

Specialized Algorithm for Navigation of a Micro Hopping Air Vehicle Using Only Inertial Sensors

Edward Scheuermann

Graduate Research Assistant, Student Member ASME
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, 30332
Email: escheuermann3@gatech.edu

Mark Costello

Professor, Member ASME
School of Aerospace Engineering
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, 30332
Email: mark.costello@aerospace.gatech.edu

ABSTRACT

The need for accurate and reliable navigation techniques for micro air vehicles plays an important part in enabling autonomous operation. Traditional navigation systems typically rely on periodic GPS updates and provide little benefit when operating indoors or in other similarly shielded environments. Moreover, direct integration of the onboard inertial measurement unit (IMU) data stream often results in substantial drift errors yielding virtually unusable positional information. This paper presents a new strategy for obtaining an accurate navigation solution for the special case of a micro hopping air vehicle, beginning from some known location and heading, using only one triaxial accelerometer and one triaxial gyroscope. Utilising the unique dynamics of the hopping vehicle, a piece-wise navigation solution is constructed by selectively integrating the inertial data stream for only those short periods of time while the vehicle is airborne. Inter-hop data post-processing and sensor bias recalibration are also used to further improve estimation accuracy. To assess the performance of the proposed algorithm, a series of tests were conducted in which the estimated vehicle position following a sequence of 10 consecutive hops was compared with measurements from an optical motion-capture system. On average, the final estimated vehicle position was within 0.70 m or just over 6% from its actual location based on a total traveled distance of approximately 11 m.

Nomenclature

\vec{a}	=	vehicle linear acceleration vector
a_x, a_y, a_z	=	body frame components of the vehicle mass center acceleration
B_{Ai}, B_{Gi}	=	i^{th} component of the accelerometer and gyroscope bias vector
C_{Ai}^j, C_{Gi}^j	=	cross-axis sensitivity of accelerometer and gyroscope i^{th} axis with respect to the j^{th} axis
\mathbf{F}	=	linearized system matrix
g	=	gravitational acceleration
\mathbf{H}	=	linearized measurement matrix
\mathbf{I}	=	identity matrix
\mathbf{K}	=	Kalman gain matrix
N_{Ai}, N_{Gi}	=	i^{th} component of accelerometer and gyroscope noise vector
$\mathbf{P}, \mathbf{Q}, \mathbf{R}$	=	error, process noise, and measurement noise covariance matrices
p, q, r	=	angular velocity components in body reference frame
$\vec{r}_{a \rightarrow b}$	=	position vector extending from generic point a to another point b

$\mathbf{S}_A, \mathbf{S}_G$	=	scaling matrices of accelerometer and gyroscope
S_{Ai}, S_{Gi}	=	scale factor for i^{th} axis of accelerometer and gyroscope
\mathbf{T}_{IB}	=	transformation matrix from body to inertial reference frame
\mathbf{T}_{SB}	=	transformation matrix from body to sensor reference frame
t	=	time
u, v, w	=	body frame components of vehicle translation velocity
\vec{v}, \vec{w}	=	measurement and process noise vectors
x, y, z	=	inertial position components of vehicle mass center
\vec{x}	=	system state vector
\vec{y}	=	system output vector
\vec{z}	=	sensor measurement vector
$\vec{\alpha}$	=	vehicle angular acceleration vector
$\vec{\delta}_A$	=	accelerometer misposition vector
ϕ, θ, ψ	=	vehicle Euler roll, pitch, and yaw angles
$\vec{\omega}$	=	vehicle angular velocity vector

Subscript

B	=	body reference frame
I	=	inertial reference frame
S	=	sensor reference frame

Superscript

*	=	raw sensor value
---	---	------------------

1 Introduction

Micro vehicle platforms are an attractive choice for many mobile applications. They are ideally suited for search and rescue, autonomous exploration, surveillance, and a host of other military missions. Accordingly, accurate and reliable navigation of these vehicles is critical to the successful execution of the intended mission. Considering the relatively small footprint and low payload capacity inherent to micro vehicle platforms, a growing need exists for a lightweight, cost-effective solution for automatic vehicle localization and navigation. Furthermore, the autonomous navigation challenge is often intensified when operating within unknown, indoor environments. Such scenarios present extreme difficulty to micro vehicle designers in that conventional localization techniques including Global Positioning System (GPS) communication are no longer accessible within certain covered structures or similarly shielded locations.

While the concept of micro vehicle navigation has been extensively investigated, a large portion of research focuses on vehicle state estimation and localization via the fusion of several different onboard sensors. One such example presented by Liu et al. [1] combines the use of GPS with a microelectromechanical (MEMS) based IMU. Vehicle acceleration and angular velocity are continuously integrated and subsequently fused together with discrete GPS positional measurements using a Kalman filter. Although direct integration of the IMU data is possible, some form of external reference or filtering algorithm is required such that the resulting error growth is sufficiently bounded. Similarly, several researchers [2–6] have extended the GPS/IMU approach by adding additional sensors including magnetometers, differential pressure transducers, and baro-altimeters to both improve accuracy and help compensate for vehicle attitude and localization errors during periods of degraded GPS signal quality. Using flight test data from a micro aerial vehicle (MAV) quadrotor equipped with GPS receiver, MEMS-based IMU, baro-altimeter, and triaxial magnetometer, Wendel et al. [3] demonstrated that GPS outages of up to 50 seconds could be successfully managed with minimal attitude drift. However, significant errors in vehicle position (on the order of 30 m) were reported within the first 10 seconds of GPS signal loss. While these navigation strategies do provide some level of defense against brief signal interruption, additional techniques must be employed to achieve sustained operation under such GPS-denied conditions. Recently, several vision based strategies for indoor navigation have been adapted to fit micro vehicles [7–9]. Using a single downward looking camera coupled with an onboard IMU, Blösch et al. [8] developed and tested a MAV quadrotor capable of navigating through unknown, indoor environments by means of a 3-D point map constructed using a simultaneous localization and mapping (SLAM) algorithm. Also, Bachrach et al. [9] developed a similar quadrotor vehicle using a scanning laser range sensor, onboard IMU, and similar visual SLAM techniques fused together with an extended Kalman filter. Unfortunately, vision-based navigation of micro vehicles requires relatively large computational power and complex sensing ability to drive such algorithms.

