Proceedings of the SPIE Aerosense Conference, Orlando, FL, April 21 - 25, 2003.

Moving Object Prediction for Off-road Autonomous Navigation[¶]

R. Madhavan and **C. Schlenoff**

Intelligent Systems Division, Manufacturing Engineering Laboratory,
National Institute of Standards and Technology (NIST),
Gaithersburg, MD 20899-8230, U.S.A.

ABSTRACT

The realization of on- and off-road autonomous navigation of **U**nmanned **G**round **V**ehicles (UGVs) requires real-time motion planning in the presence of dynamic objects with unknown trajectories. To successfully plan paths and to navigate in an unstructured environment, the UGVs should have the difficult and computationally intensive competency to predict the future locations of moving objects that could interfere with its path. This paper details the development of a combined probabilistic object classification and estimation theoretic framework to predict the future location of moving objects, along with an associated uncertainty measure. The development of a moving object testbed that facilitates the testing of different representations and prediction algorithms in an implementation-independent platform is also outlined.

Keywords: autonomous navigation, moving object prediction, uncertainty measure, path planning, estimation theory.

1. INTRODUCTION

At the National Institute of Standards and Technology (NIST), we are undertaking efforts to realize on- and off-road autonomous navigation of **U**nmanned **G**round **V**ehicles (UGVs). To efficiently plan the motion of UGVs in the presence of dynamic objects with unknown trajectories, it is crucial to maintain an accurate internal representation of current and possible future states of the environment. In particular, to successfully plan paths and to navigate in an unstructured environment, an autonomous vehicle should have the ability to predict, with some level of confidence, the future locations of moving objects that could interfere with its path. This process is not only difficult but is also computationally intensive.

To represent moving objects in a dynamic environment, a combination of symbolic, equation-based, and grid-based representations have been developed for use within the 4D/**R**eal-Time **C**ontrol **S**ystem (RCS) reference model architecture.[1] Applying this architecture, we can ensure that the methodologies being developed for representation and prediction of moving objects can accommodate different types of planners. This paper concentrates on the moving object prediction problem; representational issues have been detailed elsewhere.[2]

Before the future position of an object can be predicted and subsequently tracked, it should be detected in sensor images. The problem of moving object detection has been an area of active research and can be readily identified in the literature. Most approaches consider the detection problem from a stationary platform (e.g. surveillance applications[3]). The more difficult problem of detecting moving objects from a moving platform typically employs computer vision techniques including condensation algorithms,[4] distance transforms,[5] optical flow,[6] stereo vision,[7] and color vision.[8]

For real-time unmanned ground vehicle navigation, range imaging using LADAR is becoming popular for several reasons. Such active range sensing techniques do not have to deal with the extra burden of calibration

Send correspondence to R. Madhavan.
R. Madhavan: E-mail: raj.madhavan@nist.gov, Tel: +1-301-975-2865, Fax: +1-301-990-9688
C. Schlenoff: E-mail: craig.schlenoff@nist.gov, Tel: +1-301-975-3456, Fax: +1-301-990-9688

[¶]Commercial equipment and materials are identified in this paper in order to adequately specify certain procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

issues, computational costs and illumination problems usually associated with passive ranging techniques like stereo vision.[9] Recent developments in miniaturization of LADAR technology have led to significant improvements in these devices small enough to operate on aircraft and in ground vehicles. With evergrowing computer processing capabilities and cost reduction of these units, LADAR holds great promise[10] for navigation applications required to deal with moving objects, though the recentness of the advancements in the technology has precluded significant published results in this area.

Once objects are detected, they need to be classified either as stationary or moving. Additionally, it would be of great value if the moving object can be classified with respect to an *a priori* list of entities. The framework proposed in this paper classifies a moving object based upon similarities between its perceived attributes and those that are stored in *a priori* ontologies and knowledge bases. Based upon the classification of the moving object, its future location and velocity can be predicted. The research presented in this paper performs *probabilistic object classification* and *cross corroboration* combined with the proposed moving object prediction scheme. The object classification and cross corroboration ideas in this paper differ from the conventional approaches in the literature which do not classify an object as moving or stationary *before* attempting to predict its future position. Only a limited amount of research can be identified in the literature on this crucial aspect of motion planning in dynamic off-road environments.[11,12] Recently, Wang *et al.* have developed a Bayesian formulation of the simultaneous localization and mapping (SLAM) combined with the detection of moving objects where the authors contend that the detection of moving objects is useful for SLAM. Results from an urban environment using a 2D laser range finder are presented to demonstrate the feasibility of the EKF formulation.[13] We anticipate our framework to produce better results in light of the use of LADAR and the use of explicit object classification and cross corroboration of the moving objects with their predicted positions and velocities.

Statistical methods for estimating obstacle locations using statistical features have been proposed by other researchers such as the Hidden Markov Models (HMMs) to predict obstacle motion,[14] Poisson distribution to describe the probability of collision with obstacles,[15] autoregressive models for one-step ahead prediction of moving obstacles[16] or probability of occupancy of cells in grid maps.[17] The principal disadvantages of these methods are that they are computationally intensive thus precluding real-time implementations and perhaps most importantly have only been implemented for 2D polygonal environments.

For moving object prediction and tracking, estimation theoretic approaches have shown promising results especially in radar imagery.[18] The estimation theoretic framework to predict future location and velocity of moving objects in this paper is based on an **E**xtended **K**alman **F**ilter (EKF). Kalman filtering is a well established recursive state estimation technique where estimates the states of a system are computed using the process and observation models together with Gaussian assumptions made about process and observation noise.[19] The recursive nature of the algorithm utilizes the system's CPU more uniformly to provide estimates without the latency resulting from batch processing techniques.[20] We have envisaged a bank of object-specific EKFs for each of the classified moving objects. The notion of using a bank of Kalman filters is not entirely new but our framework has several advantages. The framework described in this paper takes a more generalized view of moving object representation and prediction, specifically developing a moving object testbed that allows one to test different representations and prediction algorithms in an implementation-independent platform. This effort represents a new way forward in concurrently integrating multiple knowledge representation approaches from disparate sources to completely model the information necessary for dynamic planning in the presence of moving objects.

