Passive Resonator Localization System for Electric Autonomous Vehicles

Isaac Froisland

Electrical and Computer Engineering Department,
Utah State University
Logan, UT 84321
isaac.froisland@gmail.com

I. Introduction

All-electric autonomous vehicles (AVs) carry many advantages over internal-combustion and non-autonomous modes of transportation. Existing obstacles to widespread adoption of electric AVs include limited battery energy storage, initial cost, and the current limitations of vehicle automation. One approach to addressing these issues is the development of "connected autonomous vehicles" (CAVs) [5]. CAVs can rely on communication with a network of other connected vehicles and information systems to determine the state of the vehicle and its surroundings. CAVs can also benefit from dynamic wireless power transfer (DWPT), sometimes called "in-lane charging." DWPT for electric vehicles uses a primary coil embedded in the roadway to transmit power to a secondary coil on the vehicle. DWPT mitigates the initial cost and limited battery energy storage of electric AVs by decreasing required battery capacity and providing opportunities to recharge without stopping the vehicle. DWPT relies on precise alignment between the coils (on a scale of centimeters), and this precise alignment requires accurate state measurements of the vehicle. Current AV technology has difficulty determining vehicle state in rain, snow, and other poor visibility conditions due to its reliance on visual line-of-sight sensors and imprecise or limited GPS. Current localization technology also suffers from high power requirements and expensive computational hardware.

A. Literature Review

Existing research on vehicle localization has considered systems that do not use line-of-sight sensors (LiDAR and cameras) or GPS. Cortes [1] developed a method to sense lateral misalignment in inductive power transmission wireless charging systems using received signal strength indicator (RSSI) measurements. Cortes's research focused on static charging systems, determining the impact of vertical and horizontal spacing of the primary and secondary coils on position measurements. The results show that misalignment direction can be determined consistently with RSSI measurements, but misalignment magnitude is difficult to determine from the non-linear signal output of the sensing coils.

In [8], a phase difference of arrival (PDOA) method is used with RFID tags to localize mobile robots in an indoor environment. Compared to using a RSSI method, the PDOA method proved through simulation to have less error. The PDOA method is less susceptible to multipath signal propagation. The researchers in [8] also implemented a Kalman Filter to fuse relative positioning data from encoders with the absolute positioning data from the RFID tags. In a similar study, Hekimian-Williams et al [3] conducted physical experiments that showed localization accuracy on a scale of millimeters is achievable with a RFID PDOA system. Hekimian-Williams et al indicated how physical experiment results are directly relevant to localization applications (as opposed to purely simulated results), but it should be taken into account that the physical experiment results may depend on ideal conditions.

Other work has focused on localization in GPS-denied or GPS-limited environments. Whitaker et al [7] conducted simulations to determine the performance of "self-describing fiducials" fused with IMU data via an extended Kalman Filter (EKF) for ground-vehicle localization in GPS-denied environments. Similar fiducials are commonly used with cell-phone cameras to give the phone information relevant to the marker location. Whitaker et al determined that such a system is heavily dependent on sensor quality, but the system could provide acceptable error levels even with a consumer grade IMU. Future work along this track could include incorporating fiducial location uncertainty into the model.

Costley and Christensen in [2] developed a navigation framework that incorporates LiDAR object detection with IMU measurements to work in GPS-limited environments. The LiDAR object detection and IMU measurements are fused via an EKF to estimate the state of simulated vehicle in an orchard. The paper provides a performance evaluation that compares the position estimate error covariance of the system with the LiDAR measurements and without LiDAR. The results show a minimum of 90% improvement in 3-σ deviation for the system incorporating LiDAR over the system that did not incorporate LiDAR.

B. Proposed Contribution

This work proposes a simulation to predict the performance of a localization system for road vehicles utilizing an industrial grade IMU, GPS, and PDOA method using passive circuit resonators. The PDOA method is similar to the technology used in [8] and [3]. IMU data is used to propagate the state of the vehicle while GPS and PDOA data is used for state updates. Data from each subsystem is fused via an EKF to estimate vehicle state. Performance is measured by position estimate error covariance, and the system incorporating PDOA measurements is compared to a system that uses only IMU

and GPS data. The sensitivity of the design to measurement errors, road coil spacing, and vehicle coil configuration is analyzed. After determining the performance of the proposed system, the feasibility of using such a system for DWPT alignment and general vehicle automation is considered.

II. State Vectors

A. Coordinate System

The coordinate system, truth state, design state, navigation state, and error state is defined in this section. Mappings between states to be used in the simulation are determined and verified via a numerical example.

To predict the performance of the localization system, a Monte Carlo simulation is used. The simulated vehicle has dimensions similar to a common road vehicle. To simplify the problem, it is assumed that the IMU, GPS, and PDOA systems are all mounted rigidly with respect to each other. This means that the position and orientation of the IMU, GPS, and PDOA system relative to each other can be determined with negligible uncertainty in the simulation. Figure 1 shows the coordinate system and vectors used to define the state vectors. (Need to add GPS vector.) The vectors in the

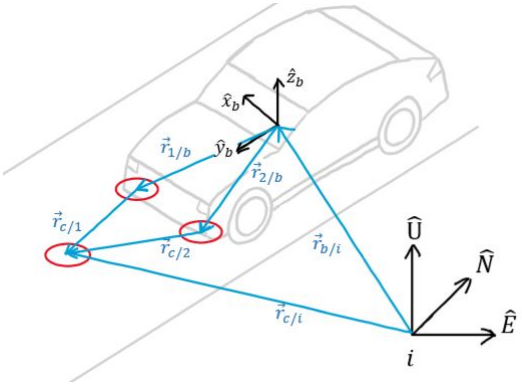

Fig. 1. Coordinate System of IMU and PDOA System

coordinate system are described as follows:

$\hat{N}, \hat{E}, \hat{U}$ = North, East, and Up unit vectors of inertial frame.
$\hat{x}_b, \hat{y}_b, \hat{z}_b$ = Unit vectors of the body frame.
$\vec{r}_{b/i}$ = Vehicle and IMU position wrt inertial frame.
$\vec{r}_{1/b}$ = Sensing coil 1 position wrt body frame.
$\vec{r}_{2/b}$ = Sensing coil 2 position wrt body frame.
$\vec{r}_{c/1}$ = Ground coil position wrt sensing coil 1 frame.
$\vec{r}_{c/2}$ = Ground coil position wrt sensing coil 2 frame.
$\vec{r}_{c/i}$ = Ground coil position wrt inertial frame.