In contrast to strictly airborne or land-based micro vehicles, several new hybrid configurations specifically designed to explore rough, uneven terrain have been developed [10–14]. These specialized vehicles offer numerous advantages over more conventional concepts given their unique ability to hop over large obstacles or gullies and also loiter on the ground for extended periods of time while conserving energy. Likewise, many of the challenges associated with indoor micro vehicle

navigation are analogous to those encountered with pedestrian tracking in urban settings. A large collection of research exists which explores various methods for indoor pedestrian tracking including vision-based systems, foot-mounted IMUs, and radio frequency identification (RFID) with the latter relying on a network of pre-installed RFID tags and variations in received signal strength (RSS) indicators to deduce specific information about the subject's location, speed, and direction of travel [15–19]. For example, Ruiz et al. [15] used a foot mounted IMU combined with RFID and an extended Kalman filter to track the movements of an individual within an office building to within less than 3 m error. Also, Yun et al. [17] used a foot-mounted IMU and tri-axis magnetometer combined with gait pattern recognition, zero velocity updates, and magnetic heading corrections to achieve similar results.

The work reported here investigates the feasibility of obtaining an accurate navigation solution for the special case of a micro hopping vehicle using only a simple, low-cost inertial sensor suite. By taking advantage of the unique dynamics of the hopping vehicle, a new algorithm has been developed capable of navigating with reasonable accuracy through unknown, GPS-denied environments using one triaxial MEMS accelerometer and one triaxial MEMS gyroscope. The sensor processing algorithm automatically detects vehicle take-off and landing and subsequently integrates the kinematic IMU equations only for the short time period while the vehicle is airborne. A novel technique of inter-hop data processing and sensor bias recalibration is also used to help minimize drift errors and improve estimation accuracy. The algorithm has been experimentally tested to assess the accuracy of the estimated navigation solution by comparing it with vehicle location measurements from an optical motion capture system with favorable results.

2 Vehicle Description

This section describes both the prototype vehicle and integrated onboard electronics used for all experiments reported in this paper.

2.1 Micro Hopping Rotochute

The micro hopping rotochute is a slightly smaller realization of a new hybrid vehicle originally proposed by Beyer and Costello [10, 11]. Designed to robustly operate within complex, irregular terrain, the micro hopping rotochute features a pair of coaxial counter-rotating propellers which, when powered in short bursts, enable the vehicle to propel itself upwards hopping from one location to the next. Its spherical external shape and low center of gravity enable the vehicle to passively reorient itself to an upright position while simultaneously protecting the rotors from any unwanted contact during ground maneuvers. Independent control of both the upper and lower rotor speeds enables precise orientation control about its vertical axis. Directional control is achieved via a small mass offset fixed to the vehicle's outer cage which creates both an initial tilt prior to takeoff and an additional pitching moment during flight to further accelerate the vehicle toward its intended target. A schematic diagram of the micro hopping rotochute is shown in Fig. 1.

The prototype vehicle used during experimental tests is formed from a 16.5 cm diameter spherical cage constructed using 1.2 mm diameter nitinol wire. The vehicle base and motor support structure are fabricated from extruded ABS plastic using a rapid prototyping machine. Total vehicle weight is 75.4 g. A pair of 13.8 cm diameter rotors powered by two independent 5 g brushless outrunner electric motors and two 6 A electronic speed controllers (ESC) provide the necessary thrust needed for takeoff. Maximum thrust of the combined rotor system is approximately 100 g. Electrical power is supplied to the vehicle from a two-cell, 300 mAh lithium polymer (LiPo) battery. Control is provided using a 2.4 GHz transmitter and six channel receiver.

2.2 GINA Wireless IMU

The inertial sensing suite used onboard the micro hopping rotochute is the Guidance and Inertial Navigation Assistant (GINA) developed by Mehta and Pister [20]. The GINA mote, shown in Fig. 2, is a small wireless IMU comprised of two separate triaxial MEMS accelerometers and one triaxial MEMS gyroscope. Also packaged within the 26.6 mm by 21.6 mm GINA platform is a 16 bit Texas Instruments MSP430F2618 microprocessor and an Atmel AT86RF231 2.4 GHz wireless transceiver. During operation, all sensors are sampled at a frequency of 333 Hz and data packets are transmitted back to an Atmel AVR RZUSBstick base station (Fig. 2) attached to a laptop computer where a custom Python script is used to read, interpret, and record all incoming data. Table 1 details the specifications for each inertial sensor used onboard the GINA mote. Note that low-pass filters have been enabled in hardware for each sensor to help minimize noise. Additional sensors including magnetometers and GPS receivers have also been adapted to fit the GINA platform but were not considered for this experiment.

3 Navigation Algorithm

The navigation algorithm presented within this paper is based upon selective integration of the vehicle kinematic equations of motion using data acquired from two onboard inertial sensors. A simple technique of inter-hop sensor bias recalibra-

tion is included such that a reasonably accurate navigation solution is obtained with minimal drift. Mathematical modeling of the inertial sensors and vehicle kinematics is presented next followed by a detailed description of the sensor processing algorithm.

3.1 Gyroscope Model

The gyroscope model depends on several key sensor parameters to relate the components of the raw gyroscope output, denoted ω_x^* , ω_y^* , and ω_z^* , to the body frame angular velocity vector $\vec{\omega}_{B/I}$ shown in Eq. (1). Denoting the body frame components of the vehicle angular velocity vector as p , q , and r , the gyroscope sensor model is given in equation Eq. (2):

$$\vec{\omega}_{B/I} = p\vec{I}_B + q\vec{J}_B + r\vec{K}_B \quad (1)$$

$$\begin{Bmatrix} \omega_x^* \\ \omega_y^* \\ \omega_z^* \end{Bmatrix} = \begin{Bmatrix} B_{Gx} \\ B_{Gy} \\ B_{Gz} \end{Bmatrix} + \begin{Bmatrix} N_{Gx} \\ N_{Gy} \\ N_{Gz} \end{Bmatrix} + [\mathbf{S}_G][\mathbf{T}_{SB}] \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \quad (2)$$

where B_{Gi} and N_{Gi} represent the i^{th} component of the gyroscope bias and noise vectors, respectively. Additionally, \mathbf{T}_{SB} represents the transformation matrix from the body to the sensor reference frame, and \mathbf{S}_G is the gyroscope cross-axis sensitivity and scaling matrix as defined in Eq. (3).

$$\mathbf{S}_G = \begin{bmatrix} S_{Gx} & C_{Gx}^y & C_{Gx}^z \\ C_{Gy}^x & S_{Gy} & C_{Gy}^z \\ C_{Gz}^x & C_{Gz}^y & S_{Gz} \end{bmatrix} \quad (3)$$

Although the measured vehicle angular velocity is subject to noise and bias offset, details regarding sensor error mitigation are discussed in later sections.

