51st IEEE Conference on Decision and Control
December 10-13, 2012. Maui, Hawaii, USA

Terrain-Based Vehicle Localization from Real-Time Data Using Dynamical Models

Emil Laftchiev, Constantino Lagoa, and Sean Brennan

Abstract— **This paper describes a novel method for the location of road vehicles using vehicle pitch data obtained from on-board sensors. The method encodes the road map data using linear dynamical models, and then, during travel, identifies the vehicle location through continuous validation of the previously obtained linear models. The approach presented has several advantages over previous approaches in the literature, namely a smaller computational burden, a more definitive location estimate, and a simplified and more direct way of handling common types of noise. These benefits have the potential to both increase the speed of the localization and to reduce the implementation cost of terrain-based localization. The method is tested in simulation using real-world road data collected in State College PA, USA. Performance is demonstrated both in a noise-free and noisy environments, and a bound is shown on the convergence distance.**

I. INTRODUCTION

This paper addresses the problem of localizing a road vehicle without using the global positioning system (GPS). Specifically, the paper focuses on terrain-based vehicle localization using only vehicle pitch information gathered from the inertial measurement unit (IMU). Localization using this data has been previously studied by other authors [1]–[4].

Similar to other forms of alternate vehicle localization which do not require an external position reference, localization based on vehicle pitch suffers from several problems in implementation. First, the volume of data generated by the vehicle's IMU is very large, requiring any localization algorithm to tackle the task of efficiently storing and parsing through the data. Second, localization through "brute force" comparison is computationally inefficient for an on-board computer. Lastly, because signal-to-noise ratio (SNR) of the acquired the data is often uncontrollable, the vehicle localization can often only be computed within some bounds of uncertainty.

This paper will present a novel approach to vehicle localization using *switched linear systems*. This approach creates an efficient structure to compress and parse road map data, while enabling localization to within an exact distance specified by the extracted road map. In addition, this paper will also present some initial approaches to mitigating the effects of noise on the developed algorithm.

Emil Laftchiev is a PhD candidate in the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16801, USA Emil.Laftchiev@psu.edu

Constantino Lagoa is a Professor in the Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16801, USA Lagoa@psu.edu

Sean Brennan is an Associate Professor in the Department of Nuclear and Mechanical Engineering, The Pennsylvania State University, University Park, PA 16801, USA snb10@psu.edu

A. Problem Statement

Vehicle pitch observed by the IMU is a filtered version of road grade. One interpretation of the mapping of vehicle position to vehicle pitch is shown in figure 1. In this diagram,

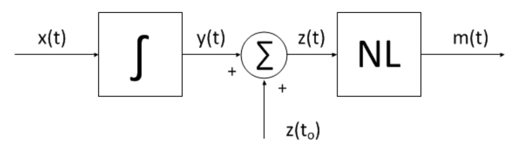


Fig. 1. Sensor Output Diagram

the variable $x(t)$ represents the vehicle velocity; the variable $y(t)$ represents the distance traveled by the vehicle; the variable $z(t)$ represents the position of the vehicle relative to its initial position, $z(t_o)$; and the variable $m(t)$ represents the vehicle pitch measurement that is non-linearly mapped from the vehicle's position.

In this paper, road pitch data will be collected *a priori*, and structured through the use of switched linear systems. The switched linear system will create a set of linear dynamical models that describe the continuous road map. During travel, the vehicle will compare the collected pitch data and determine agreement with the pre-extracted models. Because the models are linear, and extracted prior to localization, this approach will reduce the in-vehicle computational complexity. Additionally, the linear treatment of incoming data will ease the implementation of noise mitigation schemes for practical implementation.

B. Previous Research

The problem of vehicle localization is similar to the problem of autonomous robot localization. The overarching goal for robot navigation is simultaneously building an area map and locating the robot on it, termed the simultaneous mapping and localization (SLAM) problem. A good overview of SLAM is provided in [5].

The most common approaches to SLAM are based on the work of Smith and Moutarlier, which build a probabilistic framework for the problem solution [6]–[8]. This approach provides a recursive solution to the SLAM problem using an Extended Kalman Filter (EKF), and an estimate of the uncertainty in the vehicle location. However, the approach suffers from computational complexity that scales quadratically with the size of the state vector.