B. State Vector Definitions

Truth State: the actual state of the system. A motorcycle model in the 2D plane is used.
Design State: includes the components of the truth state that need to be estimated by the Kalman filter.
Navigation State: the estimated state of the system with components corresponding to those of the design state.
Error State: defines the error between the design state and navigation state.

The truth state vector is $\vec{x}_t$:

$$\vec{x}_t = \begin{bmatrix} r_{E/i} & r_{N/i} & v_{y/b} & \psi & \phi & \vec{b}_a & \vec{b}_g & \vec{r}_{c/i} \end{bmatrix}^T \quad (1)$$

The components of $\vec{x}_t$ are (component dimension in braces):

$r_{E/i}$ = vehicle East-position wrt inertial frame $\{1\}$
$r_{N/i}$ = vehicle North-position wrt inertial frame $\{1\}$
v_{yb} = vehicle forward velocity $\{1\}$
ψ = vehicle heading angle (from North) $\{1\}$
ϕ = vehicle steering angle $\{1\}$
$\vec{b}_a$ = accelerometer bias $\{3\}$
$\vec{b}_g$ = gyroscope bias $\{3\}$
$\vec{r}_{c/i}$ = ground coil position wrt inertial frame $\{3\}$

Vehicle angular rate $\vec{\omega}$ is measured directly from the gyroscope, and ϕ is directly coupled with $\vec{\omega}$. The design state vector ($\vec{x}$) can omit ϕ because it not have to be estimated. $\vec{x}_t$ can be mapped to $\vec{x}$ with the following vector expansion:

$$\vec{x} = \vec{m}(\vec{x}_t) = \begin{bmatrix} \begin{bmatrix} r_{E/i} \\ r_{N/i} \\ 0 \end{bmatrix} \\ \begin{bmatrix} v_{yb}sin(\psi) \\ v_{yb}cos(\psi) \\ 0 \end{bmatrix} \\ \begin{bmatrix} cos(\frac{\psi}{2}) \\ 0 \\ 0 \\ sin(\frac{\psi}{2}) \end{bmatrix} \\ \vec{b}_a \\ \vec{b}_g \\ \vec{r}_{c/i} \end{bmatrix} \quad (2)$$

$$\vec{x} = \begin{bmatrix} \vec{r}_{b/i} & \vec{v}_{b/i} & q_{b/i} & \vec{b}_a & \vec{b}_g & \vec{r}_{c/i} \end{bmatrix}^T \quad (3)$$

The design state vector $\vec{x}$ is a conventional inertial navigation system (INS) augmented with the position of the ground coil. $q_{b/i}$ is an attitude quaternion. The navigation state vector $\hat{x}$ has estimated components corresponding to those of $\vec{x}$:

$$\hat{x} = \begin{bmatrix} \hat{r}_{b/i} & \hat{v}_{b/i} & \hat{q}_{b/i} & \hat{b}_a & \hat{b}_g & \hat{r}_{c/i} \end{bmatrix}^T \quad (4)$$

The error state vector is $\delta\vec{x}$:

$$\delta\vec{x} = \begin{bmatrix} \delta\vec{r}_{b/i} & \delta\vec{v}_{b/i} & \delta\vec{\theta}_{b/i} & \delta\vec{b}_a & \delta\vec{b}_g & \delta\vec{r}_{c/i} \end{bmatrix}^T \quad (5)$$

where each component has a dimension of $\{3\}$. In $\delta\vec{x}$, $\delta\vec{\theta}_v$ is a vector that has a dimension of $\{3\}$ that replaces the q_v and $\hat{q}_v$ with dimensions of $\{4\}$ found in $\vec{x}_t$, $\vec{x}$, and $\hat{x}$. This is because only the vector component of the quaternion needs to be estimated. The scalar component will always be approximately equal to 1 because angle changes measured by the gyroscope will be very small. That is (using a small angle approximation of $sin\,\theta \approx \theta$ and $cos\,\theta \approx 1$),

$$q_{b/i} = \begin{bmatrix} cos(\theta/2) \\ \vec{k}sin(\theta/2) \end{bmatrix} \quad \rightarrow \quad \delta q_{b/i} \approx \begin{bmatrix} 1 \\ \delta\vec{\theta}_{b/i}/2 \end{bmatrix}$$

C. State Vector Mappings

$\vec{x}$ can be related to $\hat{x}$ and $\delta\vec{x}$ by defining the following error correction mapping:

$$\vec{x} = \vec{c}(\hat{x}, \delta\vec{x}) = \begin{bmatrix} \hat{r}_{b/i} + \delta\vec{r}_{b/i} \\ \hat{v}_{b/i} + \delta\vec{v}_{b/i} \\ \begin{bmatrix} 1 \\ \delta\vec{\theta}_{b/i}/2 \end{bmatrix} \otimes \hat{q}_{b/i} \\ \hat{b}_a + \delta\vec{b}_a \\ \hat{b}_g + \delta\vec{b}_g \\ \hat{r}_{c/i} + \delta\vec{r}_{c/i} \end{bmatrix} \quad (6)$$

where $\otimes$ denotes quaternion multiplication.

(6) can be manipulated to produce an error injection mapping:

$$\hat{x} = \vec{i}(\vec{x}, \vec{\delta x}) = \begin{bmatrix} \vec{r}_{b/i} - \vec{\delta r}_{b/i} \\ \vec{v}_{b/i} - \vec{\delta v}_{b/i} \\ \begin{bmatrix} 1 \\ -\vec{\delta\theta}_{b/i}/2 \end{bmatrix} \otimes q_{b/i} \\ \vec{b}_a - \vec{\delta b}_a \\ \vec{b}_g - \vec{\delta b}_g \\ \vec{r}_{c/i} - \vec{\delta r}_{c/i} \end{bmatrix} \tag{7}$$

