UNIVERSITY OF CALIFORNIA,

IRVINE

A Machine Learning-Based State Detection and Correction Algorithm

THESIS

submitted in partial satisfaction of the requirements

for the degree of

MASTER OF SCIENCE

in Mechanical and Aerospace Engineering

by

Gaetano Antonellis

Thesis Committee:
Professor Tryphon Georgiou, Chair
Professor Andrei Shkel
Assistant Professor Haithem Taha

2021

DEDICATION

To

Those who have come before me,

Those who have yet to come

And those with whom I share this wonderful experience

If I have seen further it is by standing on the shoulder of giants

(many)

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my committee chair, Professor Tryphon Georgiou, whose students gravitate towards discovery, academia, and fellowship in part because of his love for pedagogy. People like him bring life to the academic environment; without the 'soft' skills of leadership, teamwork, and communication, the work requiring the technical skills cannot get off the ground.

I would like to thank my committee members, professors Andrei Shkel and Haithem Taha. The topics of this paper drew inspiration from the steps taken toward our understanding of these systems by Professor Shkel and the MicroSystems Laboratory. Professor Haithem Taha carries an energy that awakens the academic spirit and encourages one to put one foot in front of the other in their journey.

As these scholars pass down the wisdom and knowledge given to them, and as they lead efforts in expanding this pool of a resource, they empower those yet to come by sharing their contributions.

In addition, a thank you to Michael Kazek of the California Maritime Academy, and to all of those who have reminded me to use the breaths between sprints to laugh.

My academic journey at the University of California, Irvine has been stimulating, exciting, and peculiar. Engaging in a graduate degree almost completely remotely when it is traditionally experienced in person has removed me from the academic environment. While I regret being

physically isolated from the innovation and excitement found at places of research, and while I miss the camaraderie and immersion I found during my undergraduate experience at the California Maritime Academy, I am grateful that my professors have allocated the time and effort to create a learning environment that has lent well to a variety of student lifestyles. Specifically, I am grateful to have had the opportunity afforded to me by the university and by my employer to continue my academic adventure while seeing firsthand the practical application of nearly every topic covered in the courses that have constituted this MS.

ABSTRACT

A Novel Multi-Class State Detection Algorithm

by

Gaetano Antonellis

Master of Science in Mechanical and Aerospace Engineering

University of California, Irvine, 2021

Tryphon Georgiou, Chair

One popular example of state detection and correction is the Zero Velocity Update (ZUPT), which is often used in pedestrian navigation to limit error due to sensor noise and biases. Existing methods, while they have become both efficient and effective, are often either computationally heavy or are not designed to handle the detection of a variety of states.

This study proposes an expression to be used in a state detection algorithm, produced in part by the information provided by machine learning models. The intent for the state detection algorithm is the classification of inertial sensor data into various important trajectory states. In this study, the goal was to distinguish zero-velocity states and left- and right-handed orbital trajectories out of sets of IMU data using the simplest effective methods.

First, some existing methods are evaluated in their utility as candidates as a singular data feature of a classifier for the expected navigational trajectories. None of the methods examined were suitable candidates for implementing a piecewise classifier for the trajectory states. In turn, machine learning techniques are employed to identify important data features for the proposed classifier. To assist in the implementation of a method using these key data features, a novel test statistic for a likelihood test similar to the Stance Hypothesis Optimal Detection method is proposed. Finally, the proposed classifier is tested using collected IMU data, and the potential of incorporating such a state detector is explored.

The results show that the novel state detector is able to classify the expected trajectory states with over 95% accuracy. In the data used to test the sensor, it did not experience any false classifications or missed detections. While there is still work to be done to improve the robustness and applicability of the algorithm, this study acts as a proof of concept, providing a starting point for future work.

INTRODUCTION

Most general consumer devices employ inexpensive, compact Micro Electro-Mechanical Systems (MEMS) as sensors. The state estimate of an Inertial Navigation System (INS) using these sensors is subject to noise, bias, and other errors that are inherent to the sensors due to vibration, manufacturing tolerances, and resolution [1]. In navigation systems, these errors propagate over time, resulting in the divergence of position and velocity estimates. Rather than spending resources to implement a more accurate (expensive, in size, weight, and power) Inertial Measurement Unit (IMU), one option is to offer software solutions to assist the navigational estimate. When an IMU enters a trajectory containing expected data profiles, a state detection algorithm can be implemented to make corrections to an INS solution, mitigating future error propagation [2]. The corrections to these inherent errors in the IMU due to noise, drift, and biases in the sensor depend on reliable state detection and correction algorithms.

The Zero Velocity Update (ZUPT) is one such software technique used to limit the accumulation of error in an INS [2]. When an IMU referencing a local coordinate system is stationary on Earth, it experiences only the local gravity vector and the rotation of the Earth. Because the IMU enters this known state, ZUPT can be used to correct the velocity and angular rate estimates of the INS. ZUPT is often employed in pedestrian navigation - parts of the human foot periodically stop their motion relative to the navigational coordinate frame - and is comparable to robotic odometry if the step size can accurately be estimated. This can be critical in environments where GPS-aided navigation is not available.

Studies have evaluated the parameterization of the expressions used in ZUPT to increase the state detection performance [3]. Another strategy to improve the detection of these zero-velocity states is to employ data fusion with additional sensors that are often used in inertial navigation systems such as magnetometers to aid the heading estimate [4], heel-mounted pressure sensors to help aid the state detection [3], and other similar sensors. Some of these studies have proven to increase the accuracy of the position estimate of the INS, though they add complexity to the system with the additional sensor(s). Because of this added complexity to the system, the option is not considered in this study.

Finally, machine learning techniques have been implemented in a number of studies to classify a variety of human movements, such as stance phases as in [5] and swim strokes as in [6] from IMU data. Each of these comes with its own utility and can be used for applications such as direct improvements to the ZUPT for GPS-denied navigation, health monitoring, and athletic training. This study draws from their insights on picking data features to form an expression for a more simple classifier.

Limitations of Existing Methods; Objectives

Although ZUPT and similar techniques have become powerful tools for pedestrian navigation, there are still many limitations to the development and implementation of this family of state detection algorithms:

1. ZUPT has utility in determining when an IMU is stationary, though for a system that might not stop moving for a long period of time, the velocity and position estimates for an INS that relies solely on MEMS IMUs will quickly diverge.

2. For a system that moves with constant velocity, some ZUPT detection algorithms experience false detections - the IMU data will have small values of magnitude and variance. An example of this is when a pedestrian is stationary on a moving escalator, where a false positive state detection may cause divergence in the state estimate. For these conditions, ZUPT is limited in its utility.

3. Although Machine Learning algorithms can be powerful tools for the classification of data instances such as the ZUPT detector in [7], they often rely on a computationally heavy framework. While this may not be a problem for some systems, other systems do not have the processing bandwidth to train or predict classes using a machine learning algorithm, especially if an INS requires timely decision-making.

This study attempts to address some of these limitations by building on a family of common ZUPT detection algorithms to systems that undergo other expected trajectories - orbits of relatively constant radius and velocity, about an axis parallel to the IMU's Z-axis. Though pedestrian and automobile navigation sequences do not often exactly match these trajectories, the detection and classification of these trajectories has its own utility - the localization of these turn events. Vehicles navigating a known road network can be localized based on the distances between turns and whether the turn made is left or right in direction. Similar concepts were explored in [9] using a particle filter to track pedestrian motion using smartphone sensors and in [10] using Kalman filtering and a probabilistic state detection algorithm with a separate detector for each motion class. This study attempts to improve upon this last technique by using a piecewise multi-class classifier to detect similar motion classes. For vehicles that don't navigate a road network, implementation of a simple algorithm to handle the detection of dynamic trajectories lays down a framework for creating detection algorithms for motion sequences of greater complexity.

In section 1, existing detection algorithms for ZUPT are evaluated for their utility in detecting the zero-velocity states and the orbital trajectories. The magnitude and variance of the detection methods' test statistic features are examined. It is determined that each of the existing test statistics are unfit to be used as a standalone data feature of a multi-class classifier, though they offer insight about data features that might support such a classifier. It is concluded that a new test statistic or a different framework must be generated to construct a classifier for the expected trajectories in consideration.

In section 2, machine learning techniques are used to identify important data features that support classification models built to act as interim state detectors. IMU data was collected and then processed using Principal Component Analysis to determine an appropriate number of data features to consider for the test statistic expression. Then, machine learning models of various classes are trained and tested to validate the feasibility of using a reduced feature space and to determine feature importance.

In section 3, the important features from the machine learning models are used to design a few test statistics. Through evaluation and iterative design, the study begins to converge on an effective expression. The intention is to use such a test statistic as the data feature for a simple multi-class classifier. Each of them is evaluated against IMU data sets containing zero-velocity and orbital trajectory states.

