

Mountain Bike Trail Mapping with Mobile Phone Sensors

Nick Goodson,*Stanford University*

BIOGRAPHY

Nick Goodson is a Masters student in Stanford University's Department of Aeronautics & Astronautics. Previously, Nick completed a Bachelor of Engineering with Honours in Mechanical Engineering at the University of Canterbury, New Zealand.

Source code: https://github.com/NickG-NZ/MTB_Track_Mapping

ABSTRACT

A Multiplicative Extended Kalman filter is employed for state estimation of a mountain bike with the intention of providing a detailed trail map and statistics using only sensors available on a smartphone. Mountain bike trail mapping is currently a time intensive procedure that yields unsatisfactory results. Furthermore, riders are interested in collecting in-depth dynamic statistics for evaluating their own performance. The algorithms employed in this work are suitable for implementation on mobile devices so that they can be used by riders. Testing with data collected on a smart phone demonstrates that the algorithm is effective at estimating the complete state of the mountain bike. In particular the geometric path closely matches a known baseline and the attitude estimate is significantly more stable than naive integration of sensor readings.

I. INTRODUCTION

Mountain biking is undergoing a period of rapid growth as part of a wider trend in the global cycling industry. An increase in riders has lead to significant expansion of dedicated trail networks within mountain biking parks. Navigating these growing bike parks has subsequently become more challenging. In parallel with this growth, digital technologies for mountain bikers have also been widely adopted with dedicated trail map and performance measurement mobile applications (apps) becoming commonplace. Map apps allow riders to plan their rides and navigate once they are out on the trails, while performance apps provide timing measurements and other statistics. Both of these services require accurate trail maps to function. Current approaches to trail mapping typically involve collecting GNSS data along the trail which is then post-processed with aerial images and GIS data to obtain an accurate estimate of the trail path. There are two key limitations with this approach which the work in this paper aims to address. The first issue is that mountain-bike trails are frequently being built and modified; hence, it is undesirable to require a dedicated mapping effort each time a new trail is opened or an old one is changed. The second limitation addressed is the lack of detailed information provided other than the geometric path. Riders who use performance measurement applications are increasingly searching for more in-depth analyses which the current static maps do not enable.

In this work, GNSS and IMU measurements collected using a typical smartphone are fused in an Extended-Kalman Filter (EKF) to recover a trail map. By restricting the mapping algorithm to work exclusively with this data source, trail maps can be constructed entirely from data collected by trail users, thereby addressing the first problem. Furthermore, the use of an INS enables the recovery of detailed information including the velocity, acceleration and bicycle orientation. The task is therefore to estimate the state of the bicycle at all points along the trail. Previous approaches to GNSS/INS state estimation are discussed in section II.. The navigation problem is formulated and modelling decisions are outlined in section III.. Following this, the filtering algorithm is developed in section IV.. Finally, hardware testing and experimental results are discussed in section V..

II. RELATED WORK

One of the few state estimation examples for two wheeled vehicles was from Teerhuis and Jansen [1], who applied an EKF to estimate the lateral dynamics of a motorcycle including the effects of tyre contact forces. Modelling tyre forces for a mountain bike is infeasible due to the enormously varying terrain and lack of manufacturer data; hence, in this work the estimation problem is posed such that the dynamics of the bicycle need not be fully considered. Alam, López et al. [2] used an EKF for loosely coupled GNSS/INS fusion to estimate full 6DOF state of a boat along with gyroscope and accelerometer biases. Estimating the biases online enables cancellation of the drift inherent to MEMS sensors. If raw GNSS pseudoranges are available, they can be tightly coupled with inertial measurements as in [3] [4] [5]. However, raw pseudoranges are unavailable on many smart phones, so this approach is unsuitable given the objectives of this project. Girrbach, Hol et al. [6] take an alternative approach

to the sensor fusion problem, developing a moving horizon estimator which performs an optimization using direct collocation. The performance is shown to be superior to a Kalman filter at the expense of increased computational cost. In the spirit of mobile phone trail mapping with limited computational power, a loosely coupled EKF is adopted for GNSS/INS fusion in this work.

