A Comparison of Methods for Online Lever Arm Estimation in GPS/INS Integration

Nick Montalbano
Department of Aerospace Engineering
and Engineering Mechanics
University of Texas at Austin
Email: nick30075@utexas.edu

Todd Humphreys
Department of Aerospace Engineering
and Engineering Mechanics
University of Texas at Austin
Email: todd.humphreys@mail.utexas.edu

***Abstract*—A comparison of neural network, state augmentation, and multiple model-based approaches to online location of inertial sensors on a vehicle is presented that exploits dual-antenna carrier-phase-differential GNSS. The best technique among these is shown to yield a significant improvement on *a priori* calibration with a short window of data. Estimation of Inertial Measurement Unit (IMU) parameters is a mature field, with state augmentation being a strong favorite for practical implementation, to the potential detriment of other approaches. A simple modification of the standard state augmentation technique for determining IMU location is presented that determines which model of an enumerated set best fits the measurements of this IMU. A neural network is also trained on batches of IMU and GNSS data to identify the lever arm of the IMU. A comparison of these techniques is performed and it is demonstrated on simulated data that state augmentation outperforms these other methods.**

***Keywords*—GPS/INS integration, neural networks**

I. Introduction

Online calibration of sensors is a persistent problem for multi-sensor integration. Estimation of IMU parameters must be performed to determine the biases and location of the device on the vehicle; failing to accurately determine these parameters creates large errors in state estimates produced from inertial measurements. However, measuring the location on the vehicle of the IMU relative to the GNSS antenna may be difficult, due to the design of the vehicle, or time-consuming, for platforms in the prototyping phase. For many applications, it is preferable to locate the IMU on a vehicle during standard operation. Bias estimation must be performed during device operation and, though many solutions to the problem of simultaneous estimation have been developed, several techniques have not been applied to this problem.

Active calibration of IMUs has been extensively studied from the perspective of state augmentation. Reference [1] analytically determined necessary conditions on vehicle motion to determine the lever arm and biases of an IMU mounted on a car equipped with carrier-phase differential GNSS. Reference [2] studied the full nonlinear problem on a car with a single-antenna GNSS setup and derived a series of sufficient conditions to determine intrinsic parameters. The approach presented herein will estimate biases and mounting errors from an enumerated set. A multiple-model filter (MMKF) should be able to accurately determine the true location faster and more efficiently than conventional methods by extracting information from discarded knowledge. For example, the vehicle's approximate size constrains the search space for sensor location. While it may be more accurate to explicitly handle the constraint, underlying distributions become strongly non-Gaussian. Selection from an enumerated set does not encounter this problem. Multiple model estimation has been held back by its prohibitive computational cost compared to other adaptive estimation methods [3]; however, modern processing power is quickly compensating for this increased cost. Reference [4] presents a modern comparison of a multiple model estimator with a 15-state extended Kalman filter (EKF) and demonstrates gains in accuracy and convergence time, a clear indicator of the potential of this approach.

The equations for dynamics propagation of the coupled INS/GNSS system are linear within the limits of the IMU employed for model replacement, if biases are exactly known. The measurement equation, however, may behave poorly, depending on the integration architecture employed [5]. Tightly coupled estimation incorporates the pseudorange to each GNSS satellite into its measurement equation and, as a result, is highly nonlinear. Loosely coupled estimation instead incorporates the position solution provided by a GNSS receiver and results in a measurement equation that can be linearized without large errors. The extended Kalman filter is typically applied to this architecture ([1], [6]). This work will concern itself with loosely coupled estimation. State augmentation is typically performed for integrating these sensors but, given the performance benefits of multiple-model filters over EKFs, it is possible that a multiple-model filter with a fine discretization over possible lever arm locations will outperform the standard state augmentation approach. In addition to the MMEKF, machine learning may be beneficial for solving this problem; a neural network may be able to resolve nonlinearities where a standard filter would experience difficulties.

Aforementioned improvements to processing power have enhanced the accuracy and availability of neural networks for a wide range of problems. The literature is rich with algorithms improving the accuracy of machine learning, ranging from improvements to the training process such as dropout [7] to modifications to network architectures such as batch normalization [8] and convolutional networks [9]. These refinements to standard approaches have greatly increased the

number of applications that are amenable to neural network-based solutions. It is well-established that neural networks can successfully perform dynamic estimation for well-posed non-linear problems [10]. Machine learning has also demonstrated successes at system identification for later application within a Kalman Filter ([11], [12]). A neural network may be able to estimate this parameter without need for well-tuned or more sophisticated filters.

Reference [12] noted several problems with convergence of the network employed in system identification–in particular, that an H_∞-based technique outperforms the neural network in near-linear cases. They experimented with a one-layer network; it is likely that using a more modern network, larger in size and trained with methods developed after [12]'s initial publication, a better estimate for system parameters can be obtained. On the other hand, the loosely coupled estimation problem with unknown lever arm is only weakly observable under most movement patterns [1] and a neural network may have difficulty recognizing those weaker areas.

This paper makes two closely related contributions. First, it establishes that both multiple model-based and neural network-based solutions can be successfully applied to the problem of lever arm estimation in GPS/INS integration. These approaches are tested at attitudes lower than those against which standard techniques are typically validated. Second, it compares these techniques to established techniques and demonstrates that their performance is of middling quality compared to tried-and-true methods. There are some problems that neural networks cannot solve as accurately as existing techniques; it is important that the current excitement surrounding neural networks does not lead to their application to problems for which they show poor performance.

