d \quad (7)$$

Substituting relative velocity and acceleration,

$$\dot{\mathbf{r}}_d = \frac{d\mathbf{x}_d}{dt} - \frac{d\mathbf{x}_h}{dt} - \boldsymbol{\omega} \times \mathbf{r}_d \quad (8)$$

$$\ddot{\mathbf{r}}_d = \frac{d^2\mathbf{x}_d}{dt^2} - \frac{d^2\mathbf{x}_h}{dt^2} - \dot{\boldsymbol{\omega}} \times \mathbf{r}_d - 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_d - \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_d \quad (9)$$

The helicopter's IMU measures acceleration with respect to the inertial frame (expressed in the body frame) and the projection of gravity into the body frame:

$$\frac{d^2\mathbf{x}_h}{dt^2} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \boldsymbol{\omega} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{z}_{imu} - \mathbf{T}_h \mathbf{g} \quad (10)$$

where \mathbf{z}_{imu} is the measured acceleration and \mathbf{T}_h defines the transformation from the inertial frame to the helicopter frame. The deck's relative acceleration is thus a function of the deck's acceleration in the inertial frame, the deck's relative position and velocity, the helicopter's angular rate, and the helicopter's acceleration:

$$\ddot{\mathbf{r}}_d = \frac{d^2\mathbf{x}_d}{dt^2} - \dot{\boldsymbol{\omega}} \times \mathbf{r}_d - 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_d - \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_d - (\mathbf{z}_{imu} - \mathbf{T}_h \mathbf{g}) \quad (11)$$

One must be careful here: the above equation has mixed coordinate frames. The first term on the right hand side is expressed in the inertial frame; the rest are in the helicopter frame.

Here it is assumed that the deck moves at constant velocity, perturbed by zero-mean Gaussian acceleration.

$$\frac{d^2\mathbf{x}_d}{dt^2} = \mathcal{N}(0, \mathbf{Q}_{trans}) \quad (12)$$

where $\mathbf{x}_d = [x_d \ y_d \ z_d]^T$ (the position of the deck in the inertial frame) and \mathbf{Q}_{trans} defines the covariance of the Gaussian noise term used to model deck translational acceleration.

A similar derivation is used for relative orientation.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} (\mathbf{T}_d \boldsymbol{\omega}_d - \boldsymbol{\omega}) \quad (13)$$

where

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathcal{N}(0, \mathbf{Q}_{rot}) \quad (14)$$

Here $\boldsymbol{\omega}_d$ defines angular rates of the deck in the inertial frame; \mathbf{T}_d defines the transformation from helicopter frame to deck frame, $\boldsymbol{\omega}$ defines angular rotations of the helicopter frame, and \mathbf{Q}_{rot} defines the covariance of the Gaussian noise used to model deck rotational acceleration.

Note that this constant velocity deck motion model does not accurately describe the complex dynamics of ship motion but with a large enough Gaussian noise term, the model can capture the motion of a wide variety of ships. The attractiveness of this model stems from its simplicity and non-specificity: it is not dependent on a particular ship.

Vision System

The vision system tracks a set of markers on the ship and measures bearing and bearing rate (when markers are seen in consecutive frames). A modified pinhole camera model is used for bearing measurements (azimuth and depression angles). For the i^{th} marker

$$\mathbf{z}_i^{cam} = \begin{bmatrix} \arctan\left(\frac{f s_{i,y}}{s_{i,x}}\right) \\ \arctan\left(\frac{f s_{i,z}}{s_{i,x}}\right) \end{bmatrix} + \mathbf{n}_c \quad (15)$$

where \mathbf{n}_c is zero mean Gaussian noise. The vector \mathbf{s}_i is from the focal point of the camera to the i^{th} feature on the ship deck expressed in the camera's coordinate frame (Figure 4).

Recall that the position of the marker in the deck coordinate frame is assumed known. Hence

$$\mathbf{s}_i = \mathbf{T}_{cam}(\mathbf{r}_d + \mathbf{T}_d^T \mathbf{b}_i - \mathbf{s}_c) \quad (16)$$

Here \mathbf{T}_{cam} is the rotation from the helicopter frame to the camera frame and \mathbf{T}_d is the rotation from the helicopter frame to the deck frame. The vector \mathbf{b}_i defines the position of the i^{th} marker in the deck frame and the vector \mathbf{s}_c is the position of the camera in the helicopter frame.

Bearing rates are found by taking the derivative of equation (15). The bearing rate for the i^{th} marker is

$$\dot{\mathbf{z}}_i^{cam} = \begin{bmatrix} \frac{s_{i,x}s_{i,y} - \dot{s}_{i,x}s_{i,y}}{s_{i,x}^2 + s_{i,y}^2} \\ \frac{s_{i,x}s_{i,z} - \dot{s}_{i,x}s_{i,z}}{s_{i,x}^2 + s_{i,z}^2} \end{bmatrix} + \mathbf{n}_v \quad (17)$$

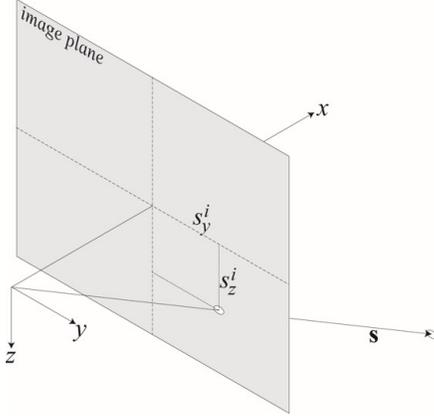


Fig. 4. Bearing measurement to the the i th feature.

where $\dot{s}_{i,x}$, $\dot{s}_{i,y}$, and $\dot{s}_{i,z}$ are the components of the velocity of the i^{th} marker in the camera frame. It is assumed that bearing rate measurements are corrupted by zero-mean Gaussian noise \mathbf{n}_v .

The velocity of the i^{th} marker in the camera frame is found by taking the derivative of the equation (16).

$$\dot{\mathbf{s}}_i = \mathbf{T}_{cam}(\dot{\mathbf{r}}_d + \boldsymbol{\omega} \times \mathbf{r}_d - \mathbf{T}_d^T \boldsymbol{\Omega}^T \mathbf{s}_i) \quad (18)$$

where $\boldsymbol{\Omega}$ is the skew symmetric matrix representing the cross product of the difference between the helicopter body rate and ship deck body rate in the helicopter frame, that is

$$\boldsymbol{\Omega} = (\boldsymbol{\omega}_d + \mathbf{T}_d \boldsymbol{\omega}) \times \quad (19)$$

UNSCENTED KALMAN FILTER

Deck motion predictions, measurements from the IMU, and measurements from the vision system are fused together using an unscented Kalman filter. UKFs are capable of estimating systems with highly nonlinear dynamics and have been used in several vision based estimation problems (Refs. 1, 3).

