World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:5, No:8, 2011

Low-Cost and Highly Accurate Motion Models for Three-Dimensional Local Landmark-based Autonomous Navigation

Gheorghe Galben and Daniel N. Aloi

Abstract**—**Recently, the Spherical Motion Models (SMM's) have been introduced [1]. These new models have been developed for 3D local landmark-base Autonomous Navigation (AN). This paper is revealing new arguments and experimental results to support the SMM's characteristics. The accuracy and the robustness in performing a specific task are the main concerns of the new investigations. To analyze their performances of the SMM's, the most powerful tools of estimation theory, the extended Kalman filter (EKF) and unscented Kalman filter (UKF), which give the best estimations in noisy environments, have been employed. The Monte Carlo validation implementations used to test the stability and robustness of the models have been employed as well.

Keywords**—**Autonomous Navigation; Extended Kalman Filter; Unscented Kalman Filter; Localization Algorithms

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

I. Introduction

ALMOST all applications of Autonomous Navigation (AN) require a vehicle capable of moving accurately and repeatedly to a particular location within its environment while executing a specific task including autonomous navigation. With the commercial development of Autonomous Vehicles AV's in applications such as surface and underground mining , agriculture , and cargo handling , there has been a corresponding development of navigation systems [3], [4]. Landmark-based navigation systems are widely adopted for applications requiring high accuracy and adaptability [5], [6], [7]. Landmarks are distinct features that an AV can recognize from its sensory input. In general, landmarks have a fixed and known position relative to which an AV can localize itself.

Through [1], the 2D reference system motion model which corresponds the most with the control and estimation technique has been extended to 3D reference system. The Spherical Motion Models (SMM's) have been introduced in deterministic and probabilistic forms [1], [2]. It is proven already that the SMM's are highly accurate, robust, and low-cost models.

This paper first, presents minimum necessary to make the subject comprehensive and then brings new scientific arguments and experimental results to support the SMM's characteristics. To analyze their performances of the SMM's, the Localization Algorithms (LA's), using the extended Kalman filter (EKF) and unscented Kalman filter (UKF) are employed. The Monte Carlo validation implementations strategy is used to prove the robustness and stability of models.

Gheorghe Galben is with Electrical and Computer Engineering Department, Oakland University, Rochester, MI, 48309 USA. He is also with iTrack LLC, 2200 North Squirrel Road, Rochester, MI, 48309 USA (phone: 248-613-1826; fax: 248-648-4799; e-mail: ggalben@ sbcglobal.net).

Daniel N. Aloi is an Associate Professor and Acting Chair in the Electrical and Computer Engineering Department and Founder/Director of the Applied Electromagnetic and Wireless Laboratory at Oakland University, Rochester, MI, 48309 USA (e-mail: aloi@oakland.edu).

II. Virtual Vehicle

Given a control protocol independently specifying speeds for translation and rotation, it is natural to build a control cycle operating in terms of these parameters [6], [7], [8]. This may be accomplished by translating the vehicle geometry into a virtual vehicle, which operates in terms of rotation and translation.

Consider a differentially steered vehicle having a rectangular form as shown in Fig 1. The vehicle is translated using two independently controlled power wheels located on the axis aligned with the base. The origin of the vehicle coordinate system is located midway between the power wheels. All vehicle position measurements are specified with respect to this reference point. Two large caster wheels are mounted in the front of the vehicle to maintain balance. A Self Leveled Platform (SLP) is mounted on top of the vehicle. The SLP acts as a neck for a Laser Scanner (LS).

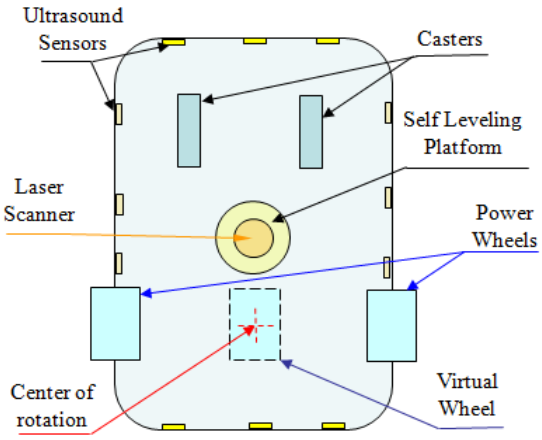


Fig. 1 Illustration of virtual vehicle

III. 2D Models

Vehicle kinematics for an AV operating in a planar (horizontal) 2D environment will now be examined.

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

A. Kinematics Configuration

Kinematics is the calculus describing the effect of control actions on the configuration of a vehicle. The *configuration* of a rigid vehicle is commonly described by its two-dimensional Cartesian coordinates (x, y) and angular orientation θ relative to an external coordinate frame.

The kinematic state is summarized by three variables (x, y, θ), referred to as *position*. The *position* of the AV in a plane described by the vector: $(x, y, \theta)^T$.

The orientation of the AV is called the bearing, or heading direction. By convention orientations of $\theta = 0$ and $\theta = \pi / 2$ point in the direction of the x-axis and y-axis, respectively.

B. Model Process

The models are developed based on six constraints [12]: (1) The AV can be adequately represented as a two-dimensional vehicle whose motion is restricted to a plane. (2) It is assumed that the ground plane is inertial so the motion and rotation of the Earth may be neglected. (3) The vehicle body is rigid, so the effects of deflections caused by the suspension and the steering linkage are ignored. (4) The vehicle obeys the rolling constraint that all points on the vehicle rotate with the same angular speed about the Instantaneous Center of Rotation (C). (5) There is no motion along the transversal direction and the direction normal to the path surface. (6) It is assumed that there is no slip between the tires and the ground

Given these constraints, as the vehicle negotiates a turn, this motion can be described as a pure rotation with respect to C. Since the vehicle body is rigid, the motion of each wheel is a pure rotation with respect to the C. Given the "no slip" condition, the steer angle of each wheel is orientated along the tangent of its arc, as shown in Fig. 2.