II. Loosely Coupled Error Models and Estimators

State augmentation is typically performed for online estimation of biases for a platform with integrated INS/GNSS. A lever arm error state, $\delta \boldsymbol{l}_{ab}$, can be appended to the full state if the exact value of the lever arm is not known. This work will concern itself with the loosely coupled formulation of the problem, with GNSS position included in the measurement equation rather than raw psuedoranges. This section will introduce models for error states in GPS/INS integration and measurement equations corresponding to their use. A review of the interacting multiple-model Kalman filter will then be performed.

A. Loosely Coupled Estimation with Unknown Location

Using [6]'s formulation, the equations of motion of an object moving in the Earth-Centered Earth-fixed (ECEF) frame can be expressed as

$$\begin{aligned} \dot{\boldsymbol{P}}^e &= \boldsymbol{V}^e \\ \dot{\boldsymbol{V}}^e &= R_b^e \boldsymbol{f} - 2\boldsymbol{\omega}_e \times \boldsymbol{V}^e + \boldsymbol{g} \\ \dot{R}_b^e &= R_b^e \Omega \end{aligned} \tag{1}$$

where $\boldsymbol{P}$, $\boldsymbol{V}$, and R_b^e are position vector, velocity vector, and rotation matrix of the body's local frame with respect to the ECEF frame, $\boldsymbol{\omega}_e$ is the angular velocity of the Earth, $\boldsymbol{f}$ is the specific force acting on the body's center of gravity expressed in the ECEF frame, and Ω is the cross-product matrix of angular rates of the body relative to the ECEF frame, expressed in the body frame.

Denote estimated states for position, velocity, and the rotation matrix as $\hat{\boldsymbol{P}}$, $\hat{\boldsymbol{V}}$, and $\hat{R}$ and likewise denote $\delta \boldsymbol{P}$, $\delta \boldsymbol{V}$ as the errors in position and velocity. Γ is the cross product matrix of $\boldsymbol{\gamma}$, the attitude estimation error, $\boldsymbol{f}$ is the specific force on the body and $\hat{\boldsymbol{f}}$ its measurement, $\boldsymbol{\omega}_B$ the true angular rate of the body frame with respect to the inertial frame and $\hat{\boldsymbol{\omega}}_B$ its measurement, $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ are accelerometer and gyro biases, respectively, and $\boldsymbol{w}_a$ and $\boldsymbol{w}_g$ denote noise. Though these biases evolve quickly in low-quality sensors, they can be treated as constant over sufficiently short periods of time. Scale factor and alignment errors are ignored for this examination. A multi-antenna solution is capable of estimating alignment errors which, if not in yaw, are typically small. Scale factor errors are of lesser impact than alignment errors for aerial vehicles, producing position errors almost an order of magnitude lower [13]. The estimation errors of state estimates and biases can be modeled as

$$\begin{aligned} \hat{\boldsymbol{P}}^e &= \boldsymbol{P}^e + \delta \boldsymbol{P} \\ \hat{\boldsymbol{V}}^e &= \boldsymbol{V}^e + \delta \boldsymbol{V} \\ \hat{R}_b^e &= R_b^e (I_3 + \Gamma) \\ \hat{\boldsymbol{f}} &= \boldsymbol{f} + \boldsymbol{b}_a + \boldsymbol{w}_a \\ \hat{\boldsymbol{\omega}}_B &= \boldsymbol{\omega}_B + \boldsymbol{b}_g + \boldsymbol{w}_g \end{aligned} \tag{2}$$

The GNSS position solution is denoted $\boldsymbol{P}$. A complementary filter output $\hat{\boldsymbol{P}}$ is used as an additional estimate of $\boldsymbol{P}$. The difference between the two is taken as the measurement. Defining the vector in the body frame describing the IMU's location relative to the GPS unit as $\boldsymbol{L}_{ab}$ and its estimation error as $\delta \boldsymbol{l}_{ab}$, the linearized measurement equation is

$$\boldsymbol{z} = \hat{\boldsymbol{P}} - \boldsymbol{P} = \delta \boldsymbol{P} + R_b^e \boldsymbol{L}_{ab} \boldsymbol{\gamma} + R_b^e \delta \boldsymbol{l}_{ab} - \boldsymbol{v} \tag{3}$$

Combining 1 and 2, the linearized dynamics can be expressed in terms of state $\boldsymbol{x} = [\delta \boldsymbol{P}^T \ \delta \boldsymbol{V}^T \ \boldsymbol{\gamma}^T \ \boldsymbol{b}_g^T \ \boldsymbol{b}_a^T \ \delta \boldsymbol{l}_{ab}^T]^T$ with additive IMU noise $\boldsymbol{w}(t)$ as

$$\dot{x}(t) = \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & -R_b^e F_b & 0_3 & R_b^e & 0 \\ 0_3 & 0_3 & -\Omega_b & I_3 & 0 & 0 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix} \boldsymbol{x}(t) + \boldsymbol{w}(t) \tag{4}$$

wherein Ω_b and F_b are the cross product matrices of body-frame acceleration and angular rate. This must be discretized when implemented in the filter; the authors have found that a first-order approximation in time is accurate enough for (simulated) real-time use. The measurement equation of Eq. 3 can be written in terms of $\boldsymbol{x}$ as

$$\boldsymbol{z}(t) = \begin{bmatrix} I_3 & 0_3 & -R_b^e \boldsymbol{L}_{ab} & 0_3 & 0_3 & R_b^e \end{bmatrix} \boldsymbol{x}(t) - \boldsymbol{v}(t) \quad (5)$$

Linearization errors are small. The Kalman filter described by these equations has unobservable modes for most trajectories and care must be taken during validation to guarantee that the vehicle experiences pitching and/or rolling motions during operation [1].