The dynamics of the time update step of the UKF can be written compactly as

$$\dot{\mathbf{x}}_r = f(\mathbf{x}_r, \mathbf{z}_{imu}) \quad (20)$$

Equation (11) and Equation (13) define deck motion in the helicopter frame; Equation (12) and Equation (14) define deck acceleration in the inertial frame (which are driven by zero-mean Gaussian noise). Since we are estimating deck state in the helicopter frame, the motion model is not affine in the process noise (which is acting on the deck in the inertial frame).

In the hardware implementation used here, a fourth order Runge Kutta integration is used to discretize the time update equations.

The measurement update can be described compactly as

$$\mathbf{z}_{vision} = \begin{bmatrix} \mathbf{z}_1^{cam} \\ \dot{\mathbf{z}}_1^{cam} \\ \mathbf{z}_2^{cam} \\ \dot{\mathbf{z}}_2^{cam} \\ \vdots \end{bmatrix} \quad (21)$$

where \mathbf{z}_i^{cam} is given by Equation (15) and $\dot{\mathbf{z}}_i^{cam}$ is given by Equation (17).

Note that the measurement update step assumes either: (1) features are labeled, so that measurements can be properly associated; or (2) a method for data association has been implemented. Here we have explicitly implemented data association.

APPROACH TRAJECTORY GENERATION

Given an estimate of relative deck state, the vehicle must now compute a safe, dynamically feasible trajectory to touchdown.

Trajectories were generated using the ecologically inspired Tau theory (Ref. 4). Kendoul (Ref. 2) applied applied this theory to generate a biologically inspired TauPilot. Tau theory has been successful in describing the landing behaviors of hummingbirds (Ref. 6), pigeons (Ref. 5), and bats (Ref. 7). Tau theory is based on the evidence that time to contact is commonly used in the guidance of movement. The strategy used by birds is to keep tau dot constant. So

$$\dot{\tau} = k \quad (22)$$

where τ is the ratio between the distance to the goal and the velocity towards it. Solving this equality for position, velocity, and acceleration yields

$$\chi(t) = \chi_0 \left(1 + k \frac{\dot{\chi}_0 t}{\chi} \right)^{\frac{1}{k}} \quad (23)$$

$$\dot{\chi}(t) = \dot{\chi}_0 \left(1 + k \frac{\dot{\chi}_0 t}{\chi} \right)^{\frac{1-k}{k}} \quad (24)$$

$$\ddot{\chi}(t) = \chi_0 \frac{\dot{\chi}_0^2}{\chi} (1-k) \left(1 + k \frac{\dot{\chi}_0 t}{\chi} \right)^{\frac{1-2k}{k}} \quad (25)$$

In the Tau-dot equations, trajectories begin with a positive velocity towards the destination. In many situations there is no initial velocity. The solution proposed by tau theory is to have the gap follow an intrinsically generated guiding gap

$$\frac{\chi}{\dot{\chi}} = k \frac{1}{2} \left(t - \frac{T_g^2}{t} \right) \quad (26)$$

Table 1. Touchdown characteristics with different ranges of k .

k range	touchdown $\dot{\chi}$	touchdown $\ddot{\chi}$
$0 < k < 0.5$	0	0
$.5 < k < 1.0$	> 0	inf
$1.0 = k$	$\dot{\chi}_0$	0
$1.0 < k$	inf	inf

where T_g is the time elapsed while closing the distance. This gap closing matched to the gap was based on the behavior of electricity through neurons (Ref. 4). Solving for the trajectory gives

$$\chi = \frac{\chi_0}{T_g^{\frac{2}{k}}} (T_g^2 - t^2)^{\frac{1}{k}} \quad (27)$$

$$\dot{\chi}(t) = -2 \frac{\chi_0}{T_g^{\frac{2}{k}}} \frac{1}{k} t (T_g^2 - t^2)^{\frac{1-k}{k}} \quad (28)$$

$$\ddot{\chi}(t) = 2 \frac{\chi_0}{T_g^{\frac{2}{k}}} \frac{1}{k} t \left(\frac{2-k}{k} t^2 - T_g^2 \right) (T_g^2 - t^2)^{\frac{1-2k}{k}} \quad (29)$$

where the value of k determines the trajectory's characteristics. For values of k less than zero there is no touchdown, the distance grows larger and never touches down. For k values between 0 and 0.5 the touchdown has zero velocity and zero acceleration. Table 1 contains the different touchdown possibilities as a function of k .

The trajectory generated to the deck was defined by three separate trajectory's, one in each axis of the deck's reference frame. A value of $k = 0.3$ was used for each component as this provided attractive touchdown properties.

TRAJECTORY FOLLOWING CONTROLLER

Trajectory following control uses a feed forward/feedback approach, where the acceleration from the tau-guidance approach trajectory is used as a feed forward term and feedback is done using the sequence of velocities and positions from the tau-guidance trajectory. This is equivalent to proportional-derivative control of position along the trajectory combined with commanded acceleration. In effect, the helicopter is connected to the desired trajectory by a virtual spring that provides desired accelerations to the inner loop controller (Figure 5).

The resulting commanded accelerations are

$$\ddot{x}_{cmd} = \frac{k_x}{m} (x_{traj} - x) + \frac{b_x}{m} (\dot{x}_{traj} - \dot{x}) + \ddot{x}_{traj} \quad (30)$$

$$\ddot{y}_{cmd} = \frac{k_y}{m} (y_{traj} - y) + \frac{b_y}{m} (\dot{y}_{traj} - \dot{y}) + \ddot{y}_{traj} \quad (31)$$

$$\ddot{z}_{cmd} = \frac{k_z}{m} (z_{traj} - z) + \frac{b_z}{m} (\dot{z}_{traj} - \dot{z}) + \ddot{z}_{traj} \quad (32)$$

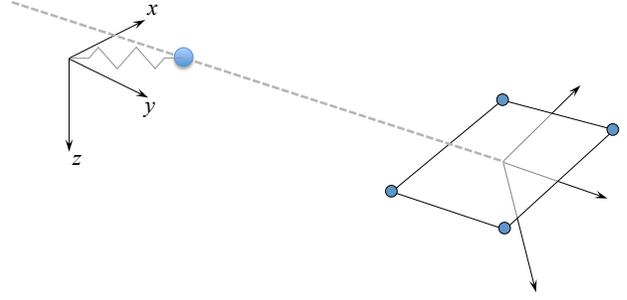


Fig. 5. Schematic of trajectory following control showing the "virtual spring."