3.2 Accelerometer Model

Similar to the gyroscope, the accelerometer model also depends on several key sensor parameters including bias, scale factor, cross-axis sensitivity, and misposition, to relate the raw accelerometer output (a_x^* , a_y^* , and a_z^*) and the linear acceleration of point P at which the IMU is located. Denoting the acceleration of point P as the vector $\vec{a}_{P/I}$ with body frame components a_{Px} , a_{Py} , and a_{Pz} , as shown in Eq. (4), the general accelerometer sensor model is provided in Eq. (5):

$$\vec{a}_{P/I} = a_{Px}\vec{I}_B + a_{Py}\vec{J}_B + a_{Pz}\vec{K}_B \quad (4)$$

$$\begin{Bmatrix} a_x^* \\ a_y^* \\ a_z^* \end{Bmatrix} = \begin{Bmatrix} B_{Ax} \\ B_{Ay} \\ B_{Az} \end{Bmatrix} + \begin{Bmatrix} N_{Ax} \\ N_{Ay} \\ N_{Az} \end{Bmatrix} + [\mathbf{S}_A][\mathbf{T}_{SB}] \left(\begin{Bmatrix} a_{Px} \\ a_{Py} \\ a_{Pz} \end{Bmatrix} - g \begin{Bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{Bmatrix} \right) \quad (5)$$

where B_{Ai} and N_{Ai} denote the i^{th} component of the accelerometer bias and noise vectors, respectively, \mathbf{S}_A is the accelerometer cross-axis sensitivity and scaling matrix shown in Eq. (6), g is the gravitational constant, and θ and ϕ are the vehicle pitch and roll angles, respectively. Note the trigonometric shorthand notation: $c_x \equiv \cos x$, $s_x \equiv \sin x$, $t_x \equiv \tan x$.

$$\mathbf{S}_A = \begin{bmatrix} S_{Ax} & C_{Ax}^y & C_{Ax}^z \\ C_{Ay}^x & S_{Ay} & C_{Ay}^z \\ C_{Az}^x & C_{Az}^y & S_{Az} \end{bmatrix} \quad (6)$$

Using Eq. (5), the linear acceleration of point P with respect to the inertial frame is easily computed from the raw accelerometer data given the current vehicle tilt angles ϕ and θ . Furthermore, the vehicle mass center acceleration $\vec{a}_{CG/I}$ is related to the acceleration at point P as shown in Eq. (7).

$$\vec{a}_{CG/I} = \vec{a}_{P/I} + \vec{\alpha}_{B/I} \times (\vec{r}_{P \rightarrow CG} + \vec{\delta}_A) + \vec{\omega}_{B/I} \times (\vec{\omega}_{B/I} \times (\vec{r}_{P \rightarrow CG} + \vec{\delta}_A)) \quad (7)$$

where $\vec{\alpha}_{B/I}$ is the vehicle angular acceleration vector. Additionally, the position vector extending from the IMU point P to the vehicle mass center is denoted as $\vec{r}_{P \rightarrow CG}$, and $\vec{\delta}_A$ is the small accelerometer misposition vector. Although the measured vehicle acceleration is subject to sensor noise and bias offset, details regarding sensor error mitigation are discussed in later sections.

3.3 Vehicle Kinematics

Quaternions are used in place of Euler angles to avoid singularity problems when computing vehicle orientation during ground maneuvers. The corresponding kinematic equations of motion are provided in Eqs. (8) and (9):

$$\begin{Bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{Bmatrix} = [\mathbf{T}_{IB}] \begin{Bmatrix} a_{CGx} \\ a_{CGy} \\ a_{CGz} \end{Bmatrix} \quad (8)$$

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \quad (9)$$

where a_{CGx} , a_{CGy} , and a_{CGz} denote the body frame components of the vehicle mass center acceleration, and the matrix \mathbf{T}_{IB} shown in Eq. (10) represents the rotational transformation matrix from the body to the inertial reference frame using quaternion form. Note that Eq. (8) is a second order linear differential equation governing the vehicle translational kinematics.

$$\mathbf{T}_{IB} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (10)$$

3.4 Sensor Processing Algorithm

The basic algorithm used to process data received from the onboard IMU is shown in Fig. 3. Beginning from some known location and heading, a piece-wise navigation solution is constructed by post-processing all data acquired from the immediately preceding hop and subsequently updating the estimated vehicle state before any further hops are initiated. In contrast to continuous integration of the IMU data stream, this technique of selective integration and inter-hop data processing offers several key advantages over other real-time, recursive algorithms including the ability to detect and compensate for slight changes in gyroscope sensor bias. Also, a variety of off-line filtering techniques can be applied to help minimize the effects of sensor noise including zero phase error filters and linear regression analysis.

Starting from rest, the sensor processing algorithm proceeds as follows. First, a 1-2 second delay is needed prior to takeoff such that a short segment of static sensor data is acquired. Using this portion of static sensor data, gyroscope bias for the subsequent hop is computed using the mean value acquired from each sensor axis. Note that time averaged sensor readings are used to minimize any possible noise contamination. Next, a brief throttle pulse is sent to the vehicle to initiate a new hop. Each hop typically results in 1-3 seconds of sustained flight during which the onboard inertial sensors are continuously sampled and the corresponding timestamped data stored for later processing. Once ground impact is detected, evidenced by an abrupt change in vehicle acceleration, data acquisition is halted as the vehicle attempts to passively reorient itself to an upright position. Once a static state is reached, defined by sufficiently small changes between consecutive sensor samples, post-processing of the previously recorded flight data begins with numerical integration of the measured vehicle

angular velocity. The resulting orientation time history is then used to transform the measured accelerometer data from the body to the inertial reference frame using the transformation matrix \mathbf{T}_{IB} defined in Eq. (10). Following transformation, the accelerometer data is first filtered or curve fit to minimize noise and then numerically double integrated. Lastly, the navigation solution is then updated using the newly estimated vehicle location, and the entire process is repeated until the desired waypoint is reached.