Accurate attitude estimation is crucial for inertial navigation. Feng, Li et al. [7] developed an EKF measurement model in which small weighted rotations are applied to the attitude quaternion estimate, first using the measured gravity vector then the measured magnetic field. Marins, Yun et al. [8] use a Gauss-Newton optimization, rather than applying sequential small rotations, to determine the quaternion to best explain the accelerometer and magnetometer measurements. This approach has an increased computational overhead but requires less tuning. To overcome the computational burden of the Gauss-Newton optimization step, the authors extended their original work in [9] by adapting the widely used QUEST attitude determination algorithm to find the optimal quaternion. All of these EKF attitude estimation algorithms treat measured accelerations other than gravity as random noise, limiting their applicability to highly dynamic systems. The approach presented in [10] overcomes this limitation by estimating the true acceleration using GNSS measurements and subtracting it from the accelerometer measurement to better estimate the gravity vector. Each of the algorithms discussed use a quaternion attitude representation but employ an additive correction in the EKF update step. Vector addition of quaternions has no clear physical meaning and a filtering paradigm which better encapsulates the group properties of three dimensional rotations is the Multiplicative-Extended Kalman Filter (MEKF) [11] [12] [13]. A multiplicative approach is adopted in this paper for attitude estimation which enables more natural expression of rotation errors and eliminates the need to re-normalize the quaternion at each step. A similar computation of the gravity vector to [10] is used in the measurement model.

III. NAVIGATION PROBLEM FORMULATION

The navigation objective is to estimate the position, velocity and attitude of a bicycle using the sequence of measurements provided by the internal MEMS IMU and GNSS receiver of a smart phone rigidly mounted to the bike's frame. The IMU includes a 3-axis accelerometer, gyroscope and magnetometer.

A body frame b is fixed relative to the front triangle of the bicycle and a right-handed bike coordinate system B is adopted. The coordinate system is specified with the x-y plane tangent to the tyre contact patches and the y-direction unit vector aligned between their centres at zero steering angle. The z-axis passes through the approximate centre of gravity of the combined bike and rider system. Sensor measurements are expressed in a predefined sensor coordinate system S as dictated by the phone's internal IMU orientation. The relationship between the bike coordinates and the sensor coordinates is illustrated in Figure 1. Although ultimately the objective is to estimate the state with respect to the bike coordinates B , the algorithm is simplified when applied directly to the sensor coordinates S . The known constant transformation ${}^B H^S$ can be applied as required to convert vectors to bike coordinates. For the remainder of this paper only the sensor coordinates will be considered.

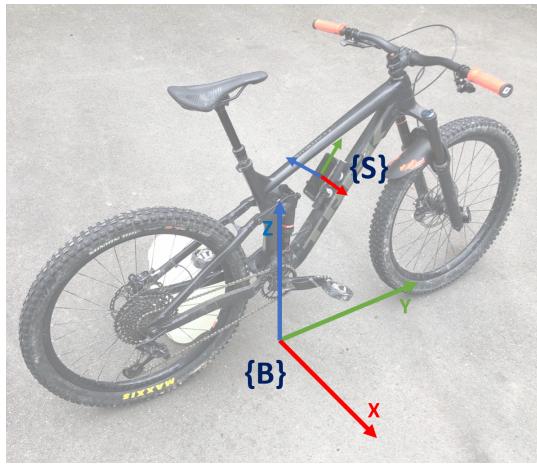


Figure 1: The bike coordinates B and sensor coordinates S are both fixed to the physical bicycle frame and hence they are related by a constant homogeneous transformation ${}^B H^S$. The y-z planes of both systems are co-planar.

A North-East-Down (N) coordinate system is fixed in a navigation frame n moving with the sensor and has its position on the WGS-84 spheroid specified by geodetic latitude φ and longitude λ as shown in Figure 2. The geodetic altitude h constrains the final degree of freedom of the sensor coordinates relative to Earth-Centred Earth-Fixed coordinates (E), and completes the position $p^L \in \mathbb{R}^3$. Velocity of the sensor relative to an Earth fixed frame is specified in North-East-Down coordinates, $v^N \in \mathbb{R}^3$.

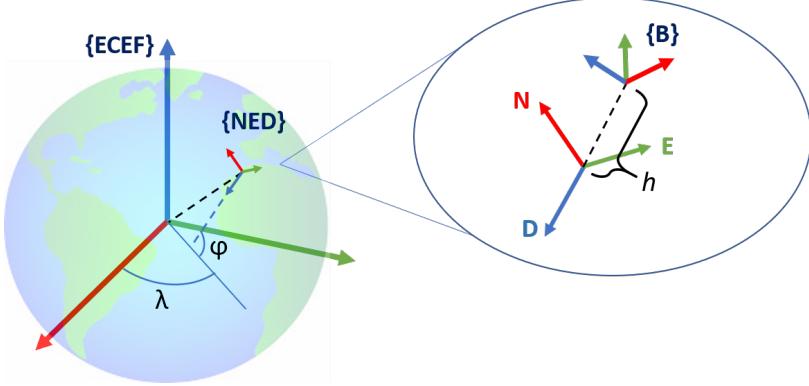


Figure 2: The NED coordinate system is fixed to the navigation frame moving along the Earth’s surface with the rider. The velocity is expressed in NED coordinates and the attitude is also referenced relative to it. The navigation frame itself is referenced to the ECEF frame via geodetic coordinates.