If an additional GPS unit is added to the system at a known location $\boldsymbol{L}_{21}$ in the body frame, then the measurement equation can be modified to incorporate it. If the antenna and IMU frames are aligned, incorporating $\boldsymbol{z}_2(t) = \delta \boldsymbol{P}_{21} = \boldsymbol{P}_2 - \boldsymbol{P}_1$ into the measurement equation results in

$$\boldsymbol{z}(t) = \begin{bmatrix} I_3 & 0_3 & -R_b^e \boldsymbol{L}_{ab} & 0_3 & 0_3 & R_b^e \\ 0_3 & 0_3 & -R_b^e \boldsymbol{L}_{21} & 0_3 & 0_3 & 0_3 \end{bmatrix} \boldsymbol{x}(t) - \boldsymbol{v}(t) \quad (6)$$

This modification provides better attitude observability, better constraining the gyro biases.

B. Multiple-Model Estimation

The multiple-model filter adaptively determines which dynamics model of an enumerated set best fits collected data by comparing the outputs of a bank of filters with different dynamics and/or measurement models. This work will experiment with an interacting multiple-model EKF (IMMEKF), which differs slightly from the standard MMEKF in the size of its bank of filters–the IMMEKF mixes all data from the previous iteration of the filter before performing the next update step, reducing the number of filters that must operate simultaneously.

From [14], the probability that the i^{th} model was in effect at time $k-1$ given measurement $\boldsymbol{z}^k$ and that the j^{th} model is in effect at time k is

$$\begin{aligned} \mu_{i|j}(k-1|k-1) &= P(m_i(k-1)|m_j(k), \boldsymbol{z}^k) \\ &= \frac{1}{c} p_{i,j} \mu_i(k-1) \end{aligned} \quad (7)$$

where c_j is the normalizing constant such that $\frac{1}{c_j}\sum_{i=1}^n p_{i,j}\mu_i(k-1) = 1$. $p_{i,j}$ represents the probability of switching models from i to j.

The pre-measurement state estimate $\hat{\boldsymbol{x}}^j(k-1)$ and covariance matrix P^j for the j^{th} filter at time k is

$$\begin{aligned} \hat{\boldsymbol{x}}_0^j(k-1|k-1) &= \sum_{i=1}^n \mu_{i|j}(k-1|k-1)\hat{\boldsymbol{x}}_i(k-1|k-1) \\ P_0^j(k-1|k-1) &= \sum_{i=1}^n \mu_{i|j}(k-1|k-1)\times \\ &\quad \Big(P_i(k-1|k-1) + [\hat{\boldsymbol{x}}_0^j(k-1|k-1) \\ &\quad - \hat{\boldsymbol{x}}_i(k-1|k-1)][\,.\,]^T \Big) \end{aligned} \quad (8)$$

A measurement is taken and the likelihood function of the j^{th} filter is updated to

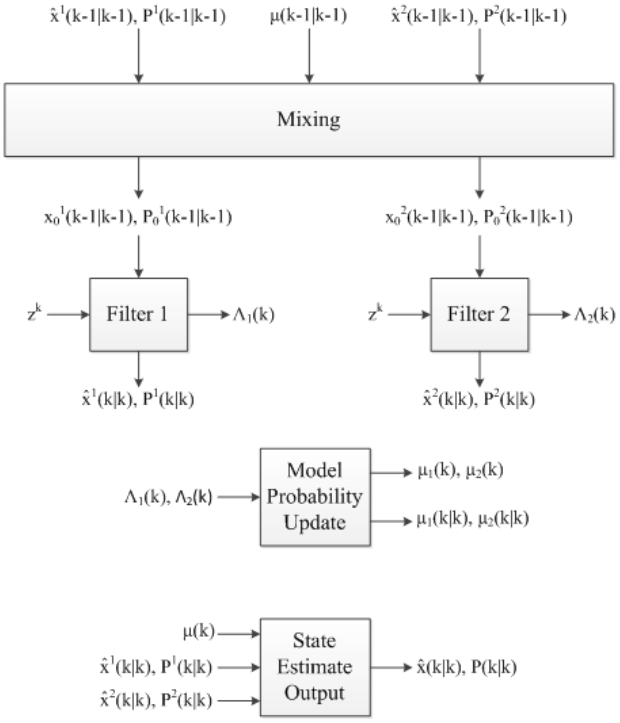

Fig. 1. A single propagation and measurement step of the IMM algorithm with two models.

$$\begin{aligned} \Lambda_j(k) &= p(z(k)|m_j(k), \hat{z}^{k-1}) \\ &= N\Big(z(k); \hat{z}(k|k-1, \hat{\boldsymbol{x}}_0^j(k-1|k-1)), \\ &\qquad S^j(k, P_0^j(k-1|k-1)) \Big) \end{aligned} \quad (9)$$

The model probabilities can be updated as

$$\mu_j(k) = \frac{1}{d} c_j \Lambda_j(k)$$

where d is another normalization constant such that $\frac{1}{d}\sum_{i=1}^n c_j \Lambda_j(k) = 1$.