Although accurate orientation estimation is critical to precise vehicle localization, yaw angle estimation is inherently difficult without some form of external heading reference or specialized sensor. In contrast to both vehicle roll and pitch which are integrated only during flight, vehicle heading is integrated continuously through ground impact until a sufficiently static state is reached. As a result, the only available estimate of vehicle heading following each hop is simply this final integrated value. Similarly, pre-hop vehicle roll and pitch angles are not explicitly measurable quantities given the limited onboard sensor suite and must be known prior to angular velocity integration. Fortunately, this problem of unknown initial vehicle tilt is easily overcome by neglecting any change in accelerometer bias between consecutive hops and directly solving for both vehicle roll and pitch (ϕ and θ) using the time-averaged static accelerometer data and the accelerometer sensor model presented in Eq. (5).

3.5 Extended Kalman Filter

In addition to the sensor processing algorithm described in the previous section, an extended Kalman filter (EKF) may also be used to estimate vehicle state from discrete inertial sensor measurements. Although several different forms of the EKF exist for state estimation, one such example is briefly presented here and will be used for comparison purposes in later sections. In this application, a hybrid extended Kalman filter [21] was selected which features continuous models of plant dynamics, observation, and covariance propagation combined with discrete-time equations for gain computation and measurement updates. The kinematic differential equations describing vehicle motion are of the form:

$$\dot{\vec{x}}(t) = f(\vec{x}, t) + \vec{w}(t) \quad (11)$$

$$\vec{y}(t) = h(\vec{x}, t) + \vec{v}(t) \quad (12)$$

where \vec{w} is the process noise vector and \vec{v} is the measurement noise vector. Note that both process noise and measurement noise are assumed to be zero mean Gaussian random sequences with diagonal covariance matrices \mathbf{Q} and \mathbf{R} , respectively.

The plant dynamics, represented by the nonlinear vector function $f(\vec{x}, t)$, are shown in Eqs. (13)-(17):

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = [\mathbf{T}_{IB}] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad (13)$$

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \quad (14)$$

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \begin{Bmatrix} a_{CGx} \\ a_{CGy} \\ a_{CGz} \end{Bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \quad (15)$$

$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \quad (16)$$

$$\begin{cases} \dot{a}_{CGx} \\ \dot{a}_{CGy} \\ \dot{a}_{CGz} \end{cases} = \begin{cases} 0 \\ 0 \\ 0 \end{cases} \quad (17)$$

where u , v , and w represent the body frame components of vehicle translational velocity. Note the dynamics in Eqs. (16) and (17) imply that current estimates of vehicle angular velocity and mass center acceleration are simply held fixed between measurement updates from the onboard IMU.

Propagation of the error covariance matrix \mathbf{P} is performed according to Eq. (18) where \mathbf{F} is the Jacobian linearization of the plant model $f(\vec{x}, t)$ shown in Eq. (19).

$$\dot{\mathbf{P}}(t) = [\mathbf{F}(t)][\mathbf{P}(t)] + [\mathbf{P}(t)][\mathbf{F}(t)]^T + [\mathbf{Q}] \quad (18)$$

$$\mathbf{F}(t) = \frac{\partial f(\vec{x}, t)}{\partial \vec{x}} \quad (19)$$

Computation of the Kalman gain matrix \mathbf{K} is performed using Eq. (20) where \mathbf{H} is the Jacobian linearization of the system measurement model $h(\vec{x}, t)$ shown in Eq. (21).

$$\mathbf{K}(t) = [\mathbf{P}(t)][\mathbf{H}]^T \left[[\mathbf{H}][\mathbf{P}(t)][\mathbf{H}]^T + [\mathbf{R}] \right]^{-1} \quad (20)$$

$$\mathbf{H} = \frac{\partial h(\vec{x}, t)}{\partial \vec{x}} = \begin{bmatrix} [\mathbf{0}]_{6 \times 10} & [\mathbf{I}]_{6 \times 6} \end{bmatrix} \quad (21)$$

State and error covariance updates, denoted ${}^+\vec{x}$ and ${}^+\mathbf{P}$, are calculated as shown in Eqs. (22) and (23):

$${}^+\vec{x}(t) = \vec{x}(t) + [\mathbf{K}(t)] \left\{ \vec{z}(t) - [\mathbf{H}]\vec{x}(t) \right\} \quad (22)$$

$${}^+\mathbf{P}(t) = \left[[\mathbf{I}] - [\mathbf{K}(t)][\mathbf{H}] \right] [\mathbf{P}(t)] \quad (23)$$

where \vec{z} is the six element measurement vector comprised of vehicle angular velocity and mass center acceleration as measured by the onboard IMU.

4 Results

To quantify the performance of the proposed navigation algorithm, a series of tests were conducted in which the estimated navigation solution was compared with exact vehicle location measurements obtained using a 3-D optical motion capture system. The test vehicle and experimental setup used for all studies reported in this paper are shown in Figs. 4 and 5, respectively. Note that small spherical retroreflective markers are attached to the outer frame of the test vehicle to facilitate position and orientation tracking during in-flight and ground maneuvers. The optical motion tracking system consists of 12 wall-mounted cameras that emit and collect visible infrared light at up to 2000 frames/s. Optical correlation techniques are employed in the system software to locate in 3-D space the position of each marker to within 1 mm accuracy. Using this information, time-stamped vehicle position and orientation measurements are obtained in real-time.

4.1 Example Hop

A single example hop is first considered to illustrate the trajectory estimation accuracy and overall mechanism of the proposed navigation algorithm. For this example, a short throttle pulse lasting 1.25 s was used to initiate hop. Figures 6 and 7 show the resulting 3-D trajectory and overhead cross range versus range time history of the vehicle measured using the optical motion-capture system. During flight, a maximum height of approximately 0.40 m was achieved with a total horizontal distance traveled of 1.12 m and 0.21 m in range and cross range, respectively. Ground impact was detected 1.79 s after takeoff followed by several short bounces and oscillations as the vehicle attempts to passively upright itself. As a result of vehicle bounce following landing, a minimum radial distance of 0.19 m was measured between the vehicle impact point and its final at rest location.