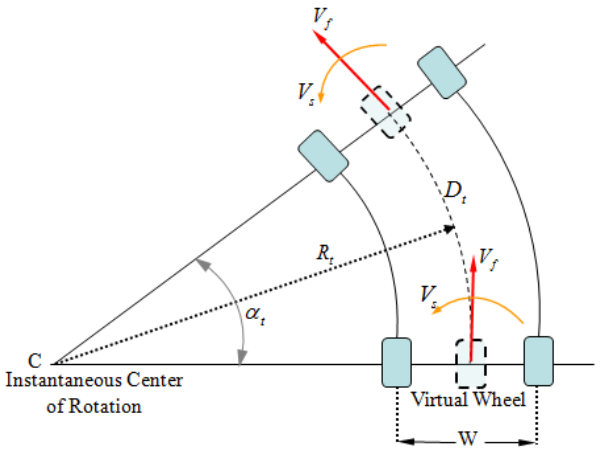


Fig. 2 Illustration of the model process of an AV

For modeling purposes, the constraints also make it possible to combine the effects of the pairs of wheels on their common axis and replace them by a single virtual wheel placed on the center of the axle. This reduces the number of degrees of freedom from four (drive speed and steer angle of the two wheels) to two (steer velocity v_S and the forward speed v_f) of the virtual wheel, or their displacement equivalents: translational (D_t) and circular displacement (α_t).

C. Velocity Motion Model

The velocity motion model assumes that the AV can be controlled through two velocities: a translation forward velocity v_f and a rotational steer velocity v_S . Denote the translation velocity at time *t* by v_{f_t} and the rotational velocity by v_{S_t} . The control u_t at time *t* is: $u_t = \left(v_{f_t} \quad v_{S_t}\right)^T$.

By definition, positive rotation velocities v_{S_t} introduce a counterclockwise rotation (left turn), and positive translation velocities v_{f_t} correspond to forward motion.

1) Direct Integration Velocity Motion Model

Denoting the position of the AV at a moment of time *t* as $X_t = \left(x_t \quad y_t \quad \theta_t\right)^T$, as shown in Fig. 3,

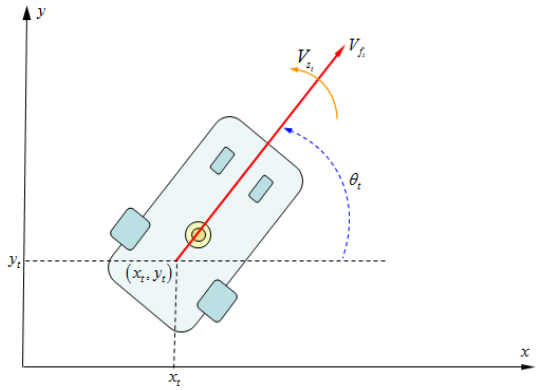


Fig. 3 Illustration of the position of an AV at a moment of time *t*

the kinematics equations can be written as [3], [11]:

$$\begin{cases} v_{x_t} = \dot{x}_t \approx \dfrac{x_t - x_{t-1}}{T} = v_{f_t} \cos\theta_t \\ v_{y_t} = \dot{y}_t \approx \dfrac{y_t - y_{t-1}}{T} = v_{f_t} \sin\theta_t \\ \dot{\theta}_t \approx \dfrac{\theta_t - \theta_{t-1}}{T} = v_{S_t} \end{cases} \tag{1}$$

This model, called often the 2D Initial Velocity (IV) model, is perfect when $T \to 0$, which is actually a significant approximation for autonomous navigation applications.

In this practical implementation, the navigation algorithm improves localization accuracy by integrating odometry data with laser scanner data for a 2D environment, and odometry-inertial data with a laser scanner data for a 3D environment. The rotational speed of the laser scanner is $1/6\ s \cong 167ms$ for a full rotation of 360 degrees, while the odometry measurements are available every 100ms [7]. The accelerometer readings are available with a frequency typically of 100 Hz (10ms) [7]. Sonar data is acquired at a rate

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

of about 60 ms per range measure [8]. Motor control loops operate at hundreds of nano-seconds, while device controllers can operate at a rate of tens of ms, and some higher performance servo controllers operate at a rate of hundreds of ms [7], [8]. It is known that when using the kinematics models, measurements should be fused prior to the time update for better accuracy performance. Since T is not sufficiently small, a different model is needed to remove modeling approximations.

2) Circular Velocity Motion Model

Let $u_t = \begin{pmatrix} v_{f_t} & v_{S_t} \end{pmatrix}^T$ denote the control over the entire sampling time interval (t-1, t]. Both velocities have fixed values for entire sampling time interval (t-1, t], then the AV moves on a circle with radius $R_t = \left| v_{f_t} / v_{S_t} \right|$ around an instantaneous center of rotation, $C(x_C, y_C)$. The form of R_t encompasses the case where the AV does not turn at all (i.e., $v_S = 0$), in which case the AV moves on a straight line. A straight line corresponds to a circle with infinite radius.

Considering $X_{t-1} = \begin{pmatrix} x_{t-1} & y_{t-1} & \theta_{t-1} \end{pmatrix}^T$ as initial position of the AV, after a time of motion $\Delta t = T$, the AV will be at $X_t = \begin{pmatrix} x_t & y_t & \theta_t \end{pmatrix}^T$ as shown in Fig. 4.

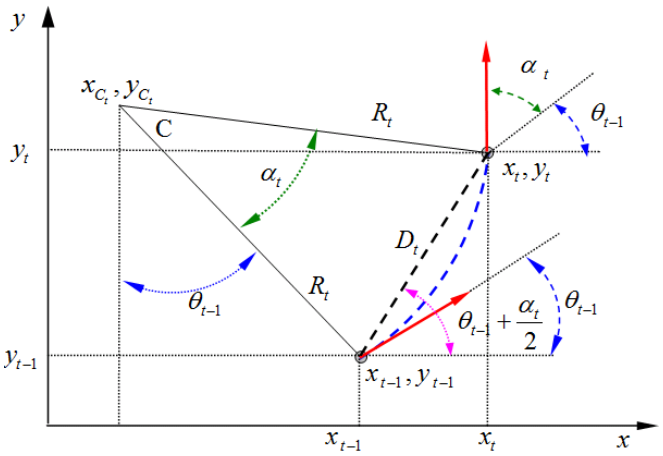


Fig. 4 Illustration of the AV's circular motion during a sampling time interval

Hence

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} \left| \frac{v_{f_t}}{v_{S_t}} \right| [-\sin\theta_{t-1} + \sin(\theta_{t-1} + v_{S_t} T)] \\ \left| \frac{v_{f_t}}{v_{S_t}} \right| [\cos\theta_{t-1} - \cos(\theta_{t-1} + v_{S_t} T)] \\ v_{S_t} T \end{pmatrix} \quad (2)$$