The paper is structured as below: Section 2 introduces the moving object testbed. Section 2.1 describes the OneSAF testbed that provides processed sensor data to validate the classification and prediction algorithms in addition to the testbed itself. Section 2.2 develops a LADAR noise model for use in the EKF. Section 3 details the probabilistic object classification algorithm. Section 4 details the EKF-based prediction of moving objects with a complete estimation cycle in Section 4.1 and results in Section 4.2. Section 5 concludes the paper by discussing ongoing and further research efforts.

2. THE MOVING OBJECT TESTBED

In order to test the prediction algorithms developed in this effort, we have designed and developed a moving object testbed. The purpose of this testbed is to provide an implementation-independent platform in which to

apply various prediction algorithms to determine the results that can be expected to be achieved in different environments. The testbed allows for testing algorithms that predict in the near future (within two seconds) and in further time horizons. For short-term prediction, we have implemented EKF-based prediction algorithms.

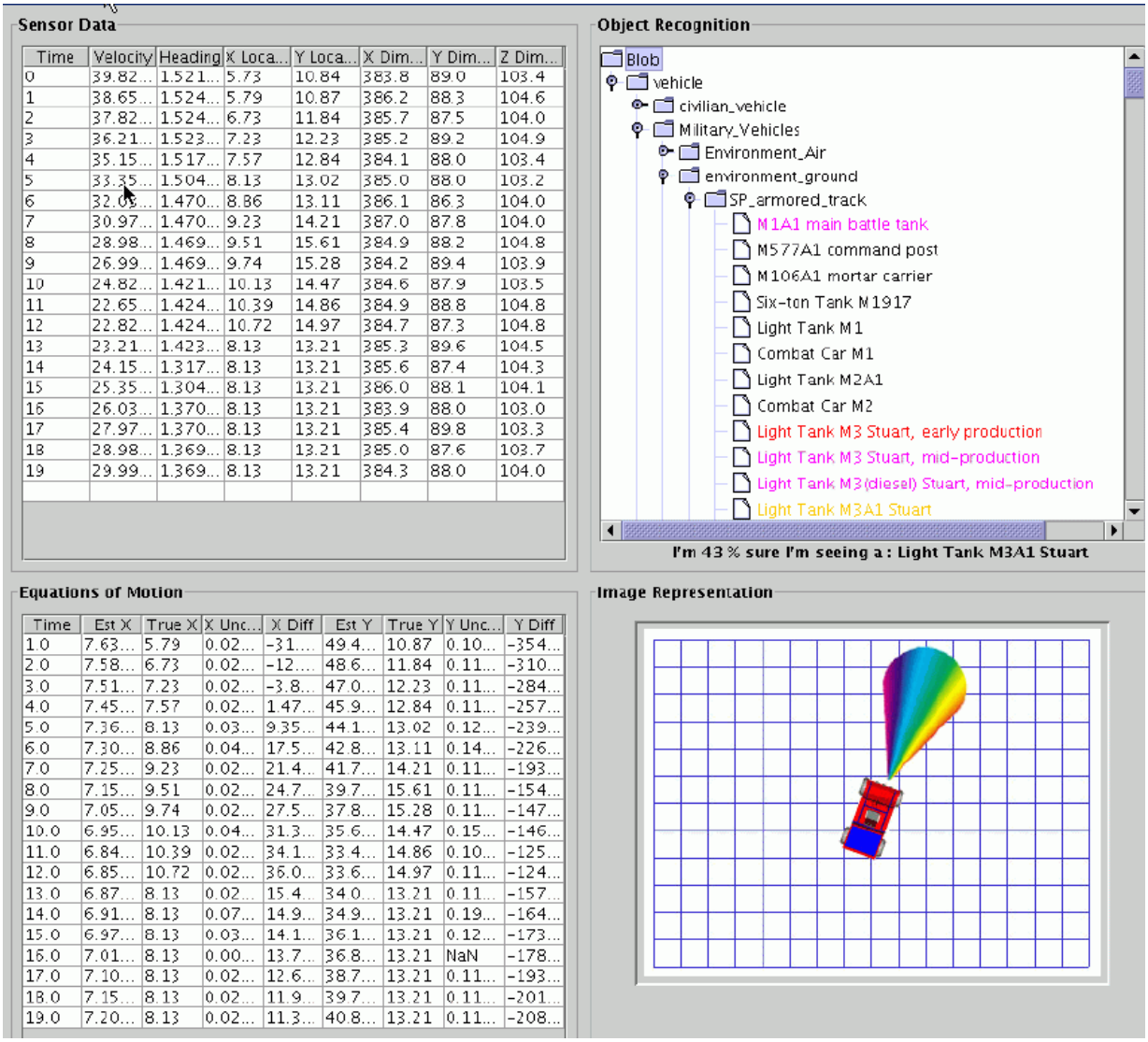

Figure 1. The Graphical Front-end of the Moving Object Testbed.

Figure 1 shows a graphical front-end of the moving object testbed. Each of the four panes in Figure 1 serves a unique purpose:

- The *top-left pane* receives processed input through the Neutral Messaging Language (NML)[21] from the vehicle's sensors as a function of time. The data that is being received include the moving object's:
  - ⋆ **Current Location**: x, y, and z locations as well as the type of terrain in which the object resides such as land, road, water, or air.
  - ⋆ **Dimensions**
  - ⋆ **Velocity**
  - ⋆ **Color**: Although any object can have multiple colors, this represents the most predominant color of the object, if appropriate.
  - ⋆ **Motion Pattern**: Based upon multiple concurrent sets of position data.

- The *top-right* pane attempts to classify this object based on attribute-matching by comparing perceived object attributes to those stored in an *a priori* knowledge base. In our experiments, we have relied heavily on the types of information we expect to have available, or could be deduced, from a LADAR sensor.
- The *bottom-left* pane grid applies an EKF-based algorithm to the sensed data to predict the future location of that moving object with an associated confidence measure.
- The *bottom-right* pane instantiates the results of the EKF into an occupancy grid, thus representing the probability of the object occupy certain cells at pre-defined times in the future. In this pane, we can interactively set the time we wish to predict the location of the moving object, and the predicted location and associated uncertainty will interactively display on the grid.