Also shown on Fig. 7 are the corresponding estimated vehicle trajectories calculated using three different data post-processing schemes. The first method is simply direct integration of the transformed vehicle acceleration data using a fourth order Adams-Bashforth multistep integration routine [22]. Second, the transformed vehicle acceleration along the \vec{I}_I and \vec{J}_I axes is fit using least squares to a second order polynomial function prior to integration such that any sensor noise contamination is minimized. Higher order polynomial functions were also considered but provided little or no improvement in position estimation accuracy. To further compensate for sensor noise and to serve as a comparison model for the previous two direct integration techniques, the final data processing scheme employs an extended Kalman filter (EKF) as described in Sec. 3 to estimate vehicle orientation and position from the onboard inertial sensor measurements. Although filter performance is highly dependent on the selection of several tuning parameters, estimated values for process and measurement noise variance were 0.05 and 0.015 rad^2/s^2 for the vehicle angular dynamics, and 0.5 and 0.23 m^2/s^4 for the translational dynamics. Also, the filter error covariance matrix was initialized to zero although minimal improvement in performance was noted when using higher values. In each case, the estimated trajectories correlate well with the measured vehicle motion resulting in final positional errors of 0.026 m, 0.031 m, and 0.032 m for the direct integration, polynomial fit, and EKF data processing schemes, respectively. Note these error values are computed based on the minimum radial distance between the final measured and estimated vehicle locations.

Figures 8 and 9 show the measured and estimated orientation time histories using both direct integration and the EKF. In contrast to vehicle acceleration, measured angular velocity is not polynomial fit prior to integration. For this example hop, the measured vehicle tilt just prior to takeoff was approximately zero before reaching a maximum value of 10 deg and -5 deg in pitch and roll, respectively. Maximum error between the measured and estimated vehicle pitch and roll prior to ground impact was less than 2.4 deg for both direct integration and the EKF. Similarly, estimated vehicle heading closely matches that of the measured data for the in-flight portion of the hop; however, 6.9 deg error is evident once the vehicle has landed and returned to an upright, static state. As expected, the amount of estimation error is slightly higher in yaw considering the fact that vehicle heading must be integrated from takeoff through ground impact. Consequently, the rapid changes in vehicle attitude following ground impact, noted in Fig. 9, combined with possible sensor saturation and other nonlinear effects make accurate yaw angle estimation difficult without some form of reference for heading determination.

4.2 Multi-Hop Navigation Accuracy Study

To assess the accuracy of the proposed navigation algorithm, 20 separate test sequences, each consisting of 10 consecutive hops, were performed. Although the proposed navigation algorithm is capable of estimating position for a sequence of dissimilar vehicle trajectories, a 1.25 s throttle pulse was used for simplicity to initiate each hop in all tests. No attempt was made to actively control vehicle heading except to ensure that each hop remained inside the optical motion capture volume at all times. The tabulated results for all 20 test sequences are provided in Table 2. Note that error values reported for each test represent the minimum radial distance between the measured and estimated vehicle positions following completion of the last hop in each sequence.

On average, the proposed algorithm was able to estimate the position of the vehicle to within 0.70 m based on a mean traveled distance of approximately 11 m. Standard deviations of 0.249 m, 0.269 m, and 0.271 m were calculated using the direct integration, polynomial fit, and EKF data processing schemes, respectively. Note the EKF approach does improve estimation accuracy by as much as 16% when compared to the other two direct integration techniques; however, the increased computational cost required to support such an algorithm must be considered when intended for use in low-power embedded systems. The average bounce displacement following landing in each test was 0.116 m with a standard deviation of 0.076 m. For demonstration purposes, Fig. 10 provides an overhead view of the estimated and measured vehicle trajectory for an example 10 hop sequence. Also shown for comparison in Fig. 10 is the estimated vehicle position obtained from continuous integration of the inertial sensor data between several consecutive hops. Note that nearly 10 m error is evident after just 10 seconds of continuous integration.

Table 3 details the vehicle orientation estimation accuracy for each hop in all 20 test sequences (200 hops in total). The maximum error between the measured and estimated vehicle pitch and roll was 9.14 deg and 9.22 deg for the direct integration and EKF data processing schemes, respectively. Also, the maximum error in pre-hop estimation of vehicle tilt angles (ϕ and θ) calculated using the time averaged static accelerometer data was 3.17 deg with a mean error and standard deviation of

0.04 deg and 0.93 deg in pitch and 0.15 deg and 1.05 deg in roll, respectively. Furthermore, yaw estimation accuracy was considered on both an individual hop basis and in the combined sense following completion of each 10 hop sequence. The maximum yaw angle error for any single hop was found to be just over -22 deg with a mean error of -0.81 deg and a standard deviation of 5.77 deg for the direct integration case. Note that minimal improvement in yaw estimation accuracy is achieved when using the EKF. In the combined sense, maximum yaw angle estimation error is significantly higher at approximately -34.3 deg and -36.6 deg for direct integration and the EKF cases, respectively. This degradation in yaw estimation accuracy following multiple consecutive hops comes at no surprise when considering that the resulting heading error from two consecutive hops could, in some cases, add together and double or simply sum to zero. However, it must be noted that despite the inherent difficulties in vehicle heading estimation, a reasonably accurate navigation solution is still achieved. In a more practical application, some form of heading reference (i.e. magnetometer feedback, vision, etc.) is recommended to ensure the resulting error growth is sufficiently bounded.

5 Conclusions

The work reported here explores the feasibility for obtaining an accurate navigation solution for a micro hopping vehicle using only one triaxial MEMS accelerometer and one triaxial MEMS gyroscope. By exploiting the unique dynamics of the micro hopping rotochute, a new algorithm was developed based on selective integration of the inertial sensor data and inter-hop data processing and sensor bias recalibration. Additionally, a series of tests were conducted in which the results from the estimated navigation solution were compared with those measured using a 3-D optical motion capture system. On average, vehicle position error was just over 0.70 m following a sequence of 10 consecutive hops. In order to minimize the effects of sensor noise, polynomial fit expressions for vehicle horizontal acceleration and Kalman filtering were also investigated and found to provide at best 16% accuracy improvement in the final estimated vehicle position. Overall, the proposed navigation algorithm has demonstrated that obtaining reasonably accurate estimates of vehicle localization in complex, GPS-denied environments is possible using only inexpensive, lightweight inertial sensors.