Attitude is parameterized with a unit quaternion representing the active rotation (sensor to navigation) ${}^N q^S \in \mathbb{H}$. Many practitioners are not aware that there are two forms of quaternion multiplication currently in use, one of which is rarely seen outside of the aerospace literature [14]. To be explicit, in this paper the original and more widely adopted quaternion multiplication will be used $\odot : \mathbb{H} \times \mathbb{H} \rightarrow \mathbb{H}$

$$q \odot r := \begin{bmatrix} q_s r_s - q_v^\top r_v \\ q_s r_v + r_s q_v + q_v \times r_v \end{bmatrix} \quad (1)$$

where $q := [q_s \ q_v]^\top$ with subscripts s and v denoting the scalar and vector components respectively of the quaternion.

MEMS gyroscope and accelerometer measurements are known to suffer from bias drift and it is common to estimate them online in order to apply the necessary correction. However, the biases of each sensor add three additional state variables to be estimated, increasing the computational burden. As a compromise, only the gyroscope biases, $b_\omega = [b_x \ b_y \ b_z]^\top$ will be estimated as GNSS measurements should help to correct any accelerometer bias dynamics. The full state vector for the estimation problem becomes

$$X = \begin{bmatrix} {}^N q^S & b_\omega & v^N & p^L \end{bmatrix}^\top \in \mathbb{H}, \mathbb{R}^9 \quad (2)$$

$$= \begin{bmatrix} q_s & q_v^\top & b_x & b_y & b_z & v_N & v_E & v_D & \varphi & \lambda & h \end{bmatrix}^\top \quad (3)$$

All IMU measurements are expressed in sensor coordinates and include:

- $\omega_{b/i}^S \in \mathbb{R}^3$, angular velocity of the body frame relative to the inertial frame
- $f^S \in \mathbb{R}^3$ specific force on the body in the inertial frame
- $h^S \in \mathbb{R}^3$ local Earth magnetic field

The GNSS measurements are:

- $p^L \in \mathbb{R}^3$, geodetic position
- $v^N \in \mathbb{R}^3$, velocity of sensor relative to Earth fixed frame

IV. MULTIPLICATIVE GNSS-INS FUSION

1. Filter Preliminaries

Given a probabilistic prior over the initial state $p(x_0)$, a recursive Bayesian filter estimates a posterior distribution $p(x_t | y_{1:t}, u_{1:t-1})$ at each point on the trajectory conditioned on the sequence of measurements $y_{1:t}$ and inputs $u_{1:t-1}$. The EKF requires that both the state and measurement distributions are approximated as Gaussian. Standard forms for the state transition model and

measurement model are given by 4 and 5 respectively. Here w_t and v_t are Gaussian white noise approximations of the process and measurement noise.

$$X_{t+1} = f(X_t, U_t) + w_t \quad w_t \sim \mathcal{GWN}(0, Q_t) \quad (4)$$

$$Y_t = g(X_t, U_t) + v_t \quad v_t \sim \mathcal{GWN}(0, R_t) \quad (5)$$

The non-linear dynamics used as the basis of the state transition model are given in 6. Rather than modelling the highly complex rotational dynamics of the bike and rider system, the attitude kinematics are used as a motion model, treating the gyroscope measurements like a control input. This approach simplifies the filtering problem but reduces the smoothing effect on the posterior due to the gyroscope measurement noise.

$$\begin{bmatrix} {}^N\dot{q}^S \\ \dot{b}_\omega \\ {}^N\dot{v}^N \\ \dot{p}^L \end{bmatrix} = \begin{bmatrix} \frac{1}{2} {}^Nq^S \odot \omega_{b/n}^S \\ 0 \\ {}^N C^S f^S + g^N - (2\omega_{e/i}^N + \omega_{n/e}^N) \times v^N \\ Dv^N \end{bmatrix} \quad (6)$$

The matrix $D \in \mathbb{R}^{3 \times 3}$ relates the velocity in North-East-Down coordinates to the time rate of change of the geodetic position and is defined using the *meridian radius of curvature* M and the *prime vertical radius of curvature* N

$$D = \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos(\varphi)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (7)$$

$$M = \frac{a(1-e^2)}{(1-e^2\sin^2\varphi)^{(3/2)}} \quad (8)$$

$$N = \frac{a}{(1-e^2\sin^2\varphi)^{(1/2)}} \quad (9)$$

where a is the semi-major axis and e is the eccentricity of the WGS-84 Earth spheroid model. The required angular velocity $\omega_{b/n}^S$ is found from the gyroscope measurement $\omega_{b/i}^S$ and the other state variables

$$\omega_{b/n}^S = \omega_{b/i}^S - ({}^S C^N \cdot {}^N C^E) \omega_{e/i}^E - {}^S C^N \omega_{n/e}^N \quad (10)$$

$${}^N C^E = \begin{bmatrix} -s\varphi c\lambda & -s\varphi s\lambda & c\varphi \\ -s\lambda & c\lambda & 0 \\ -c\varphi c\lambda & -c\varphi s\lambda & -s\varphi \end{bmatrix} \quad (11)$$

$$\omega_{n/e}^N = \begin{bmatrix} \dot{\lambda}\cos(\varphi) \\ -\dot{\varphi} \\ -\dot{\lambda}\sin(\varphi) \end{bmatrix} \quad (12)$$