In section 4, a simulated INS is built in order to explore the potential benefits of the novel state detection algorithm. The detector correctly distinguishes the expected trajectory states from other motion sequences. While further effort is required to make any practical use of a state correction algorithm using the proposed expression, the framework has been set in anticipation of its implementation.

SECTION 1: Existing ZUPT State Detectors

In this section, IMU data is collected and the likelihood test statistic values for detecting the zero-velocity states and the orbital trajectories in this data are evaluated. It is determined that the existing test statistics are unfit to be the data feature for a multi-class classifier and it is concluded that a new test statistic or a different framework must be generated to construct such a classifier.

Many studies have been performed to improve the performance of zero-velocity state detection algorithms through parameterization [11]. By examining different data parameters, such as the magnitudes and variances of the IMU data, algorithms can be effective to meter-accuracy for an unfiltered INS over navigational sequences spanning hundreds of meters. Among the most effective of these ZUPT detection methods is the Stance Hypothesis Optimal dEtector (SHOE), which evaluates the accelerometer and gyro data. Instantaneous accelerometer readings, $y_k^a \in R^3$, and gyroscope readings, $y_k^\omega \in R^3$, for time instance k are used in these calculations, though because of sensor noise, sometimes z_n^a and $\boldsymbol{z}_n^\omega$, the measurements over a time epoch W from time $k = n$ to time $k = n + W - 1$, are considered. Alternatives to the SHOE method include the Acceleration Moving Variance (MV) detector, which examines the variance of the acceleration data, the Acceleration Magnitude (MAG) detector, and the Angular Rate Energy (ARE) detector [3].

Each of these state detectors uses a different expression based on data provided by the IMU, typically to be used in a binary hypothesis test. Given a predetermined threshold γ and the test statistic L, the hypothesis that the system is experiencing a zero-velocity condition, H_1, is assumed to be true if: $L(z_n) = \frac{p(z_n | H_1)}{p(z_n | H_0)} \succ \gamma$ [12]. This is the exact functionality of a linear classifier in Machine Learning terminology, where the linear model is of constant value γ, determining if the test statistic representing a Maximum Likelihood indicator is sufficiently low to assume the zero velocity condition. This study attempts to use the test statistic $T(z_n)$ as a data feature for a multi-class classifier. The proposed linear model has two values, γ_{ZUPT} and γ_{ORBIT}, where a zero-velocity condition is assumed if $T(z_n)$ is less than γ_{ZUPT}, and where an orbital trajectory about an axis parallel to the IMU's Z-axis is assumed if $T(z_n)$ is greater than γ_{ORBIT}.

In practice, γ_{ZUPT} and γ_{ORBIT} may need to be adjusted depending on the body mechanics of the subject, the tempo of their gait, and the mounting of the sensor, among other things. An adaptive threshold is explored in [13], which aids the detection algorithm in making the correct classification on a wider range of step tempos. This was done by fitting a logarithmic function to the step tempo to determine the appropriate threshold, a means of employing the empirical data to inform future regression or classification.

The dataset used in this section is three separate data collections from a TDK InvenSense Smartbug IMU strung together - a sequence where the IMU is carried by a human and transported in an orbit by robot arm, a sequence where a human carries the IMU and rotates it in an arm-extended orbit, and a sequence where the IMU is mounted to the boot of a subject walking at a pace of approximately 80 paces per minute. The robot arm is a UR5 from Universal Robots, used to create 'ideal' orbital trajectories of constant velocity and constant angular rate about the arm's main axis. In the plots, the sections of the data that contain the pedestrian motion and the orbital trajectories (for both the robot arm and the hand-carry case) are color-coded. In these calculations of the test statistic, sliding windows of .05 seconds were used in calculating mean and variance values.

The Smartbug IMU uses an ICM-42688-P, which has accelerometer noise levels of 65 microgravities per sample rate Hz for its X-axis and Y-axis measurements, and a noise level of 75 microgravities per sample rate Hz for its Z-axis measurements [14]. For data collections in this study, a sample rate of 100 Hz was used, with the sensor's standard ranges of ±16g and ±2000 degrees per second.

SHOE Detector

The Stance Hypothesis Optimal dEtector is among the top performing zero-velocity state detectors. It uses a weighted average over a time epoch of the accelerometer data (with a term accounting for the gravity vector) and the gyroscope data, based on the sensors' respective variance values:

SHOE Test Statistic: $$T(z_n^a, z_n^\omega) = \frac{1}{W}\sum_{k=n}^{n+W-1}\frac{1}{\sigma_a^2}\left|\left|y_k^a - g\frac{\bar{y}_n^a}{||\bar{y}_n^a||}\right|\right|^2 + \frac{1}{\sigma_\omega^2}||y_k^\omega||^2$$

SHOE detector test statistic values for the orbital trajectories were not significantly distinguishable from values for the movement phase of the pedestrian motion sequence. This can be seen in Figure 1 - the data points that correspond to the zero-velocity states in the pedestrian motion sequence are periodic local minima. Some of these secondary peaks have the same magnitude as the orbital trajectory data. Because of this, creating a piecewise classifier based on the SHOE test statistic alone is not a viable option for creating a state detector for these trajectories. The SHOE detector does present some merit for classifying the orbital trajectories - the amount of variance in the test statistic for the orbital trajectory IMU data is much smaller than when using the MV or MAG detectors. In the robot-transport case, the variance of the test statistic on a log scale across 0.5 seconds is .01, and in the hand-carry case this variance is 0.007. For comparison, the typical variance of a sliding window of test statistic of data from non-classified motion across a half second window is 2.25.

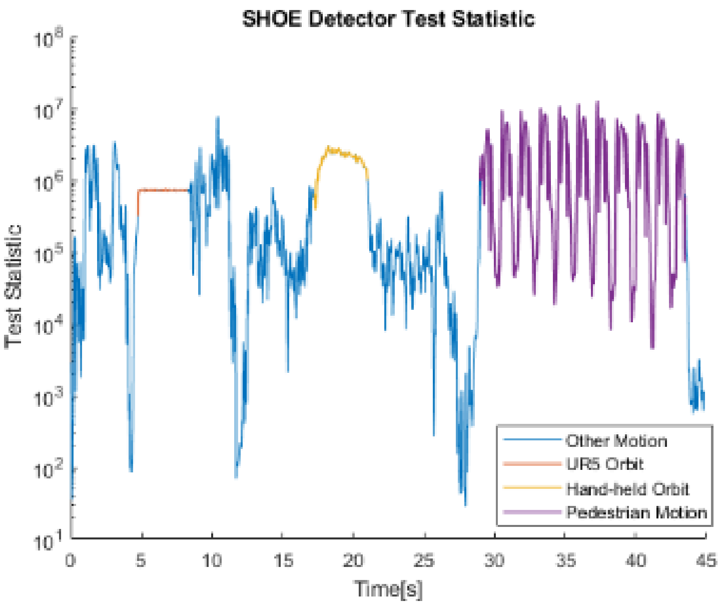


Figure 1 SHOE Test Statistic Values during IMU Navigation Sequences

MV and MAG Detectors

The MV and MAG test statistics are calculated using only the accelerometer data- the gyroscope data is not used in these calculations. This allows simple implementation of the detector since only one sensor is required (and a MEMS accelerometer also typically is less subject to noise and drift than a MEMS gyroscope). As seen in the equation below, the MV detector's test statistic uses the variance of the accelerometer data: the mean of the squared difference between each data point within the time epoch W, and the mean of all data points in the time epoch. Then, this value is scaled by the precision of the sensor, σ_a^2. Similarly, the MAG detector uses the magnitude of the accelerometer while accounting for the gravity vector.

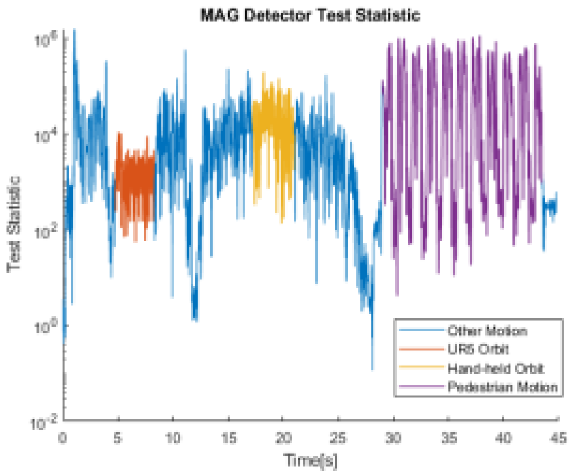
MV Test statistic:

$$T(z_n^a) = \frac{1}{\sigma_a^2 W}\sum_{k=n}^{n+W-1}\left|\left|y_k^a - \bar{y}_n^a\right|\right|^2$$

MAG Test Statistic:

$$T(z_n^a) = \frac{1}{\sigma_a^2 W}\sum_{k=n}^{n+W-1}\left(\left|\left|y_k^a\right|\right| - g\right)^2$$

When evaluating the MV test statistic and the MAG test statistic, it was found that there is almost no discernible difference between the orbital trajectory data and the non-classified navigation sequence data. For this reason, neither of these detection test statistics is feasible to use as a standalone feature for multi-class classification of the navigational trajectories in question.