Alternative approaches to SLAM are either qualitative or computationally based. For qualitative methods, the map and vehicle location estimates are generated using the relative

978-1-4673-2066-5/12/$31.00 ©2012 IEEE

location of nearby landmarks [9], [10]. Computational approaches use varying types of uncertainty descriptions. For example, Thrun [11] uses a Bayesian approach to avoid the Gaussian assumption normally required by Kalman filters. On the other hand, Gustaffson [12] avoids the same Gaussian assumption through the use of particle filters.

The approaches of greatest interest here are the landmark or map matching approaches where map data is collected *a priori* [13]. In general these approaches assume the capability to extract a set of features from sensor data which are then used for localization [14]. As a broad outline, map data is first collected offline and a pre-determined set of features is extracted. The choice of features is designer and sensor specific and plays a pivotal role in the subsequent performance of the algorithm. Typically, feature extraction algorithms are non-linear and thus computationally expensive. There exist many examples of features in the literature [1], [15]–[18]. These features serve the dual purpose of compressing the map data while retaining only the most robust data points with respect to sensor noise. Then, in the online phase, the probability of the robot's location and of the detected features is calculated using any of the previously discussed methods.

As previously noted, alternative vehicle localization algorithms carry significant computational costs. In fact, it was not until the advent of low-cost fast microcomputers that the solutions to this problem could be implemented. The leading approach to reducing this computational cost is to use approximations that maintain the sparsity of the information matrix [19]. More recently, empirical observations about the information matrix have led to the use of sparse extended Kalman filters in localization [20]–[23]. This approach allows for constant time updates but can lead to inconsistencies in the global map.

The underlying assumption in all approaches to vehicle localization is that there is an inherently non-linear process that describes the environment of the vehicle. Therefore, probabilities of landmark and vehicle locations are determined computationally in the vehicle itself, requiring significant investment in on-board vehicle technology that can perform those calculations.

II. Algorithm Description

This section describes the development of the localization algorithm. Section II-A describes the extraction of the switched linear system from pre-recorded vehicle pitch data. Then section II-B describes the method by which the switched linear systems are organized for efficient parsing during vehicle travel. Section II-C develops the on-line vehicle localization algorithm. Lastly, section II-D describes the tracking of a vehicle whose location has been identified.

A. Model Extraction

The pre-recorded road pitch data is modeled using a switched linear system. The switched linear system is a set of auto-regressive (AR) linear dynamical models that describe non-overlapping segments of collected data. Auto-regressive models use data points collected prior to an instant to estimate the data point at that instant. The error between the estimate and the collected data is bounded by a chosen bound ε. The general form of an AR model is

$$\begin{aligned} m[d] &= a_1 m[d-1] + ... + a_N m[d-N] + \varepsilon[d] \\ &= \mathbf{A}\,\mathbf{x}[d] + \varepsilon[d], \qquad |\varepsilon[d]| \le \varepsilon \end{aligned} \tag{1}$$

where $m[d]$ represents the current sample of data that is collected, $\mathbf{A} = [a_1, ..., a_N]$ is a vector that contains the coefficients of the linear model of order N, the vector $\mathbf{x}[d] = [m[d-1], ..., m[d-N]]$ contains the previous N samples of data, i.e., the so-called regressor vector, and $\varepsilon[d]$ is the model output error bounded by the above mentioned bound ε.

The greedy algorithm for switched system identification developed by Ozay et. al. in [24] is used to both obtain the AR model specified above and to simultaneously segment the collected map data into non-overlapping segments. This algorithm is shown in table I.

The algorithm begins at the $(N+1)^{th}$ data point, labeled d_0. Starting at this initial data point, the algorithm searches for the largest interval for which it is possible to obtain a single AR system that satisfies the error bound for every point. Once this is not possible, a transition is declared and the corresponding data index is labeled τ_o. The segment is removed and this process is repeated until d_{max} is reached. The resulting set of data segments spans the values of consecutive transition points: (d_0,τ_0),(τ_0,τ_1), etc. The optimality of this algorithm is described by the following *Proposition* [24].