(7) is used to inject errors into the state estimation vector $\hat{x}$ in the simulation. Those errors are computed using an estimation error mapping, which can be found by manipulating (7):

$$\vec{\delta x} = \vec{e}(\vec{x}, \hat{x}) = \begin{bmatrix} \vec{r}_{b/i} - \hat{r}_{b/i} \\ \vec{v}_{b/i} - \hat{v}_{b/i} \\ 2 \begin{bmatrix} 0_{3\times1} & I_{3\times3} \end{bmatrix} q_{b/i} \otimes (\hat{q}_{b/i})^* \\ \vec{b}_{b/i} - \hat{b}_{b/i} \\ \vec{b}_a - \hat{b}_a \\ \vec{r}_{c/i} - \hat{r}_{c/i} \end{bmatrix} \tag{8}$$

D. Mapping Verification

Table I shows the initial conditions ($\vec{x}_0$) and injected errors ($\vec{\delta x}_i$) used to verify consistency between mappings (2), (6), (7), and (8):

TABLE I
INITIAL CONDITIONS OF $\vec{x}$ AND VALUES OF INJECTED $\vec{\delta x}$ USED TO VERIFY MAPPINGS BETWEEN STATE VECTORS

$\vec{x}$	$\vec{x}_0$	Units	$\vec{\delta x}$	$\vec{\delta x}_i$	Units
$\vec{r}_{b/i}$	$\begin{bmatrix} 0 \\ 0 \\ 0.15 \end{bmatrix}$	$[m]$	$\vec{\delta r}_{b/i}$	$\begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix}$	$[m]$
$\vec{v}_{b/i}$	$\begin{bmatrix} 0.25 \\ 1 \\ 0 \end{bmatrix}$	$[\frac{m}{s}]$	$\vec{\delta v}_{b/i}$	$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$	$[\frac{m}{s}]$
$q_{b/i}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	[unitless]	$\vec{\delta\theta}_{b/i}$	$\begin{bmatrix} 0.01 \\ 0.02 \\ 0.03 \end{bmatrix}$	$[rad]$
$\vec{b}_a$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$[g]$	$\vec{\delta b}_a$	$\begin{bmatrix} 0.001 \\ 0.002 \\ 0.003 \end{bmatrix}$	$[g]$
$\vec{b}_g$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$[\frac{deg}{hr}]$	$\vec{\delta b}_g$	$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$	$[\frac{deg}{hr}]$
$\vec{r}_{c/i}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$[m]$	$\vec{\delta r}_{c/i}$	$\begin{bmatrix} 0.11 \\ 0.22 \\ 0.33 \end{bmatrix}$	$[m]$

The mapping verification process is as follows:

1) Mapping (2) is used to map the initial conditions to the design state vector (3).
2) Mapping (7) is used to inject errors into the design state vector to produce the navigation state vector (4).
3) Mapping (8) is used to calculate the errors in the navigation state vector to produce the error state vector (5).
4) Mapping (6) is used to correct the errors in the navigation state vector to reproduce the design state vector (3).

MATLAB code is used to perform all of the calculations. Within the MATLAB code, the injected errors (shown in Table I) are compared to the errors calculated in step 3. The navigation state vector produced in step 2 is also compared with the design state vector produced in step 4. Verification steps in the code confirm that corresponding vectors have differences of less than 1×10^{-10}.

Table II compares the design state vector produced in step 1 ($\vec{x}_1$) to the design state vector produced in step 4 ($\vec{x}_4$). The table shows high computational fidelity through the sequence of state vector mappings.

TABLE II
DESIGN STATE VECTOR COMPARED ACROSS MAPPING SEQUENCE

$\vec{x}_1$	$\vec{x}_4$	Difference
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.150000000000000	0.150000000000000	0.000000000000000
0.250000000000000	0.250000000000000	0.000000000000000
1.000000000000000	1.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
1.000000000000000	1.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000
1.000000000000000	1.000000000000000	0.000000000000000
0.000000000000000	0.000000000000000	0.000000000000000

III. NONLINEAR STATE PROPAGATION AND NONLINEAR MEASUREMENT MODELING

A. Nonlinear State Propagation

This section defines the differential equations used to model the dynamics of the truth, design, and navigation vectors (1), (3), and (4), respectively. State vector propagation is based on continuous measurements from the IMU. The next section defines the measurement models used for state updates from the PDOA system.

The following continuous nonlinear differential equation represents the evolution of the truth state vector over time:

$$\dot{\vec{x}}_t = \vec{f}_t(\vec{x}_t, \vec{u}_t) + B_t \vec{w}_t \tag{9}$$

where $\vec{u}_t$ is the true input to the system, B_t is the noise-coupling matrix for process noise, and $\vec{w}_t$ is white process

noise. For the purposes of this paper, $\vec{u}_t$ is composed of vehicle acceleration a (power supplied from the vehicle's engine) and steering rate ξ:

$$\vec{u}_t = \begin{bmatrix} a \\ \xi \end{bmatrix} \tag{10}$$

The steering model in the simulation is based on Figure 2. L is the wheelbase of the vehicle and ψ is heading angle. Assuming no wheel-slip occurs, the vehicle z-axis angular rate

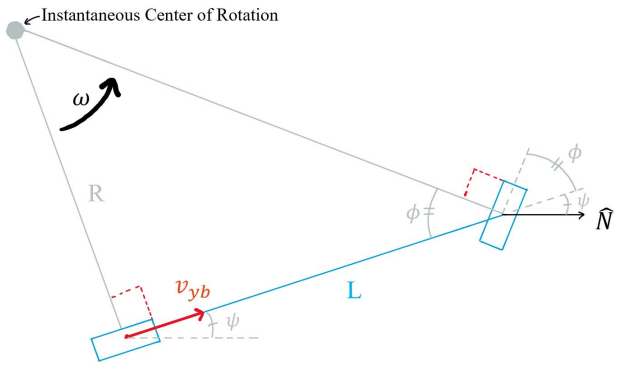

Fig. 2. Front wheel steering model

ω_z is equal to velocity v_{yb} divided by steering radius R. The radius R shown in Figure 2 can be calculated by dividing wheelbase L by $tan(\phi)$.