References

- [1] Liu, J., Li, R., Niu, X., and Qiao, L., 2006. "MEMS-based Inertial Integrated Navigation Technology for Micro Air Vehicles". In AIAA Guidance, Navigation, and Control Conference and Exhibit, pp. 1–8.
- [2] de La Parra, S., and Angel, J., 2005. "Low-Cost Navigation System for UAV's". *Aerospace Science and Technology*, **9**(6), Sept., pp. 504–516.
- [3] Wendel, J., Meister, O., Schlaile, C., and Trommer, G. F., 2006. "An Integrated GPS/MEMS-IMU Navigation System for an Autonomous Helicopter". *Aerospace Science and Technology*, **10**(6), Sept., pp. 527–533.
- [4] Oh, S.-M., 2010. "Multisensor fusion for autonomous uav navigation based on the unscented kalman filter with sequential measurement updates". In 2010 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 217–222.
- [5] Lei, X., Liang, J., Wang, S., and Wang, T., 2008. "An integrated navigation system for a small uav using low-cost sensors". In 2008 IEEE International Conference on Information and Automation, pp. 765–769.
- [6] Winkler, S., Buschmann, M., Kruger, L., Schulz, H.-W., and Vorsmann, P., 2005. "Multiple sensor fusion for autonomous mini and micro aerial vehicle navigation". In TENCON 2005 2005 IEEE Region 10, pp. 1–6.
- [7] Schlaile, C., Meister, O., Frietsch, N., Keler, C., Wendel, J., and Trommer, G. F., 2009. "Using natural features for vision based navigation of an indoor-vtol mav". *Aerospace Science and Technology*, **13**(7), pp. 349–357.
- [8] Blösch, M., Weiss, S., Scaramuzza, D., and Siegwart, R., 2010. "Vision based mav navigation in unknown and unstructured environments". In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 21–28.
- [9] Bachrach, A., de Winter, A., He, R., Hemann, G., Prentice, S., and Roy, N., 2010. "Range - robust autonomous navigation in gps-denied environments". In 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 1096–1097.
- [10] Beyer, E., and Costello, M., 2009. "Performance of a Hopping Rotochute". *International Journal of Micro Air Vehicles*, **1**(2), pp. 121–137.
- [11] Beyer, E., and Costello, M., 2009. "Measured and Simulated Motion of a Hopping Rotochute". *Journal of Guidance, Control, and Dynamics*, **32**(5), pp. 1560–1569.
- [12] Fiorini, P., and Burdick, J., 2003. "The development of hopping capabilities for small robots". *Autonomous Robots*, **14**(2/3), pp. 239–254.
- [13] Morrey, J., Lambrecht, B., Horchler, A., Ritzmann, R., and Quinn, R., 2003. "Highly Mobile and Robust Small Quadruped Robots". In Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03), pp. 82–87.
- [14] Stoeter, S., Rybski, P., Gini, M., and Papanikolopoulos, N., 2002. "Autonomous Stair-Hopping with Scout Robots". In 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 721–726.

- [15] Jiménez Ruiz, A. R., Seco Granja, F., Prieto Honorato, J. C., and Guevara Rosas, J. I., 2011. “Accurate pedestrian indoor navigation by tightly coupling foot-mounted imu and rfid measurements”. *IEEE Transactions on Instrumentation and Measurement*, **PP**(99), pp. 1–12.
- [16] Robertson, P., Angermann, M., and Krach, B., 2009. “Simultaneous Localization and Mapping for Pedestrians Using Only Foot-Mounted Inertial Sensors”. In Proceedings of the 11th International Conference on Ubiquitous Computing (UbiComp '09), ACM Press, p. 93.
- [17] Yun, X., Bachmann, E. R., Moore, H., and Calusdian, J., 2007. “Self-Contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules”. In 2007 IEEE International Conference on Robotics and Automation, IEEE, pp. 2526–2533.
- [18] Foxlin, E., 2005. “Pedestrian tracking with shoe-mounted inertial sensors”. *IEEE Computer Graphics and Applications*, **25**(6), pp. 38–46.
- [19] Hide, C., Botterill, T., and Andreotti, M. “Low cost vision-aided IMU for pedestrian navigation”. In 2010 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), IEEE, pp. 1–7.
- [20] Mehta, A., and Pister, K. S. J., 2010. “Warpwing: A Complete Open Source Control Platform for Miniature Robots”. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5169–5174.
- [21] Stengel, R., 1994. *Optimal Control and Estimation*. Dover Publications.
- [22] Kreyszig, E., 2005. *Advanced Engineering Mathematics*, 9th ed. John Wiley.

List of Tables

1	Inertial sensor specification for GINA wireless IMU	12
2	Vehicle position error comparison and general statistics for all 20 hop sequences	13
3	Vehicle orientation accuracy comparison and general statistics	14

Table 1. Inertial sensor specification for GINA wireless IMU

Sensor Type	Accelerometer (Analog)	Accelerometer (Digital)	Gyroscope (Digital)
Manufacturer	STMicroelectronics	Kionix	InvenSense
Model	LIS344ALH	KXSD9-1026	IGT-3200
Configuration	3-axis	3-axis	3-axis
Full-Scale Range	$\pm 2g$	$\pm 8g$	$\pm 2000^\circ/s$
Noise	$50 \mu g/\sqrt{Hz}$	$750 \mu g/\sqrt{Hz}$	$0.03^\circ/s/\sqrt{Hz}$
Low-Pass Filter	44 Hz	50 Hz	42 Hz
Operating Temp.	$-40 / 80^\circ C$	$-40 / 85^\circ C$	$-40 / 85^\circ C$

Table 2. Vehicle position error comparison and general statistics for all 20 hop sequences

Test No.	Total Dist. (m)	Mean Error (m)		
		Original Accel.	2 nd Order Poly.	EKF
1	10.307	0.841	0.793	0.524
2	12.943	0.577	0.548	0.435
3	12.076	0.490	0.591	0.252
4	12.852	1.258	1.348	1.069
5	11.440	0.564	0.829	0.619
6	10.734	0.346	0.338	0.384
7	12.234	0.626	0.714	0.668
8	11.177	0.530	0.256	0.747
9	9.056	0.603	0.628	0.620
10	11.475	1.358	1.331	0.246
11	11.904	0.702	0.766	0.944
12	10.799	0.504	0.537	0.806
13	9.967	0.901	0.675	0.520
14	11.006	0.706	0.758	0.814
15	11.717	0.809	0.723	0.941
16	10.277	0.444	0.398	0.140
17	9.005	0.749	0.726	0.425
18	10.874	0.648	0.668	0.146
19	10.424	0.681	0.593	0.665
20	9.917	0.727	0.546	0.820
Average	11.009	0.703	0.688	0.589
Std. Deviation	1.103	0.249	0.269	0.271