In the case of a multi-rotor (such as the 3DRobotics Y6 used here), these accelerations are achieved via pitch, roll, and total thrust. The relation between attitude, throttle, and acceleration is

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \mathbf{T}_h^T \begin{bmatrix} 0 \\ 0 \\ \frac{-T}{m} \end{bmatrix} + \mathbf{g} \quad (33)$$

Note that z is positive down and the vehicle thrust vector is directed upwards in the vehicle's body $-z$ direction. Commanded accelerations can thus be achieved (assuming they are within the limits of vehicle actuation capabilities) by solving

$$\begin{bmatrix} m\ddot{x}_{cmd} \\ m\ddot{y}_{cmd} \\ mg - m\ddot{z}_{cmd} \end{bmatrix} = \begin{bmatrix} -T \sin \theta \cos \phi \\ T \sin \phi \\ T \cos \theta \cos \phi \end{bmatrix} \quad (34)$$

Here we assume that the vehicle is near hover at all times, hence pitch and roll angles are small. Commanded thrust, roll, and pitch can then be solved in closed form:

$$T_{cmd} = m (g - \ddot{z}_{cmd}) \quad (35)$$

$$\phi_{cmd} = m \frac{\ddot{y}_{cmd}}{T_{cmd}} \quad (36)$$

$$\theta_{cmd} = -m \frac{\ddot{x}_{cmd}}{T_{cmd}} \quad (37)$$

These commands are then sent to the autopilot module.

TOUCHDOWN CONTROLLER

In the final portion of the flight, the desired position of the vehicle switches from the state generated by the tau module to a position 10 cm below the deck directly under the vehicle. The same trajectory following controller was used for the touchdown portion of the flight, though in this portion there was no feedforward portion of the trajectory. This change in the desired position allowed the vehicle to land more softly than it would have with a motor kill at the end

of the trajectory. It also protected the vehicle from catching landing gear on the deck if the vehicle arrived at the deck off the center of the deck. This touchdown controller allowed the vehicle to terminate flights on the level deck safest and consistently. Note that Kendoul used a similar approach: tau guidance was used to control an action gap to zero, and a controller to regulate the gap to zero was activated when the gap was adequately closed (Ref. 2).

HARDWARE IMPLEMENTATION AND TEST RESULTS

Hardware Description

A 3DRobotics Y6 multi-copter was used for all flight tests described here. It is equipped with a PixHawk autopilot module (running the PX4 flightstack), an ODroid XU-3 single board computer, and an mvBlueFox camera. A six degree of freedom Stewart platform is used to model the moving ship deck (see Figure 6).

Figure 7 shows how information moves through the system. The ODroid hosts all algorithms required for flight: deck state estimation, vision processing, trajectory following control, path planning, and high-level control. Robot Operating System (ROS) version Indigo manages all processes and messaging.

The PixHawk autopilot module includes rate gyros, accelerometers, a magnetometer, and GPS module. It passes inertial measurements to the ODroid via a serial port; a ROS node parses inertial data and publishes it to the ROS network.

A vision node obtains raw images from the camera and publishes bearings and bearing rates. The estimator node subscribes to both inertial and vision measurements, and publishes the deck state estimate.

A trajectory generator node subscribes to the estimates and uses tau-guidance to compute trajectories to touchdown; it publishes the trajectory as a sequence of nominal positions, velocities, and accelerations.

A trajectory following controller node subscribes to the trajectory and to the deck state estimates; it sends commanded throttle, roll, and pitch commands to the autopilot.

Estimated states are also sent to a ground station via wifi, where they can be monitored by a human supervisor. Higher level commands (such as enabling autonomy) are sent via the ground station to the PixHawk, which then sends the command to the ODroid. We can thus enable and disable autonomy, passing control between a human operator and the autonomy system carried on board the vehicle.

The locations of the lights on the deck can be seen in table 3. The distance in position between the lights on the y axis was 20.4 cm and the distance between the lights on the x axis was 18 cm.

Table 2. Gains for trajectory following controller

	x	y	z
$k_{(\cdot)}$	1.0	1.0	1.0
$b_{(\cdot)}$	2.0	2.0	2.0

Table 3. light locations in the deck frame (cm)

x	y	z
-12.8	0	0
5.2	0	0
0	10.2	0
0	-10.2	0

Test Process

Deck state estimates were sent to the ground station so the validity of deck estimates could be verified before each flight. If deck estimates were reasonable, the vehicle entered autonomous mode. As the vehicle enters autonomous mode, trajectories were generated in the tau module. These trajectories were then tracked by the trajectory following controller whose outputs, desired attitude and throttle, were passed to the Pixhawk. Once the vehicle had landed, manual control was resumed and the vehicle was flown back above the deck. Once the vehicle was close to the nominal starting position, the UKF was reset from the ground station and the process was started again.

Stationary deck

Sixty four approaches to a stationary deck were conducted, with approach paths shown in Figure 8 (subfigure (a) shows a top-down view, subfigure (b) shows height vs. distance to the deck). Note that “distance to deck” is computed with respect to the center of gravity of the vehicle, which at the moment of touchdown is approximately 10 centimeters above the deck. All approaches resulted in safe touchdown.

Error in the deck state estimate as a function of distance to the deck is shown in Figure 9. Each run is shown as a grey line; the average over all runs is shown as black dots. Note that the error steadily decreases as the vehicle approaches the deck.

The top plot of Figure 9 shows the 2-norm of the total estimate error, where the units for position, attitude, velocity, and body rate are meters, radians, meters per second, and radians per second respectively.

The middle plot shows the history of position error. Initial errors are large because of estimator initialization, but estimates rapidly converge to centimeter-level accuracy.

The bottom plot shows the error in attitude estimates. The worst-case attitude error is roughly fifteen degrees:

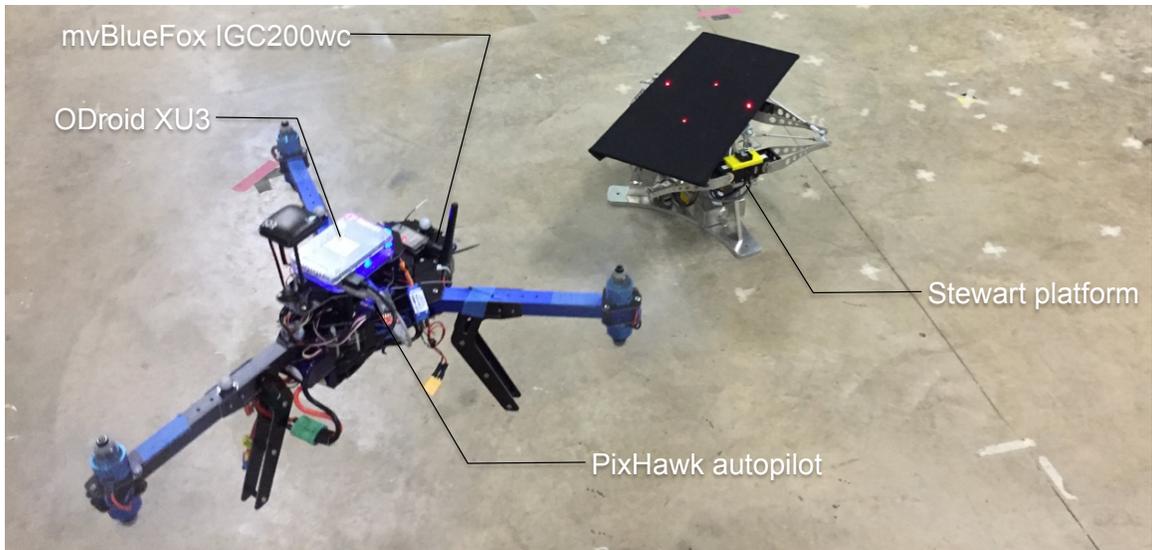


Fig. 6. Small hex-copter used for hardware tests with Stewart platform landing pad in background. All vision processing and state estimation are done on the single-board computer mounted on top of the vehicle; the camera is mounted underneath. The Stewart platform is capable of six degree of freedom motion and can be programmed to mimic arbitrary ship deck motion.

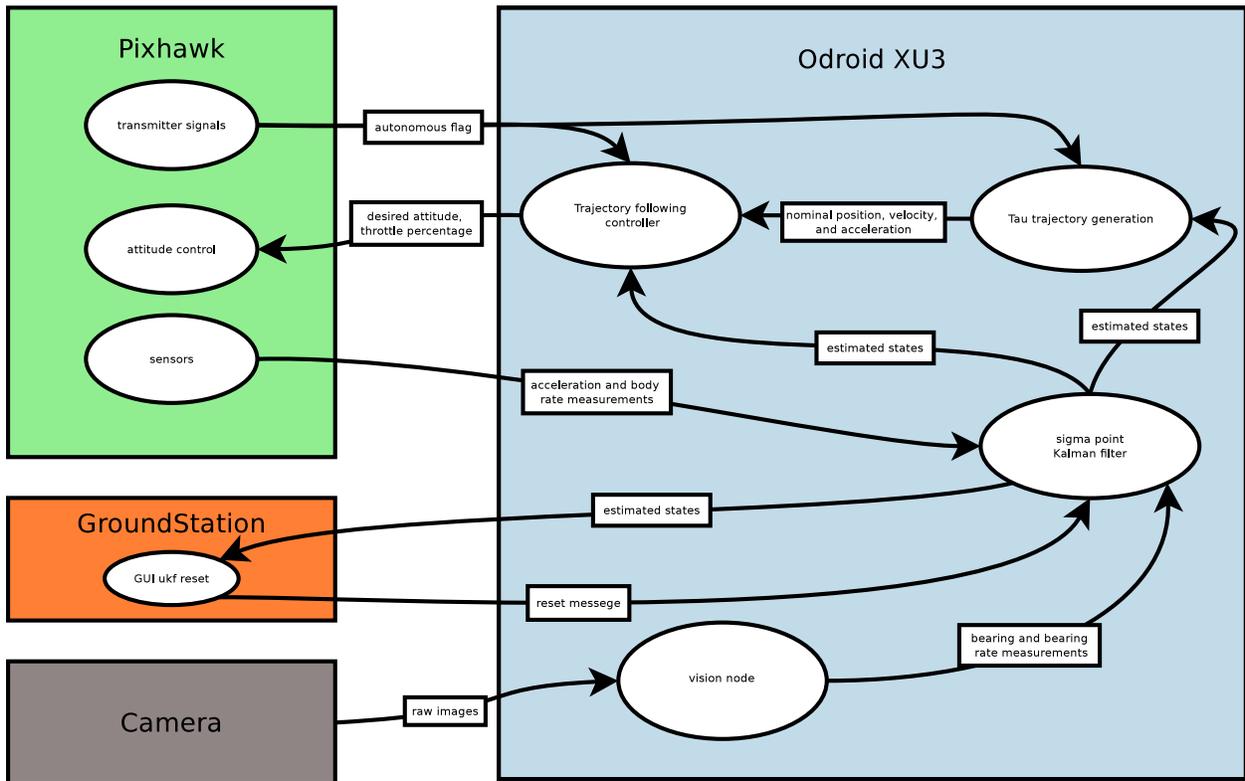


Fig. 7. How information migrates through the system. The Pixhawk, Odroid, and Camera are carried on the hex-copter pictured in figure 6. The ground station is used to verify the estimates from the UKF and to reset the system between runs.

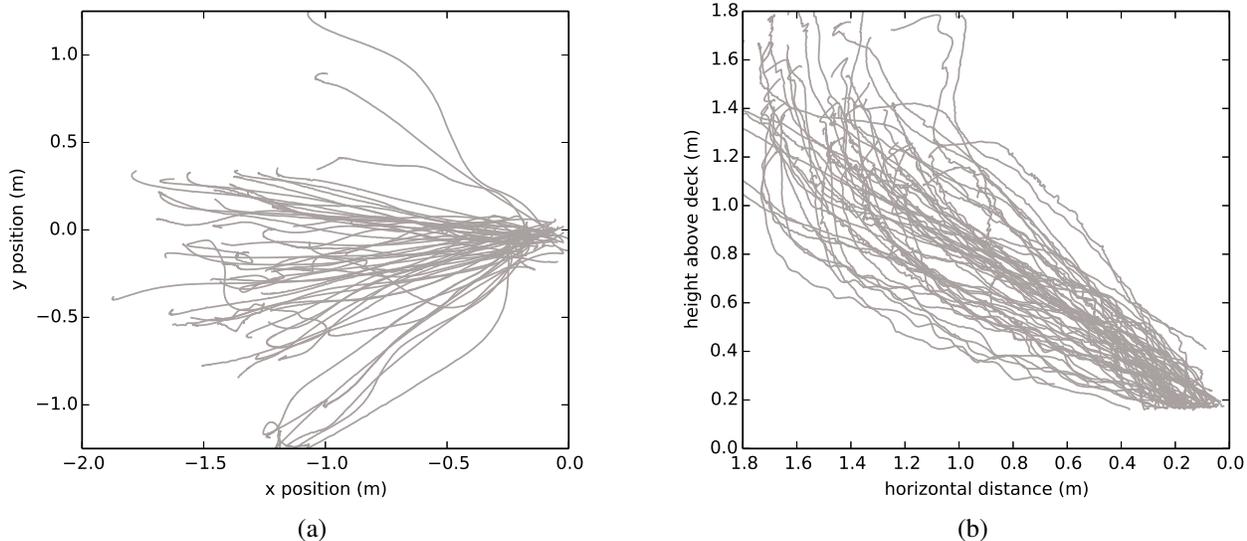


Fig. 8. The paths taken to the deck in the horizontal plane (a) and the height above the deck against the distance to the deck in the horizontal plane.

even with this large attitude error the vehicle still landed safely. Note that the angle error tends to shrink during the initial approach, and then begins to grow during the final phase of the approach. This is most likely due to imperfect camera calibration: as the distance to the deck gets small, features tend to appear at the edges of the field of view, where calibration errors have a large effect. Further, it is more likely that a feature will leave the field of view when the deck is close, reducing the information that can be obtained from the vision system.