The equations of this model give good accuracy except for the case when $v_{S_t} \to 0$ which is now examined. Since $v_{S_t} \to 0, \alpha_t \to 0$ the distance traveled on circle from (x_{t-1}, y_{t-1}) to (x_t, y_t) is actually equal to the linear distance between these positions, D_t; therefore, as can be seen in Fig. 4:

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} v_f T \cos(\theta_{t-1} + \frac{v_{S_t} T}{2}) \\ v_f T \sin(\theta_{t-1} + \frac{v_{S_t} T}{2}) \\ v_{S_t} T \end{pmatrix} \quad (3)$$

The model will be a composite of (2) as a general form, and (3) for particular cases of very small steering speed values. It may be recommended to switch from form (2) to form (3) for $R_t \geq 10^6 m$. It is easy to prove that form (2) and form (3) are equivalent for very small steering speed values.

a) Real circular velocity motion model

In reality, vehicle motion is subject to noise [10]. The actual velocities differ from the commanded ones (or measured ones). This difference will be modeled as a zero-centered random variable with finite variance. Assume the actual velocities are given by:

$$\begin{pmatrix} \tilde{v}_{f_t} \\ \tilde{v}_{S_t} \end{pmatrix} = \begin{pmatrix} v_{f_t} \\ v_{S_t} \end{pmatrix} \pm \begin{pmatrix} \varepsilon_{\alpha_1 |v_{f_t}| + \alpha_2 |v_{S_t}|} \\ \varepsilon_{\alpha_3 |v_{f_t}| + \alpha_4 |v_{S_t}|} \end{pmatrix} \quad (4)$$

Here ε_b is a zero-mean error variable with a standard deviation *b*. Thus, the true velocity equals the commanded velocity plus or minus some small error or noise. The parameters $\alpha_1 \div \alpha_4$ are vehicle-specific error parameters. The common choice for ε_b is the normal distribution.

The assumption of circular motion leads to an important degeneracy. In particular, the support of the density $p(X_t | u_t X_{t-1})$ is two dimensional, within a three-dimensional embedded position space. Unfortunately, this degeneracy has important ramifications when applying a Bayes filter for the state estimation. To generalize the model accordingly, it is assumed that the vehicle performs a rotation $\tilde{\delta}$ when it arrives at its final position. Thus the final orientation is modeled by:

$$\theta_t = \theta_{t-1} + \tilde{v}_{S_t} T + \tilde{\delta}_t T \quad (5)$$

with $\tilde{\delta}_t = \varepsilon_{\alpha_5 |v_f| + \alpha_6 |v_S|}$, and where α_5 and α_6 are additional vehicle-specific parameters that determine the standard deviation of additional rotational noise. The *probabilistic* form of the circular velocity motion model is obtained by using (5) in (2) and (3) while replacing v_{f_t} with $\tilde{v}_{f_t}$, and v_{S_t} with $\tilde{v}_{S_t}$.

3) Circular Odometry Motion Model

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

The odometry-based motion model assumes that an AV can be controlled through two odometry measurements: the translation (*D*) and rotational displacement (α).

Denoting the translation displacement at time *t* by D_t, and the rotational displacement at time *t* by α_t, then we have the control as, $u_t = (D_t \quad \alpha_t)^T$. By replacing $R_t = \left| v_{f_t} / v_{S_t} \right|$ with $R_t = \left| D_t / \alpha_t \right|$, $v_{S_t} T = \alpha_t$ and $v_{f_t} T = D_t$ in (2) and (3) the odometry form of the motion model is obtained. In reality, vehicle motion and its measurements are subject to noise. The true measurements equal the measurements plus or minus a small error or noise. Using the same procedure as in section *III, 2, a,* it is easy to find the probabilistic form of the circular odometry motion model.

4) *The effect of using Inertial Velocity Motion Model*

As it is mentioned earlier, denoting the control $u_t = (v_{f_t} \quad v_{S_t})^T$ over the entire sampling time interval $(t-1, t] = T$, both velocities will have fixed values for entire sampling time interval (*t-1, t*], then the AV moves on a circle with radius $R_t = \left| v_{f_t} / v_{S_t} \right|$ around an instantaneous center of rotation $C(x_C, y_C)$. The result will be a linear displacement $D_t = v_f T$ and an angular displacement $\alpha_t = v_{S_t} T$. Using the IV model under these circumstances as it is shown in Fig. 5 instead of arriving in location (x_t, y_t), the AV will arrive in location $(x_t^{\bullet}, y_t^{\bullet})$ or if the argument of sine and cosine in (1) is just θ_{t-1}, the AV will arrive in (x_t^*, y_t^*) [3], [11]. Hence the IV model is not a correct description of control actions on the configuration of a vehicle.

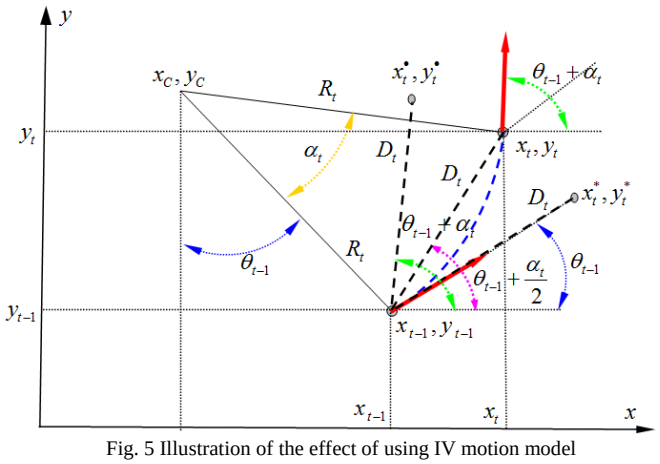


Fig. 5 Illustration of the effect of using IV motion model

IV. 3D Models

Vehicle kinematics for a AV operating in a 3D environment will now be examined.

A. Kinematics Configuration

The same "virtual vehicle" used for 2D models will be used in developing a 3D model. Having an external coordinate frame (*x, y, z)* relative to it, each position in space at a certain moment *t* in time can be represented by its three-dimensional Cartesian coordinate (*x, y, z*) and the spherical parameters representation (θ, γ, R).