Table 3. Vehicle orientation accuracy comparison and general statistics

	Orientation Error (deg)		
	[Min, Max]	Average	Std. Deviation
<i>Pre-hop Tilt:</i>			
Pitch	[-2.66, 2.81]	0.04	0.93
Roll	[-2.65, 3.17]	0.15	1.05
<i>Direct Integration:</i>			
Pitch	[-8.96, 7.02]	0.27	1.01
Roll	[-9.05, 9.14]	0.16	1.30
Yaw	[-21.75, 15.14]	-0.81	5.77
Yaw (combined)	[-34.29, 27.71]	-7.91	14.96
<i>Kalman Filter (EKF):</i>			
Pitch	[-9.22, 7.03]	0.24	1.02
Roll	[-9.15, 9.20]	0.16	1.28
Yaw	[-22.04, 14.99]	-1.04	6.08
Yaw (combined)	[-36.61, 26.58]	-10.77	15.55

List of Figures

1	Micro hopping rotochute schematic diagram	16
2	AVR RZUSBstick base station (left) with GINA wireless IMU (right)	17
3	Navigation algorithm flowchart	18
4	Micro hopping rotochute prototype vehicle	19
5	Indoor flight facility with optical motion capture system [11]	20
6	Altitude versus cross range versus range for example hop	21
7	Cross range versus range for example hop	22
8	Roll and pitch angle versus time for example hop	23
9	Heading (yaw) angle versus time for example hop	24
10	Cross versus range for example 10 hop sequence	25

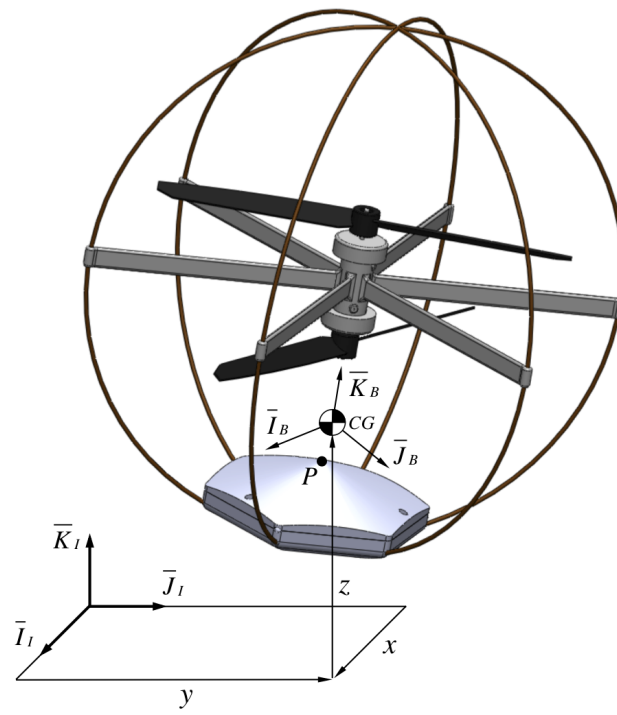


Fig. 1. Micro hopping rotochute schematic diagram

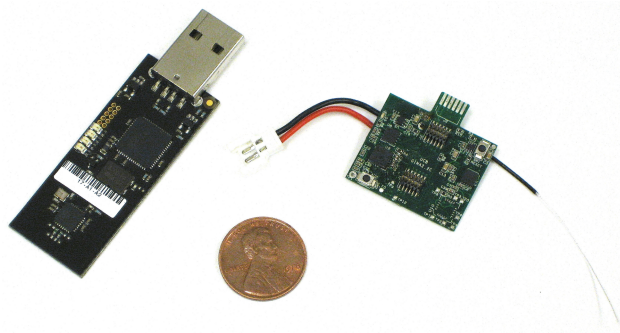


Fig. 2. AVR RZUSBstick base station (left) with GINA wireless IMU (right)

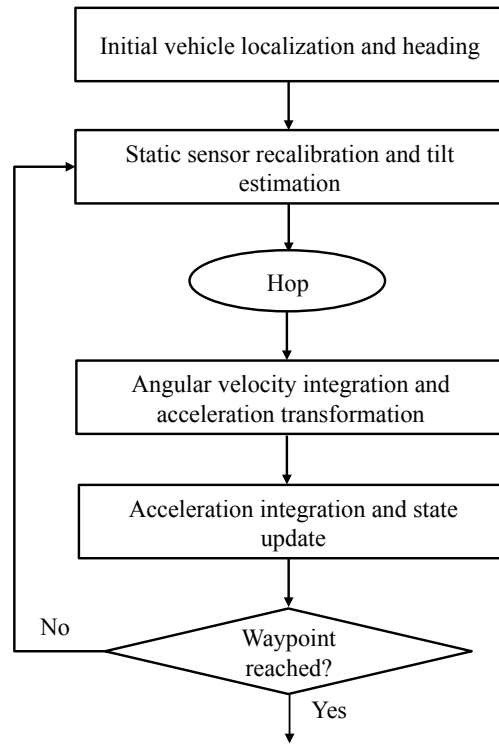


Fig. 3. Navigation algorithm flowchart

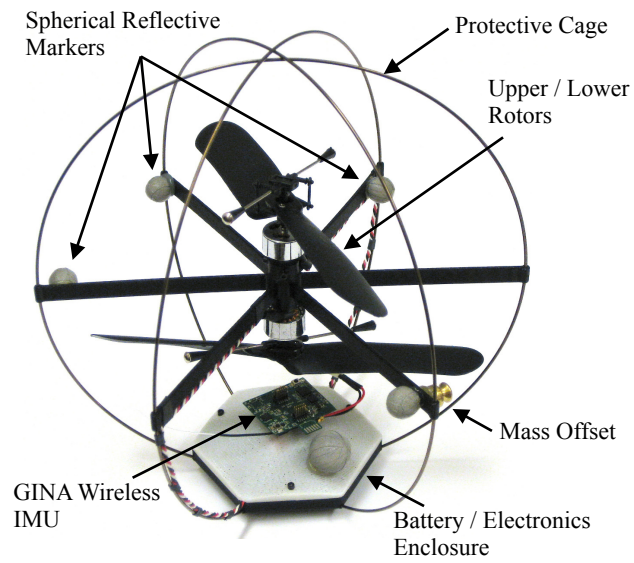


Fig. 4. Micro hopping rotochute prototype vehicle

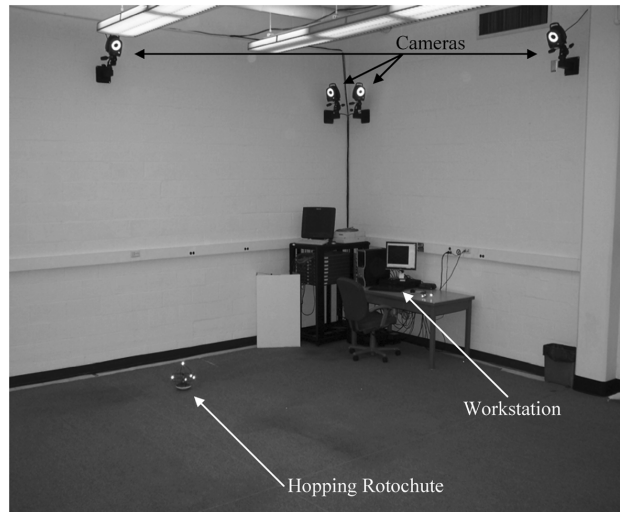


Fig. 5. Indoor flight facility with optical motion capture system [11]