The DCM ${}^S C^N$ encodes the opposite rotation to ${}^N q^S$ and can be therefore be obtained directly from the state. The discrete time state transition model $f(X_t, U_t)$ is approximated by a simple forward Euler integration of the non-linear dynamics.

$$f(X_t, U_t) = I + f(x(t), u(t))\delta t \quad (13)$$

The combined IMU and GNSS measurement model is given by 14. The relationships between the gravity and Earth magnetic field vectors in sensor and NED coordinates encode attitude information. The gravity vector in the sensor frame g^S is not directly measured and must be computed using the accelerometer and GNSS measurements according to 15. The acceleration \dot{v}^N is also not directly measured and is computed using a first order difference between the GNSS velocity measurements at the

previous and current time steps.

$$Y_t = \begin{bmatrix} h^S \\ g^S \\ v^N \\ p^L \end{bmatrix} = \begin{bmatrix} {}^S C^N h^N \\ {}^S C^N g^N \\ v^N \\ p^L \end{bmatrix} \quad (14)$$

$$g^S = -f^S + {}^S C^N (\dot{v}^N + (2\omega_{e/i}^N + \omega_{n/e}^N) \times v^N) \quad (15)$$

At each time step, the measurement vector Y_t may only contain readings from a subset of the four sensors due to their different operating rates. To enable the navigation algorithm to handle measurements received asynchronously, two separate measurement models are used in the filter. The first model is for the magnetometer only and is given by the first row of 14. The second model is the combined accelerometer and GNSS model given by the last three rows of 14.

The standard EKF recursion is given by 16. At each time step, the filter performs a prediction followed by a measurement update to compute the new posterior belief over the state. A and C are the discrete time jacobians of the state transition model and measurement model respectively.

Predict:

$$\begin{aligned} X_{t|t-1} &= f(X_{t-1|t-1}, U_{t-1}) \\ \Sigma_{t|t-1} &= A_{t-1} \Sigma_{t-1|t-1} A_{t-1}^\top + Q_{t-1} \end{aligned}$$

Update:

$$\begin{aligned} L_t &= \Sigma_{t|t-1} C_t^\top (C_t \Sigma_{t|t-1} C_t^\top + R_t)^{-1} \\ X_{t|t} &= X_{t|t-1} + L_t (Y_t - g(X_{t|t-1}, U_{t-1})) \\ \Sigma_{t|t} &= (I - L_t C_t) \Sigma_{t|t-1} (I - L_t C_t)^\top + L_t R_t L_t^\top \end{aligned} \quad (16)$$

2. Multiplicative Error Representations

The vanilla EKF algorithm presented in the previous section does not properly account for the group structure of three dimensional rotations. Unlike vectors which compound through addition, rotations compound through multiplication. This leads to a fundamental difference in how rotation errors should be represented. Consider a normally distributed random vector $x \in \mathbb{R}^3$ with mean μ and covariance Σ_x . The following relationships hold

$$\delta x = (x - \mu) \quad (17)$$

$$\Sigma_x = \mathbb{E}[\delta x \delta x^\top] \quad (18)$$

The equivalent error relationship for a rotation can be developed by introducing a differential rotation $\phi \in \mathbb{R}^3$. There are several possible representations for ϕ and in this work the Gibbs Vector will be used.

$$\phi = \mathbf{n} \tan(\theta/2) \quad (19)$$

where \mathbf{n} and θ are the axis and rotation angle of the equivalent angle-axis vector respectively. Consider a Gaussian distribution over rotation errors with mean q_μ . The rotation error and the covariance of the distribution are given by

$$\delta q = q_\mu^\dagger q = \frac{1}{\sqrt{1 + \|\phi\|}} \begin{bmatrix} 1 \\ \phi \end{bmatrix} \quad (20)$$

$$\Sigma_q = \mathbb{E}[\phi \phi^\top] \quad (21)$$

The mapping between the Gibbs vector and the error quaternion is called the Cayley map.