The *position* of the vehicle is described by the vector: $(x, y, z, \theta, \gamma)^T$. The orientation of the SGV is described by θ and γ. By definition, an AV with orientation $\theta = \pi/2$, $\gamma = \pi/2$ points in the direction of x-axis, an orientation of $\theta = \pi$, $\gamma = \pi/2$ points in the direction of y-axis, and an orientation of $\theta = 0$, $\gamma = \pi$ points in the direction of the z axis, where $0 \le \theta \le 2\pi$ and $0 \le \gamma \le 2\pi$.

B. Model Process

The models are developed based on the same constraints used for a 2D environment except the first one. As the vehicle negotiates a turn at a specific moment *t* of time, this motion can be described as a pure rotation with respect to the C. If β_t (the angle between the circular motion plane and the horizontal plane) is not zero, the circular motion becomes a spatial motion on a circle lying on a sphere with radius R_t. Since the vehicle body is rigid, the motion of each wheel is a pure rotation with respect to C. Given the "no slip" condition, the steer angle of each wheel is orientated along the tangent of its arc as shown in Fig. 6.

The constraints reduce the number of degrees of freedom to two: The steer velocity v_S and the forward speed v_f for the virtual wheel or their displacement equivalents: translation D_t and circular displacement α_t. Two more rotational motions will be used to represent the motion in 3D: one in the horizontal plane and another in the vertical plane, described by the horizontal steering velocity $v_{S\theta}$ or horizontal circular displacement $\Delta\theta$ and vertical steering velocity $v_{S\gamma}$ or vertical circular displacement $\Delta\gamma$. These motions can be sensed and measured by inertial sensors.

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

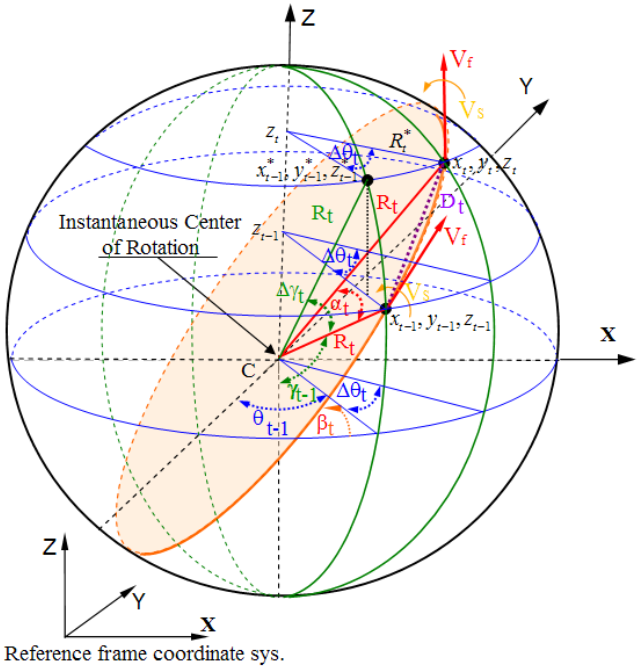


Fig. 6 Illustration of the spherical motion of an AV in 3D environment over a sampling time period

C. Kinematics Equations

Let $X_{t-1} = (x_{t-1} \quad y_{t-1} \quad z_{t-1} \quad \theta_{t-1} \quad \gamma_{t-1})^T$ be the initial position of the AV. It moves keeping the velocities constant at v_{f_t} and v_{S_t} over a sampling interval of time $\Delta t = t_t - t_{t-1} = T$, arriving at $X_t = (x_t \quad y_t \quad z_t \quad \theta_t \quad \gamma_t)^T$. The AV moves on a circle with radius $R_t = \left| D_t / \alpha_t \right| = \left| v_{f_t} / v_{S_t} \right|$ around C. If there is a slope of $\beta_t \neq 0$, the circle C is part of a sphere with radius R_t, as shown in Fig. 6.

The 3D kinematics' equations may be found by first decoupling the motion into two motions: one in the vertical plane (from $x_{t-1}, y_{t-1}, z_{t-1}$ to $x^*_{t-1}, y^*_{t-1}, z^*_{t-1}$), and one in the horizontal plane (from $x^*_{t-1}, y^*_{t-1}, z^*_{t-1}$ to x_t, y_t, z_t) and then using the same methodology as for 2D models.

Note, in the 2D model that the orientation of the AV is defined as the angle between the heading direction and the *x*-axis. It is also the angle between R_t and the extension of *y*-axis if the origin of the reference system is translated to *C*. Thus, the position of the AV at time *t* is given by the equations:

$$\begin{cases} x_t = x^*_{t-1} - R^*_t[\sin\theta_{t-1} - \sin(\theta_{t-1} + v_{S\theta_t} T)] \\ y_t = y^*_{t-1} + R^*_t[\cos\theta_{t-1} - \cos(\theta_{t-1} + v_{S\theta_t} T)] \\ z_t = z^*_t = z_{t-1} + R_t[\cos\gamma_{t-1} - \cos(\gamma_{t-1} + v_{S\gamma_t} T)] \\ \theta_t = \theta_{t-1} + v_{S\theta_t} T \\ \gamma_t = \gamma_{t-1} + v_{S\gamma_t} T \end{cases} \quad (6)$$

For very small variations of steer speed, the position at time *t* is given by the equations:

$$\begin{cases} x_t = x^*_{t-1} + D_{\theta_t} \cos(\theta_{t-1} + \dfrac{\Delta\theta_t}{2}) \\ y_t = y^*_{t-1} + D_{\theta_t} \sin(\theta_{t-1} + \dfrac{\Delta\theta_t}{2}) \\ z_t = z^*_{t-1} = z_{t-1} + D_{\gamma_t} \sin(\gamma_{t-1} + \dfrac{\Delta\gamma_t}{2}) \\ \theta_t = \theta_{t-1} + \Delta\theta_t \\ \gamma_t = \gamma_{t-1} + \Delta\gamma_t \end{cases} \quad (7)$$

where:

$$\begin{cases} x^*_{t-1} = x_{t-1} - [R_t \sin\gamma_{t-1} - R_t \sin(\gamma_{t-1} + \Delta\gamma_t)]\sin\theta_{t-1} \\ y^*_{t-1} = y_{t-1} + [R_t \sin\gamma_{t-1} - R_t \sin(\gamma_{t-1} + \Delta\gamma_t)]\cos\theta_{t-1} \end{cases} \quad (8)$$

D_{θ_t}, and D_{γ_t} are the linear translations on the horizontal and vertical planes, and $\Delta\theta_t$ and $\Delta\gamma_t$ are the rotational translations on the horizontal and vertical planes.