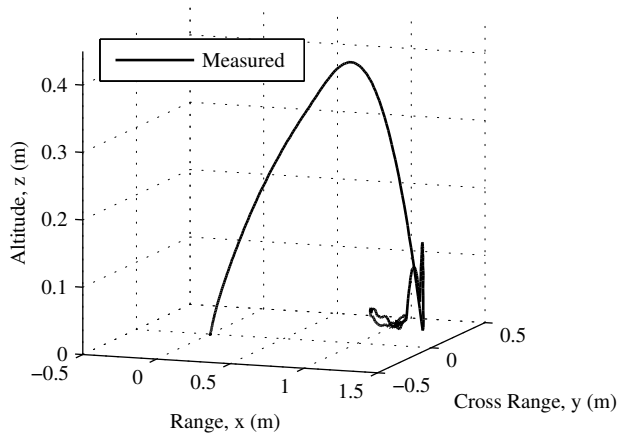


Fig. 6. Altitude versus cross range versus range for example hop

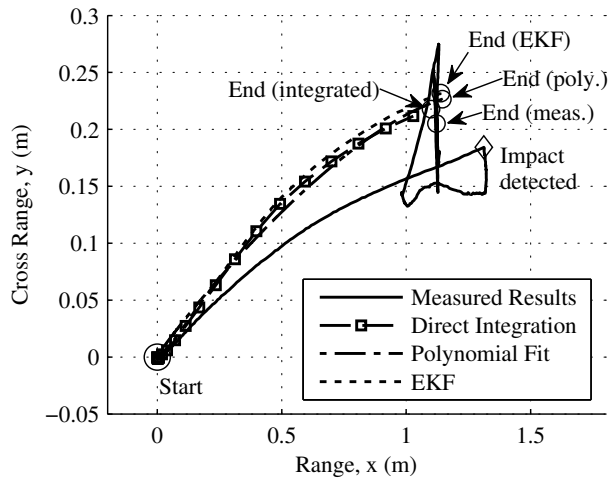


Fig. 7. Cross range versus range for example hop

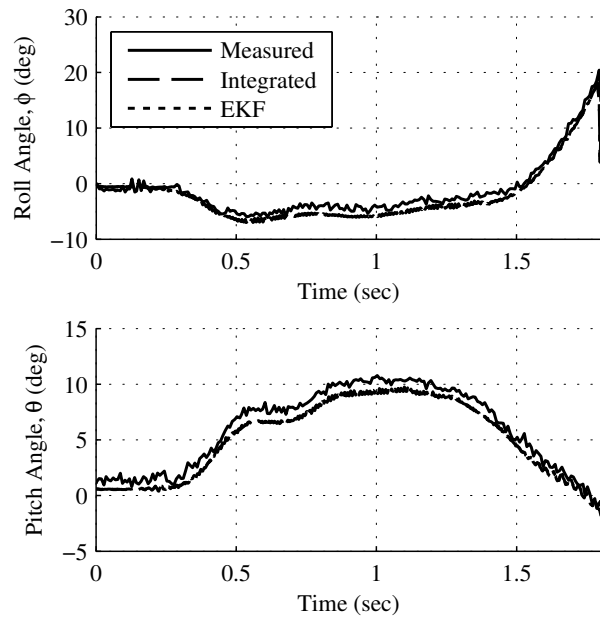


Fig. 8. Roll and pitch angle versus time for example hop

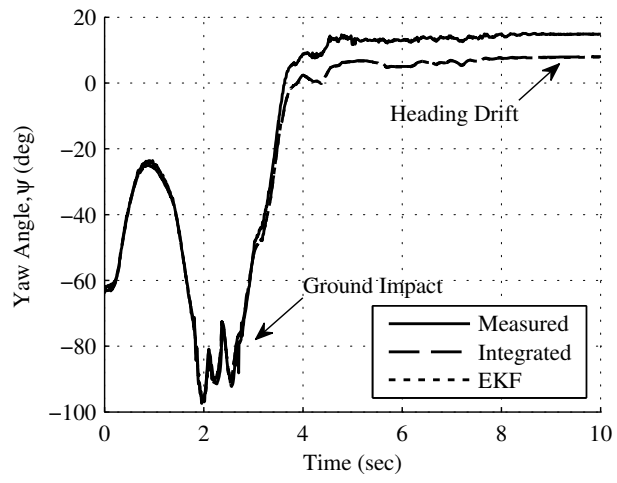


Fig. 9. Heading (yaw) angle versus time for example hop

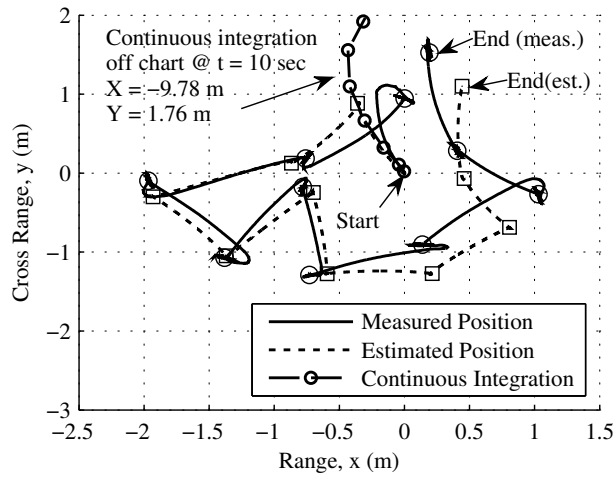


Fig. 10. Cross versus range for example 10 hop sequence