Figure 10 shows position estimates, error, two sigma bounds, truth, and the autonomy flag over seven consecutive runs (with a run consisting of hover, estimate initialization, hand-over to autonomy, approach, landing, and manual repositioning to hover). The hand-over to autonomy is shown by the autonomy indicator going high. Note that estimate error remains generally within the 2σ bounds once the estimator is initialized, and the rapid approach to the deck once autonomy is enabled. A complete sequence takes approximately 20 seconds to complete.

Figure 11 shows position estimate error and the computed covariance of the estimate error as a function of distance for a single representative run. The initial error and covariance is large due to estimate initialization, but rapidly approaches centimeter levels as the vehicle approaches the deck. Further, the estimate error stays generally within 2σ bounds. Close examination of the 2σ plot shows a saw-tooth pattern that corresponds to the vision updates.

Moving deck

A total of eight approaches to a moving deck were performed. Figure 12 shows a sample of deck motion about the mean deck state. As a reference scale, the hexacopter is approximately 30 centimeters across. All eight approaches resulted in safe touchdown.

Deck motion in these tests was fairly slow, with roll and pitch having a period of approximately 20 seconds, and surge and sway having a period of approximately 17 seconds. A single approach is flown in about 8 seconds from the moment of autonomy activation.

Figure 13 shows error in deck state estimates for all eight runs (as a grey line for each run) and the average over all runs (as black dots). Qualitatively, there is very little difference between the moving deck cases and the stationary deck cases. Position error begins at approximately thirty centimeters and rapidly converges to centimeter level accuracy at touchdown; angle errors begin large and converge rapidly to touchdown.

Figure 14 shows the deck true relative position, estimated relative position, true error, and estimated error versus time for a representative run. The autonomy flag is high when autonomy is enabled. Note that error in estimated position remains generally within 2σ bounds and converges rapidly to zero. Touchdown occurs at approximately 133 seconds, and the vehicle is manually flown back to an initial hover position.

Similar behavior is shown for attitude estimates in Figure 15. Errors remain within 2σ bounds and remain below five degrees.

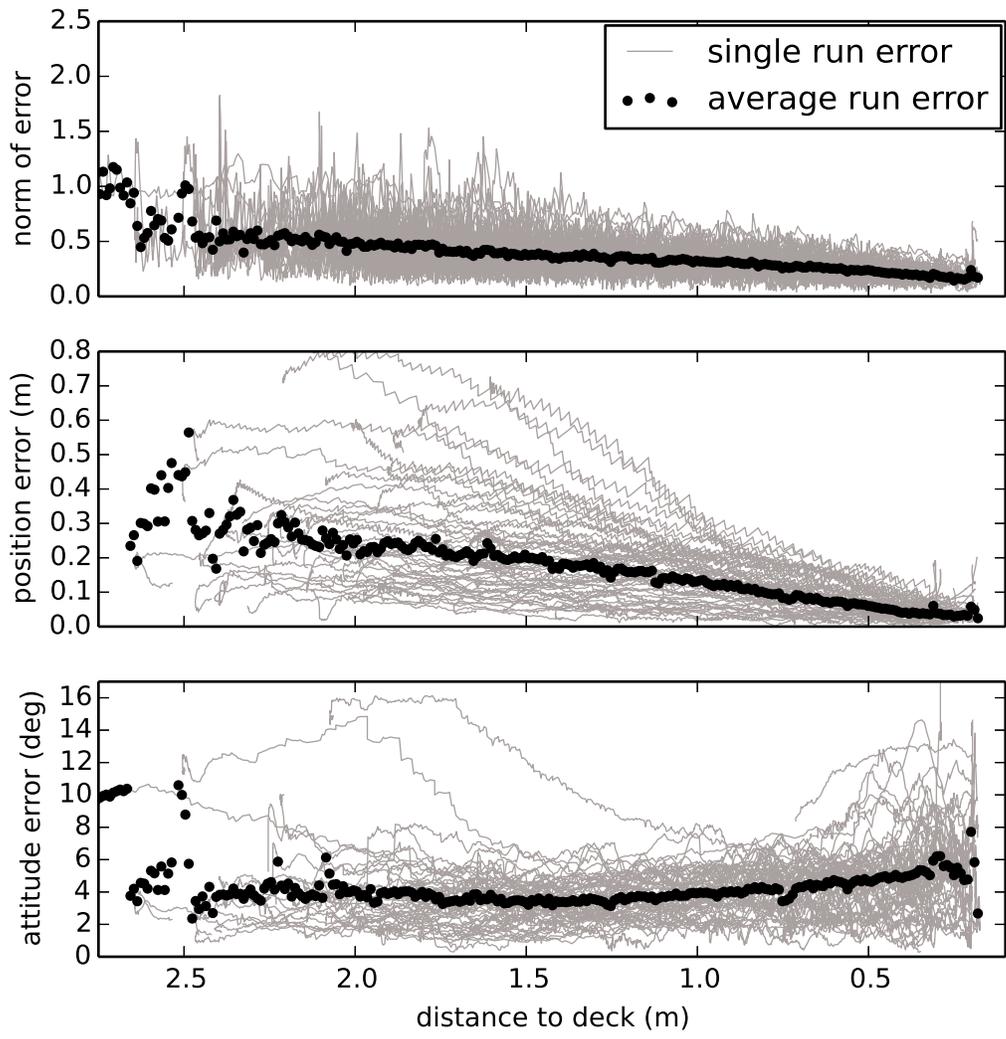


Fig. 9. The errors in the estimator over 64 flights. Individual flights are plotted in grey. The black scatter plot shows the average error in 1 cm bins. Shown above in descending order are the norm of all errors, the norm of the position errors, and the norm of the attitude errors.