It is easy to observe that for $\gamma_{t-1} = \pi/2$ and $\Delta\gamma_t = 0$ ($D_{\gamma_t} = 0$), then (6) and (7) of the 3D model are equivalent with (2) and (3) of the 2D model, where $D_{\theta_t} = D_t$.

D. 3D Spherical Odometry-Inertial Motion Model

The odometry-inertial motion model assumes that the AV can be controlled through the odometry measurements of the linear translation D, rotational translation α, and the gyro measurements of the rotational displacements $\Delta\theta$ and $\Delta\gamma$. Denote at a certain time *t* the translation displacement by D_t and the rotational displacements by α_t, $\Delta\theta_t$ and $\Delta\gamma_t$. The spherical motion has the radius $R_t = \left| \dfrac{D_t}{\alpha_t} \right|$ and the control $u_t = (D_t \quad \alpha_t \quad \Delta\theta_t \quad \Delta\gamma_t)^T$. Assume that the AV has to travel from position $X_{t-1} = (x_{t-1} \quad y_{t-1} \quad z_{t-1} \quad \theta_{t-1} \quad \gamma_{t-1})^T$ to $X_t = (x_t \quad y_t \quad z_t \quad \theta_t \quad \gamma_t)^T$ during the sampling time *T*, and at moment *t* the sensors reading provide D_t, α_t, $\Delta\theta_t$ and $\Delta\gamma_t$. Replacing R^*_t by $R^*_t = \left| \dfrac{D_t}{\alpha_t} \right| \sin(\gamma_{t-1} + \Delta\gamma_t)$, $v_{S\theta_t} T = \Delta\theta_t$ and $v_{S\gamma_t} T = \Delta\gamma_t$ in (6) and (8), the spherical odometry-inertial motion model version is:

$$\begin{pmatrix} x_t \\ y_t \\ z_t \\ \theta_t \\ \gamma_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ \theta_{t-1} \\ \gamma_{t-1} \end{pmatrix} + \begin{pmatrix} \left|\frac{D_t}{\alpha_t}\right|[-\sin\gamma_{t-1}\sin\theta_{t-1} + \sin(\gamma_{t-1}+\Delta\gamma_t)\sin(\theta_{t-1}+\Delta\theta_t)] \\ \left|\frac{D_t}{\alpha_t}\right|[\sin\gamma_{t-1}\cos\theta_{t-1} - \sin(\gamma_{t-1}+\Delta\gamma_t)\cos(\theta_{t-1}+\Delta\theta_t)] \\ \left|\frac{D_t}{\alpha_t}\right|[\cos\gamma_{t-1} - \cos(\gamma_{t-1}+\Delta\gamma_t)] \\ \Delta\theta_t \\ \Delta\gamma_t \end{pmatrix} \quad (9)$$

Using the same procedure for (7) and (8), the form of spherical odometry-inertial motion model for very small variations of steer speed can be derived:

$$\begin{cases} x_t = x_{t-1} - \left|\frac{D_t}{\alpha_t}\right|\left\{\sin\theta_{t-1}[] - v_{S\theta_t}T\sin(\gamma_{t-1}+v_{S\gamma_t}T)\cos(\theta_{t-1}+\frac{v_{S\theta_t}T}{2})\right\} \\ y_t = y_{t-1} + \left|\frac{D_t}{\alpha_t}\right|\left\{\cos\theta_{t-1}[] + v_{S\theta_t}T\sin(\gamma_{t-1}+v_{S\gamma_t}T)\sin(\theta_{t-1}+\frac{v_{S\theta_t}T}{2})\right\} \\ z_t = z_{t-1} + \left|\frac{D_t}{\alpha_t}\right| v_{S\gamma_t}T\sin(\gamma_{t-1}+\frac{v_{S\gamma_t}T}{2}) \\ \theta_t = \theta_{t-1} + Tv_{S\theta_t} \\ \gamma_t = \gamma_{t-1} + Tv_{S\gamma_t} \end{cases} \quad (10)$$

Where: $[] = [\sin\gamma_{t-1} - \sin(\gamma_{t-1}+v_{S\gamma_t}T)]$.

E. 3D Spherical Velocity Motion Model

The velocity motion model assumes that the AV can be controlled through four velocities: translation velocity v_f and three rotational velocities v_S, $v_{S\theta}$, and $v_{S\gamma}$. Denote the translation velocity at time t by v_{f_t}, and the rotational velocities by v_{S_t}, $v_{S\theta_t}$ and $v_{S\gamma_t}$. The spherical motion has the radius at time t, $R_t = \left|v_{f_t} \middle/ v_{S_t}\right|$ and the control $u_t = \left(v_{f_t} \quad v_{S_t} \quad v_{S\theta_t} \quad v_{S\gamma_t}\right)^T$. By definition, the positive rotational velocities v_{S_t}, $v_{S\theta_t}$, and $v_{S\gamma_t}$ introduce counterclockwise rotation (left turns), and the positive linear translation velocity v_{f_t} corresponds to forward motion. Replacing R_t^* by $R_t^* = \left|\frac{v_{f_t}}{v_{S_t}}\right|\sin(\gamma_{t-1}+v_{S\gamma_t}T)$, $\Delta\theta_t = v_{S\theta_t}T$ and $\Delta\gamma_t = v_{S\gamma_t}T$ in (6) and (7), there is going to be the spherical velocity motion model forms.

Using the same methodology described in section *III, 2, a*, the probabilistic forms of the 3D spherical motion models may be determinate.

V. AV LOCALIZATION

Mobile vehicle localization is the problem of determining the position of the AV relative to a given map of the environment. It is often called *position estimation* [1], [10].