$$\omega_z = \frac{v_{yb}}{R} = \frac{v_{yb}}{L/tan(\phi)}$$

This results in the following rate of change of heading angle ψ:

$$\omega_z = \frac{d\psi}{dt} = \frac{v_{yb}}{L}\, tan(\phi) \tag{11}$$

Using (10) and (11), (9) can be expressed in the following way:

$$\dot{\vec{x}}_t = \frac{d}{dt}\begin{bmatrix} r_{E/i} \\ r_{N/i} \\ v_{y/b} \\ \psi \\ \phi \\ \vec{b}_a \\ \vec{b}_g \\ \vec{r}_{c/i} \end{bmatrix} = \begin{bmatrix} v_{yb} sin(\psi) \\ v_{yb} cos(\psi) \\ a \\ \frac{v_{yb}}{L}\, tan(\phi) \\ \xi \\ -\frac{1}{\tau_a}\vec{b}_a + \vec{w}_a \\ -\frac{1}{\tau_g}\vec{b}_g + \vec{w}_g \\ 0 \end{bmatrix} \tag{12}$$

where

1) τ_a and τ_g are exponentially-correlated random variable (ECRV) time-constants corresponding to accelerometer and gyroscope measurements

2) $\vec{w}_a$ and $\vec{w}_g$ are white noise vectors corresponding to accelerometer and gyroscope measurements

A nonlinear model representing the evolution of the design state vector over time has the following form:

$$\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}) + B\vec{w} \tag{13}$$

On level ground, the total angular rate vector $\vec{\omega}_{b/i}$ in the body frame is

$$\vec{\omega}_{b/i} = \begin{bmatrix} 0 & 0 & \frac{v_{yb}}{L}\, tan(\phi) \end{bmatrix}^T \tag{14}$$

Centripetal acceleration is equal to angular rate squared times radius. This means that the x-component of acceleration of the vehicle is

$$a_x = R\omega_z^2 = \frac{L}{tan(\phi)}\left(\frac{v_{yb}}{L}\, tan(\phi)\right)^2$$

Simplifying and adding the y and z-components, the total acceleration vector of the vehicle is

$$\vec{a}_b = \begin{bmatrix} \frac{v_{yb}^2}{L}\, tan(\phi) & a & g \end{bmatrix}^T \tag{15}$$

To transform the body frame acceleration vector $\vec{a}_b$ to the inertial frame, the following transformation is used:

$$\vec{a}_{b/i} = T_b^i \vec{a}_b \tag{16}$$

T_b^i is a direction cosine matrix (DCM) extracted from the attitude quaternion. Using (14), (15), and (16), (13) can be expressed in the following way:

$$\dot{\vec{x}} = \frac{d}{dt}\begin{bmatrix} \vec{r}_{b/i} \\ \vec{v}_{b/i} \\ q_{b/i} \\ \vec{b}_a \\ \vec{b}_g \\ \vec{r}_{c/i} \end{bmatrix} = \begin{bmatrix} \vec{v}_{b/i} \\ T_b^i \vec{a}_b \\ \frac{1}{2}\begin{bmatrix} 0 \\ \vec{\omega}_{b/i} \end{bmatrix} \otimes q_v \\ -\frac{1}{\tau_a}\vec{b}_a + \vec{w}_a \\ -\frac{1}{\tau_g}\vec{b}_g + \vec{w}_g \\ 0 \end{bmatrix} \tag{17}$$

A nonlinear model representing the evolution of the navigation state vector over time has the following form:

$$\dot{\hat{x}} = \vec{f}(\hat{x}, \vec{u}) \tag{18}$$

Measurements from the accelerometer and gyroscope are composed of measured, bias, and noise components. The measured angular rate $\tilde{\omega}_{b/i}$ is

$$\tilde{\omega}_{b/i} = \begin{bmatrix} 0 & 0 & \frac{v_{yb}}{L}\, tan(\phi) \end{bmatrix}^T + \vec{b}_g + \vec{w}_g \tag{19}$$

The measured acceleration $\tilde{a}_b$ of the vehicle is

$$\tilde{a}_b = \begin{bmatrix} \frac{v_{yb}^2}{L}\, tan(\phi) & a & g \end{bmatrix}^T + \vec{b}_a + \vec{w}_a \tag{20}$$

In the case of the navigation model, the state vector accounts for estimated bias components. The estimated acceleration $\hat{a}_b$ of the vehicle in the body frame is

$$\hat{a}_b = \tilde{a}_b - \hat{b}_a \tag{21}$$

Similar to (16), the following operation transforms the estimated acceleration vector from the body frame to the inertial frame:

$$\hat{a}_{b/i} = T_b^i \hat{a}_b \tag{22}$$

The estimated angular rate $\hat{\omega}_{b/i}$ of the vehicle on level ground is

$$\hat{\omega}_{b/i} = \tilde{\omega}_{b/i} - \hat{b}_g \tag{23}$$

Using (21), (22), and (23), (18) can be expressed in the following way:

$$\dot{\hat{x}} = \frac{d}{dt}\begin{bmatrix} \hat{r}_{b/i} \\ \hat{v}_{b/i} \\ \hat{q}_{b/i} \\ \hat{b}_a \\ \hat{b}_g \\ \hat{r}_{c/i} \end{bmatrix} = \begin{bmatrix} \hat{v}_{b/i} \\ \hat{T}_b^i \hat{a}_b - \vec{g} \\ \frac{1}{2}\begin{bmatrix} 0 \\ \hat{\omega}_{b/i} \end{bmatrix} \otimes \hat{q}_v \\ -\frac{1}{\tau_a}\hat{b}_a \\ -\frac{1}{\tau_g}\hat{b}_g \\ 0 \end{bmatrix} \tag{24}$$

B. Nonlinear Measurement Modeling

This section defines the discrete measurement models used for state updates from the PDOA system. For the purposes of this paper, it is assumed that the PDOA system has already been implemented and the phase measurements can be injected into the simulation. The estimated phase difference $\hat{\alpha}$ equal to the maximum likelihood estimate is

$$\hat{\alpha} = \phi_1 - \phi_2 \tag{25}$$

where ϕ_1 and ϕ_2 are the phase measurements associated with each sensing coil.

The associated variance $var(\hat{\alpha})$ can be approximated as follows:

$$var(\hat{\alpha}) \approx \frac{2}{T}\left[\frac{Q_1^2}{A_1^2} + \frac{Q_2^2}{A_2^2}\right] \tag{26}$$

where

T = Integration time. This is the time required for the analog circuit in the ground coil to begin re-transmitting.
Q_1, Q_2 = power spectral densities associated with each sensing coil measurement.
A_1, A_2 = amplitudes of the signals received by the sensing coils.