The moving object testbed has been developed in C++ and runs on a Linux system. The testbed uses a MySQL server as the database backend and communicates with the vehicle's sensory processing system using the Neutral Messaging Language (NML) protocol.

2.1. THE ONESAF TESTBED (OTBSAF)

To test the prediction approach developed in this paper, we have used simulated processed sensor data to validate the testbed and test the prediction algorithms. We have used the OneSAF Testbed (OTBSAF)[22] as the virtual sensor.

OTBSAF is a simulation package used for integrating, testing and user feedback of technology developments into the future OneSAF Objective System. It provides operational environments useful for identifying, developing, prototyping, demonstrating, and testing of enabling technologies and entity behaviors. OneSAF will be the future entity level battalion and below constructive simulation that, when linked with other virtual simulators, will seamlessly integrate live, virtual, and constructive simulations into realistic synthetic battle spaces. It will also represent the physical environment and its effect on simulated activities and behaviors.

As a simulated environment, OTBSAF is able to represent moving objects. By querying OTBSAF, we can retrieve an object's location and velocity at the current time. To validate the testbed and prediction algorithms, we are initially using this retrieved data to serve as our processed sensor data.

2.2. LADAR Noise Model

The data retrieved from OTBSAF is perfect sensor data. In other words, when we ask for the location or dimensions of the object, we are presented with the exact location and the exact dimensions without any associated uncertainty. Although convenient, this does not represent the information that we expect to get from sensors on the actual vehicle. To compensate for this, we have introduced a noise model into the data retrieved from OTBSAF. As mentioned earlier in the paper, we are using the LADAR sensor as our primary source of sensor data. The LADAR sensor provides a mechanism to measure shapes in 3D based on laser time of flight. The technology works from centimeter sized objects to aerial map generation. The general output of the LADAR is a cluster of 3D points with respect to a global/local coordinate system. The rest of this subsection describes the LADAR noise model.

Every LADAR has an associated resolution, which indicates the angle at which the laser beam is expected to expand as a function of distance. For this work, we are using the resolution of the Riegl LADAR,[23] which is 0.072 degrees/pixel. Based on this information, we compute the size of the LADAR beam at various distances from the vehicle in accordance with the schematic in Figure 2 where $b = 2$ meters (the height the LADAR is mounted above the ground on the autonomous vehicle) and D represents the angular resolution of the LADAR. The value of a is varied to determine the size of the LADAR beam (d) at various distances. Table 1 represents the LADAR beam size at distances from $10 - 150$ meters. It is important to note that, as the LADAR beam moves upward in Figure 2, the beam size will reduce in size. However, this reduction is only within a few percent, and is negligible for our purposes.

Knowing the beam size as a function of distance, we are now able to associate a noise model with the information provided by OTBSAF. For example, lets assume that we query OTBSAF for the dimensions of an

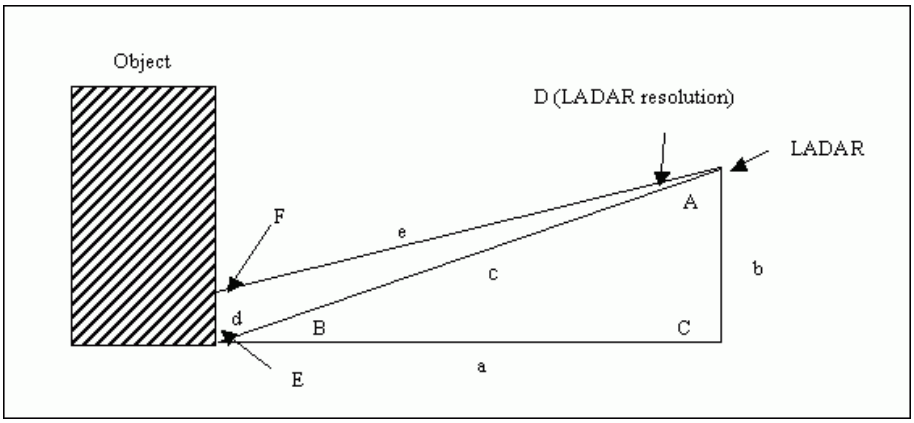

Figure 2. Schematic of the LADAR Noise Model.

Table 1. LADAR Beam Size as a Function of Distance

Distance Ahead [m]	LADAR Beam Size [cm]
10	1.3
20	2.5
30	3.8
40	5.0
50	6.3
60	7.5
70	8.8
80	10.1
90	11.3
100	12.6
110	13.8
120	15.1
130	16.3
140	17.6
150	18.9

object that is 100 meters away from our vehicle. OTBSAF returns a length value of 2.5 meters (250 centimeters), representing the actual and exact length dimension of that object. From our noise model, we know that a LADAR would introduce an error of up to 12.6 centimeters at that distance. Therefore,we introduce this noise into the returned value by introducing a random number of between -6.3 and $+6.3$ centimeters (half of the noise value). After applying this noise model, we would they get a return of a value between 243.7 and 256.3 centimeters, depending on the random number generated.

3. CLASSIFICATION OF MOVING OBJECTS

Being able to accurately classify the type of object being perceived is of immense value in determining where the object can and can not be at times in the future. Different types of objects have varying dynamics and motion capabilities that dictate characteristics such as their maximum attainable velocity and acceleration, braking capabilities, and minimum turning diameter. Knowing these object-specific characteristics helps to define, and refine, algorithms that predict the object's future location.

One can attain the above benefits without having to identify the exact object that is being perceived in the environment. Simply classifying the object by its type can go a long way in understanding and leveraging the object's motion capabilities and constraints. For example, classifying an object as a tank (even though it would

be more accurately classified as an M1A1 tank) may tell us that the vehicle moves on tracks and thus can cross negative obstacles of up to 2.1 meters wide, has a maximum attainable velocity of 65 km/h, and has a minimum turning diameter of 1.5 meters.