A. Localization Algorithm

This algorithm requires a Gaussian estimate of the AV position at time *t*-1 as its input, with mean μ_{t-1} and covariance $\sum_{t-1}$. It also requires a control u_t, a map *m* and a set of features measurement, $Z_t = \{Z_t^1, Z_t^2, ...\}$ at time *t*. The output is a new revised μ_t and $\sum_t$.

1) EKF Localization for Odometry-Inertial Motion Model

The *Prediction Step* (or control update step) modifies the belief in accordance to an action. The EKF localization algorithm uses the probabilistic form of the spherical odometry-inertial motion model, maintaining a local posterior estimate of the state represented by μ_{t-1} and covariance Σ_{t-1}. The motion is generated by the motion control, $u_t = \left(D_t \quad \alpha_t \quad \Delta\theta_t \quad \Delta\gamma_t\right)^T$ with additional Gaussian noise. To linearize the motion model, the model is decoupled into a noise-free component and a random noise component as in equation:

$$\underbrace{\begin{pmatrix} x_t \\ y_t \\ z_t \\ \theta_t \\ \gamma_t \end{pmatrix}}_{X_t} = \underbrace{\begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ \theta_{t-1} \\ \gamma_{t-1} \end{pmatrix} + \begin{pmatrix} \left|\frac{D_t}{\alpha_t}\right|[-\sin\gamma_{t-1}\sin\theta_{t-1} + \sin(\gamma_{t-1}+\Delta\gamma_t)\sin(\theta_{t-1}+\Delta\theta_t)] \\ \left|\frac{D_t}{\alpha_t}\right|[\sin\gamma_{t-1}\cos\theta_{t-1} - \sin(\gamma_{t-1}+\Delta\gamma_t)\cos(\theta_{t-1}+\Delta\theta_t)] \\ \left|\frac{D_t}{\alpha_t}\right|[\cos\gamma_{t-1} - \cos(\gamma_{t-1}+\Delta\gamma_t)] \\ \Delta\theta_t \\ \Delta\gamma_t \end{pmatrix}}_{g(u_t, X_{t-1})} + N(0, R_t) \quad (11)$$

The EKF linearization approximates the function g through a Taylor expansion that constructs a linear approximation to a function g from value and its slope.

$$g(u_t, X_{t-1}) \approx g(u_t, \mu_{t-1}) + \underbrace{g'(u_t, \mu_{t-1})}_{=:G_t}(X_{t-1} - \mu_{t-1}) =$$

$$= g(u_t, \mu_{t-1}) + G_t(X_{t-1} - \mu_{t-1}) \quad (12)$$

The remaining steps are identical to the general EKF localization algorithm [1], [5].

2) UKF Localization for Odometry-Inertial Motion Model

UKF localization is a feature-based vehicle localization algorithm using the unscented Kalman filter. The key idea is to augment the state with additional components representing control and measurement noise. The dimension L of the augmented state is 5+4+7=16. Since zero-mean Gaussian noise is assumed, the mean μ_{t-1}^a of the augmented state estimate is given by the mean of the position estimate, μ_{t-1}, and zero vectors for the control and measurement noise. The covariance, Σ_{t-1}^a, of the augmented state estimate is given by

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

combining the position covariance, Σ_{t-1}, the control covariance, M_t, and the measurement covariance Q_t. In this case, $\mathbf{X}^a_{t-1}$ contains 2L+1=33 sigma points, each having components in state, control, and measurement space:

$$\mathbf{X}^a_{t-1} = \begin{pmatrix} \mathbf{X}^{X\ T}_{t-1} & \mathbf{X}^{uT}_t & \mathbf{X}^{zT}_t \end{pmatrix}^T \quad (13)$$

Mixed time indices were chosen to make clear that $\mathbf{X}^X_{t-1}$ refers to X_{t-1}, while the control and measurement components refer to u_t and z_t, respectively. The position components $\mathbf{X}^X_{t-1}$ of these sigma points are then passed through the motion model function *g*, using the control with the added control noise component $\mathbf{X}^u_{i,t}$ of each sigma point as in (14).

$$\bar{\mathbf{X}}^X_{i,t} = \begin{pmatrix} \left|\frac{D_{i,t}}{\alpha_{i,t}}\right|[-\sin\gamma_{i,t-1}\sin\theta_{i,t-1} + \sin(\gamma_{i,t-1}+\Delta\gamma_{i,t})\sin(\theta_{i,t-1}+\Delta\theta_{i,t})] \\ \left|\frac{D_{i,t}}{\alpha_{i,t}}\right|[\sin\gamma_{i,t-1}\cos\theta_{i,t-1} - \sin(\gamma_{i,t-1}+\Delta\gamma_{i,t})\cos(\theta_{i,t-1}+\Delta\theta_{i,t})] \\ \left|\frac{D_{i,t}}{\alpha_{i,t}}\right|[\cos\gamma_{i,t-1} - \cos(\gamma_{i,t-1}+\Delta\gamma_{i,t})] \\ \Delta\theta_{i,t} \\ \Delta\gamma_{i,t} \end{pmatrix} + \mathbf{X}^X_{i,t} \quad (14)$$

The remaining updated steps are identical to the general UKF algorithm [1], [5].

VI. Experimental Results

A. Performances Comparative between the Circular Odometry Motion Model and Inertial Motion Model.

The implementation uses real input data of a path with a length of approximately 1.2 Km. The real input data has been collected from the Meadow Brook Hall and Smart Zone area of Oakland University, Michigan USA, as shown in Fig. 7.