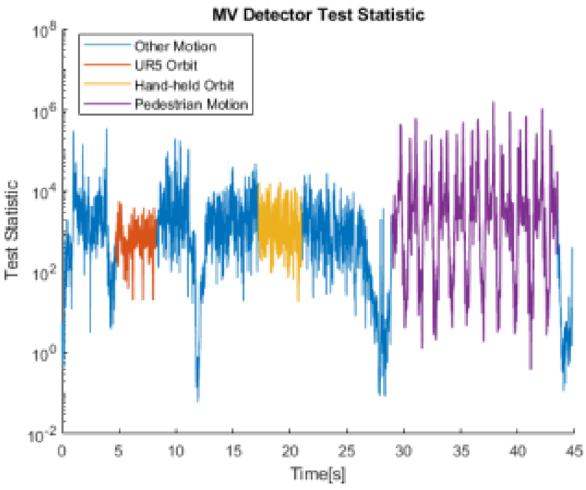


Figure 2 MAG and MV Test Statistic Values during IMU Navigation Sequences

One expectation that was held for the MV detector was an increase in the test statistic while in an orbital trajectory because such a trajectory has relatively constant values in each of the accelerometer axes. However, it seems that the noise seen by the IMU due to the vibration of the Universal Robots arm and the imperfections in the orbital trajectory when the IMU was transported by a human were enough to have a significant effect on the MV test statistic.

ARE Detector

The Angular Rate Energy detector uses just the second term of the SHOE detector:

ARE Test Statistic: $T(z_n^{\omega}) = \frac{1}{\sigma_{\omega}^2 W} \sum_{k=n}^{n+W-1} \left|\left| y_k^{\omega} \right|\right|^2$

The ARE detector has shown to be nearly as effective in detecting the zero-velocity state as the SHOE detector while only requiring the gyroscopic sensor. The ARE detector test statistic for this set of data has qualities similar to the SHOE test statistic. When the IMU is transported by the robot arm, the variance in the test statistic decreases significantly. Though not as drastically, this occurs when the IMU enters a stable orbital trajectory in the hand of a human subject. The ARE detector test statistic has more overall variance than the SHOE detector, though it does not appear to have a great difference when attempting to use it as a piecewise classifier.

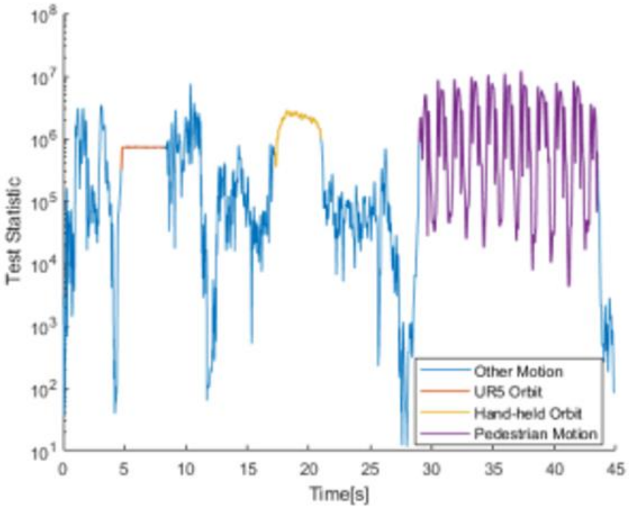


Figure 3 ARE Test Statistic Values during IMU Navigation Sequences

The existing ZUPT detectors were not designed with the intention of being used as the standalone data feature for a multi-class classifier; they were chosen with the intent to be used for a singular binary classifier with just one threshold. Although, for this study they do offer some insight on what might make a good data feature for such a model. It is apparent from the results of the MV and MAG detectors that a gyro data term will be necessary in such a test statistic, and it may also be important to retain an accelerometer magnitude term so that false positives are not detected when the system is traveling in a linear path with no rotation.

SECTION 2: Feature Extraction through Data Science

In this section, Machine learning algorithms were used to extract the important data features for classifying the IMU data. Some key parts in using machine learning tools include collecting data and preprocessing that data, engineering the feature space of the model, partitioning the data and using a subset of it to train a model(s), and using the model features and prediction results to draw inferences about the data. In this particular study, the information gained from the machine learning techniques is used to reduce the feature space. These reductions were then validated by training similar models with the new feature space, and using the models to perform predictions on IMU data in a supervised learning setting. The feature space is reduced to three key features which the models can use to classify expected trajectories with 95% accuracy.

Description of the Data Set

The data set is a compilation of accelerometer and gyroscope data collected using the Smartbug sensor, sampled at 100Hz. The data includes foot-mounted pedestrian motion, sequences where the IMU was carried by hand, and sequences where the IMU was transported by the Universal Robots UR5. Data was taken in a number of locations where the setting was surveyed so that the IMU location could be compared against ground truth, and so that the sequences could be recreated if necessary. In the pedestrian and hand-carry data sets, the motion was captured with a camera so that the data could later be labeled with the correct classes, which is an important step for machine learning. Because there was no high-speed motion capture system available, this is a possible source of error in the data set.

For the pedestrian motion data sets, the IMU was mounted to a boot as in Figure 4, where a subject walked at tempos between 75 and 90 steps per minute. A total number of 28 steps were taken. There are also some zero-velocity states that exist in the pedestrian motion datasets where the subject is standing still before and after walking. There are no orbital trajectory states that exist in the pedestrian motion datasets. Combined, these data sets offer over 3600 samples, with over 1000 instances of zero-velocity conditions.

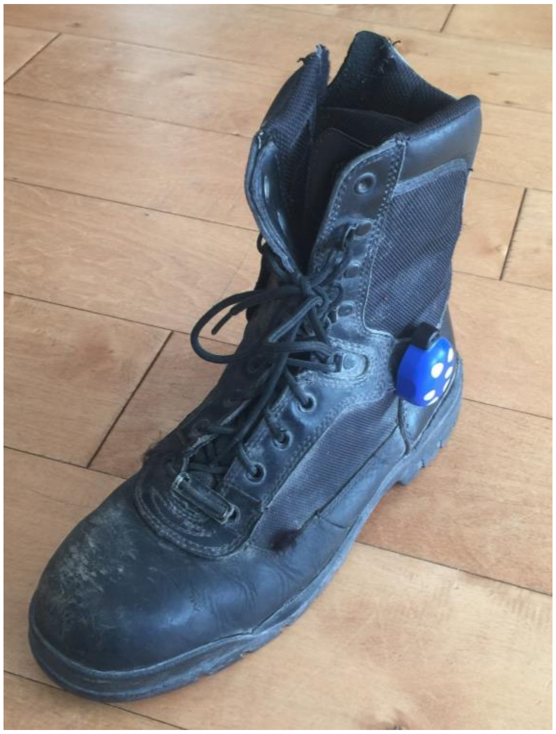


Figure 4 Mounting of the SmartBug IMU for Pedestrian Navigation

For the hand-carry data sets, the IMU was carried by hand as in Figure 5, where the subject was instructed to slide the IMU across a table, lift it off the table, walk a few paces to a designated point, turn in a circle with the hand holding the IMU extended, walk back to the table, and set the IMU down. These data sets provide a large variety of IMU motion, which offers value in training the machine learning models. Data from the hand-carry cases offer over 4820 samples, including

Figure 5 The SmartBug IMU was carried by hand in some sequences

some zero-velocity conditions and over 800 instances of hand-carried orbital trajectories

For the Universal Robots data sets, the IMU was mounted to the robot arm and the arm was commanded to turn about its main axis. The radius of the IMU trajectory while being transported by the robot arm was 1 meter, and the arm rotated at a constant rate of 120 degrees per second. These data sets include 540 degrees of robot arm rotation in each direction, for a total of 900 samples of 'ideal' orbital trajectory data, with constant velocity and angular rates. They also include some hand-carry motion sequences. The robot arm has a small amount of vibration as it moves along its path, though this did not seem to have a significant effect on the IMU data.