NIST is leveraging its expertise in sensory processing and knowledge representation to develop an object classification system based upon the principle of attribute-matching. We have developed a database of objects that we expect to encounter in different environments, along with the objects' pertinent characteristics (e.g., dimensions, maximum and average velocity, weight, minimum turning diameter, color, etc.).[2] The data is stored in an MySQL database, and is presented to the vehicle as an *a priori* data set before the vehicle begins its mission.

Using processed sensor data from a LADAR and a color camera, we are able to determine pertinent characteristics of objects in the environment (with an associated level of noise), including the sensed object's dimensions, position, velocity and heading, color, and texture. By analyzing sets of LADAR data (or simulation thereof) taken over sequential time points, we can analyze the motion pattern of the object to determine characteristics such as turning diameters. We compare the processed sensed data from the LADAR and the color camera with the characteristics of the objects stored in the *a priori* database to determine matches. We associate a confidence measure with each object in the database based on:

- The number of characteristics that match
- The number of times in the past that the characteristics matched
- How recently the characteristics matched
- How closely the characteristics matched

The object with the highest confidence measure indicates the highest probability that the object is the one that we are seeing in the environment. This process is performed in real time, so the confidence measures are constantly changing. However, experiments have shown that, over the course of $3 - 5$ seconds at 20 Hertz, the system tends to single out a small set of objects that provide the highest confidence match, and gradually increases their confidence as time progresses, separating them from the rest. Thus, through probabilistic object classification, we are able to determine which of the object-specific EKFs need to be employed to predict the future location of the object.

4. PREDICTION OF MOVING OBJECTS

Once the object is classified with a reasonable certainty, an EKF is employed to predict (estimate) the position and velocity of the moving object at a future time instant. Kalman's prediction theory, allows the computation of the best estimate of a future system state by using the most recent estimates of system state along with the system dynamic model. With appropriate interpretation, covariance analysis inherent in the Kalman filtering techniques serves as a confidence measure indicative of the uncertainty in the predicted system states. The EKF thus provides a convenient measure of prediction accuracy through the covariance matrix.

An EKF employs a process model to estimate the future location of the object of interest. Since the object classification module provides the type of moving object whose position and velocity needs to be predicted, we have envisaged a bank of EKFs for each type of classified object. In addition to the distributed nature of the estimation process, this has the added advantage of cross-corroborating the object classification itself as the uncertainty in the EKF prediction will be an indicator of the quality of the prediction. The higher the uncertainty, the lower the confidence in the selection of the correct set of object models and thus consequently decreasing the confidence of the object classification. Thus, our framework combines low-level (image segmentation/classification) and mid-level (recursive trajectory estimation) information to obtain the short-term prediction and combines it with the cross corroboration to work symbiotically in the reduction of the total uncertainty in predicting the positions and velocities of moving objects.

Currently, short-term prediction of objects moving at variable speeds and at given look-ahead time instants are predicted using the EKF from a stationary platform. The EKF employs a non-linear model derived from equations based on the kinematics of the moving objects (vehicles) to be predicted. The following section presents a complete estimation cycle for predicting the future position and orientation of a moving wheeled vehicle.

4.1. Estimation Cycle for a Wheeled Vehicle

The extension of the linear Kalman filtering ideas to a non-linear system is termed extended Kalman filtering. The extended Kalman filter (EKF) is a linear estimator for a non-linear system obtained by *linearization* of the non-linear state and observation equations. In the context of this paper, the state vector $\mathbf{x}$ comprises of the predicted position and orientation of the vehicle but can also consist of the tri-axial velocities of the vehicle. We are in the process of developing sophisticated kinematic models for vehicles that are part of the object classification module. For example, tracked vehicles like tanks require that the track-soil interactions be taken into account resulting in kinematic models that enable slip estimation.

To formulate a Kalman filter algorithm, process and observation (measurement) models are needed. In view of the availability of data at discrete instants of time from asynchronous sensors and implementation based on digital computers, a discrete-time formulation of the continuous-time vehicle kinematic models are necessary. In discrete-time, only discrete sampling instants $t_0, t_1, \ldots$ are considered. The discrete-time process model is usually derived by integrating the continuous-time process model between two consecutive time steps. A general discrete-time process model can be expressed as

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, k\right) + \mathbf{w}_k \tag{1}$$

where $\mathbf{f}\left(\cdot, \cdot, k\right)$ is a discrete function that maps the previous state and control inputs to the current state, $\mathbf{x}_k$ is the state at time instant k, $\mathbf{u}_k$ is a known control vector, and $\mathbf{w}_k$ is the discrete process noise. The process noise is assumed to be a Gaussian-distributed random variable of zero mean with covariance $\mathbf{Q}_k$ and is written as*:

$$\mathbf{w}_k \sim \mathrm{N}\left(\mathbf{0}, \mathbf{Q}_k\right); \quad \mathrm{E}\left[\mathbf{w}_k\right] = \mathbf{0} \ \ \forall\, k$$

where $\mathrm{E}\left[\cdot\right]$ is the mathematical expectation operator.

The process model is assumed to model any deterministic behavior of the state. The process noise at any time is assumed to be independent of the state of the system and the process noise at any other time so that

$$\mathrm{E}\left[\mathbf{w}_i \mathbf{x}_j^T\right] = \mathbf{0} \ \ \forall\, i, j; \quad \mathrm{E}\left[\mathbf{w}_i \mathbf{w}_j^T\right] = \delta_{ij}\mathbf{Q}_i \ \ \forall\, i, j$$

where δ_{ij} is the dirac-delta function.