Forward Euler discretization of the state transition model is adequate for vector quantities but will result in un-normalized quaternions if applied directly to the attitude kinematics. An alternate solution is to use a zero order hold on the angular velocity

to enable a direct quaternion multiplication. This computation is performed in the prediction step of the filter.

$$\begin{aligned}\theta &= \|\omega_{b/n}^S - b_\omega\| \delta t, \quad \mathbf{n} = \frac{\omega_{b/n}^S - b_\omega}{\|\omega_{b/n}^S - b_\omega\|} \\ \hat{q}_t &= \begin{bmatrix} \cos(\theta/2) \\ \mathbf{n} \sin(\theta/2) \end{bmatrix} \\ q_{t+1} &= q_t \odot \hat{q}_t\end{aligned}\tag{22}$$

3. MEKF

The Multiplicative Extended Kalman Filter architecture that was used in this work is illustrated in Figure 3. The filter adopts the Gibbs vector error representation from equation 20 as part of an error state 23. This error state is used throughout the entire measurement update step of the filter.

$$\delta X = \begin{bmatrix} \phi & \delta b_\omega & \delta v^N & \delta p^L \end{bmatrix}^\top \in \mathbb{R}^{12}\tag{23}$$

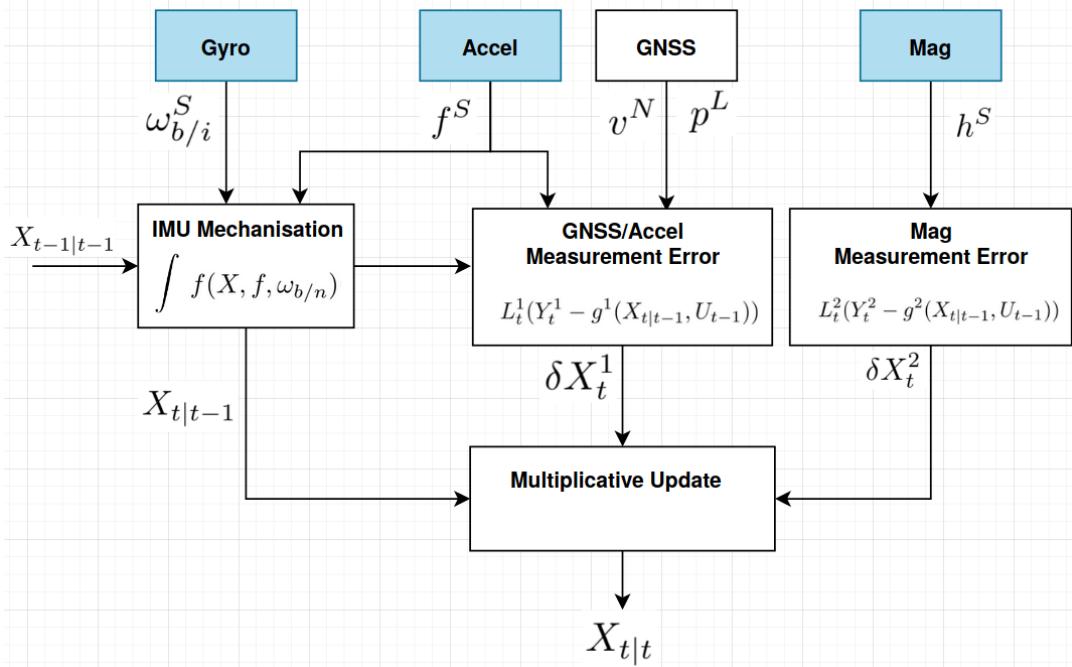


Figure 3: The MEKF architecture used for estimating the state of a mountain bike. Two separate measurement models are used to account for the different update rates of the sensors. The error states output by the measurement models are fused with the output of the prediction step in a multiplicative update

In order to define the Jacobians, some operators are introduced.

- $[\cdot]_\times$ denotes the skew symmetric cross product matrix operator
- $H := \begin{bmatrix} 0 \\ I_3 \end{bmatrix}$ transforms a vector into a quaternion with zero scalar part.
- $L(q) := \begin{bmatrix} q_s & -q_v^\top \\ q_v & sI + [q_v]_\times \end{bmatrix}$ is the matrix equivalent to left multiplication by quaternion q

- $R(q) := \begin{bmatrix} q_s & -q_v^\top \\ q_v & sI - [q_v]_\times \end{bmatrix}$ is the matrix equivalent to right multiplication by quaternion q

The Jacobian for the state transition function is defined as

$$\delta X_{t+1} = A_t \delta X_t \quad (24)$$

$$A_t = \begin{bmatrix} H^\top L^\top (\hat{q}_t) R(\hat{q}_t) H & -\frac{1}{2} I \delta t & 0 & 0 \\ 0 & I & 0 & 0 \\ -2^N C^S [f^S]_\times \delta t & 0 & I - [2\omega_{e/i}^N + \omega_{n/e}^N]_\times \delta t & 0 \\ 0 & 0 & D \delta t & \frac{\partial p_{t+1}}{\partial p_t} \end{bmatrix} \quad (25)$$

A brief derivation of the term $\frac{\partial p_{t+1}}{\partial p_t}$ can be found in Appendix 1.. The measurement Jacobian is defined as

$$C_t = \begin{bmatrix} 2[g^S]_\times & 0 & 0 & 0 \\ 2[h^S]_\times & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad (26)$$

The state covariance is defined using the error state, $\Sigma_t = \mathbb{E}[\delta X \delta X^\top] \in \mathbb{R}^{12 \times 12}$.