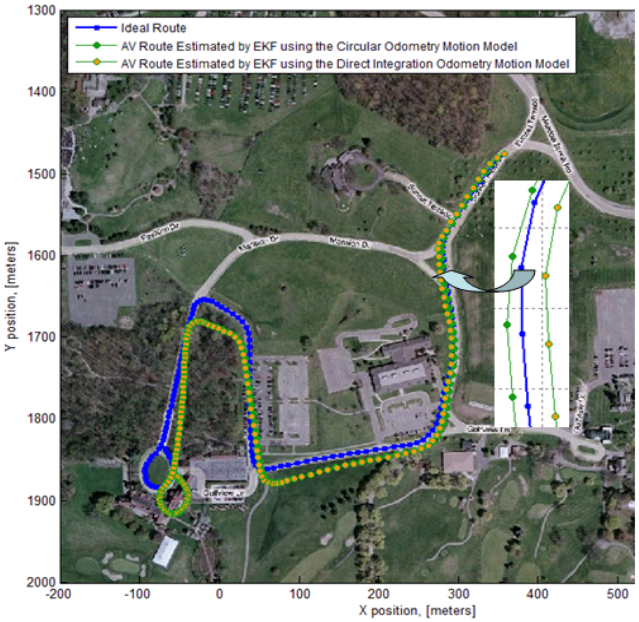


Fig.. 7 Comparative implementation of Odometry motion models (Circular/Inertial) using EKF for a real path data input (2D)

In the first implementation, the EKF has been used to compare the performance of the circular odometry motion model and the direct integration (inertial) odometry motion model. For the circular odometry motion model, the small steering speed form is used. It provides a numerical evaluation of the mean error for both models as well. The implementation results are for an initial noise of 10 % in longitudinal translation and 9 % in rotation translation. The results are in Table I. The results from Table 1 reveal that for a single run of the algorithm, the circular model provides a better accuracy of 21 cm. For 50 runs with additional incremental noise, there is a better accuracy of 93.59 cm. At 300 runs, the improvement in accuracy is 1.0313 m and for 400 runs it is 1.0371 m. The improvement is significant when the accuracy matters.

TABLE I
MONTE CARLO VALIDATION (CIRCULAR/INS)

Monte Carlo Iterations #	Position mean error in meters.	
	EKF+Circular Odometry Motion Model	EKF+Direct Integration Motion Model
1	22.42	22.63
50	23.766	24.7019
300	25.0217	26.0530
400	25.1974	26.2345

B. Performance Analysis of Spherical Motion Model

To evaluate the performance of SMM's, the Matlab simulations of the EKF localization and UKF localization algorithms in a 3D environment have been performed. The Spherical Odometry-Inertial for a low steer speed model has been used. Results of the implementation of the EKF localization algorithm, and the results of the UKF localization algorithm implementation are shown in Fig. 8. The implementation uses the same real input data of a path used for 2D implementations.

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf

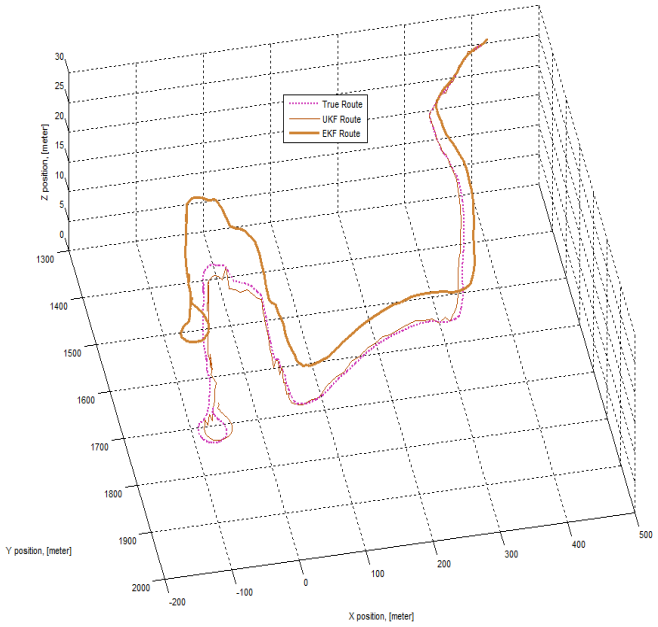


Fig. 8 3D EKF and UKF implementation of Odometry motion models

This figure shows that the localization algorithms track the reference path very accurately in the beginning of path and less accurately towards the end of the path. This behavior is exactly what is expected because of the noise accumulation during estimation process.

Even though the route is not smooth at all, it is easy to observe the efficiency and accuracy of the EKF and UKF localization algorithm implementations. To validate and evaluate the performance of the EKF localization and UKF localization, Monte Carlo and Spherical Error probable (SPE) evaluations have been performed as well. Fig. 9 shows the Monte Carlo validation of the EKF localization algorithm implementation for 1000 iterations. Each circle corresponds to the mean error of one run.

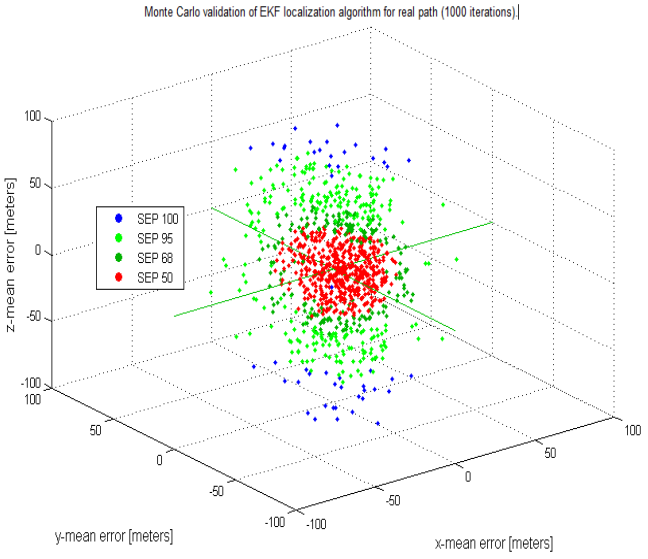


Fig. 9 Monte Carlo validation of EKF localization algorithm implementation (1000 iterations) for 3D environment

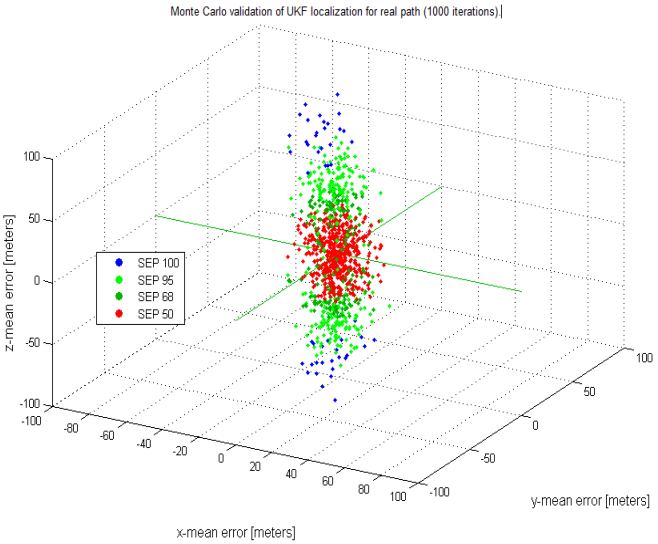


Fig. 10 Monte Carlo validation of UKF localization algorithm implementation (1000 iterations) for 3D environment

Fig. 10 shows the Monte Carlo validation of the UKF localization algorithm implementation for 1000 iterations. The SPE is defined as the radius of a sphere, centered about the general mean error, with a boundary expected to include 50% (or 68%, 95%) of the population (mean error of each Monte Carlo' iteration) within it. The red circles represent the means of iterations which are part of SPE 50, the dark green circles represent SPE 68, the light green circles represent SPE 95, and the blue circles represent SPE 100.