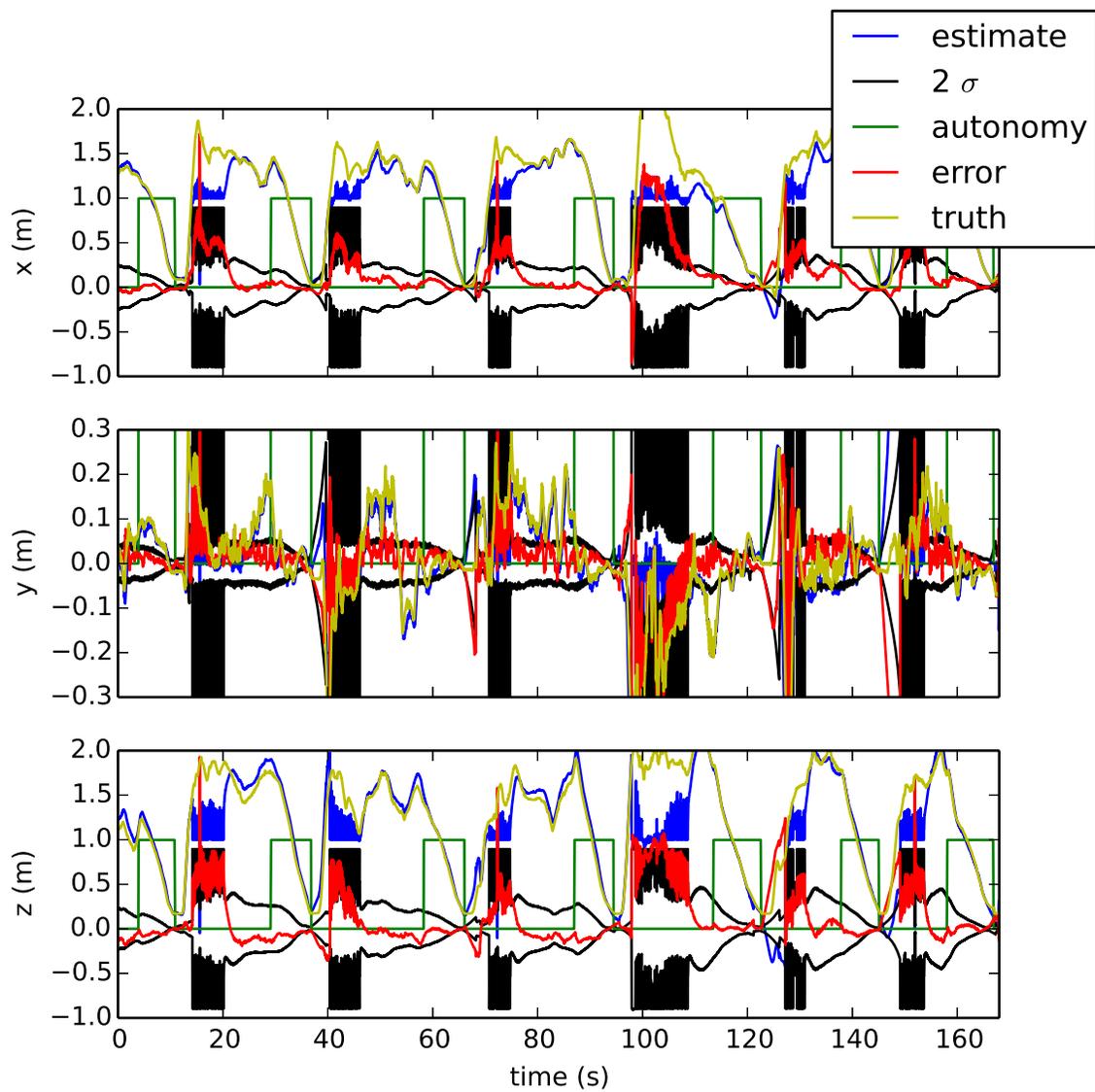


Fig. 10. The estimates and errors of the deck position in the vehicle frame over the course of seven consecutive runs. High regions in the autonomy line indicate periods in which the vehicle was autonomous. The system presented here requires minimal setup between flights which allows for rapid testing and verification.

CONCLUSION

This paper has presented a method for autonomous landing on a moving deck using a monocular vision system and an inertial measurement unit carried on the aircraft. A sigma-point Kalman filter computes an estimate of relative deck state and uses tau-guidance to compute a path to touchdown. A path-following controller follows this trajectory to touchdown. This approach is independent of GPS and uses only sensing carried aboard the vehicle.

Hardware tests were conducted in an indoor flight facility, with a motion capture system providing truth data. Sixty four touchdowns to a stationary deck showed that the deck state estimator remains well-behaved (i.e. the error in deck state estimate is well described by the estimated covariance) and all flights resulted in safe touchdown. Eight flights to a moving deck (motion was not known *a priori* to the estimator) were conducted: all resulted in well-behaved state estimates and safe touchdown.

Future work will consist of expanding the envelope of deck motion.

ACKNOWLEDGEMENTS

This research is partially funded by the Government under Agreement No. W911W6-11-2-0011. The U.S. Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

REFERENCES

- ¹A. Huster and S. Rock, "Relative position sensing by fusing monocular vision and inertial rate sensors," in *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, Coimbra, Portugal, July 2003.
- ²F. Kendoul, "Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 237–267, 2014.
- ³J. W. Langelaan, "State estimation for autonomous flight in cluttered environments," *Journal of Guidance, Control and Dynamics*, vol. 30, no. 5, pp. 1414–1426, September–October 2007.
- ⁴D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.

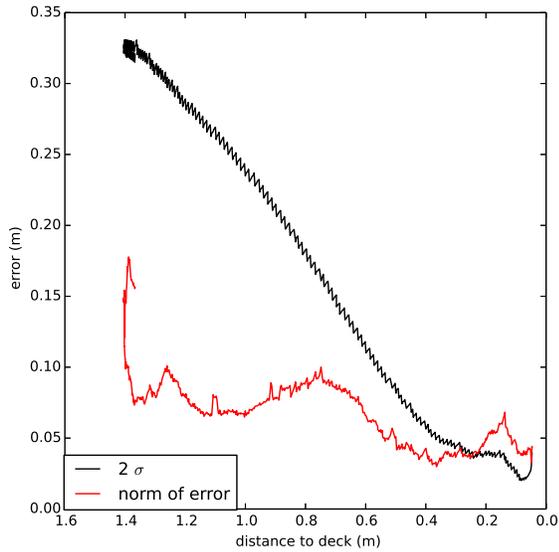


Fig. 11. The error of the estimate as the vehicle approaches the deck with 2σ bounds.

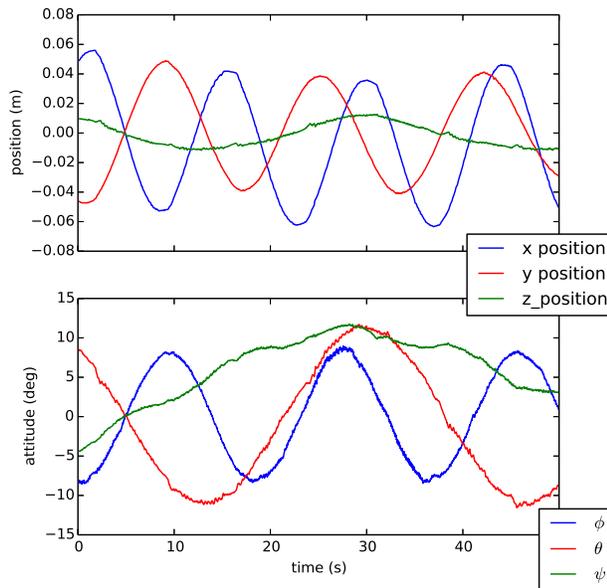


Fig. 12. The movement of the deck about its mean position and attitude.