Observations of the state $\mathbf{x}_k$ are made according to the observation model:

$$\mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k, k\right) + \mathbf{v}_k \tag{2}$$

where $\mathbf{h}\left(\cdot, k\right)$ is the discrete function that maps the current state to observations. $\mathbf{v}_k$ is the measurement noise source and plays a similar role in the observation model to the role played by the process noise in the process model. It accounts for effects that are not modeled explicitly and is assumed to be a Gaussian-distributed random variable of zero mean with covariance $\mathbf{R}_k$ and is written as

$$\mathbf{v}_k \sim \mathrm{N}\left(\mathbf{0}, \mathbf{R}_k\right); \quad \mathrm{E}\left[\mathbf{v}_k\right] = \mathbf{0} \ \ \forall\, k$$

It is assumed that the observation noise at any time is independent of the state of the system and also of the observation noise at any other time. A further assumption is that the process and observation noises are also independent.

$$\mathrm{E}\left[\mathbf{v}_i \mathbf{x}_j^T\right] = \mathbf{0} \ \ \forall\, i, j; \quad \mathrm{E}\left[\mathbf{v}_i \mathbf{v}_j^T\right] = \delta_{ij}\mathbf{R}_i \ \ \forall\, i, j; \quad \mathrm{E}\left[\mathbf{w}_i \mathbf{v}_j^T\right] = \mathbf{0} \ \ \forall\, i, j$$

The estimate of the true state $\mathbf{x}_i$ conditioned on the sequence of observations $\mathbf{Z}^j = \{\mathbf{z}_1, \ldots, \mathbf{z}_j\}$ is denoted by $\hat{\mathbf{x}}_{i|j}$. The EKF computes the estimate that minimizes the mean squared error in the estimate and is equal to the expected value of the state conditioned on the sequence of observations:

$$\hat{\mathbf{x}}_{i|j} = \mathrm{E}\left[\mathbf{x}|\mathbf{Z}_j\right]$$

*The notation $\mathbf{x} \sim \mathrm{N}\left(\mathbf{m}, \mathbf{P}\right)$ indicates that $\mathbf{x}$ is a Gaussian (normal) random vector with mean $\mathbf{m}$ and covariance $\mathbf{P}$.

The error between the true state $\mathbf{x}_i$ and the estimated state $\hat{\mathbf{x}}_{i|j}$ is denoted by $\tilde{\mathbf{x}}_{i|j}$ and the covariance of $\hat{\mathbf{x}}_{i|j}$ is denoted by $\mathbf{P}_{i|j}$

$$\tilde{\mathbf{x}}_{i|j} \stackrel{\triangle}{=} \mathbf{x}_i - \hat{\mathbf{x}}_{i|j}; \quad \mathbf{P}_{i|j} = \mathrm{E}\left[\tilde{\mathbf{x}}_{i|j}\tilde{\mathbf{x}}_{i|j}^T\right]$$

The EKF computes the estimate $\hat{\mathbf{x}}_{i|j}$ and the associated covariance $\mathbf{P}_{i|j}$. A cycle in the estimation process is now considered. The aim is to formulate an estimate $\hat{\mathbf{x}}$ of the state $\mathbf{x}$ at time step k, based only on the estimate at time step $(k-1)$ and an observation made at time step k. A prediction stage, which forms an estimate of the state at time step k based only on the estimate at time step $(k-1)$, and an update stage, which combines the predicted estimate at time step k with the observation at time step k, form an estimate at time step k. The iteration then repeats recursively.

The state estimate at time $(k-1)$ is assumed to be approximately equal to the conditional mean

$$\hat{\mathbf{x}}_{(k-1|k-1)} = \mathrm{E}[\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}]$$

The predicted state estimate $\hat{\mathbf{x}}_{(k|k-1)}$ at time k based on information up to time $(k-1)$ is given by expanding Equation (1) as a Taylor series about the estimate $\hat{\mathbf{x}}_{(k-1|k-1)}$ to obtain:

$$\mathbf{x}_k = \mathbf{f}\left(\hat{\mathbf{x}}_{(k-1|k-1)}, \mathbf{u}_k, k\right) + \nabla\mathbf{f}_{\mathbf{x}_k}\left[\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{(k-1|k-1)}\right] + O\left(\left[\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{(k-1|k-1)}\right]^2\right) + \mathbf{w}_k$$

where $\nabla\mathbf{f}_{\mathbf{x}_k}$ is the Jacobian evaluated at $\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{(k-1|k-1)}$. Neglecting second and higher order terms and taking expectations conditioned on the first $(k-1)$ observations:

$$\hat{\mathbf{x}}_{(k|k-1)} = \mathrm{E}\left[\mathbf{x}_k|\mathbf{Z}_{k-1}\right] = \mathbf{f}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k, k\right) \tag{3}$$

In an autonomous vehicle navigation context, the prediction stage uses a model of the motion of the vehicle (a process model having the form described in Equation (1)) to predict the vehicle position, $\hat{\mathbf{x}}_{(k|k-1)}$, at instant k given the information available until and including instant $(k-1)$. The state prediction function $\mathbf{f}(\cdot)$ is defined by Equation (1) assuming zero process and control noise. The prediction of state is therefore obtained by simply substituting the previous state and current control inputs into the state transition equation with no noise. For the wheeled vehicle under consideration† the state prediction is given by:

$$\begin{bmatrix} x_{v_{(k|k-1)}} \\ y_{v_{(k|k-1)}} \\ \phi_{v_{(k|k-1)}} \end{bmatrix} = \begin{bmatrix} x_{v_{(k-1|k-1)}} \\ y_{v_{(k-1|k-1)}} \\ \phi_{v_{(k-1|k-1)}} \end{bmatrix} + \Delta T \begin{bmatrix} V_k \cos\phi_{v_{(k-1|k-1)}} \\ V_k \sin\phi_{v_{(k-1|k-1)}} \\ \omega_k \end{bmatrix}$$

where ΔT is the synchronous sampling interval between states at discrete-time instants $(k-1)$ and k. The tuple given by $(x_{v_k}, y_{v_k}, \phi_{v_k})$ denotes the (x, y) coordinates and orientation of the vehicle, respectively. The control signals applied to the vehicle are $\mathbf{u}_k = [V_k, \omega_k]$ where V_k is the linear velocity and ω_k is the angular velocity of the robot at time instant k. The errors due to the control signals V and ω are modeled as simple additive noise sources, δV and $\delta\omega$, about their respective means $\bar{V}$ and $\bar{\omega}$ as $V_k = \bar{V}_k + \delta V_k$ and $\omega_k = \bar{\omega}_k + \delta\omega_k$. The error source vector is defined as: $\delta\mathbf{w}_k = [\delta V_k, \delta\omega_k]^T$ and is a direct effect of the associated modeling errors and uncertainty in control. The source errors δV and $\delta\omega$ are assumed to be zero-mean, uncorrelated Gaussian sequences with constant variances σ_V^2 and σ_ω^2, respectively. The variances are determined experimentally to reflect true noise variances. The prediction of covariance is obtained as:

$$\mathbf{P}_{(k|k-1)} = \mathrm{E}[\tilde{\mathbf{x}}_{(k|k-1)}\tilde{\mathbf{x}}_{(k|k-1)}^T|\mathbf{Z}_{k-1}] = \nabla\mathbf{f}_{\mathbf{x}_k}\mathbf{P}_{(k-1|k-1)}\nabla\mathbf{f}_{\mathbf{x}_k}^T + \nabla\mathbf{f}_{\mathbf{w}_k}\mathbf{Q}_k\nabla\mathbf{f}_{\mathbf{w}_k}^T \tag{4}$$

†Instead of predicting the trajectory of a vehicle, the general translational ($\mathbf{x}_i^o(t)$) and rotational ($\theta_i^o(t)$) motions of a moving object (assuming constant or slowly changing acceleration) can be predicted in terms of its position, velocity and acceleration as: $\mathbf{x}_i^o(t) = \begin{bmatrix} x_i(t) \\ \dot{x}_i(t) \\ \ddot{x}_i(t) = k_{1_i} \end{bmatrix}$ and $\theta_i^o(t) = \begin{bmatrix} \theta_i(t) \\ \dot{\theta}_i(t) \\ \ddot{\theta}_i(t) = k_{2_i} \end{bmatrix}$, respectively, where k_1 and k_1 are constants and the superscript o denotes the moving object. It is straightforward to extend the EKF equations to the this case.

The predicted covariance of the vehicle, $\mathbf{P}_{(k|k-1)}$, is thus computed using the Jacobian of the state propagation equation linearized about the current vehicle state estimate, $\nabla\mathbf{f}_{\mathbf{x}}$ and about the process noise model, $\nabla\mathbf{f}_{\mathbf{w}}$ with $\mathbf{Q}_k$ being the noise strength matrix and are given by:

$$\nabla\mathbf{f}_{\mathbf{x}_{v_k}} = \begin{bmatrix} 1 & 0 & -\Delta T V_k \sin\phi_{v_{(k-1|k-1)}} \\ 0 & 1 & \Delta T V_k \cos\phi_{v_{(k-1|k-1)}} \\ 0 & 0 & 1 \end{bmatrix}; \quad \nabla\mathbf{f}_{\mathbf{w}_k} = \Delta T \begin{bmatrix} \cos\phi_{v_{(k-1|k-1)}} & 0 \\ \sin\phi_{v_{(k-1|k-1)}} & 0 \\ 0 & 1 \end{bmatrix}; \quad \mathbf{Q}_k = \begin{bmatrix} \sigma_{V_k}^2 & 0 \\ 0 & \sigma_{\omega_k}^2 \end{bmatrix}$$

When an exteroceptive sensor observation becomes available, the states of the EKF (comprising of the moving object's position and velocities) are to be updated so that during the next cycle the prediction starts from a reasonably known position. This is important since the EKF computes the future position based on the current position. If the prediction starts from a wrong position, the cumulative errors will result in an estimate that will be far from the truth. If it is desired to track the position of a moving object, one should predict its future position and velocity in addition to the associated uncertainties both in position and velocity.

Once the state and covariance predictions are available, the next step is to compute a predicted observation and a corresponding innovation for updating the predicted state. Expanding Equation (2) as a Taylor series about the predicted state $\hat{\mathbf{x}}_{(k|k-1)}$

$$\hat{\mathbf{z}}_k = \mathbf{h}\left(\hat{\mathbf{x}}_{(k|k-1)}, \mathbf{u}_k, k\right) + \nabla\mathbf{h}_{\mathbf{x}_k}\left[\hat{\mathbf{x}}_{(k|k-1)} - \mathbf{x}_k\right] + O\left(\left[\hat{\mathbf{x}}_{(k|k-1)} - \mathbf{x}_k\right]^2\right) + \mathbf{v}_k$$

where $\nabla\mathbf{h}_{\mathbf{x}_k}$ is the Jacobian evaluated at $\mathbf{x}_k = \hat{\mathbf{x}}_{(k|k-1)}$. The predicted observation $\hat{\mathbf{z}}_{(k|k-1)}$ is found by using the non-linear relation described in Equation (2) and taking expectations conditioned on the first $(k-1)$ observations by considering only terms up to the first order and neglecting higher order terms such that

$$\hat{\mathbf{z}}_{(k|k-1)} \stackrel{\triangle}{=} \mathrm{E}[\mathbf{z}_k|\mathbf{Z}_{k-1}] = \mathbf{h}(\hat{\mathbf{x}}_{(k|k-1)})$$

The difference between the actual observation and the predicted observation at time step k is termed the innovation[‡] and is written as: $\nu_k = \mathbf{z}_k - \hat{\mathbf{z}}_{(k|k-1)}$. The innovation covariance is found by squaring the estimated observation error and taking expectations conditioned on the first $(k-1)$ measurements

$$\mathbf{S}_k = \mathrm{E}\left[\tilde{\mathbf{z}}_{(k|k-1)}\tilde{\mathbf{z}}_{(k|k-1)}^T\right] = \nabla\mathbf{h}_{\mathbf{x}_k}\mathbf{P}_{(k|k-1)}\nabla\mathbf{h}_{\mathbf{x}_k}^T + \mathbf{R}_k \tag{5}$$