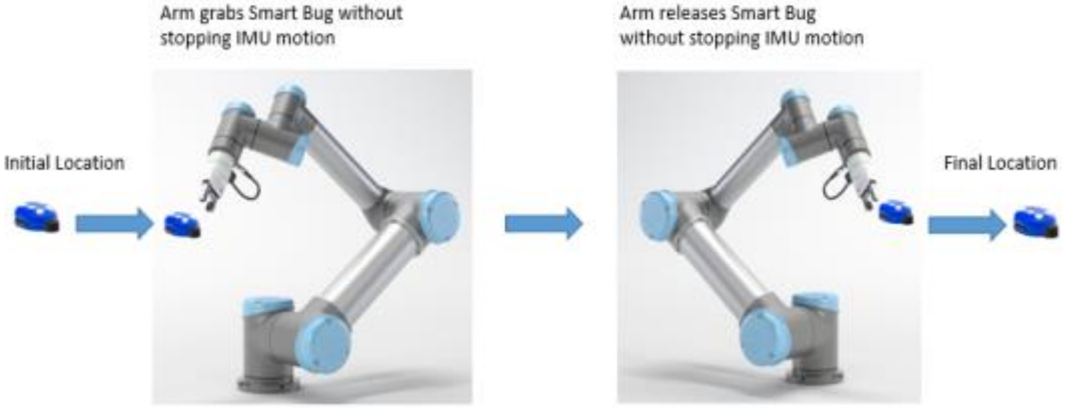

Arm grabs Smart Bug without stopping IMU motion
Arm releases Smart Bug without stopping IMU motion
Initial Location
Final Location

Figure 6 The Navigation Sequence of the SmartBug IMU

Data Preprocessing; Engineering the Feature Space

Prior to using machine learning models for regression or classification, the data must all be in the same format. This is not a large problem with multiple sets of IMU data from the same sensor. However, when importing the data, the parts from some collection sequences had to be transformed so that the IMU's axes corresponded to the subject's coordinate frame consistently. The convention used in this study is:

- Axis 1 Positive Direction: Forward
- Axis 2 Positive Direction: Starboard
- Axis 3 Positive Direction: Down
- Axis 1 Positive Rotation: Roll Right
- Axis 2 Positive Rotation: Pitch Up
- Axis 3 Positive Rotation: Yaw Right

Next, the class labels for the various trajectory states were assigned by hand. During the classification of the trajectory states, the class labels were as follows:
1 - zero-velocity condition
2 - clockwise turn
3 - counterclockwise turn
0 - other motion

Finally, MATLAB's cvpartition function was used to randomly split the data into training data and test data. 70% of the data was used for training the machine learning models, while the other 30% was used as test data to enable proper validation of the model's performance. Splitting the data randomly ensures distribution of the data classes between the training data and test data sets, and is an important piece in avoiding overfitting when using machine learning tools [15].

Machine learning algorithms can accept raw sensor data as features by which to label the instances, and they can also accept features engineered to better represent the problem at hand. Preprocessing and engineering the feature space can aid in preventing the model from over-fitting to data that is extraneous, and it can also simplify the machine learning model while maintaining performance. Engineering the feature space often consists of creating new features from combinations of existing ones, determining which features are the most important, and normalizing the features.

A number of options for engineering new data features were considered for this classification problem, including some of the following:

- The Euclidean norm of a subset of data
- The variance of a subset of the data, over a sliding window
- The effective radius of an orbital trajectory:

$$a_y = (\omega_z)^2 r \Rightarrow r = \frac{(\omega_z)^2}{a_y}$$

- The angular rate of the orbital trajectory, normalized by the total rotation magnitude:

$$\frac{y_k^{\omega_z}}{\left|\left|y_k^{\omega}\right|\right|}$$

- The acceleration term of the SHOE test statistic
  (normalized acceleration, minus the gravity vector)

$$\frac{1}{\sigma_a^2}\left|\left|y_k^a - g\frac{\bar{y}_n^a}{\left|\left|\bar{y}_n^a\right|\right|}\right|\right|^2$$

The complete list of the data features used in the machine learning models is shown below:

- X-, Y-, and Z- axis accelerometer readings
- the norm of the accelerometer readings
- X-, Y-, and Z- axis gyroscope readings
- the norm of the gyroscope readings
- the variance of each of these parameters

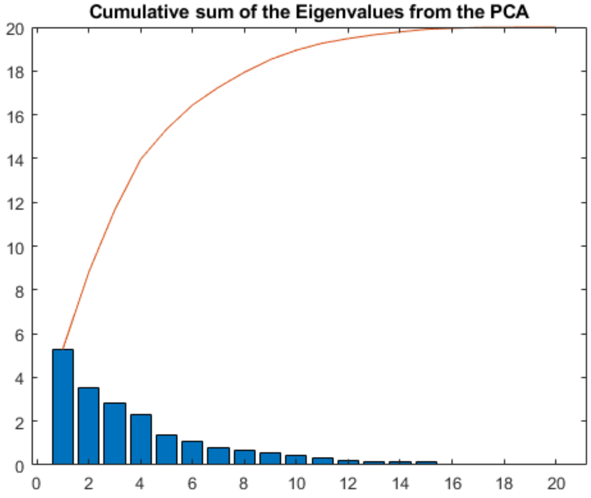
- the sign of the Z-axis gyroscope readings
- the norm of the accelerometer readings minus the gravity vector
- the angular z-axis gyroscope readings, normalized by the total rotation magnitude
- the effective radius of an orbital trajectory
- the Y-axis accelerometer readings multiplied by the square of the Z-axis gyroscope readings
  (small values of these two terms during relatively linear trajectories resulted in data with extreme variance for the effective radius feature)

A few different methods were used to perform the selection of data features for the final model. First, Principal Component Analysis (PCA) was performed on the data set with raw IMU data and the addition of the engineered features mentioned above. In order for PCA to be effective, the data was first normalized to have zero mean and a variance of 1.

$$X_{normalized} = \frac{X-\mu}{\sigma}$$

It is important to normalize the data prior to PCA since the data features with more variance will be weighed more heavily if it is not done. PCA uses a Singular Value Decomposition algorithm on the data set, and it returns the coefficients of the PCA along with the eigenvalues of the covariance matrix of the data set. The coefficients, or Principal Components, are the eigenvectors of the covariance matrix. PCA is a powerful tool that can be used to reduce the feature space of a dataset to simplify a model. However, it is easy through PCA to lose sight of which of the original features are responsible for large portions of the total variance in the dataset.

The results from the PCA were very informative regarding how many features might be optimal to use. Distinguishing the N most important eigenvectors can be done by looking at the relative magnitudes of the eigenvalues, and taking their cumulative sum relative to the total number of coefficients. The number of effective data features has been reduced to the N most important features by multiplying the data set X by the first N rows of the coefficient matrix, which represent the Eigen directions of the principal components.



When this was performed, the first 6 of 20 Eigen directions were chosen since their cumulative weight was responsible for 82% of the variance of all of the data. Since many of the terms in the feature space have lots of covariance (for example, the individual accelerometer values and the norm of the accelerometer measurement), the feature space can be effectively reduced to a few key features.

Data feature selection was completed by using the original feature space to train a decision tree model and by gathering the resulting information on feature importance. Decision tree models are valuable tools for extracting this kind of data since the trees are built by using a series of perceptrons to split the data in order to minimize entropy. By examining a decision tree model, the features most effective in reducing the entropy of the data set can be determined. In the figure below, the root of the decision tree is isolating most of the 'other movement' class based on the variance in the X-axis gyroscope data - both the zero-velocity state and the orbital trajectory states have low variance in roll rate.

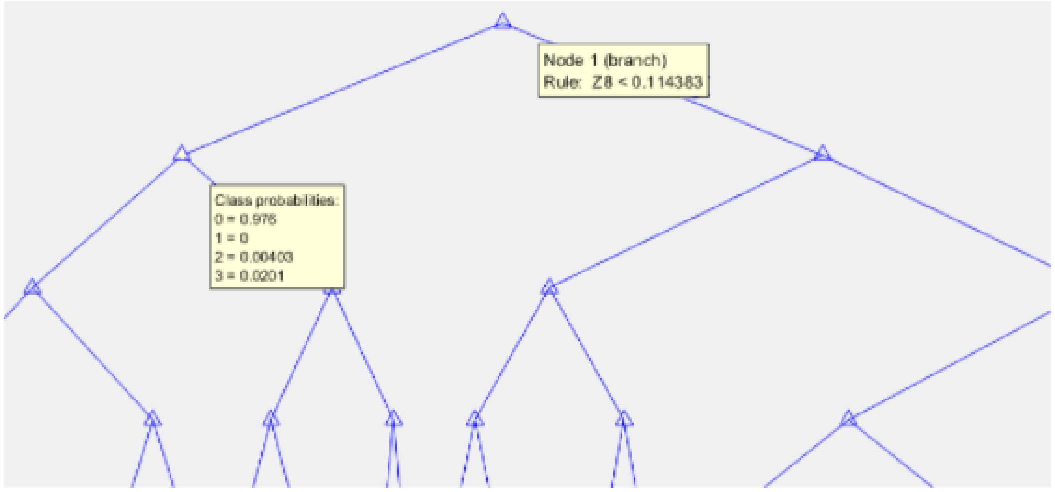


Figure 7 Part of a Decision Tree Model used to Classify the Trajectory States

Out of the feature space including the raw data and many engineered feature parameters, six features were chosen, based on the results of the PCA and the decision tree information gains. Because some of the terms were directly related through common terms, the subset was further reduced, resulting in the following feature space:

- The norm of the gyroscope measurement: $||y_k^{\omega}||$
- The variance of the gyroscope measurement: $\frac{1}{W}\sum_{k=n}^{n+W-1}||y_k^{\omega} - \bar{y}_n^{\omega}||^2$
- The Z-axis angular rate, normalized by the total gyroscope measurement: $\frac{y_k^{\omega_z}}{||y_k^{\omega}||}$

Testing the Proposed Feature Space

This new simplified data set with just three features was tested using a k-Nearest Neighbors model, a Decision Tree model, and a Support Vector Machine (SVM) model. In order to evaluate the effectiveness of the feature space the models were also trained and tested in cases with the original feature space, and also in cases where the ‘other motion’ class was not included in the data.