The Monte Carlo validation implementations show that even if the amount of noise is increased after each iteration, the model is still stable and robust.

VII. Conclusion

The goal of this paper is to present the minimum necessary to make the subject comprehensive and present new scientific argument and experimental result to support the SMM's characteristics (high accuracy, robust, low-cost, and well adapted to microprocessor process control and estimation models). The SMM's are low-cost because just by adding one additional gyroscope to a 2D navigation system and using the SMM's in localization algorithms allows it to operate in a 3D environment. The Monte Carlo validation implementations show that the SMM's are robust and stable models.

As shown in section *III, C, 1 & 4*, the traditional inertial models are not adapted to the microprocessor estimation and control process. This can affect negative the accuracy in applications where the accuracy matters.

Even though they were developed for land navigation, the SSM's models can be used for underwater or airborne navigation if the vehicle is equipped with specific sensor systems or the model is adjusted to the existing sensor system.

References

[1] G. Galben,"New Three-Dimensional Velocity Motion Model and Composite Odometry-Inertial Motion Model for Local Autonomous Navigation" IEEE Trans on Vehicular Technology, vol 60, NO.3, pp. 771-782, March 2011.

[2] G. Galben,"Robust and High Accuracy Autonomous Navigation", PhD dissertation, Dept. Elect. And Comp. Eng, Oakland Univ., Rochester, MI, 2011, pp 38-77 and 87-128.
[3] H. Huosheng, G Dongbing , "Landmark-based Navigation of Industrial Mobile Robots", International Journal of Industry Robot, Vol. 27 No 6, 2000 pp. 458-467.
[4] S. Scheding, G. Dissanayake, E. Nebot, and H. Durrant-Whyte,"An experiment in autonomous navigation of an underground mining vehicle", IEEE Trans. Robot. Automat. vol 15, Feb 1999 pp.85-95.
[5] S. Thrun, W. Burgard, D. Fox "Probabilistic Robotics" The MIT Press Cambridge, Massachusetts London, England 2005, pp 126-129.
[6] M. Ollis and A. Stentz, "Vision-based perception for an autonomous harvester" in Proc. IEEE/RSJ Int. Conf. Intelligent Robotic Systems, Sept.1997, pp.1838-1844.
[7] S. Sukkarieh, E. M. Nebot , and H. Durrant-Whyte,"Achiving integrity in an ins/gps navigation loop for autonomous land vehicle applications", in IEEE Int. Conf. Robotics and Automation, May 1998, pp 3437-3442.
[8] G. Welch and G. Bishop, "Introduction to the Kalman Filter". Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 1995, pp 19-31.
[9] J. L. Crowley, "Mathematical Foundation of Navigation and Perception for an Autonomous Mobile Robot", Tutorial presented at the International Workshop on Reasoning with Uncertainty in Robotics, University of Amsterdam, The Netherlands Dec. 4-6, 1995, pp 2-4.
[10] J. Borenstein, H. Everet, L. Feng, and D. Wehe "Mobile Robot Positioning –sensors and Techniques", Journal of Robotic Systems, Special Issue on Mobile Robots Vol. 14, No. 4, 1997 pp. 231 – 249.
[11] G. Antonelli, S. Chiaverini "A Deterministic Filter for Simultaneous Localization and Odometry Calibration of Differential-Drive Mobile Robot" Univerita deli Studi , Cassino (FR) , Italy 2008, pp. 1-3.
[12] S. J. Julier "Process Model for Navigation of High-Speed Land Vehicle" PhD. Dissertation, Dept. Eng., Oxford Univ., UK, 1997, pp 34-35



Gheorghe Galben was born in Susenii-Bargaului, jud. Bistrita-Nasaud, Romania, Europe.

He received the Diploma of Engineering degree in electronics engineering from Polytechnic Institute Cluj-Napoca Romania in 1984. The Diploma has been evaluated as a M.S. degree in electronics engineering by Educational International Inc., Wellesley, MA, U.S.A. He received the M.S. degree in dynamic systems and control engineering from Oakland University, MI, U.S.A. in 2003, as well. He is currently working toward the PhD. degree at the Electrical and Computer Engineering Department of Oakland University, Rochester, MI, U.S.A.

He is currently employed at iTrack-LLC, North Squirrel Rd. Rochester, MI, USA. as a research and development engineer. His current research interests include: wireless communications systems, digital signal processing, sensor fusion systems, tracking techniques, and autonomous navigation systems.



Daniel N. Aloi (M'95-SM2005**)** is an Associate Professor and Acting Chair in the Electrical and Computer Engineering Department and Founder/Director of the Applied Electromagnetic and Wireless Laboratory at Oakland University, located in Rochester, Michigan. His other professional assignments have included positions as a Visiting Assistant Professor in the School of Electrical Engineering and Computer Science at Ohio University and as a Senior Project Engineer at OnStar, Inc., which is a subsidiary of General Motors Corporation. His research interests have focused on various aspects of applied electromagnetics and location technology in the aviation and automotive industries. He has been awarded $1.8M in external funding a principal investigator, published over 50 technical papers and is an inventor on five patents related to automotive navigation. In addition, he served as a key technical advisor to the Federal Aviation Administration's Satellite Program Office for the Local Area Augmentation System. He received his B.S.E.E. (1992), M.S.E.E. (1996), and Ph.D. in EE (1999) all from Ohio University's School of Electrical Engineering and Computer Science.

Open Science Index, Electrical and Computer Engineering Vol:5, No:8, 2011 publications.waset.org/15444/pdf