Reference for (25) and (26)?

Nikitin et al in [6] lay out a linear relationship between phase accumulated due to electromagnetic wave propagation and the distance between the emitter and receiver. For the purposes of this paper, phase from other sources is not considered. Adapting the model from [6] for a system only concerned with one-way transmission, this relationship is

$$\phi_{prop} = \frac{-2\pi f}{c} d \tag{27}$$

where

f = carrier frequency of the ground coil circuit
c = speed of light in air
d = distance between emitter and receiver

A model of a discrete measurement $\tilde{z}_t$ for the design state (true data) has the following form:

$$\tilde{z}_t = \vec{h}_t(\vec{x}_t) + G_t\vec{\nu}_t \tag{28}$$

where G_t is the noise-coupling matrix for measurement noise and $\vec{\nu}_t$ is white measurement noise.

The discrete measurement $\tilde{z}_{p,t}$ for true phase difference can be modeled as

$$\tilde{z}_{p,t} = \hat{\alpha}_t + \nu_{p,t}$$

Substituting in (25),

$$\tilde{z}_{p,t} = \phi_{1,t} - \phi_{2,t} + \nu_{p,t}$$

Substituting in (27) for $\phi_{1,t}$ and $\phi_{2,t}$,

$$\tilde{z}_{p,t} = \frac{2\pi f}{c}(d_2 - d_1) + \nu_{p,t} \tag{29}$$

For the system shown in Figure 1, the distances d_1 and d_2 are the magnitudes of the vectors $\vec{r}_{c/1}$ and $\vec{r}_{c/2}$, respectively. To match (29) to the form of (28), vector addition can be used to express the magnitudes of $\vec{r}_{c/1}$ and $\vec{r}_{c/2}$ in terms of the state variables defined in 1. Performing "tip-to-tail" vector addition and transforming $\vec{r}_{1/b}$ to the inertial frame, the magnitude of $\vec{r}_{c/1}$ can be expressed as

$$d_1 = ||\vec{r}_c - \vec{r}_v - T_b^i\vec{r}_{1/b}|| \tag{30}$$

Similarly, the magnitude of $\vec{r}_{c/2}$ can be expressed as

$$d_2 = ||\vec{r}_{c/i} - \vec{r}_{b/i} - T_b^i\vec{r}_{2/b}|| \tag{31}$$

(30) and (31) can be substituted into (29) to produce a relationship that matches the form of (28). A model of a discrete measurement $\tilde{z}$ for the navigation state (estimated data) has the following form:

$$\tilde{z} = \vec{h}(\vec{x}) + G\vec{\nu} \tag{32}$$

Following the same process as for (29), the discrete measurement $\tilde{z}_p$ for the navigation model phase difference can be modeled as

$$\tilde{z}_p = \frac{2\pi f}{c}(d_2 - d_1) + \nu_p \tag{33}$$

(33) is used to estimate the phase difference measurements based on the navigation states while (29) is used to synthesize the measurements in the simulation.

C. Dynamics & Measurement Model Verification

To verify the dynamics and measurement models, a single simulation is run to ensure that the estimated navigation state values are close to the design state values. That is, the results of propagating the state with (17) are compared to the results of propagating the state with (24). The results of synthesizing phase difference measurements with (29) are compared with the results of estimating phase difference measurements using (33). Mapping (8) is used to calculate the errors. A fourth-order Runge-Kutta integration method is used to propagate the design and navigation states. Using a time-step of 0.001s and a carrier frequency of 122Mhz, Figure 3 shows the estimated position of the vehicle versus the true position. The trajectories effectively overlap the entire duration of the simulation.

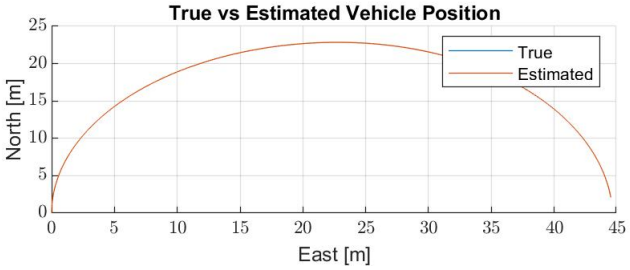

Fig. 3.

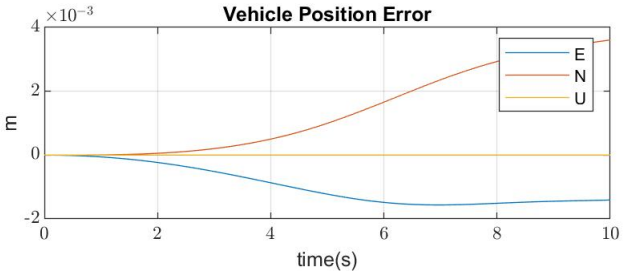

Fig. 4.

Figure 4 shows that the scale of the position errors is in millimeters for a simulation lasting 10 seconds. Figures 5, 6,

7, 8, and 9 all show similar scales of errors for estimated states in simulations lasting 10 seconds.

Figure 10 shows the estimated phase difference measurements versus the true, synthesized phase difference measurements. Again, the trajectories effectively overlap over the duration of the simulation.

Figure 11 shows the residuals of the two sets of phase difference measurements. The errors for the estimated phase difference measurements are within 1×10^{-3} radians of the true measurements.

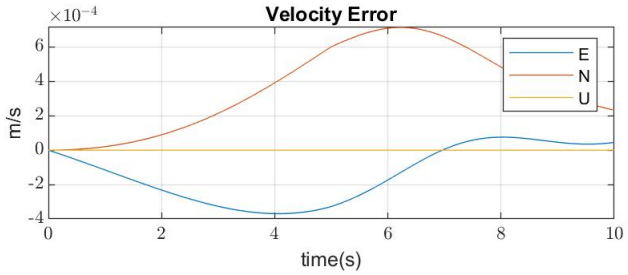

Fig. 5.

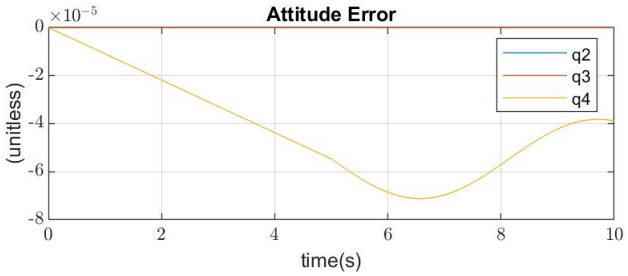

Fig. 6.