The 3×3 block of the measurement covariance R_t corresponding to the measurement of gravity g^S needs to account for the computation in equation 15

$$\xi = \begin{bmatrix} \delta f^S & \delta v^N & \delta \dot{v}^N \end{bmatrix}^\top \quad (27)$$

$$\Xi = \begin{bmatrix} -I & [2\omega_{e/i}^N + \omega_{n/e}^N]_\times & I3 \end{bmatrix} \in \mathbb{R}^{3 \times 9} \quad (28)$$

$$\delta g^S = \Xi \xi \quad (29)$$

$$Cov(g^S) = \Xi \mathbb{E}[\xi \xi^\top] \Xi^\top \quad (30)$$

where $\mathbb{E}[\xi \xi^\top] \in \mathbb{R}^{9 \times 9}$ is a diagonal matrix containing the measurement variances of each of the terms in ξ .

The MEKF recursion proceeds in a similar manner to the standard EKF algorithm given by equations 16. The prediction step is unchanged other than the modification in equation 22. The measurement update step computes an error state δX_t which is applied as a correction to the output from the prediction $X_{t|t-1}$. Note that this correction is identical to the standard EKF update for all state variables except for the attitude. The attitude correction is given by

$$q_{t|t} = q_{t|t-1} \odot \frac{1}{\sqrt{1 + \|\phi\|}} \begin{bmatrix} 1 \\ \phi \end{bmatrix}$$

V. EXPERIMENTS

A Samsung Galaxy S7 Active was mounted to a Trek Remedy 9.8 to collect data for testing the trail mapping algorithm. In this work, computations were performed on an Asus UX410U laptop after the data was collected. A future objective is to implement the algorithms in Java to run directly on the mobile hardware. The mobile device sensor update rates are given in Table 1. The rapid update rate of the accelerometer and gyroscope result in several prediction steps occurring between each measurement update. Estimation of the gyroscope bias can help reduce drift during this INS mechanization. However, to better emulate the processing limitations of the mobile hardware, the data was down-sampled to 200 Hz.

Due to time restraints, only a single trajectory was tested. Furthermore, bad weather conditions (particularly an abundance of mud) prevented testing the hardware on a proper mountain bike trail. Performing initial testing on a road is also advantageous in that it allows clear qualitative comparisons with a highly accurate prior map. The filtered path is visualized as an overlay on

Table 1: Mobile device specifications

Samsung Galaxy S7 Active	
Year	2015
Gyroscope rate	400 Hz
Accelerometer rate	400 Hz
Magnetometer rate	125 Hz
GNSS Fix rate	1 Hz

a satellite image in Figure 4. Overall the estimated path matches very closely to that which was ridden; however, the estimated start and end points deviate significantly in the lateral direction when in fact they should lie directly on top of one another. The total deviation in the lateral direction is $6.8m$. The vertical deviation is $1.7m$. The nature of the sensor data used prevents implementing any form of autonomous loop closure to reduce this error. If measurements from exteroceptive sensors such as vision or ranging were included in the estimation algorithm it would be possible to include a loop closure scheme.



Figure 4: The path estimated by the filter overlaid on a satellite image. The position vectors in geodetic coordinates were transformed into the local NED frame at the starting point of the trajectory. The origin of this frame was then positioned relative to the known coordinates of the top left corner of the image. The curvature of the Earth over the length of the trajectory is sufficiently small that a fixed NED frame can be used.

In Figure 5, the filter is compared to naive integration of the gyroscope for estimating the pitch ϕ and roll θ angles of the bike. The bike remained nearly level throughout the trajectory and the net change in elevation was $0m$ so the average pitch and roll should be close to 180° and 0° respectively. The maximum and average values are compared to the naive gyro integration approach in Table 2. The average pitch and roll for the filter are significantly lower than for naive integration. From the figure it is clear that there is significant gyro drift over the trajectory.