The state estimates and covariance can be recovered at any time from

$$\hat{\boldsymbol{x}}(k|k) = \sum_{i=1}^n \mu_i(k)\hat{\boldsymbol{x}}_i(k|k) \quad (10)$$

$$P(k|k) = \sum_{i=1}^n \mu_i(k)\Big(P_i(k|k)[\hat{\boldsymbol{x}}(k|k) - \hat{\boldsymbol{x}}_i(k|k)][\,.\,]^T \Big) \quad (11)$$

Relevant to this application, Eq. 4 can be easily modified by removing rows and columns corresponding to $\delta \boldsymbol{l}_{ab}$. This renders the equations into a form amenable to a known, fixed lever arm. The continuous space of possible locations on the body is then discretized into a mesh. The MMKF treats each node in the mesh as a separate model for the IMU's location. The computational burden of the MMKF employed to estimate lever arm is approximately $O(3)$ in the fineness of the mesh

employed in the discretization.

C. Neural Networks

A neural network consists of an input layer, followed by a chain of matrix multiplications. After each matrix multiplication, the resulting matrix is fed through some function (typically a sigmoid or ramp function) to normalize data before being fed into the next matrix multiplication. A depiction of this structure with a network with three nodes in its input layer, two hidden layers, and an output layer with two nodes is shown below in Fig. 2.

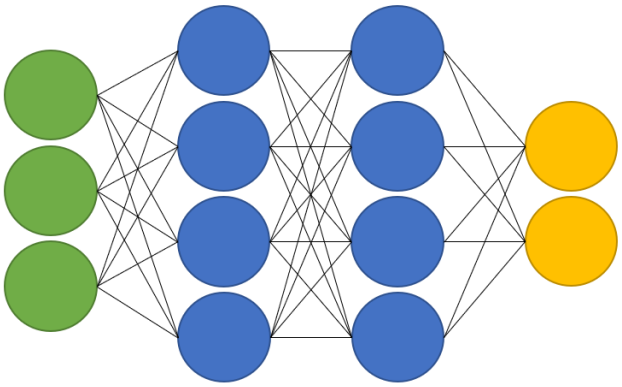

Fig. 2. Illustration of a neural network with two hidden layers, with neurons represented as colored dots and connections between neurons by lines.

The matrices used in the network are initialized randomly and trained in a process in which the network is evaluated against some known truth data, errors between the network output and the true stateare evaluated, and the network matrices are adjusted based on the backpropagated errors.

Several common modifications to network structure and training procedures exist; the results presented here only employed dropout during training, a procedure in which node outputs have a chance to become zero during training and other node outputs are reweighted to compensate [7].

III. Results

The aforementioned filters and a neural network were tested against simulated data for vehicles with one and two GNSS receivers.

A. Simulation Results: Single Antenna

A simulated quadrotor was commanded to follow random trajectories with nonzero velocities and always initialized with its local body frame aligned with the world frame. This initialization angle forces the quadrotor to pitch and/or roll through the course of its trajectory, with relatively constant pitch- and roll-rates near control saturation. The vehicle was provided with yaw commands filtered through a noisy double integrator with a constant bias, to guarantee that the vehicle rotates about its body z-axis with nonzero angular acceleration. These conditions on vehicular motion guarantee the observability of all error and bias states [1]. The trajectory against which the filters were tested is shown in Figs. 3-6 below. While some of the literature on loosely coupled extrinsic parameter estimation incorporates data taken at high attitudes of the body relative to the local frame (often in excess of 40^o), the trajectories presented here are more realistic in scale. Filter performance suffers as a result.

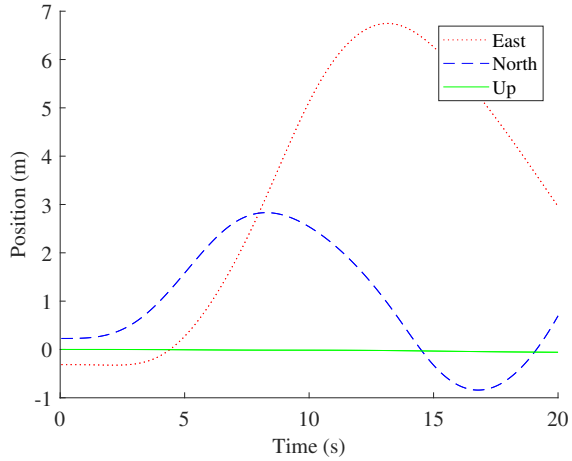

Fig. 3. Simulated true position of the quadrotor for the single- antenna validation trajectory.

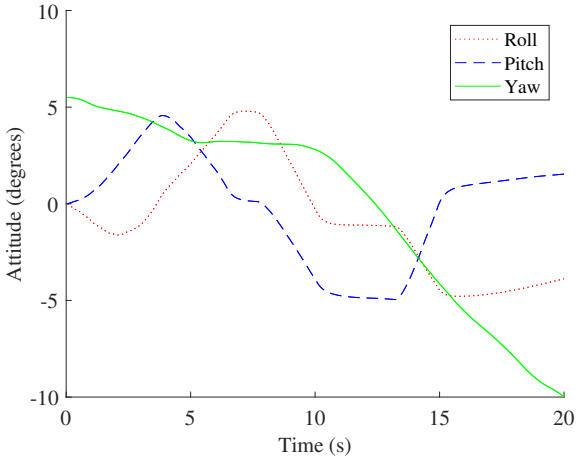

Fig. 4. Simulated true attitude of the quadrotor for the single- antenna validation trajectory. The angular motion shortly after 10 s is a strong feature for observability ([1]).