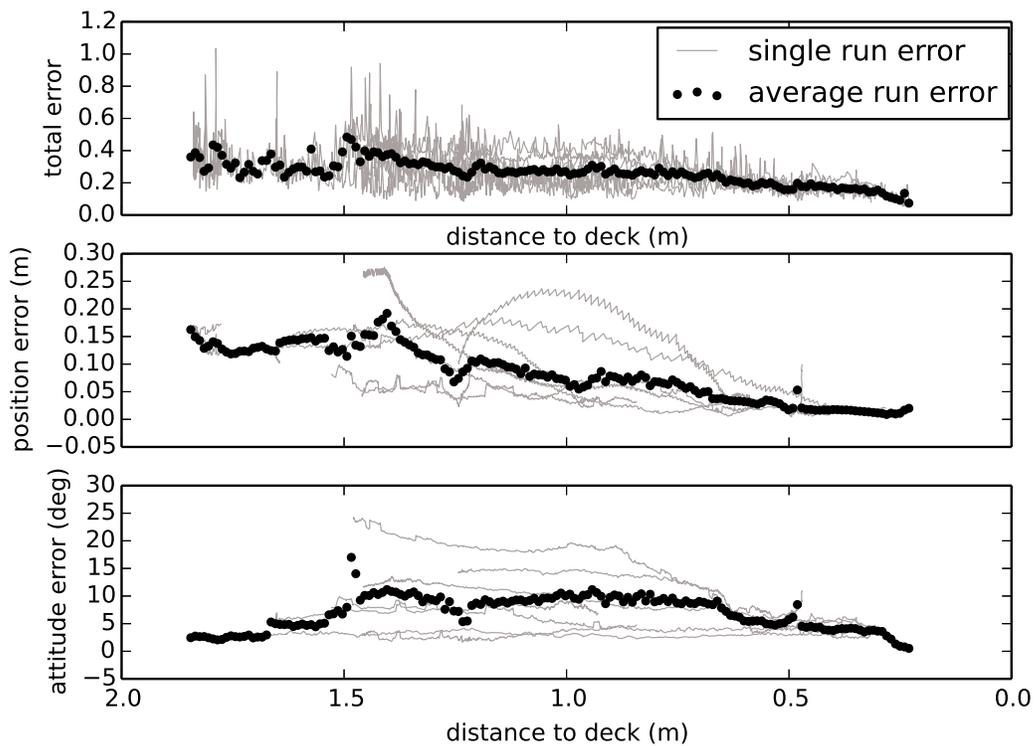


Fig. 13. The errors in the estimator in flights to a moving deck. Individual flights are plotted in grey. The black scatter plot shows the average error in 1 cm bins. Shown above in descending order are the norm of all errors, the norm of the position errors, and the norm of the attitude errors. The errors for position, attitude, velocity, and body rate are in meters, radians, meters per second, and radians per second respectively.

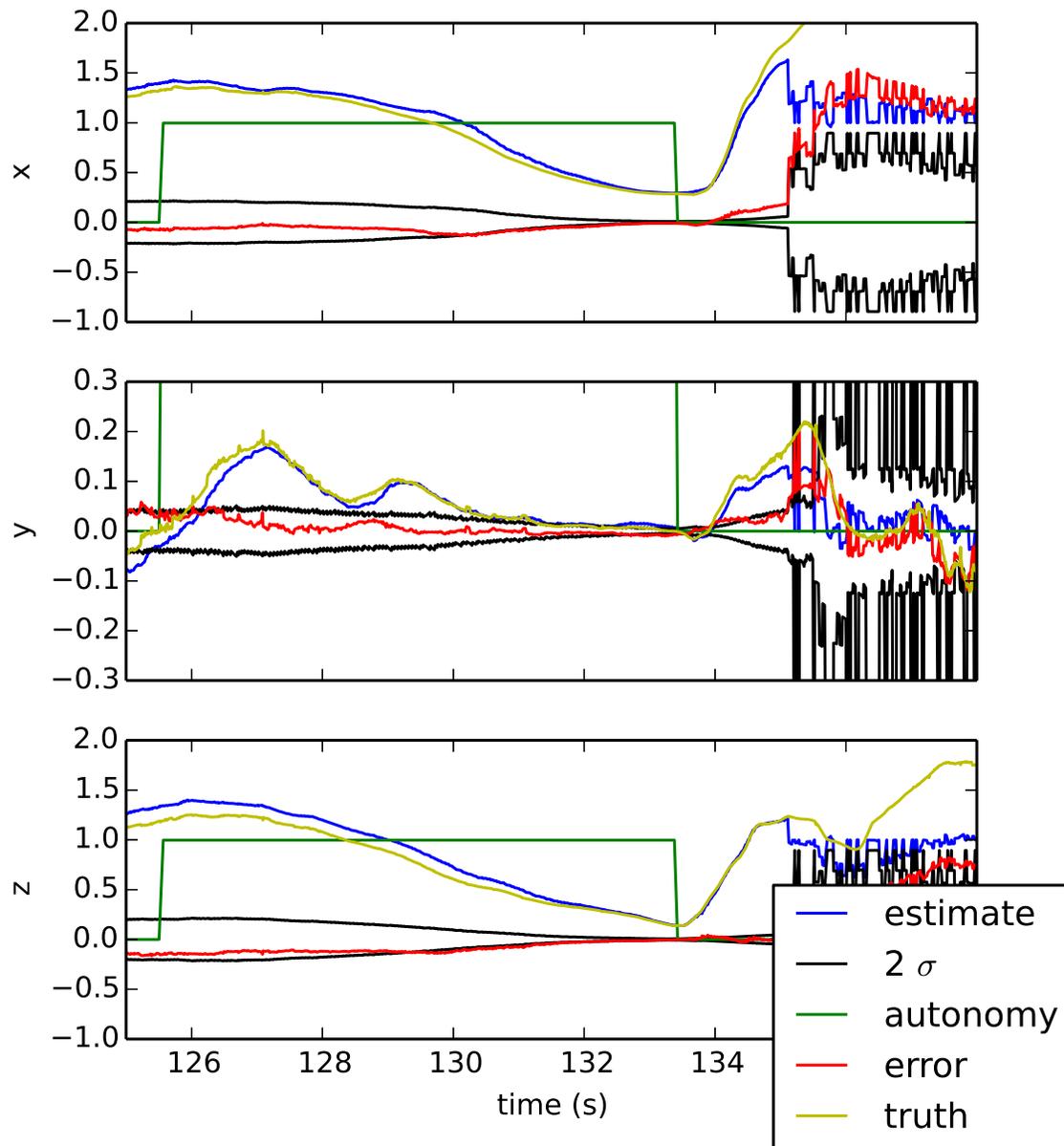


Fig. 14. The error of the position estimates a single run to a moving deck. The autonomous portion of the flight is indicated by the high region in the autonomy line.