The observations that arrive are accepted only if the observation falls inside the normalized innovation validation gate, $\nu_k^T\mathbf{S}_k^{-1}\nu_k \leq \epsilon_\gamma$, where ν_k is the innovation defined as the difference between the actual and predicted positions. The value of ϵ_γ can be chosen from the fact that the normalized innovation sequence is a χ^2 random variable with m degrees of freedom (m being the dimension of the observation).[24] Once a validated observation is available, the update of the estimate equal to the weighted sum of the observation and the prediction can be computed as:

$$\hat{\mathbf{x}}_{(k|k)} = \hat{\mathbf{x}}_{(k|k-1)} + \mathbf{W}_k\nu_k$$

where $\mathbf{W}_k$ is the Kalman gain matrix determined by the relative confidence in vehicle prediction and observation and determines the influence of the innovation on the updated estimate. The error in the updated estimate is

$$\tilde{\mathbf{x}}_{(k|k)} = \mathbf{x}_k - \hat{\mathbf{x}}_{(k|k)} = \mathbf{x}_k - \left[\hat{\mathbf{x}}_{(k|k-1)} + \mathbf{W}_k\nu_k\right] = \tilde{\mathbf{x}}_{(k|k-1)} - \mathbf{W}_k\nu_k$$

The covariance update is

$$\mathbf{P}_{(k|k)} = \mathrm{E}\left[\tilde{\mathbf{x}}_{(k|k)}\tilde{\mathbf{x}}_{(k|k)}^T\right] = \mathbf{P}_{(k|k-1)} - \mathbf{W}_k\mathbf{S}_k\mathbf{W}_k^T$$

where the Kalman gain matrix is given by

$$\mathbf{W}_k = \mathbf{P}_{(k|k-1)}\nabla\mathbf{h}_{\mathbf{x}_k}^T\mathbf{S}_k^{-1}$$

[‡]Also called residual in *filtering* terms.

4.2. Preliminary Results and Discussion

Figure 3 shows the predicted trajectory of the moving wheeled vehicle. The 2σ uncertainty ellipses are shown along certain predicted positions (denoted by diamonds) in the section considered for illustration. At **A**, there is a lower uncertainty as reflected by the size of the ellipse but region **B** of the trajectory the size of the ellipses are larger indicative of the growing uncertainty. At **C**, there is a positional fix and accordingly the size of the ellipses reduces as shown.

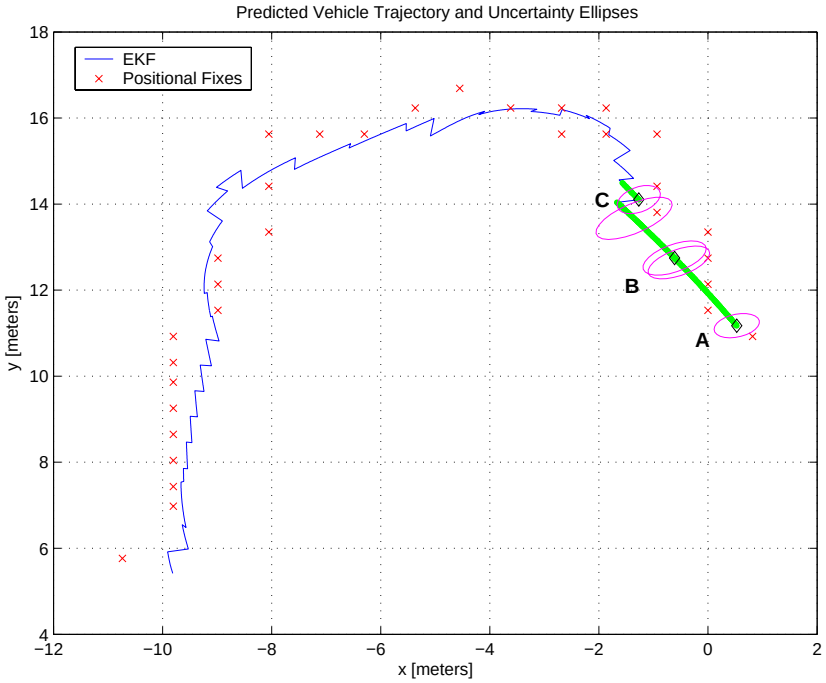

Figure 3. Predicted Trajectory of the Wheeled Vehicle. The 2σ uncertainty ellipses are also shown.

The positional fixes provide external aiding to reset the predicted estimates. The absence of such external aiding would cause the divergence[§] of the estimates of the vehicle and will not result in good prediction estimates for the next cycle. In Figure 4, the periodic rise and fall of the position standard deviations can be seen. The increase in standard deviations is due to the trajectory being estimated based on the prediction alone and the decrease is due to the corrections offered by the positional fixes. The standard deviation is at a maximum when such corrections are not available.

Currently, the predictions take place in the order of $10 - 20$ time steps into the future (where each time step is on the order of $1/20$ of a second). The associated position uncertainty gradually increases as the time horizon increases. Because the confidence values progressively decrease as the planning horizon increases, we plan to use a threshold value to determine the point at which the framework no longer provides non-divergent estimates to allow for safe path planning. Although the emphasis of this paper is on vehicles, the proposed framework can be easily extended to include various moving objects. After object classification and prediction, the resulting information is stored as a set of mathematical equations and instantiated in multi-dimensional occupancy grids that are used for path planning and collision avoidance in the dynamic environment. The equations and the resulting instantiation are constantly updated as new information becomes available. These aspects are beyond the scope of this paper and are being submitted for publication in an upcoming conference.

[§]Filter divergence occurs when the computed error covariance is too optimistic and misleading.

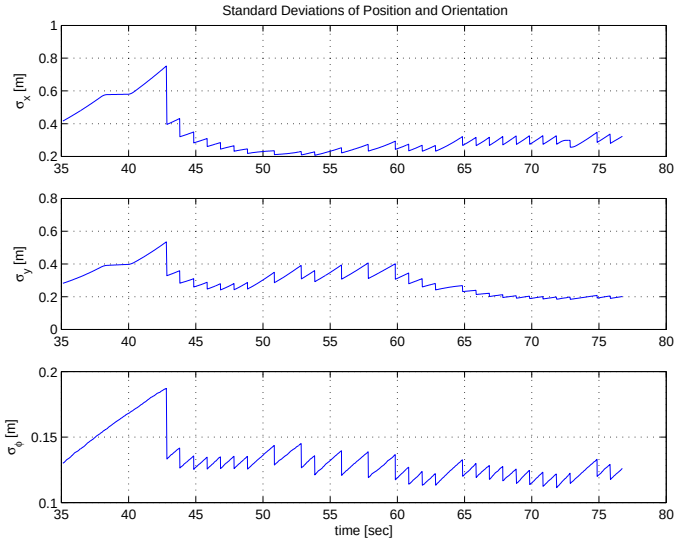

Figure 4. Standard Deviation of the Wheeled Vehicle's Position and Orientation Estimates.