Data were simulated with a 80 Hz navigation grade IMU with each bias state modeled as a two-state Markov process ([14]) with simulated decorrelation times of 10 s (accelerometer and gyro) and 100s (accelerometer). 20 Hz GNSS measurements were simulated with a standard deviation of 2cm in each of east/north/up. The true lever arm from the GNSS unit to the IMU was also generated randomly on $N(0, I_{3x3})$ and had a value of $[0.865\ 0.321\ -0.080]^T$ meters. Two discretizations for the MMKF are presented–one grid of 50 cm intervals and

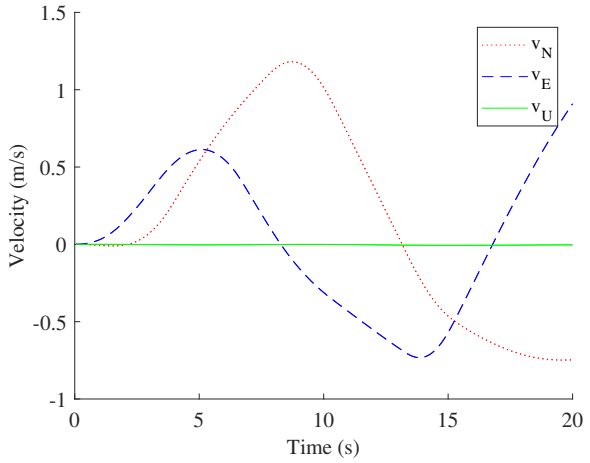

Fig. 5. Simulated true velocity of the quadrotor for the single- antenna validation trajectory.

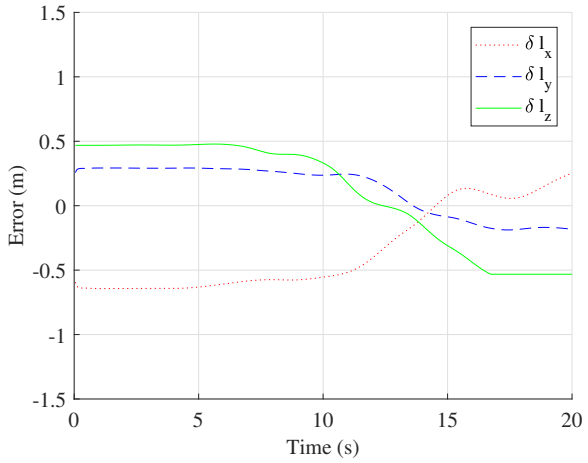

Fig. 7. Error in lever arm estimation for a simulated single-antenna quadrotor with a state augmentation EKF.

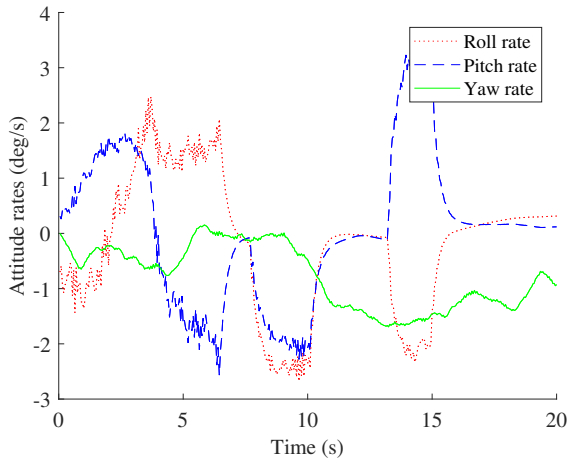

Fig. 6. Simulated true attitude rate, expressed in the body frame, of the quadrotor for the single-antenna validation trajectory. Noisy peaks in pitch and roll are the product of saturation in the low-level controller.

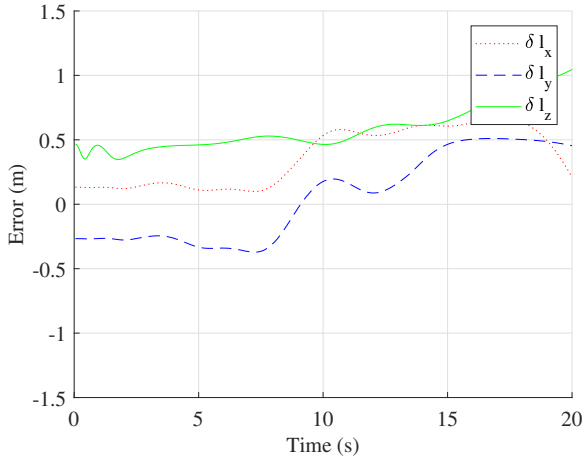

Fig. 8. Error in lever arm estimation for a simulated single-antenna quadrotor with a multiple-model filter with a 50 cm discretization.

another with a 10 cm grid. The performance of the various filters is shown below in Figs. 7-9.