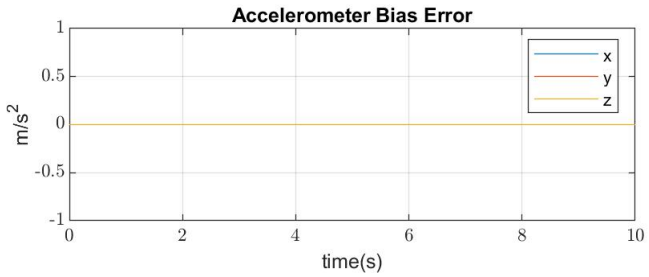

Fig. 7.

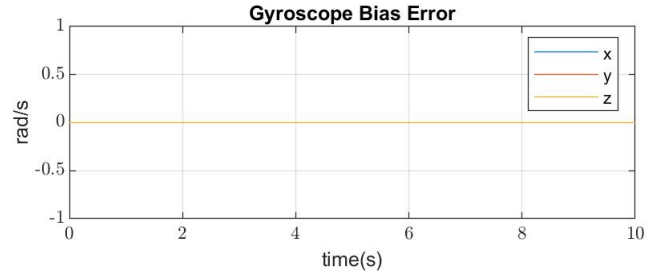

Fig. 8.

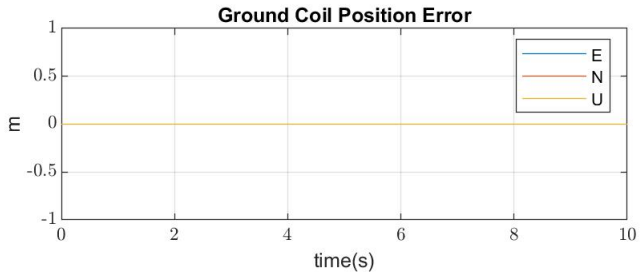

Fig. 9.

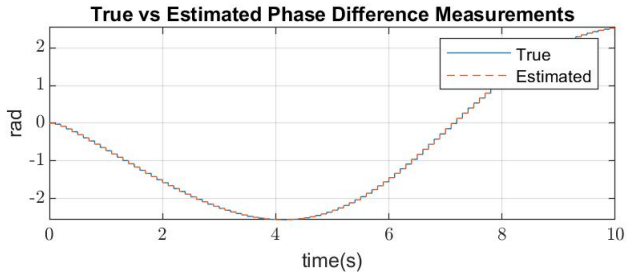

Fig. 10.

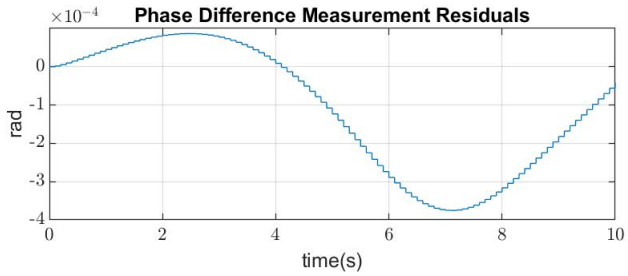

Fig. 11.

IV. Linear Error State Modeling and Verification

A. Linear Error State Modeling

This section defines the linear perturbation models of the state dynamics. These linearizations are performed about the current state estimate in the EKF. The linearized error state dynamics have the following form:

$$\dot{\vec{\delta x}} = F(\hat{x})\vec{\delta x} + B\vec{w} \tag{34}$$

The purpose of this linearization is to generate the matrix $F(\hat{x})$ which is used to the propagate covariances. For the vehicle, let

$$\vec{r}_{b/i} \triangleq \hat{r}_{b/i} + \vec{\delta r}_{b/i} \tag{35}$$

and

$$\vec{v}_{b/i} \triangleq \hat{v}_{b/i} + \vec{\delta v}_{b/i} \tag{36}$$

The time-derivative of (35) is

$$\dot{\vec{r}}_{b/i} = \dot{\hat{r}}_{b/i} + \dot{\vec{\delta r}}_{b/i} \tag{37}$$

Setting (37) equal to (36),

$$\dot{\hat{r}}_{b/i} + \dot{\vec{\delta r}}_{b/i} = \hat{v}_{b/i} + \vec{\delta v}_{b/i}$$

The first term on the left is equivalent to the first term on the right side. Cancelling these two terms produces the linear error state for the vehicle position matching the form of (34):

$$\dot{\vec{\delta r}}_{b/i} = \vec{\delta v}_{b/i} \tag{38}$$

The time derivative of (36) is

$$\dot{\vec{v}}_{b/i} = \dot{\hat{v}}_{b/i} + \dot{\vec{\delta v}}_{b/i} \tag{39}$$

The time derivative of $\hat{v}_{b/i}$ can be expressed as

$$\dot{\hat{v}}_{b/i} = \hat{T}_b^i \tilde{a}_b - \hat{T}_b^i \hat{b}_a - \vec{g} \tag{40}$$

The time derivative of the vehicle's true velocity is

$$\dot{\vec{v}}_{b/i} = T_b^i(\tilde{a}_b - \vec{b}_a - \vec{n}_a) - \vec{g} \tag{41}$$

Let

$$\vec{b}_a \triangleq \hat{b}_a + \vec{\delta b}_a \tag{42}$$

The transformation matrix T_b^i used in the nonlinear design state vector(17) can be expressed as

$$T_b^i = \hat{T}_b^i[I + (\delta\theta)\times] \tag{43}$$

where $\times$ is the cross operator. Setting (39) equal to (41), substituting (40) into (39), and substituting (43) and (42) into (41) produces

$$\hat{T}_b^i\tilde{a}_b - \hat{T}_b^i\hat{b}_a - \vec{g} + \dot{\vec{\delta v}}_{b/i} = \hat{T}_b^i[I+(\delta\theta)\times](\tilde{a}_b - \hat{b}_a - \vec{\delta b}_a - \vec{w}_a) - \vec{g}$$