The models, without optimization through hyper-parameterization, performed up to 95% classification on the test data set. Below is a confusion chart from the SVM classifying the test data.

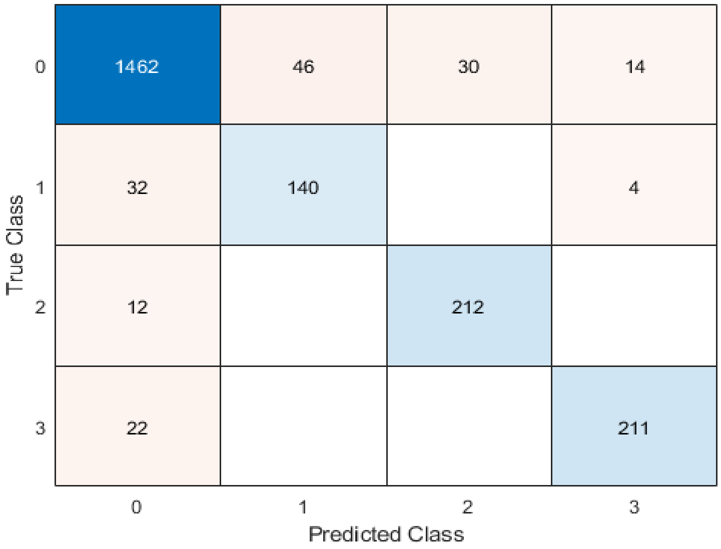


Figure 8 The Confusion Chart for a Support Vector Machine Classifier

The reduction to this feature space is intuitive when considering the trajectories that the models are to distinguish from one another: A zero-velocity condition will have small gyro measurements and accelerometer measurements. An orbit will have gyro measurements mainly composed of the z-axis measurement, and relatively small accelerometer measurements. Distinguishing left and right turns can be done using the sign of the Z-axis gyro (in practice this should be done in the combination with the sign of a velocity estimate).

SECTION 3: Test Statistic Design

In this section, a new test statistic is proposed using the key features identified from the analysis of the existing ZUPT detectors in the first section and the key features extracted from the machine learning techniques employed in the previous section. The test statistic is intended to distinguish zero velocity conditions and orbital trajectory states from other non-distinguished motion.

Formulation of a test statistic was performed by combining the results of the previous two sections. During this part of the study, it was important to maintain the convention of a unit-less test statistic (all acceleration squared terms are divided by accelerometer variance squared terms, etc.). This ensures that the test statistic scales properly when a different sensor is used. By thinking about the data seen by the IMU in the expected trajectories, a table can be formed to help narrow down the data features that will be important to include in the test statistic, which has been provided in the Appendix.

The results from Section 1 communicate that a test statistic designed to be a single data feature for a multi-class classifier for these expected trajectories using accelerometer data alone would likely not be very effective - the orbital trajectory test statistics from the MAG and MV detectors were very noisy, and were characteristic of the undistinguished motion. However, the results of the machine learning techniques from Section 2 indicated neither the magnitude nor the variance of the y-axis accelerometer data was of great importance, regardless of it being a key component in defining an orbital trajectory. For this reason, intuition said that maintaining the first term of the SHOE test statistic would be of great merit.

The results from section 2 offered a few options for the angular rate term of the test statistic. It was clear that including a z-axis gyroscope term would be imperative for detecting the orbital trajectory, so (3) was considered first. The expression first looked like this:

$$T(z_n^a, z_n^\omega) \ = \frac{1}{W} \sum_{k=n}^{n+W-1} \frac{1}{\sigma_a^2} \left\| y_k^a - g \frac{\bar{y}_n^a}{||\bar{y}_n^a||} \right\|^2 + \frac{1}{\sigma_\omega^2} \frac{y_k^{\omega_z}}{\left||y_k^\omega|\right|}$$

In order to be consistent with remaining unit-less, however, the z-axis gyro measurement would have to be cubed. In addition, the absolute value of the z-axis gyro measurements were used, since those measurements are negative when making a left-handed turn. The angular rate term of the test statistic was updated:

$$\frac{1}{\sigma_\omega^2} \frac{y_k^{\omega_z}}{\left||y_k^\omega|\right|} \Rightarrow \frac{1}{\sigma_\omega^2} \frac{\left|y_k^{\omega_z}\right|^3}{\left||y_k^\omega|\right|}$$

This state detector test statistic was evaluated in the same manner as the existing detectors, as in Section 1. The threshold used for the zero-velocity condition was $7x10^3$ and the threshold for the orbital trajectory condition was $7x10^5$. The detector successfully classified both the robot arm orbit and the hand-carry orbit, and it also successfully classified all of the zero-velocity conditions in the pedestrian motion. There is one zero-velocity condition prior to the robot-driven orbit due to the moment where the IMU was set into the grasp of the Universal Robots gripper, before the robot started moving, and an additional zero-velocity condition at the end of the hand-carry sequence when the subject stops sliding the IMU across the table. These are not misclassifications.

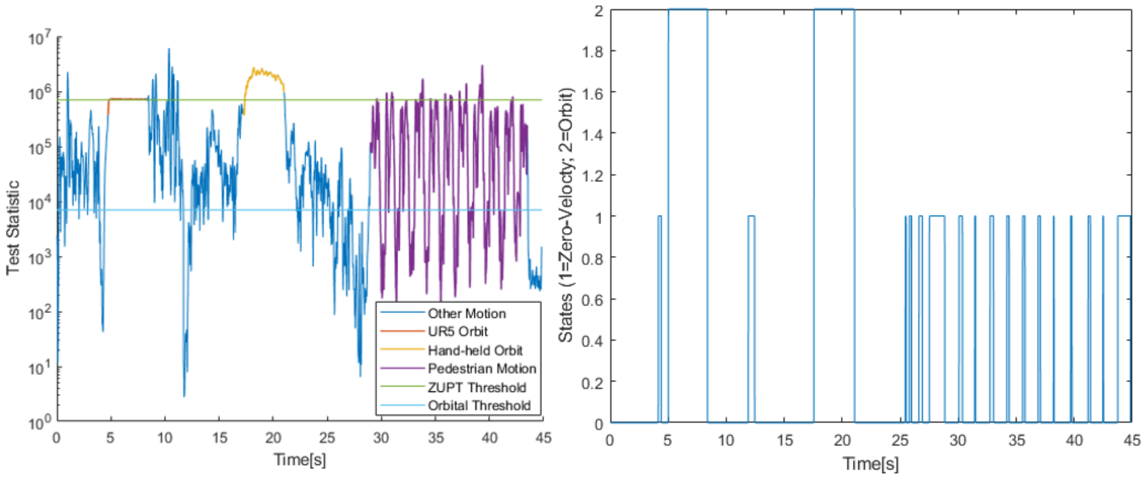


Figure 9 The Test Statistic values and State Detections using the Proposed Expression

Because of the Z-axis' contribution to the total angular rate of the IMU, the angular rate term distinguished the orbital trajectories from other test statistic data points, without becoming noisy as in the MV and MAG detector test statistics. Even though there is a larger difference in the test statistic for the orbital trajectory states and the swing phase of the pedestrian motion, the magnitude of the individual data points would be near indistinguishable for orbital trajectories of slower rotation if not for the variance in the test statistic. This family of detectors requires a number of consecutive data points to be greater than the threshold for the state to be detected; although the blue data points reading higher than the threshold in the plot below were not misclassified, it is reasonable to believe that the detector could falsely classify data as belonging to an orbital trajectory. For this detector, the number of consecutive test statistic values required for a state detection was set to 20, requiring 0.20 seconds of characteristic data to trigger a state detection.