The standard state augmentation EKF appears to have the best performance, with a final estimation error of 63cm. Both forms of the MMKF provide poorer performance, with final estimation errors of 99cm and 116cm for the 10cm and 50cm grids, respectively. All estimators arrive at an incorrect final value for δl_z; the vehicle's roll and pitch were roughly constant during this motion, rendering the system partially unobservable for long periods of motion. The state augmentation filter (SA) and MMKFs were initialized with the same weighted mean errors; the SA makes a large initial deviation, likely misinterpreting the initial roll necessary to begin flying the trajectory. The MMKFs are resistant to this error but are more susceptible to later errors, failing to fully reject inaccurate models.

The trajectory here is less information-rich than others in the literature. The simulated trajectories of [6] are not dynamically feasiable for quadrotors which can only apply accelerations along the body frame z-axis. Furthermore, a single-antenna solution only weakly couples the attitude state to the measurement, resulting in poorer accuracy than that of a multi-antenna design.

A neural network was trained on 14000 flights (in batches of 100) with full IMU and GNSS data provided as network inputs and the lever arm as the output state. The objective of the network was to absorb all of the data that would normally be sent sequentially to a filter for post-processing and attempt to perform the job of the filter, without knowledge of the underlying process. Though the EKF does an admirable job of handling the nonlinearities present in the problem, it was hoped that the neural network would learn the underlying filter equations in addition to finding a more accurate approximation

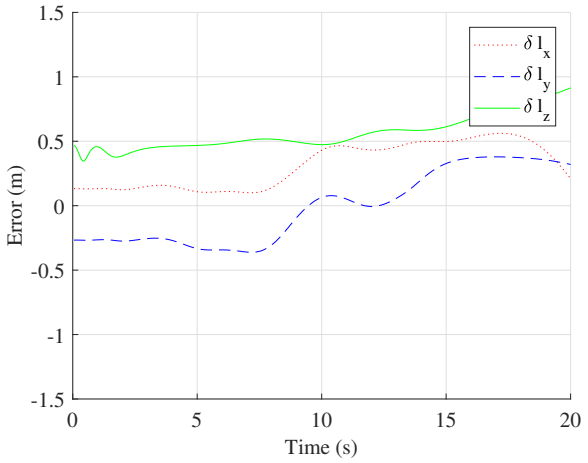

Fig. 9. Error in lever arm estimation for a simulated single-antenna quadrotor with a multiple-model filter with a 10 cm discretization.

than the standard linearization. The network was trained to view the data as a standard bandit problem, treating each input as a new data set unconnected to previous data. The most successful network tested was composed of seven layers, with 14000 nodes in its outermost layer and 40 nodes in its final layer. Intermediate layers had 11000, 8000, 5000, 2000, and 500 nodes. The network was trained using dropout with $P_{keep} = 0.75$. Other modifications to network architecture, such as convolution, did not produce noticeable effects on network accuracy.

After training, the data provided to the filters was input to the network. The neural network produced the output estimate $L_{ab,NN} = [0.1761\ 0.2729\ 0.5510]'$, with a final estimation error of 106 cm. This is close to the performance of the MMKF but slightly below the performance of the standard EKF with state augmentation.

The tried-and-true technique of state augmentation outperforms both multiple-model estimation and neural network-based approaches for navigation-grade IMU units. The results are summarized below in Table I.

TABLE I
SUMMARY OF ESTIMATION ERRORS FOR LEVER ARM ESTIMATION ON A SINGLE-ANTENNA SIMULATED QUADROTOR.

Estimation	Final error (cm)
EKF	63
MMKF (fine discretization)	99
MMKF (coarse discretization)	116
Neural network	106

B. Simulation Results: Dual Antenna

The quadrotor trajectory is less aggressive than many of those shown in the literature ([6]) and has smaller maximum attitude deviations. Furthermore, the attitude rates are noisy due to frequent control saturations encountered in the simulated controller. A second set of simulations was performed on a simulated point mass with two antennas in order to present a best-case scenario for IMU integration.

A simulated point mass was commanded to follow random trajectories with nonzero velocities and trajectories with slowly varying attitude states in order to guarantee observability. For training the neural network, as was done with the quadrotor simulation, double integrators with biased noise were used to generate all attitude components. Measurements were collected with two antennas simulated with noise of the same quality as in the single-antenna experiment. The validation trajectory, presented below, was composed of a heavy rolling sinusoid with small yawing and pitching actions used to provide observability.

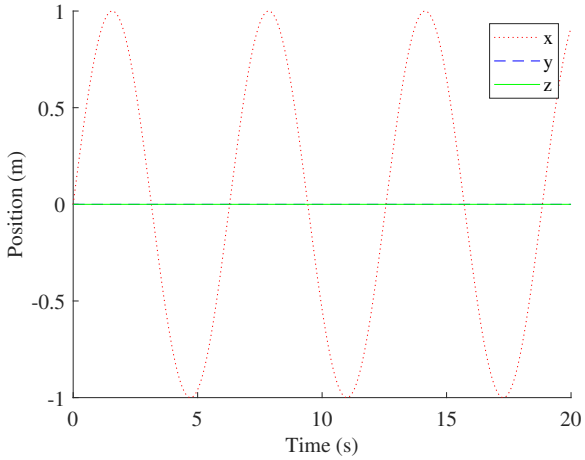

Fig. 10. Simulated true position of the point mass for dual- antenna validation trajectory.