Distributing terms,

$$\begin{aligned}\hat{T}_b^i\tilde{a}_b - \hat{T}_b^i\hat{b}_a - \vec{g} + \dot{\vec{\delta v}}_{b/i} = {} & \hat{T}_b^i\tilde{a}_b - \hat{T}_b^i\hat{b}_a - \hat{T}_b^i\vec{\delta b}_a - \hat{T}_b^i\vec{w}_a \\ & + \hat{T}_b^i(\delta\theta)\times\tilde{a}_b - \hat{T}_b^i(\delta\theta)\times\hat{b}_a \\ & - \hat{T}_b^i(\delta\theta)\times\vec{\delta b}_a - \hat{T}_b^i(\delta\theta)\times\vec{w}_a \\ & - \vec{g}\end{aligned}$$

The terms $\hat{T}_b^i\tilde{a}_b$, $\hat{T}_b^i\hat{b}_a$, and $\vec{g}$ appear on both sides of the equation and can be cancelled out. The terms $\hat{T}_b^i(\delta\theta)\times\vec{\delta b}_a$ and $\hat{T}_b^i(\delta\theta)\times\vec{w}_a$ can be approximated as zero. This approximation is valid because a very small number multiplied by another very small number produces a negligible value. These simplifications produce

$$\dot{\vec{\delta v}}_{b/i} = -\hat{T}_b^i\vec{\delta b}_a - \hat{T}_b^i\vec{w}_a + \hat{T}_b^i(\delta\theta)\times\tilde{a}_b - \hat{T}_b^i(\delta\theta)\times\hat{b}_a$$

Combining terms and rearranging,

$$\dot{\vec{\delta v}}_{b/i} = \hat{T}_b^i(\delta\theta)\times(\tilde{a}_b - \hat{b}_a) - \hat{T}_b^i\vec{\delta b}_a - \hat{T}_b^i\vec{w}_a$$

A cross product can be negated and reversed for an equivalent form. Performing this operation on $(\delta\theta)\times(\tilde{a}_b - \hat{b}_a)$ produces the linear error state for the vehicle velocity matching the form of (34):

$$\dot{\vec{\delta v}}_{b/i} = -\hat{T}_b^i(\tilde{a}_b - \hat{b}_a)\times\delta\theta - \hat{T}_b^i\vec{\delta b}_a - \hat{T}_b^i\vec{w}_a \tag{44}$$

Markley and Crassidis in [4] give the following relationship for rate of change of attitude errors:

$$\dot{\vec{\delta\theta}} = \vec{\omega}_{b/i} - \hat{\omega}_{b/i} - \hat{\omega}_{b/i}\times\vec{\delta\theta} \tag{45}$$

The measured angular rate defined in (19) can be substituted into the estimated angular rate defined in (23):

$$\hat{\omega}_{b/i} = \vec{\omega}_{b/i} + \vec{b}_g + \vec{w}_g - \hat{b}_g$$

Substituting this into the first $\hat{\omega}_{b/i}$ term of (45),

$$\dot{\vec{\delta\theta}} = \vec{\omega}_{b/i} - (\vec{\omega}_{b/i} + \vec{b}_g + \vec{w}_g - \hat{b}_g) - \hat{\omega}_{b/i}\times\vec{\delta\theta}$$

Cancelling out terms and rearranging,

$$\dot{\vec{\delta\theta}} = -(\vec{b}_g - \hat{b}_g) - \hat{\omega}_{b/i}\times\vec{\delta\theta} - \vec{w}_g$$

The term $(\vec{b}_g - \hat{b}_g)$ is equal to $\vec{\delta b}_g$. Substituting in this definition produces the linear error state for the vehicle attitude matching the form of (34):

$$\dot{\vec{\delta\theta}} = -\vec{\delta b}_g - \hat{\omega}_{b/i}\times\vec{\delta\theta} - \vec{w}_g \tag{46}$$

The time derivative of (42) is

$$\dot{\vec{b}}_a = \dot{\hat{b}}_a + \dot{\vec{\delta b}}_a \tag{47}$$

From the design vector (17),

$$\dot{\vec{b}}_a = -\frac{1}{\tau_a}\vec{b}_a + \vec{w}_a \tag{48}$$

Setting (47) equal to (48),

$$\dot{\hat{b}}_a + \dot{\vec{\delta b}}_a = -\frac{1}{\tau_a}\vec{b}_a + \vec{w}_a$$

Substituting in (42) for $\vec{b}_a$,

$$\dot{\hat{b}}_a + \dot{\vec{\delta b}}_a = -\frac{1}{\tau_a}(\hat{b}_a + \vec{\delta b}_a) + \vec{w}_a$$

Distributing terms,

$$\dot{\hat{b}}_a + \dot{\vec{\delta b}}_a = -\frac{\hat{b}_a}{\tau_a} - \frac{\vec{\delta b}_a}{\tau_a} + \vec{w}_a$$

Because $\dot{\hat{b}}_a = -\frac{\hat{b}_a}{\tau_a}$, the first term on the left cancels out the first term on the right. This results in the linear error state for the accelerometer bias matching the form of (34):

$$\dot{\vec{\delta b}}_a = -\frac{1}{\tau_a}\vec{\delta b}_a + \vec{w}_a \tag{49}$$

Letting $\vec{b}_g \triangleq \hat{b}_g + \vec{\delta b}_g$ and following the same process for the gyroscope bias as for the accelerometer bias, the linear error state for the gyroscope bias matching the form of (34) is

$$\dot{\vec{\delta b}}_g = -\frac{1}{\tau_g}\vec{\delta b}_g + \vec{w}_g \tag{50}$$

Because the ground coil position does not change over time, the linear error state for the ground coil position matching the form of (34) is

$$\dot{\vec{\delta r}}_{c/i} = 0 \tag{51}$$

(38), (44), (46), (49), (50), and (51) can all be combined to form a matrix equation matching the form of (34):

$$\dot{\vec{\delta x}} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\hat{T}_b^i(\tilde{a}_b - \hat{b}_a)\times & -\hat{T}_b^i & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -\hat{\omega}_{b/i}\times & 0_{3\times3} & -I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_a}I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & -\frac{1}{\tau_g}I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}\vec{\delta x} + \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ -\hat{T}_b^i & 0_{3\times3} \\ 0_{3\times3} & -I_{3\times3} \\ I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}\begin{bmatrix}\vec{w}_a \\ \vec{w}_g\end{bmatrix} \tag{52}$$

B. Linear Error State Model Verification

To verify the matrix $F(\hat{x})$, the error state vector is propagated over one Kalman cycle using (52). The estimated state $\hat{x}$ and truth state $\vec{x}_t$ are propagated using (24) and (12, respectively. The estimation error mapping (8) is used to calculate the error associated with propagating the states via the nonlinear dynamics. This "nonlinear error" is compared with the "linear error" generated via (52) to confirm that the two are close enough.