To increase the margin between the orbital trajectory test statistic values and the non-classified test statistic values, the angular rate term was modified once again. In Section 2, all of the data was normalized when the machine learning techniques were applied - data features such as accelerations, which are typically in the range of -20 to $20m/s^2$ were normalized to the same scale as the Z-axis angular rate, normalized by the total angular rate of the IMU, which by definition is in the range of 0 to 1. This enabled the machine learning models to more effectively classify the data. In the likelihood ratio test, it helps to have the variance in the data accentuated. To increase the variance of the angular rate term of the test statistic, the Z-term was removed from the angular rate norm. In this test statistic, the angular rate term scales cubically with the Z-axis angular rate term, rather than having a nearly quadratic relationship, and it still gets penalized for large X-axis or Y-axis gyroscope readings. The angular rate term of the test statistic expression was updated again:

$$\frac{1}{\sigma_\omega^2}\frac{\left|y_k^{\omega_z}\right|^3}{\left|\left|y_k^{\omega}\right|\right|} \Rightarrow \frac{1}{\sigma_\omega^2}\frac{\left|y_k^{\omega_z}\right|^3}{\left|\left|y_k^{\omega_{xy}}\right|\right|}$$

The test statistic was once again evaluated as in Section 1. The threshold used for the zero-velocity condition remained at $7x10^3$ and the threshold for the orbital trajectory condition was set to $3x10^6$. Once again, the detector successfully classified both the robot arm orbit and the hand-carry orbit, and it also successfully classified all of the zero-velocity conditions in the pedestrian motion.

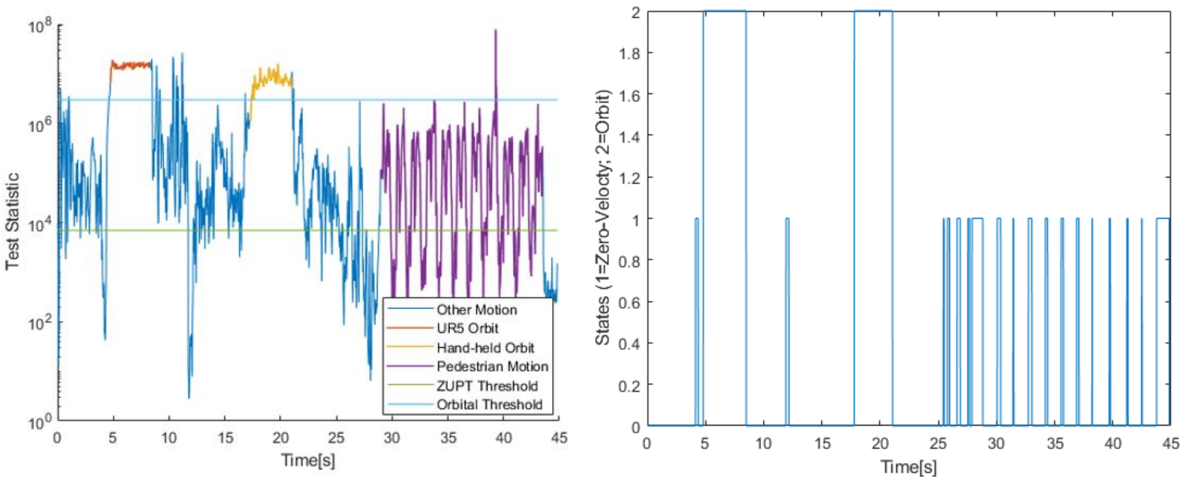


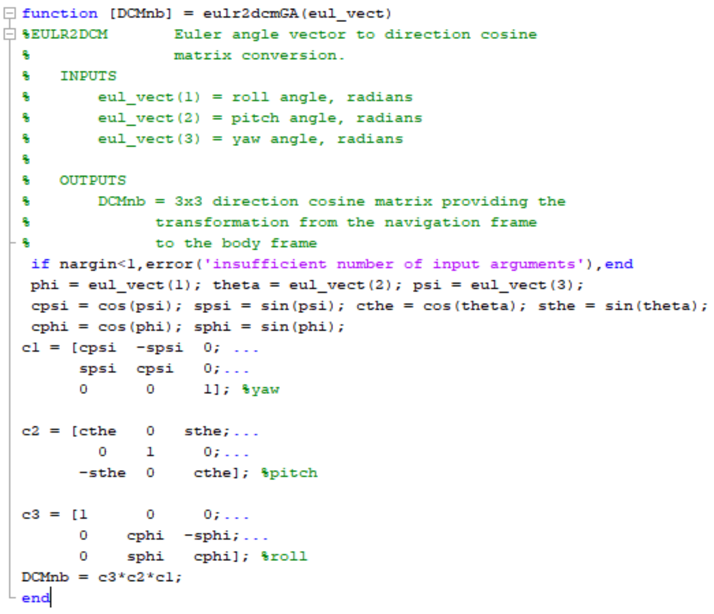
Figure 10 The Test Statistic values and State Detections using the Proposed Expression

Using this new test statistic, named the Gyroscope Axis Energy (GAE) detector for its utility in examining orbits with a specific rotational axis, there was an increased margin between the orbital trajectory test statistic values and the undistinguished motion test statistic values. Except for just a few outliers, the test statistic values of greatest magnitude belong to the orbital trajectories.

SECTION 4: Simulation and Application

In this section, the proposed expression for detection of zero velocity and orbital trajectory states was evaluated and the implications of the capability are explored. A simulated INS was built, and the state estimate of a simulated 'ideal' INS without noise or bias was used to act as ground truth. Then, for comparison, empirical data gathered using a sensor package was fed into the simulated INS employing a state correction based on the proposed algorithm.

A simulation was built in MATLAB to produce INS state estimates [16] [18]. The input of the script is a data file of the format used by the Smartbug IMU, along with some initial position details such as orientation and heading. The output of the script is the velocity, position, and attitude estimates of the INS. In the simulation, the system's navigational coordinates follow the North-East-Down (N-E-D) conventional reference system, also known as the n-frame, since the range of the navigation sequences examined are relatively local. The simulated INS uses Euler angles to express the initial attitude of the unit. It then generates a Directional Cosine Matrix, C_b^n, with a function:



```
function [DCMnb] = eulr2dcmGA(eul_vect)
%EULR2DCM        Euler angle vector to direction cosine
%                matrix conversion.
%    INPUTS
%        eul_vect(1) = roll angle, radians
%        eul_vect(2) = pitch angle, radians
%        eul_vect(3) = yaw angle, radians
%
%    OUTPUTS
%        DCMnb = 3x3 direction cosine matrix providing the
%              transformation from the navigation frame
%              to the body frame
 if nargin<1,error('insufficient number of input arguments'),end
 phi = eul_vect(1); theta = eul_vect(2); psi = eul_vect(3);
 cpsi = cos(psi); spsi = sin(psi); cthe = cos(theta); sthe = sin(theta);
 cphi = cos(phi); sphi = sin(phi);
c1 = [cpsi  -spsi  0; ...
      spsi  cpsi   0;...
      0      0     1]; %yaw

c2 = [cthe   0   sthe;...
        0    1     0;...
      -sthe  0    cthe]; %pitch

c3 = [1      0     0;...
      0    cphi  -sphi;...
      0    sphi   cphi]; %roll
DCMnb = c3*c2*c1;
end
```

Figure 11 The Function used to Transform Euler Angles to a Directional Cosine Matrix

...which is then used to transform any vector $\vec{a}$ from the body frame into the n-frame according to the following equation:

$$\vec{a}^n = C_b^n \vec{a}^b$$

This transformation is applied to the specific forces and angular rates in the body frame to put them in the context of the navigational frame as they are integrated over time to produce attitude and position estimates. The directional cosine matrix gets updated via a skew-symmetric matrix made from the rotations in the body frame relative to the navigational frame. The Euler angles get updated based on these rotations as well. The Euler angles are not used in the INS except for providing a user-friendly way to represent the orientation of the IMU in the n-frame.

IMU Navigation Sequence

In both the simulation and during the empirical data collection, the IMU's navigation sequence includes a few different types of segments:

- Stationary initialization, where sensor data is evaluated to define threshold values
- Under-defined trajectories as the IMU is carried by hand
- Zero-velocity states where ZUPT can be applied if desired
- Predefined orbital trajectories where the IMU moves in a near perfect orbit about a single axis parallel to the IMU’s Z-axis, where a correction can be applied if the trajectory is well-defined.

For the simulation of a state correction using the GAE detector, the physical setup of the lab space where the data collection occurred was modeled. It involves two tables along which to slide the IMU, two spaces where the IMU is carried by hand, and an area where a robotic arm moves the IMU in a predefined orbital trajectory. The ‘ideal’ sensor data that the IMU will likely see as it is slid, carried, and transported was similarly modeled so that it could act as noiseless ground truth. The accelerations and angular rates seen by the IMU are characteristically different in each part of the navigation sequence.

Sensor data was once again collected using a TDK InvenSense SmartBug IMU to capture the navigation sequence described above. This motion sequence is also captured in Figure 6.