5. CONCLUSIONS AND FURTHER RESEARCH

To successfully plan paths and to navigate in an unstructured environment, an UGV should have the ability to predict the future locations of moving objects that could interfere with its path. This paper described our current efforts towards achieving moving object prediction for unmanned ground vehicle navigation in dynamic environments. The short-term predictions were based on an EKF working symbiotically with a probabilistic object classification scheme. The proposed framework was shown to deliver reliable position estimates for a wheeled vehicle and it differs from those cited in the literature by adopting a more generalized view of moving object representation and prediction in concurrently integrating multiple knowledge representation approaches from disparate sources to completely model the information necessary for dynamic planning. The probabilistic object classification and the cross corroboration offered by the EKF in our framework are both novel and efficient.

The future research efforts will concentrate on:

- Supplementing the estimation theoretic framework to allow for long term predictions by including information about the moving objects' intentions, personalities, terrain of travel, etc. with an emphasis on on-road navigation.
- Embedding our framework with multiple target tracking capabilities using techniques such as multiple hypothesis tracking.[25] We are also currently investigating the use of the Interacting Multiple Model algorithm[26] for model management used in the bank of EKFs to facilitate motion analysis in both quantitative and qualitative senses similar to the work described by Zhao *et al.*[12]
- Instantiating the results within a three-dimensional temporal occupancy grid (where the dimensions represent two spatial dimensions (x, y and time dimensions). Based on the output of our framework, each cell within the grid will contain a predictive probability measure indicating the likelihood of the cell being occupied at that given time instant.

REFERENCES

1. J. Albus *et al.*, "4D/RCS Version 2.0: A Reference Model Architecture for Unmanned Vehicle Systems," Tech. Rep. NISTIR 6910, National Institute of Standards and Technology, Gaitherburg, MD 20899, U.S.A., 2002.
2. C. Schlenoff, R. Madhavan, and S. Balakirsky, "Representing Dynamic Environments for Autonomous Navigation," in *Submitted to the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2003.

3. R. Pless, T. Brodsky, and Y. Aloimonos, "Detecting Independent Motion: The Statistics of Temporal Continuity," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(8), pp. 39–57, 2000.
4. V. Philomin, R. Duraiswami, and L. Davis, "Pedestrian Tracking from a Moving Vehicle," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 350–355, Oct. 2000.
5. D. Gavrila and V. Philomin, "Real-time Object Detection for "Smart" Vehicles," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 87–93, 1999.
6. W. Thompson and T. Pong, "Detecting Moving Objects," *International Journal of Computer Vision* **4**, pp. 39–57, 1990.
7. K. Saneyoshi and K. Tone, "Stereo Image Understanding System for Autonomous Vehicle to Avoid Collision," in *Proceedings of the Asian Conference on Vision*, pp. 257–260, 2002.
8. B. Heisels, U. Krebel, and W. Ritter, "Tracking Non-rigid, Moving Objects Based on Color Cluster Flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 257–260, 1997.
9. M. Hebert, "Active and Passive Range Sensing for Robotics," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 102–110, 2000.
10. H. Kenyon, "LADAR Illuminates Optical Sensors," *Signal AFCEA's Journal for Communications, Electronics, Intelligence, and Information Systems Professionals* , p. 63, Aug. 2002. `http://www.us.net/signal`.
11. C. Thorpe, O. Clatz, D. Duggins, J. Gowdy, R. MacLachlan, J. Miller, C. Mertz, M. Siegel, C. Wang, and T. Yata, "Dependable Perception for Robots," in *Proceedings of the International Advanced Robotics Programme, IEEE Robotics and Automation Society*, May 2001.
12. L. Zhao and C. Thorpe, "Qualitative and Quantitative Car Tracking from a Range Image Sequence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 496–501, June 1998.
13. C.-C. Wang and C. Thorpe, "Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas," in *Proceedings of the IEEE International Conference on Robotics and Automation (to appear)*, May 2003.
14. Q. Zhu, "Hidden Markov Model for Dynamic Obstacle Avoidance of Mobile Robot Navigation," *IEEE Transactions on Robotics and Automation* **7**(3), pp. 390–396, 1991.
15. R. Sharma, "Locally Efficient Path Planning in an Uncertain, Dynamic Environment using a Probability Model," *IEEE Transactions on Robotics and Automation* **8**(1), pp. 105–110, 1992.
16. A. Elnagar and K. Gupta, "Motion Prediction of Moving Objects Based on AutoRegressive Model," *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans* **28**(6), pp. 803–810, 1998.
17. H. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *AI Magazine* **9**(2), pp. 61–74, 1988.
18. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, New York, 1988.
19. R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME-Journal of Basic Engineering* **82**(Series D), pp. 35–45, 1960.
20. N. Kehtarnavaz and S. Li, "A Collision-free Navigation Scheme in the Presence of Moving Obstacles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 808–813, June 1988.
21. W. Shackleford, F. Proctor, and J. Michaloski, "The Neutral Message Language: A Model and Method for Message passing in Heterogeneous Environments," in *Proceedings of the World Automation Conference*, 2000.
22. "`http://www.onesaf.org`."
23. "`http://www.riegl.co.at/scanner_menu.htm`."
24. P. Maybeck, *Stochastic Models, Estimation, and Control Vol. 1*, Academic Press, New York, June 1979.
25. D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Transactions on Automatic Control* **24**(6), pp. 843–854, 1979.
26. H. Blom, "An Efficient Filter for Abruptly Changing Systems," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 656–658, Dec. 1984.