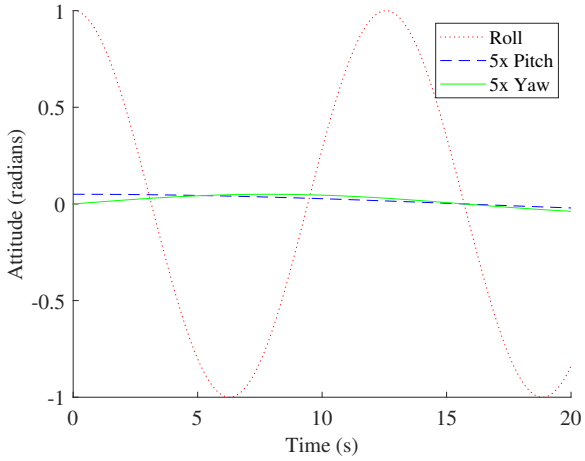

Fig. 11. Simulated true attitude of the point mass for dual- antenna validation trajectory.

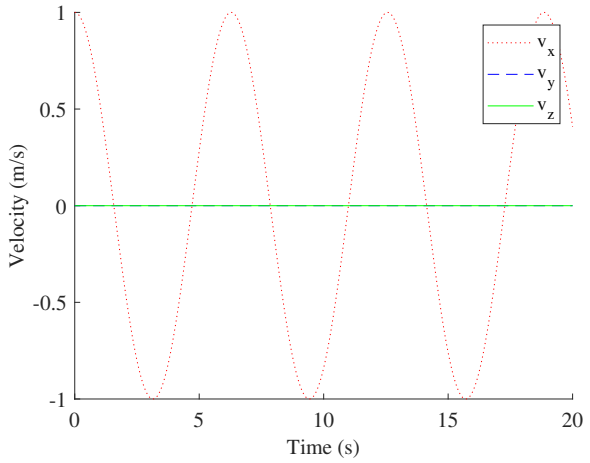

Fig. 12. Simulated true velocity of the point mass for dual- antenna validation trajectory.

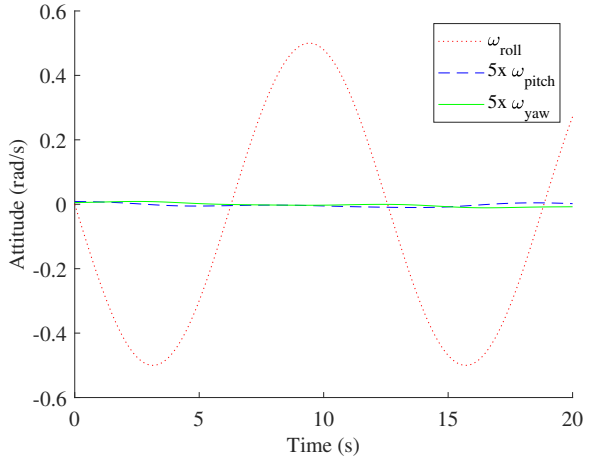

Fig. 13. Simulated true attitude rate, expressed in the body frame, of the point mass for dual-antenna validation trajectory.

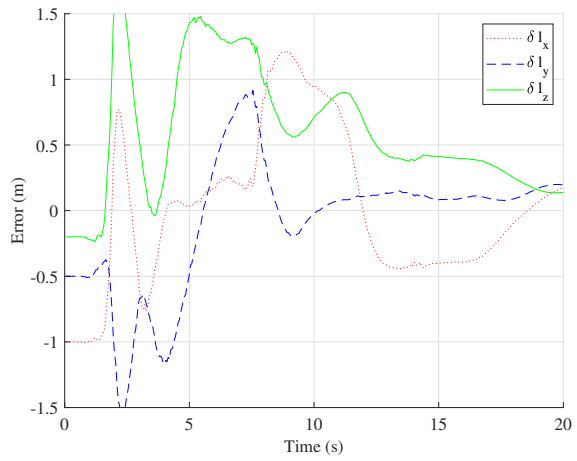

Fig. 14. Error in lever arm estimation for simulated dual-antenna point mass with a state augmentation UKF.

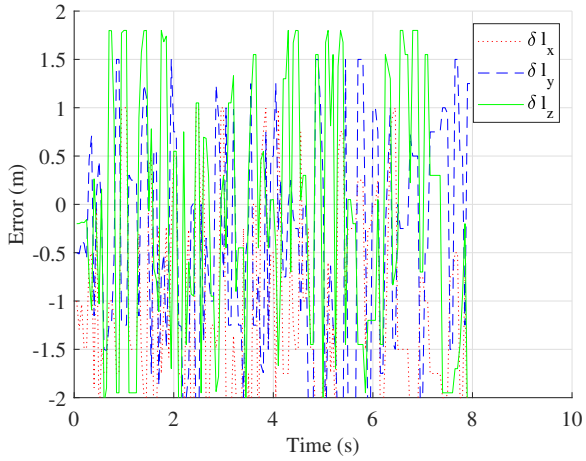

Fig. 15. Error in lever arm estimation for simulated dual-antenna point mass with a multiple-model filter with a 25 cm discretization.