The kinematics of a bicycle are such that the heading angle ψ should rotate the forward axis to nearly line up with the velocity

Table 2: Pitch and roll angles comparison between filter and naive gyroscope integration

	average pitch	max pitch	average roll	max roll
Naive	27.3	77.1	11.2	71.0
Filter	7.98	29.4	2.60	32.4

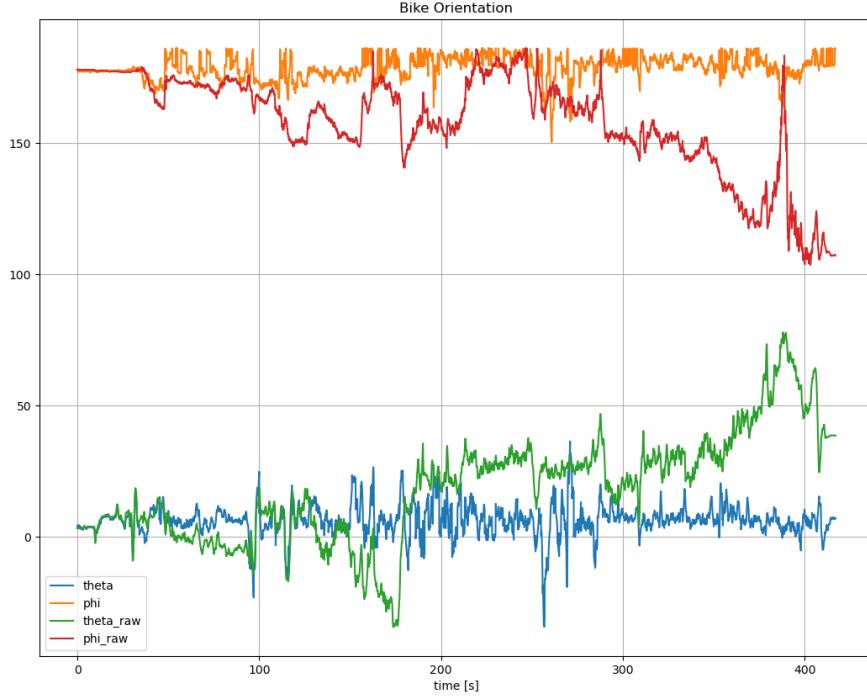


Figure 5: A comparison of the pitch ϕ and roll θ angles output from the filter to those obtained by naive integration the gyroscope

vector at all times. Results of computing the deviation in this heading angle from the velocity vector are summarized in Table 3. Once again the filter significantly outperforms naive gyroscope integration. Note that the velocity vector used for both computations was that of the filtered case as without filtering the trajectory quickly diverges.

Table 3: Deviation of the heading angle from the velocity vector

	average deviation	max deviation
Naive	14.2	122.1
Filter	8.4	86.3

The comparison between the filter and direct integration of the gyroscope shows a significant improvement in stability and accuracy of the state estimate. However, without a ground truth available it is challenging to directly quantify the error. To improve upon these results, kinematic constraints could be included within the filter. It is unlikely the bike will ever have a substantial velocity component in the vertical direction for instance. And in general the heading and velocity should nearly coincide. The accelerometer bias could also be estimated online to help reduce drift.

It is interesting to note that the complete state is recovered along the length of the trail along with uncertainty information in the form of the covariance. With multiple trail users providing data from their smartphones, an optimal map estimate could be obtained by weighted averaging of multiple maps based on the uncertainty at each point. Implementation and testing of this

algorithm is left for future works

VI. CONCLUSION

A Multiplicative Extended Kalman filter was proposed for estimating the full 6DOF state of a mountain bike on a trail. The filter uses two separate measurement models to accommodate different sensor rates and fuses the available information at each time step using a multiplicative update. The filter was tested with data collected on a typical smartphone and demonstrated a significant improvement in attitude estimation over naive integration of the gyroscope. However, the lack of exteroceptive information meant that the start and end points of the closed test trajectory had a considerable error. Future steps include implementing the algorithm on mobile hardware and considering state constraints to improve filter performance.

VII. APPENDIX

1. State Transition Jacobian

The state transition jacobian term $\frac{\partial p_{k+1}}{\partial p_k}$ was omitted in the main text and is derived below

$$p_{t+1}^L = p_t^L + \dot{p}_t^L \delta t \quad (31)$$

$$\rightarrow \frac{\partial p_{t+1}}{\partial p_t} = I + \frac{\partial \dot{p}_t^L}{\partial p_t} \delta t \quad (32)$$

Recall that

$$\dot{p}_t^L = Dv_t^N \quad (33)$$

$$D = \begin{bmatrix} \frac{1}{(M+h)} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos(\varphi)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (34)$$

$$M = \frac{a(1-e^2)}{(1-e^2 \sin^2 \varphi)^{(3/2)}} \quad (35)$$

$$N = \frac{a}{(1-e^2 \sin^2 \varphi)^{(1/2)}} \quad (36)$$

So we can write

$$\frac{\partial Dv_t^N}{\partial p_t} = \frac{\partial}{\partial p_t} \begin{bmatrix} \frac{1}{(M+h)} v_N \\ \frac{1}{(N+h)\cos(\varphi)} v_E \\ -v_D \end{bmatrix} \quad (37)$$