Greedy Algorithm
Initialize Constants:
model order : N
precision variable: ε
segment index: n = 0
initial loop index: d_0 = N + 1
first "transition point": τ_0 = d_0
Algorithm Loop:
FOR i = d_0:d_{max}
Find a vector $\mathbf{A}$ such that:
$\mathcal{F} : \{\|m[d] - \mathbf{A}\,\mathbf{x}[d]\| \le \varepsilon \;\; \forall d \in [\tau_n, i]\}$
IF $\mathcal{F}$ is infeasible
Store $\mathbf{A}$ from index i-1, set the data index bounds:
I_n = [τ_n,i], iterate the segment index: n = n + 1,
Store the transition point: τ_n = i
END IF
END FOR
I_n = [τ_n,d_{max}] and $\tau = \{I_j\}_{j=0}^{n}$
Return n and τ

TABLE I

Optimal Greedy Algorithm

Proposition: Given a bound on the error, ε, and a model of order N, the algorithm described in Table I breaks the collected data set into the smallest possible number of segments.

B. Vehicle Localization Structures

The switched linear system can be used to both identify the vehicle's exact location and its proximity. For instance, a feasible model is a set of possible vehicle locations, while a switch between two consecutive models identifies an exact location. The latter occurs because at a model transition point the data no longer agrees with one model but agrees with the next.

There are relatively few transition points on any data set. Furthermore, because the models describe the irregularities in the road surface, and because these regularities are correlated to specific locations, it is unlikely that two transition points would satisfy the same data interval. Therefore, the frequency of transitions between models is an important factor in the speed of vehicle localization.

The frequency of transitions can be increased by lowering the error bound ε. Ideally, the data would be segmented into a large number of small segments with tight ε bounds. This leads to two practical implementation problems. First, small error bounds increase the susceptibility to noise and second, increasing the number of segments directly increases the number of computations performed initially.

These implementation problems can be mitigated through the extraction of a series of progressively tighter switched linear systems. These systems can be structured in a tree-like structure with “coarse” models at the top and “fine” models at the bottom. Figure 2 shows an example of this type of model structure. The models are labeled as $\mathbf{A}_{k,n}$, where k denotes the model structure level, n denotes the segment index k^{th} level, and α denotes the error bound contraction constant. Each model structure will have L levels, and M_k models per level.

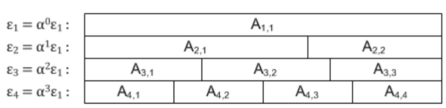


Fig. 2. An example model structure

C. Locating a Vehicle

During the vehicle travel, the observed pitch data is used to determine agreement with the pre-extracted models. This process is described in table II.

Similar to the greedy algorithm, the localization procedure in Table II begins with the collection of N + 1 pitch data points. The data index is set to $d = N + 1$, and the model output error is calculated. For each model the output error is compared to the corresponding error bound, $\varepsilon_{k,n}$. This comparison can be written as,

$$|m[d] - \mathbf{A}_{k,n}\mathbf{x}[d]| \gtreqless \varepsilon_{k,n} \tag{2}$$

If the model error does not exceed the bound, then the current segment is a possible, or feasible, set of vehicle positions; otherwise the segment is infeasible. At each successive level, only segments whose data lies in a feasible segment from the previous level are tested. This method of elimination rapidly and monotonically reduces the computational burden of localization with respect to distance traveled.

The localization loop continues until either a single segment on level L remains feasible or a transition point between models is determined at some data index, d. The latter case is preferable because at a transition point, multiple inequalities are satisfied. These inequalities are shown below in equation 3.

$$\begin{aligned} |m[d-1] - \mathbf{A}_{k,n}\mathbf{x}[d-1]| &\leq \varepsilon_{k,n} \\ |m[d] - \mathbf{A}_{k,n}\mathbf{x}[d]| &> \varepsilon_{k,n} \\ |m[d] - \mathbf{A}_{k,n+1}\mathbf{x}[d]| &\leq \varepsilon_{k,n+1} \end{aligned} \tag{3}$$

The first inequality shows that the vehicle was in segment n on level k at the data point immediately preceding point d. The next inequality shows that the vehicle is no longer in segment $\{k, n\}$, and the third inequality shows that the vehicle is now in segment $\{k, n+1\}$. The latter two inequalities have domains that only overlap at the transition point and therefore provide strong evidence about the location of the vehicle. Thus in order to quickly converge to the correct vehicle location, the bottom structure level, L, must be finely segmented, with multiple transition points.