Data were simulated with a 80 Hz navigation grade IMU with each bias state modeled as a two-state Markov process ([14]) with simulated decorrelation times of 100 s. 20 Hz GNSS measurements for two antennas located at $[\pm 0.18\ 0\ -0.08]^T$ cm were simulated with a standard deviation of 2 cm in each of east/north/up. The true lever arm from the center of gravity to the IMU had a value of $[1.00\ 0.50\ 0.20]^T$ meters. A MMKF with 25 cm discretizations was compared against a UKF. An EKF was implemented for this case but had poor performance due to the nonlinearities in the measurement equation. The performance of the various filters is given in Figs. 14-15.

The UKF converged to an estimate with a final error of 36 cm.

The MMKF had poor performance due to large errors introduced at the start of the estimation phase caused by overweighting of incorrect lever arms. It was hoped that restricting the models used by employing a smaller lever arm set would improve performance. The MMKF was re-evaluated with a second model set with a known direction from the IMU's location to the center of gravity but an unknown length. After tuning, the best results were obtained with a discretization of 15 cm intervals. The performance of this filter is shown in Fig. 16.

The additional information provided to the multiple-model filter greatly improves performance. The final error with this approach was 51 cm, slightly worse than the tried-and-true UKF.

A neural network was trained on 16000 flights (in batches of 10 flights per training step) with full IMU and GNSS data provided as network inputs and the lever arm as the output state. The network was trained to view the data as a standard bandit problem, treating each input as a new data set unconnected to previous data. The most successful network

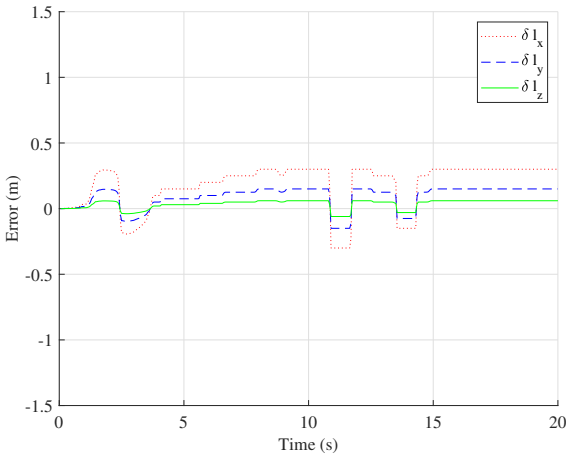

Fig. 16. Error in lever arm estimation for simulated dual-antenna point mass with a multiple-model filter with a known direction from the IMU location to the body CG.

tested was smaller than the network used in the single-antenna test, with 14000 nodes in its outermost layer and 25 nodes in its final layer. The two intermediate layers had and 270 and 190 nodes. The network was trained using dropout with $P_{keep} = 0.85$.

After training, the data provided to the filters was input to the network. The neural network provided the output estimate $L_{ab,NN} = [-0.0317\ 0.3922\ 0.1801]'$, with a final estimation error of 104 cm. The neural network had the worst performance of all estimation methods examined in this work. Kalman filtering is near-optimal (in an MMSE sense) for problems with only mild nonlinearities; any other approach would be hard-pressed to outperform a well-designed Kalman filter.

Estimator performance is summarized below in Table II.

TABLE II
SUMMARY OF ESTIMATION ERRORS FOR LEVER ARM ESTIMATION WITH DUAL-ANTENNA SIMULATED POINT MASS.

Estimation	Final error (cm)
UKF	36
MMKF (known direction)	51
Neural network	104

IV. Conclusions

A simulated experiment has been performed to examine various algorithms for loosely-coupled estimation of an IMU- and GNSS-equipped platform with unknown IMU location. Multiple model estimation over a fine discretization demonstrated similar performance to a neural network; both were outperformed by state augmentation. The neural network results presented here should be viewed as a lower bound on performance, due to the tuning-intensive and unpredictable nature of networks; due to the near-optimality of the Kalman filter for this problem, however, it is unlikely that a well-tuned or better-trained network will exceed the performance of the Kalman filter.

References

[1] S. Hong, M. H. Lee, H.-H. Chun, S.-H. Kwon, and J. L. Speyer, "Observability of error states in gps/ins integration," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 2, pp. 731–743, 2005.

[2] Y. Tang, Y. Wu, M. Wu, W. Wu, X. Hu, and L. Shen, "Ins/gps integration: Global observability analysis," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1129–1142, 2009.

[3] C. Hide, T. Moore, and M. Smith, "Adaptive kalman filtering algorithms for integrating gps and low cost ins," in *Position Location and Navigation Symposium, 2004. PLANS 2004*, pp. 227–233, IEEE, 2004.

[4] Q. M. Lam and J. L. Crassidis, "A close examination of multiple model adaptive estimation vs extended kalman filter for precision attitude determination," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, p. 5175, 2013.

[5] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[6] S. Hong, M. H. Lee, S. H. Kwon, and H. H. Chun, "A car test for the estimation of gps/ins alignment errors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 208–218, 2004.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, 2015.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] S. E. Vt and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 594–603, 1994.

[11] A. G. Parlos, S. K. Menon, and A. Atiya, "An algorithmic approach to adaptive state filtering using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1411–1432, 2001.

[12] S. Lu and T. Basar, "Robust nonlinear system identification using neural-network models," *IEEE Transactions on Neural networks*, vol. 9, no. 3, pp. 407–429, 1998.

[13] W. Flenniken, J. Wall, and D. Bevly, "Characterization of various imu error sources and the effect on navigation performance," in *Ion Gnss*, pp. 967–978, 2005.

[14] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.