As the IMU is slid along the table and carried to the robot arm, the unit accumulates some error. When the unit is carried by the robotic arm, the simulated INS calibrates the noisy, error-filled data coming from the sensor to the known motion of the IMU using an anticipated acceleration and angular rate. It does this by taking the difference between the average of what the IMU reports and what the IMU should see from the expected orbital trajectory, and giving all future data an equivalent, opposite offset. This study assumes that the trajectory dictated by the robot arm only has known angular rates, velocities, and accelerations, as if the IMU motion dictated by the robot arm was actually induced by a person or vehicle - in reality its precise location is also known.

When the unit is transported by the Universal Robots robotic arm, it is placed one meter from the axis of rotation, which is aligned with the IMU's Z-axis. The arm rotates at a rate of 120 degrees per second for a total of one and a half rotations. From the dynamics equation below, the expected Y-axis acceleration can be determined:

$$y_k^{a_y} = r\left(y_k^{\omega_z}\right)^2$$

This expected acceleration and angular rate were used to inform the state correction upon the GAE detection of an orbital trajectory.

A low pass filter was applied to the raw data to decrease the effects of noise seen by the sensor. Bias stability error was mitigated by using the mean of the data when the IMU was at rest to zero the data so that it would be more usable for the INS, similar to the effect of ZUPT. This helped to limit the effects of turn-on bias stability. This data was fed into the INS to get estimates of velocity and position data by integrating the angular rates and accelerations using the N-E-D mechanization.

The section from 4.5 seconds to 8.25 seconds is the period of time where the sensor is transported by the UR5. Here, the average roll and pitch rates are close to zero, and the accelerations in the body X-axis and Z-axis are close to zero. The variance of the Z-axis angular rate and the Y-axis acceleration are also small when the UR5 has established a constant, orbital trajectory for the IMU.

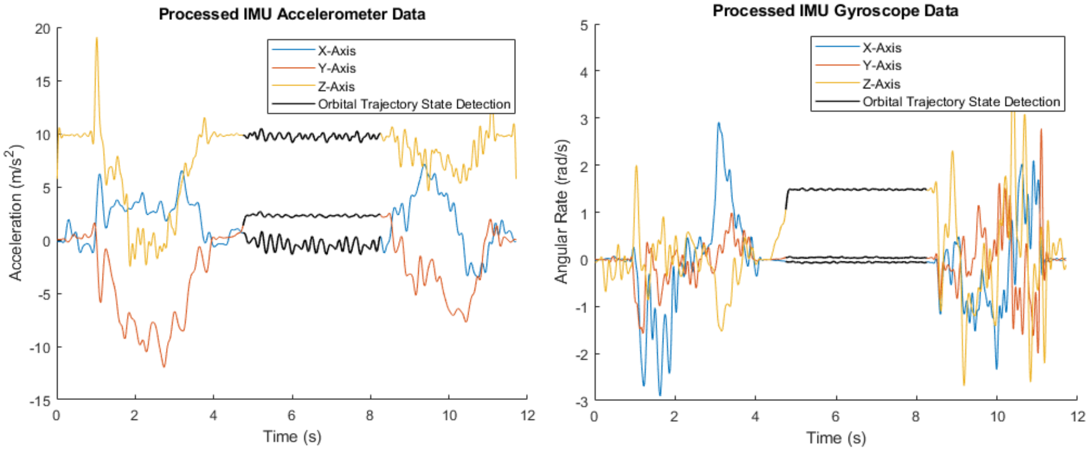


Figure 12 Processed IMU data used to validate the proposed detector

The proposed state detector was used on the IMU data collected during this navigation sequence. The detection algorithm picked up the zero-velocity condition that took place when the IMU was placed in the grasp of the robot arm, and then the orbital trajectory of the robot arm-defined motion. The plots of the test statistic and the states detected by the algorithm are displayed in Figure 14.

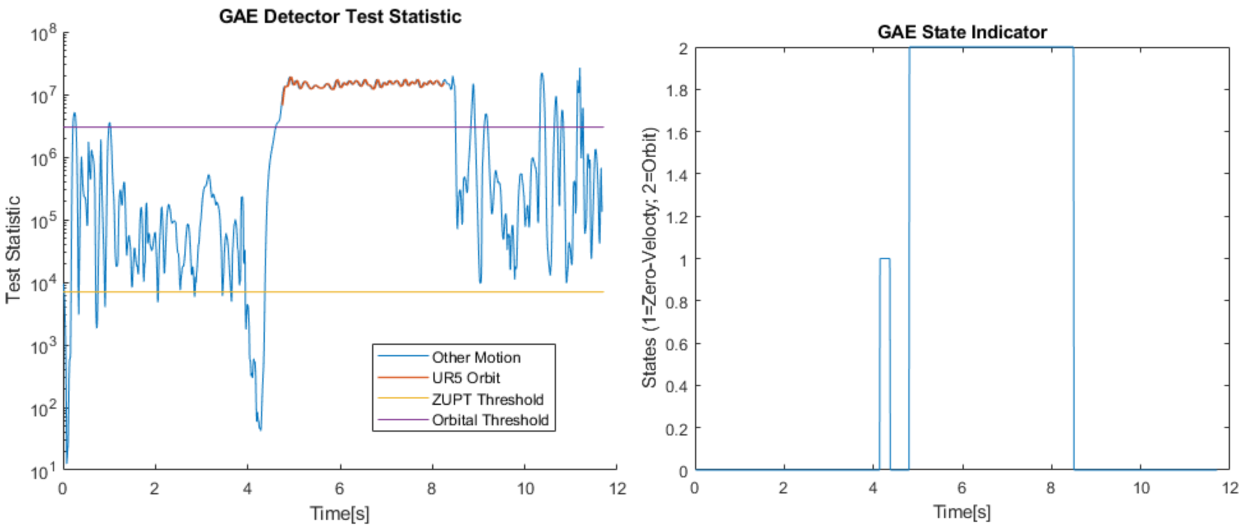


Figure 13 The Test Statistic values and State Detections using the Proposed Expression

Position estimates from the simulated INS using the SmartBug data are displayed in Figure 15. The paths of the two inertial navigation systems (with and without using a state correction based on the GAE detection) start by matching exactly, until the IMU is transported by the robotic arm. At this point, the data sets begin to diverge since one simulated INS is updated with the expected angular rates and accelerations when it recognizes the orbital trajectory state. The INS that does not get updated uses the angular rate recorded by the IMU, which has some error, causing the trajectory estimate to have a larger orbit with less total rotation.

One factor that was noticed with the Smartbug IMU was an offset in the accelerometer data after the sensor experienced a prolonged acceleration in any axis. For this reason, the path of the IMU after being transported by the robot arm is a curve, according to the INS, when in reality it was removed from the robot arm and moved in a linear path without ever stopping. One way to mitigate this may have been to make use of the GAE ZUPT detection just as the IMU was set into the grasp of the UR5.

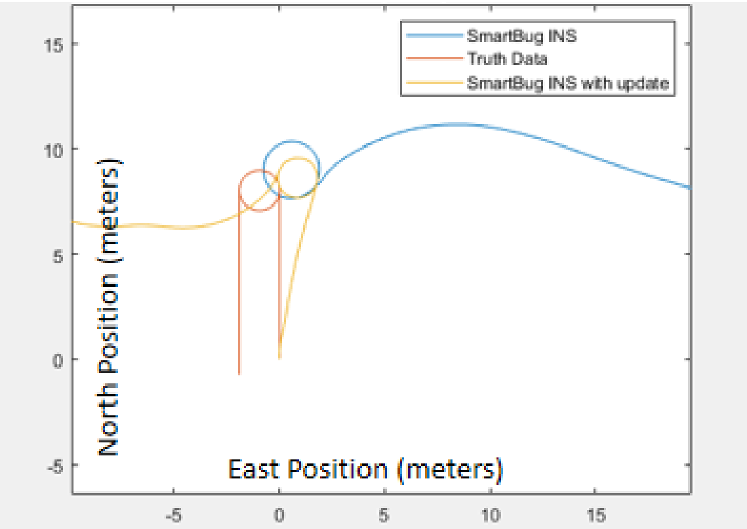


Figure 14 The 2D Trajectory Estimate of the INS in the NE Plane

The standard INS has a position estimate error of 21.5 meters. The INS data when fed the expected trajectory and provided with an update calibration has a position estimate error of 11.72 meters. The navigation sequence was split into two distinct legs for error propagation, which reset the rate of divergence. The error between the ground truth and the INS solutions in the last leg of the navigation sequence. The greatest determining factor of the error in both of these cases is the heading estimate - the distance tangent to the orbit of the final location of the simulated system that used a correction similar to ZUPT is just 1.4 meters different from that of the starting location, where the homologous error in the uncorrected INS is 4.8 meters.