D. Vehicle Tracking

Once the vehicle has been localized, the localization algorithm can be used for tracking. Similar to localization,

Vehicle Localization

Initialization

Collect N + 1 Data points:
set loop index: d = N + 1
initialize data vector: $\mathbf{x}[\cdot]$ = $m[1 : N]$

Begin Localization Loop

WHILE localization flag == 0
FOR k = 1:L
  FOR n = 1:$M_k - 1$
    IF the parent segment of $\mathbf{A}_{k,n}$ is feasible:
    $e_{k,n}$ = $|m[d] - \mathbf{A}_{k,n}\mathbf{x}[\cdot]|$
    $e_{k,n+1}$ = $|m[d] - \mathbf{A}_{k,n+1}\mathbf{x}[\cdot]|$
    IF $e_{k,n} < \varepsilon_{k,n}$
      set $\mathbf{A}_{k,n}$ as feasible
    ELSEIF $e_{k,n} > \varepsilon_{k,n}$ AND $e_{k,n+1} < \varepsilon_{k,n+1}$
      then the data point $m[d]$ is a transition point
      obtain from map the corresponding τ_n
      localization flag = 1
    ELSE
      set $\mathbf{A}_{k,n}$ as infeasible
    END IF
    END IF
  END FOR
END FOR
Collect an additional data point
iterate the index: $d = d + 1$
iterate the data vector: $\mathbf{x}[\cdot]$ = $m[d - N : d - 1]$
END WHILE

Following a Detected Transition Point

FOR k = 1:L
  FOR n = 1:M_k
    IF τ_n is inside the segment
      Set the segment as feasible
    ELSE
      Set the segment as infeasible
    END IF
  END FOR
END FOR

Exit to Tracking Loop

TABLE II
VEHICLE LOCALIZATION ALGORITHM

the observed data is used to verify model agreement with the remaining feasible models. However, during vehicle tracking only a single feasible segment is validated for each model structure level.

Typically tracking begins at an identified transition point. Each successive data point is then evaluated using equation (2). During travel within a segment, the position on the map is updated using odometery readings. When the vehicle records a data point that does not agree with the model, two tests are performed. First, vehicle travel since the last transition point is compared to the segment size. Then second the transition point check in equation (3) is performed. If both tests agree, the models are iterated and the process is repeated. Otherwise, the initial localization is assumed to be erroneous, and the vehicle localization is repeated. In general, once a correct transition point is located, only large changes in the road surface or unexpected maneuvers lead to errors in the tracking phase of the algorithm.

III. Measurement Noise

In practice there are many sources of interference that will affect the algorithm performance. The most common and pervasive source of noise is the IMU. An IMU contains three gyroscopes that are placed on the 3-dimensional coordinate axes to provide an angular rate measurements. These rate measurements are highly accurate over time, but when integrated to obtain orientation and position estimates, even small deviations lead to an accumulation of large estimate errors. IMU noise is characterized by the manufacturers in terms of Allan variance with primary components of angle random walk noise and bias noise.

With respect to noise characteristics there is a wide range of commercially available IMUs. A comparison of the noise characteristics for several sensors was performed by Jerath [3]. Figure 3 illustrates this analysis for a low-cost IMU, ADIS16367; a mid-cost IMU, Crossbow 440; and a high priced IMU, Honeywell HG1700. The top of the figure shows the angle random walk component of the IMU noise and the bottom of the figure shows the bias noise component.