Using the same injected errors as was used in mapping verification, Tables III, IV, and V show the errors corresponding to Kalman cycles of 1, 0.1, and 0.01 seconds, respectively. The residuals are the differences between the linear and nonlinear error states.

TABLE III
ERROR STATE VECTOR COMPARISON WITH KALMAN CYCLE OF 1 SECOND

Linear Error State	Nonlinear Error State	Residual
1.0961	1.197	-0.10094
2.1941	2.1203	0.073841
3.2891	3.2987	-0.0095737
0.99249	1.1888	-0.19628
1.9906	1.8341	0.1565
2.9813	3.0012	-0.019854
-0.005642	0.014287	-0.019929
-0.026358	0.017193	-0.04355
0.025942	0.02998	-0.0040382
0.0036089	0.0036089	0
0.0072178	0.0072178	0
0.010827	0.010827	0
1.7835e-06	1.7835e-06	0
3.5671e-06	3.5671e-06	0
5.3506e-06	5.3506e-06	0
0.11	0.11	0
0.22	0.22	0
0.33	0.33	0

TABLE IV
ERROR STATE VECTOR COMPARISON WITH KALMAN CYCLE OF 0.1 SECOND

Linear Error State	Nonlinear Error State	Residual
0.19998	0.20101	-0.001034
0.3999	0.39926	0.00064812
0.59986	0.59995	-8.9753e-05
0.99956	1.0202	-0.020625
1.9982	1.9851	0.01315
2.9972	2.999	-0.001809
-0.013731	0.010437	-0.024168
0.02146	0.019774	0.0016859
0.027401	0.029998	-0.002597
0.0088765	0.0088765	0
0.017753	0.017753	0
0.026629	0.026629	0
4.3868e-06	4.3868e-06	0
8.7736e-06	8.7736e-06	0
1.316e-05	1.316e-05	0
0.11	0.11	0
0.22	0.22	0
0.33	0.33	0

Table IV shows residuals on a scale of millimeters for the position error while Table III has corresponding residuals up to 100 millimeters. This indicates that the required Kalman update cycle for the system would need to be somewhere between 1 and 0.1 seconds in order to maintain a position estimate accurate on a scale of centimeters for the errors injected in this simulation. In a real system, smaller errors from higher precision sensors would allow for longer Kalman update cycles.

TABLE V
ERROR STATE VECTOR COMPARISON WITH KALMAN CYCLE OF 0.01 SECOND

Linear Error State	Nonlinear Error State	Residual
0.11	0.11001	-1.041e-05
0.22	0.21999	6.3154e-06
0.33	0.33	-8.8368e-07
0.99995	1.002	-0.002081
1.9998	1.9985	0.0012648
2.9997	2.9999	-0.0001769
0.0078276	0.010044	-0.002216
0.021203	0.019978	0.0012248
0.029819	0.03	-0.00018102
0.0097124	0.0097124	0
0.019425	0.019425	0
0.029137	0.029137	0
4.7999e-06	4.7999e-06	0
9.5998e-06	9.5998e-06	0
1.44e-05	1.44e-05	0
0.11	0.11	0
0.22	0.22	0
0.33	0.33	0

REFERENCES

[1] Ivan Cortes. "Automatic Positioning System for Inductive Wireless Charging Devices and Application to Mobile Robot". en. Accepted: 2018-02-05T21:16:43Z. Thesis. July 2017. URL: https://oaktrust.library.tamu.edu/handle/1969.1/165925 (visited on 09/20/2021).

[2] Austin Costley and Randall Christensen. "Landmark Aided GPS-Denied Navigation for Orchards and Vineyards". In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. ISSN: 2153-3598. Apr. 2020, pp. 987–995. DOI: 10.1109/PLANS46316.2020.9110130.

[3] Cory Hekimian-Williams et al. "Accurate localization of RFID tags using phase difference". In: *2010 IEEE International Conference on RFID (IEEE RFID 2010)*. ISSN: 2374-0221. Apr. 2010, pp. 89–96. DOI: 10.1109/RFID.2010.5467268.

[4] F. Landis Markley and John L. Crassidis. *Fundamentals of spacecraft attitude determination and control*. Space technology library 33. OCLC: ocn882605422. New York: Springer, 2014. ISBN: 978-1-4939-0801-1.

[5] Margarita Martínez-Díaz and Francesc Soriguera. "Autonomous vehicles: theoretical and practical challenges". en. In: *Transportation Research Procedia*. XIII Conference on Transport Engineering, CIT2018 33 (Jan. 2018), pp. 275–282. ISSN: 2352-1465. DOI: 10.1016/j.trpro.2018.10.103. URL: https://www.sciencedirect.com/science/article/pii/S2352146518302606 (visited on 09/21/2021).

[6] Pavel V. Nikitin et al. "Phase based spatial identification of UHF RFID tags". en. In: IEEE, Apr. 2010, pp. 102–109. ISBN: 978-1-4244-5742-7 978-1-4244-5743-4. DOI: 10.1109/RFID.2010.5467253. URL: http://ieeexplore.ieee.org/document/5467253/ (visited on 07/16/2018).

[7] Justin Whitaker, Randall Christensen, and Greg Droge. "Global Localization of Ground Vehicles Using Self-Describing Fiducials Coupled with IMU Data". In: *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. ISSN: 2153-3598. Apr. 2020, pp. 186–196. DOI: 10.1109/PLANS46316.2020.9109829.

[8] Yunlei Zhang et al. “Localization and Tracking of an Indoor Autonomous Vehicle Based on the Phase Difference of Passive UHF RFID Signals”. en. In: *Sensors* 21.9 (May 2021), p. 3286. ISSN: 1424-8220. DOI: 10.3390/s21093286. URL: https://www.mdpi.com/1424-8220/21/9/3286 (visited on 09/20/2021).