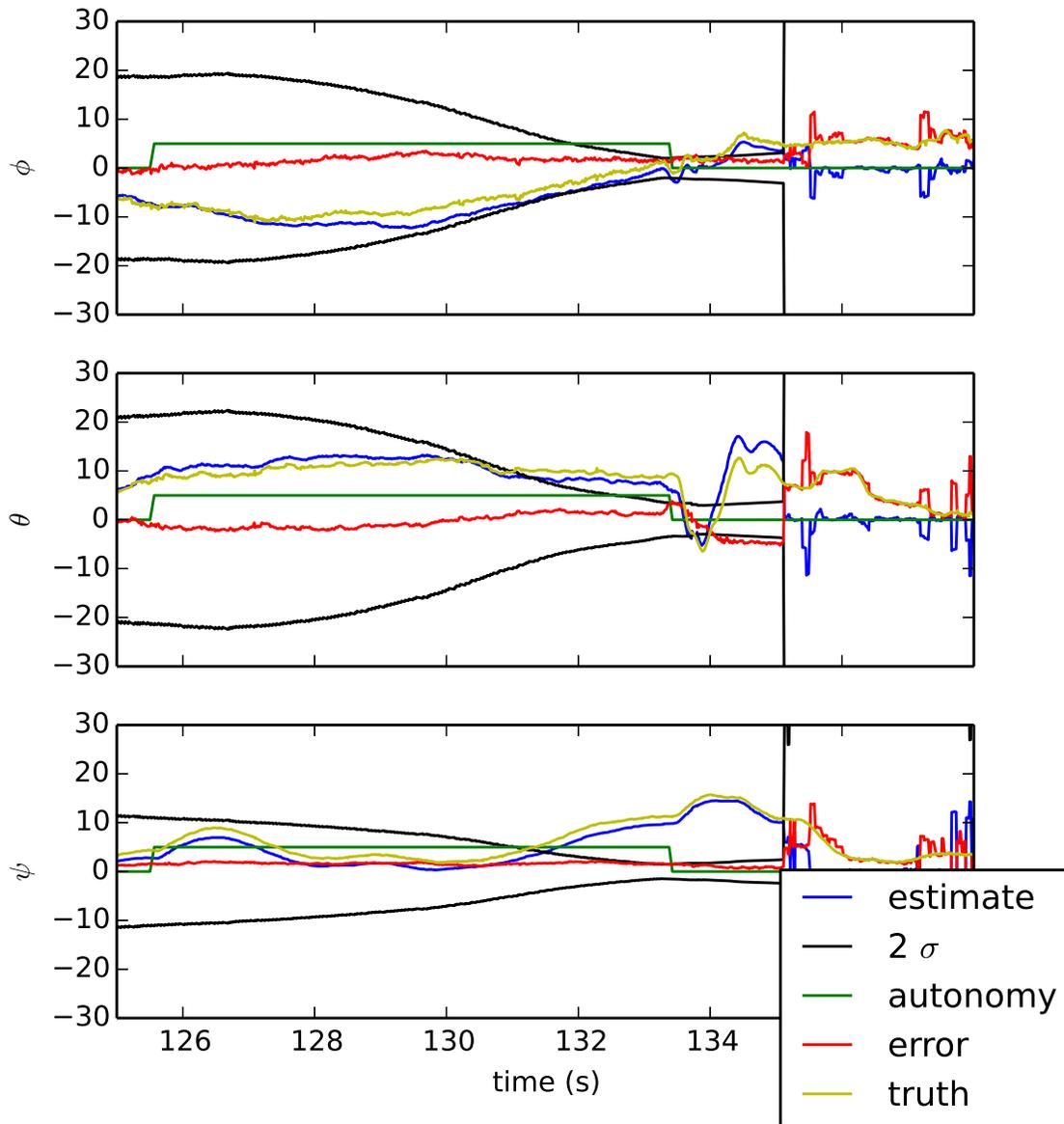


Fig. 15. The error of the attitude estimates a single run to a moving deck. The autonomous portion of the flight is indicated by the high region in the autonomy line.

⁵D. N. Lee, M. N. Davies, P. R. Green, *et al.*, “Visual control of velocity of approach by pigeons when landing,” *Journal of Experimental Biology*, vol. 180, no. 1, pp. 85–104, 1993.

⁶D. N. Lee, P. E. Reddish, and D. Rand, “Aerial docking by hummingbirds,” *Naturwissenschaften*, vol. 78, no. 11, pp. 526–527, 1991.

⁷D. N. Lee, J. A. Simmons, P. A. Saillant, and F. Bouffard, “Steering by echolocation: a paradigm of ecological acoustics,” *Journal of Comparative Physiology A*, vol. 176, no. 3, pp. 347–354, 1995.

⁸S.-R. Oh, K. Pathak, S. K. Agrawal, H. R. Pota, and M. Garratt, “Approaches for tether guided landing of an autonomous helicopter,” *IEEE Transactions on Robotics*, vol. 22, no. 3, 2006.

⁹A. Proctor and E. N. Johnson, “Vision-only approach and landing,” in *AIAA Guidance, Navigation and Control Conference*. San Francisco, California: AIAA Paper 2005-5871, American Institute of Aeronautics and Astronautics, August 2005.

¹⁰J. L. Sanchez-Lopez, J. Pestana, S. Saripalli, and P. Campoy, “An approach toward visual autonomous ship board landing of a vtol uav,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 113–127, 2014.

¹¹J. L. Sanchez-Lopez, S. Saripalli, and P. Campoy, “Autonomous ship board landing of a vtol uav,” in *69th American Helicopter Society International Annual Forum 2013*, 2013.

¹²S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, “Vision-based autonomous landing of an unmanned aerial vehicle,” in *International Conference on Robotics and Automation*. Washington DC: IEEE, May 2002, pp. 2799–2804.

¹³S. Saripalli and G. S. Sukhatme, “Landing a helicopter on a moving target,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 2030–2035.

¹⁴B. Truskin and J. W. Langelaan, “Vision-based deck state estimation for autonomous ship-board landing,” in *Proceedings of the AHS Forum*, 2013.