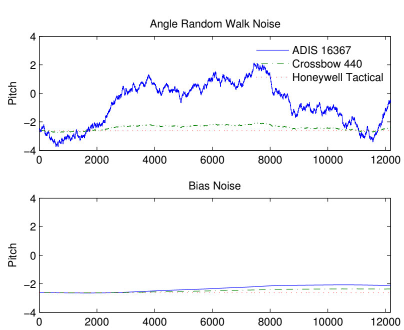


Fig. 3. IMU noise by sensor price. Top: Angle random walk noise Bottom: Bias noise

The map data used in this paper to generate the localization model tree was collected using the HG1700 IMU. Because the noise coefficients of the HG1700 are at least two orders of magnitude smaller than the coefficients of the ADIS16367, this data is regarded as noise free. Then the data was then corrupted using the simulation developed for [3] and the parameters of the Crossbow 440 IMU. Testing with the parameters of the ADIS16367 is beyond the scope of this paper because SNR produced by this sensor is poor and requires advanced mitigation strategies.

A. Bias Noise

The first mitigation strategy addresses the bias noise in the IMU. From the bottom of figure 3, note that sensor bias is relatively constant for neighboring data points. Thus it is reasonable to assume that bias in neighboring points will be approximately the same, and generate a map of differences where each point is the difference between two neighboring pitch values. This is illustrated by the following equation,

$$\Delta m[d] = m[d] + \beta[d] - m[d+1] - \beta[d+1] \tag{4}$$

where $m[d]$ and $m[d+1]$ are adjacent pitch values with bias $\beta[d]$ and $\beta[d+1]$, respectively. The variable $\Delta m[\cdot]$ is the difference and is insensitive to bias noise. Hence, $\Delta m[\cdot]$ can be used during the extraction of the road map models and structure and during the online localization to mitigate the effects of bias noise.

B. Angle Random Walk Noise

The second mitigation strategy addresses the angle random walk noise in the IMU. This noise is modeled as white noise added to the angular rate. Otherwise stated, angle random walk noise adds small perturbations to each acquired data point. These perturbations cannot be eliminated by subtraction like the bias noise. An alternative mitigation strategy is to introduce a tolerance for each data point.

Assume that the perturbations can be bounded by some η_B. Let each data point have a perturbation $|\eta[d]| \leq \eta_B$. Then a model is said to be feasible or agree with the data if a set of constants $\bar{\eta} = [\eta[d], ...\eta[d+N-1]]$ can be found such that,

$$|\Delta m[d] - \mathbf{A}(\mathbf{x}[d] + \bar{\eta})| \leq \varepsilon. \tag{5}$$

The optimal value for η_B must be sufficiently small such that the errors at the transition points cannot be described. In simulation it was determined that η_B values below $\varepsilon/2$ are sufficient for accurate detection of transitions.

One important observation is that because of the initial integration into pitch values, the perturbations at each data point are not independent. Therefore, testing a large horizon of data points simultaneously provides the greatest degree of noise mitigation. Practically, the number of simultaneously tested points is limited by computational power. In this paper, three consecutive points were used to demonstrate the algorithm.

IV. Numerical Results

A. Testing Parameters

When setting up the numerical experiments the two critical parameters are model order, N, and error bound, ε. Unfortunately, there do not exist optimal values for the model order and error bound. Instead these parameters can be varied with respect to one another to create different model structures with similar localization properties. For this paper, the model order was held constant at five, and the error bound was allowed to vary for each model.

The initial value for the error bound is 0.2556 degrees of pitch. This is the smallest value of ε such that at the top of the model structure, a five-coefficient model can be used to describe the entire data segment. For each successive level, the error bound was contracted by a factor of 0.75. This contraction factor was chosen so that each successive level would have approximately twice the number of segments as the previous level. This type of segmentation was chosen for clarity in testing, but can be further optimized for a data-driven in-vehicle process.

B. Noise-Free Environment Results

This section presents the numerical localization results given a noise free environment and knowledge of vehicle odometry. Without noise the localization algorithm converges to the correct location for every experiment. For this reason, the evaluation is focused on the localization speed. The figures below demonstrate both the convergence of the algorithm and the method by which the vehicle location is discerned.

Figure 4 illustrates the convergence of the algorithm to a single possible segment. In the figure, the horizontal axis shows the number of steps d, and the vertical axis shows the number of remaining feasible segments. Note that the algorithm converges to a single segment in 18 steps. The speed of convergence can be increased by segmenting the map into a greater number of segments.