$$(38)$$

The individual partial derivatives are

$$\frac{\partial \dot{\varphi}}{\partial \varphi} = \frac{\partial \dot{\varphi}}{\partial M} \cdot \frac{\partial M}{\partial \varphi} = \frac{-v_N}{(M+h)^2} \cdot \frac{3a(1-e^2)\sin(\varphi)\cos(\varphi)}{(1-e^2\sin^2(\varphi))^{5/2}} \quad (39)$$

$$\frac{\partial \dot{\varphi}}{\partial \lambda} = 0 \quad (40)$$

$$\frac{\partial \dot{\varphi}}{\partial h} = \frac{-v_N}{(m+h)^2} \quad (41)$$

$$\frac{\partial \dot{\lambda}}{\partial \varphi} = \frac{\partial \dot{\lambda}}{\partial N} \cdot \frac{\partial N}{\partial \varphi} + \frac{\partial \dot{\lambda}}{\partial \varphi} = \frac{-v_E}{(N+h)^2\cos(\varphi)} \cdot \frac{ae^2\sin(\varphi)\cos(\varphi)}{(1-e^2\sin^2(\varphi))^{3/2}} + \frac{2v_E\sin(\varphi)}{(N+h)(\cos(2\varphi)+1)} \quad (42)$$

$$\frac{\partial \dot{\lambda}}{\partial \lambda} = 0 \quad (43)$$

$$\frac{\partial \dot{\lambda}}{\partial h} = \frac{-v_E}{(N+h)^2\cos(\varphi)} \quad (44)$$

$$\frac{\partial \dot{h}}{\partial \varphi} = 0 \quad (45)$$

$$\frac{\partial \dot{h}}{\partial \lambda} = 0 \quad (46)$$

$$\frac{\partial \dot{h}}{\partial h} = 0 \quad (47)$$

REFERENCES

- [1] A. Teerhuis and S. Jansen, "Motorcycle state estimation for lateral dynamics," *Vehicle System Dynamics - VEH SYST DYN*, vol. 50, pp. 20–22, 11 2010.
- [2] M. Alam, G. López, M. Sipos, and J. Rohac, "Ins/gnss localization using 15 state extended kalman filter," 11 2016.
- [3] Y. Dong, D. Wang, L. Zhang, Q. Li, and J. Wu, "Tightly coupled gnss/ins integration with robust sequential kalman filter for accurate vehicular navigation," *Sensors*, vol. 20, p. 561, 01 2020.
- [4] Y. Liu, F. Liu, Y. Gao, and L. Zhao, "Implementation and analysis of tightly coupled global navigation satellite system precise point positioning/inertial navigation system (gnss ppp/ins) with insufficient satellites for land vehicle navigation," *Sensors (Basel, Switzerland)*, vol. 18, 2018.
- [5] J. Sasiadek and P. Hartana, "Gps/ins sensor fusion for accurate positioning and navigation based on kalman filtering," *IFAC Proceedings Volumes*, vol. 37, no. 5, pp. 115 – 120, 2004, 7th IFAC Symposium on Cost-Oriented Automation (COA 2004), Gatineau, Québec, Canada, 6-9 June 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474667017323534>
- [6] F. Girrbach, J. Hol, G. Bellusci, and M. Diehl, "Optimization-based sensor fusion of gnss and imu using a moving horizon approach," *Sensors*, vol. 17, 05 2017.
- [7] K. Feng, J. Li, Z. Xiaoming, C. Shen, Y. Bi, T. Zheng, and J. Liu, "A new quaternion-based kalman filter for real-time attitude estimation using the two-step geometrically-intuitive correction algorithm," *Sensors*, vol. 17, p. 2146, 09 2017.
- [8] J. Marins, X. Yun, E. Bachmann, R. Mcghee, and M. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," *IEEE International Conference on Intelligent Robots and Systems*, vol. 4, 01 2002.
- [9] X. Yun and E. Bachmann, "Design, implementation, and experimental results of a quaternion-based kalman filter for human body motion tracking," *Robotics, IEEE Transactions on*, vol. 22, pp. 1216 – 1227, 01 2007.
- [10] D.-M. Ma, J.-K. Shiau, I.-C. Wang, and Y.-H. Lin, "Attitude determination using a mems-based flight information measurement unit," *Sensors (Basel, Switzerland)*, vol. 12, pp. 1–23, 01 2012.
- [11] L. Markley, "Attitude error representations for kalman filtering," 01 2002.
- [12] ———, "Multiplicative vs. additive filtering for spacecraft attitude determination," 07 2004.

- [13] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, “Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds,” *ArXiv*, vol. abs/1107.1119, 2013.
- [14] R. Zanetti, “Rotations, transformations, left quaternions, right quaternions?” *Journal of the Astronautical Sciences*, 01 2019.