A state correction using a direct feed of the expected angular rate and acceleration is just one possible application of the GAE detector. Depending on how well-defined the dynamics of the system employing the detector are, it can be used to eliminate biases and drift accumulated during a navigation sequence without the need for a zero-velocity condition. Other possible applications and improvements are further discussed in the following section.

It is important to note that these results were determined without techniques such as the Kalman filter. While the Kalman filter is not computationally heavy for a navigation system, this section's intent was to explore the effects of using the GAE detector to implement a state correction, the results of which may have been less clear if such a filter was also employed.

Recommendations for Further Improvements

The proposed GAE detector algorithm has proven to correctly classify zero-velocity conditions orbital trajectories about an axis parallel to a specific IMU axis. There are many steps that should be taken to improve the detector before implementing it in a practical application.

A Robust Detection Method

Improvements can be made to decrease the GAE detector's likelihood of false positive detections and missed detections. While its performance has not yet been thoroughly tested, it is anticipated that sustained accelerations or angular rates of sufficient magnitude could trigger the false detection of an orbital trajectory. Here are a couple of suggested modifications to prevent false detections:

- Limit the accelerometer and gyroscope terms' contributions to the value of the test statistic, so that a combination of both is necessary for an orbital trajectory detection to be triggered.
- Require a certain magnitude from the accelerometer data in order for the gyroscope term to be enabled. This would prevent a stationary, spinning system from detecting an orbital trajectory while maintaining a detector similar to the MAG detector when the gyroscope term is not enabled.
- Similarly, experiment with an axis-specific accelerometer term in the test statistic to ensure detection is only triggered by near-orbital trajectories. Specifically, use a sign

indicating function on a cross product between the acceleration and the angular rate contributing to the orbital trajectory.

- Perform a regression (which could be linear, polynomial, or inverse in nature) based on the acceleration and the angular rate contributing to the orbital trajectory. This could be used to draw a feasibility measure from the orbit radius, based on the dynamics of the system.
- Implement an adaptive threshold for the ZUPT detection, and compare the measured orbit-axis angular rate to the expected dynamics of the system when determining the likelihood of an orbital trajectory
- Expand the set of host systems for validating the algorithm from foot-mounted, hand-carried, and robot-transported sequences to include automobiles and other vehicles

Potential Applications and other Improvements

This study has already mentioned a few potential applications of the GAE detector. Here the possible expansions and practical applications of the method are reiterated. Some implications of its implementation and other improvements to the algorithm could help develop our capabilities in GPS-denied navigation, among other technical challenges:

- Detection of turns to localize a sensor within a road network
- Live re-calibration of sensors s based on a known turn rate and radius
- Similarly, live tuning of control gain based on a desired turn rate and radius
- Data fusion using additional sensors to aid the system's state estimates
- Reduce the need for the axis of rotation to be parallel to an IMU axis by using parallel component and perpendicular component projections

Of course, these applications often rely on a thorough understanding of, and rely on an accurate model of the system dynamics, especially when considering more complex motion sequences to detect. Combining these methods with an observer-based data filter may be the next step in expanding the capability.

Summary

In this study, a state detection algorithm expression is proposed to serve as the data feature for a multi-class classifier for zero-velocity conditions and orbital trajectories about an axis parallel to one of the system's IMU axes. The terms of the expression were extracted from machine learning techniques, and although the key terms for the expression developed could have been intuited from general knowledge of the trajectories, a framework has been set for developing similar expressions. The algorithm proposed has proven to function well in performing state detection for foot-mounted pedestrian navigation sequences, as well as for hand-carried IMU sequences. The algorithm should be studied further and iterated before being employed for practical use, though the next steps in doing so are both apparent and feasible.

Bibliography

[1] Andrei M. Shkel, MAE240 Inertial Navigation Lectures https://canvas.eee.uci.edu/courses/30177/files/folder/Lectures?

[2] Yusheng Wang, Andrei Chernyshoff, Andrei M. Shkel, “Study on Estimation Errors in ZUPT-Aided Pedestrian Inertial Navigation Due to IMU Noises” 2019

[3] Skog, Isaac & Nilsson, John-Olof & Händel, Peter. (2010). Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. International Conference on Indoor Positioning and Indoor Navigation (IPIN). 1 - 6. 10.1109/IPIN.2010.5646936.

[4] W. Zhang, X. Li, D. Wei, X. Ji and H. Yuan, "A foot-mounted PDR system based on IMU/EKF+HMM+ZUPT+ZARU+HDR+compass algorithm," 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2017, pp. 1-5, doi: 10.1109/IPIN.2017.8115916.

[5] Park, S., Ju, H. and Park, C., 2016. Stance Phase Detection of Multiple Actions for Military Drill Using Foot-mounted IMU. International Conference on Indoor Positioning and Indoor Navigation, [Online] Available at: <http://www3.uah.es/ipin2016/usb/app/descargas/201_WIP.pdf> [Accessed 23 November 2021].

[6] Gino Brunner, Darya Melnyk, Birkir Sigfússon, and Roger Wattenhofer. 2019. Swimming style recognition and lap counting using a smartwatch and deep learning. In Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19). Association for Computing Machinery, New York, NY, USA, 23–31. DOI: https://doi.org/10.1145/3341163.3347719

[7] Yacouba Kone, Ni Zhu, Valérie Renaudin, Miguel Ortiz. Machine Learning-Based ZeroVelocity Detection for Inertial Pedestrian Navigation. IEEE Sensors Journal, 2020, 11p. ff10.1109/JSEN.2020.2999863ff. ffhal-02879906f

[8] Md. Syedul Amin, Mamun Bin Ibne Reaz, Salwa Sheikh Nasir, Mohammad Arif Sobhan Bhuiyan, Mohd. Alauddin Mohd. Ali, "A Novel Vehicle Stationary Detection Utilizing Map Matching and IMU Sensors", The Scientific World Journal, vol. 2014, Article ID 597180, 13 pages, 2014. https://doi.org/10.1155/2014/597180

[9] L. Köping, M. Grzegorzek and F. Deinzer, "Probabilistic step and turn detection in indoor localization," IET Conference on Data Fusion & Target Tracking 2014: Algorithms and Applications (DF&TT 2014), 2014, pp. 1-7, doi: 10.1049/cp.2014.0526.

[10] Chen, K.-T. Cho, S. Han, Z. Jin, and K. G. S. G. Shin, Invisible Sensing of Vehicle Steering with Smartphones. [Online]. Available: https://tagi98.github.io/files/publication/VSense.pdf. [Accessed: 23-Nov-2021].

[11] Wahlstrom, Johan & Skog, Isaac. (2020). Fifteen Years of Progress at Zero Velocity – A History of ZUPTs. IEEE Sensors Journal. PP. 1-1. 10.1109/JSEN.2020.3018880.

[12] Van Trees, Harry L. *Detection, Estimation, and Modulation Theory*. Wiley, 2001.

[13] Y. Wang and A. M. Shkel, "Adaptive threshold for zero-velocity detector in ZUPT-aided pedestrian inertial navigation," IEEE Sensors Letters, vol. 3, no. 11, pp. 1–4, 2019.

[14] InvenSense "ICM-42668-P Datasheet" DS-000347 datasheet, Nov. 2019 [Revised May 2021]

[15] "MATLAB," *MATLAB Documentation*. [Online]. Available: https://www.mathworks.com/help/matlab/. [Accessed: 03-Dec-2021].

[16] "OpenShoe" *Open source embedded foot-mounted INS* [Online]. Available: http://www.openshoe.org/ [Accessed: 03-Dec-2021].

[17] Stefano Zanella, "Effect of Gyroscope and Accelerometer Datasheet Parameters on Inertial Navigation Systems Accuracy" InvenSense MEMS Sensor Group 10.1.2018

[18] Wen Zhang, Mounir Ghogho, Baolun Yuan, "Mathematical Model and Matlab Simulation of Strapdown Intertial Navigation System" Hindawi Publishing Corporation, Volume 2012, Article ID 264537

[19] V-Ray 3.6, "UR5 with Zimmer Group Collaborative Gripper 3D model" https://www.cgtrader.com/3d-models/industrial/machine/ur5-with-zimmer-group-collaborative-gripper

[20] SmartBug Image, https://www.mouser.com/new/invensense/tdk-invensense-smartsense/

Appendix

Table of Evaluated Data Feature Values

	zero-velocity condition	**Pedestrian swing phase/other motion**	**Orbital Trajectory**
gyro magnitude	Low	High	Moderate
gyro variance	Low	High	Low
z-axis gyro magnitude	Low	High	Moderate
z-axis gyro variance	Low	High	Low
accelerometer magnitude	Low	High	Moderate
accelerometer variance	Low	High	Low
y-axis accelerometer magnitude	Low	High	Moderate
y-axis accelerometer variance	Low	High	Low