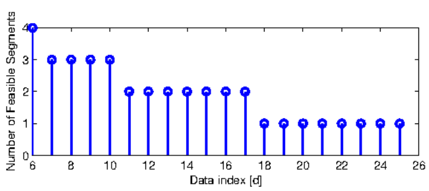


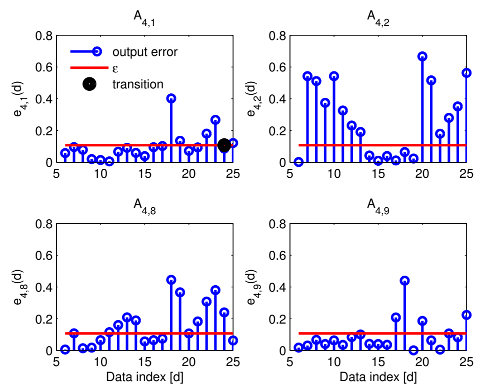
Fig. 4. Algorithm Convergence

The agreement of the data is evaluated using equation (2). The errors for several models and data indices are shown in figure 5. In these plots the horizontal axis shows the number of steps d and the vertical axis shows the model errors. Model errors are plotted even outside of the correct segment to illustrate model agreement and transitions.

By observing the model errors in figure 5, and the elimination of segments in figure 4, it is possible to see the exact time at which feasible segments are eliminated. During the first time instant, d = 6, only models: $A_{4,1}$, $A_{4,2}$, $A_{4,8}$ and $A_{4,9}$ are consistent with the collected data. These segments are labeled feasible and reconsidered at the next data point. At the next time instant, $d = 7$, the error associated with $A_{4,2}$ exceeds $\varepsilon_{4,2}$. This point is labeled as candidate transition point.

At each candidate transition point, the data is evaluated using equation (3), and the next model is checked for agreement. Furthermore, the model error which exceeded the precision bound, is compared to the error observed during model extraction. If all conditions are satisfied a transition point has been found.

At instant d = 7, $A_{4,2}$ does not satisfy the data. Since model $A_{4,3}$, also does not satisfy the data, d = 7 is not a valid transition point. Then at instant d = 11, $A_{4,8}$ no longer agrees with the data. However, since the observed error does not correspond to the expected segment end error, this is also not a transition point. For the next several time instants $A_{4,1}$ and $A_{4,9}$ continue to agree with the data. At d = 18, $A_{4,1}$ no longer agrees with the data. Because both $A_{4,2}$ is feasible, and the expected error for $A_{4,1}$ was observed, this point is a transition point. At this point, the vehicle has been localized, and tracking begins. The convergence of the algorithm is directly proportional to the frequency of the transition points on the pre-extracted map. For a noise-



(a) Feasible model errors at step 6

Fig. 5. Model errors used during localization

free environment, it is possible to place an upper bound on the speed of algorithm convergence. Because localization happens most often at transition points, this bound is the size of the largest segment at the bottom of the model structure. For the example, suppose the model structure from above has 10 levels; then, the longest possible distance a vehicle can travel before selecting a candidate location is 165 data points. Since the data used in this paper has a sample rate of 0.5 meters, this means that the slowest possible convergence rate in this scenario is 82.5 meters.

C. Results in the Presence of Noise

To test the algorithm in the presence of noise, the data is corrupted using the Crossbow 440 noise characteristics. Both mitigation strategies from section III are used. To illustrate the effects of noise, the estimated vehicle position is plotted against the true vehicle position. An example of this plot is shown in figure 6.

As in previous plots, the horizontal axis represents the number of steps taken during localization. The vertical axis represents the index point on the map corresponding to the

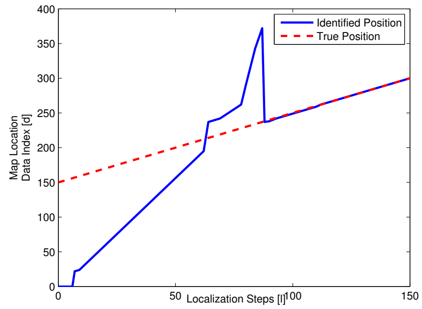


Fig. 6. Position Convergence in the Presence of Noise

vehicle's location. The plot shows that the vehicle began traveling at map index 150. Prior to convergence, the estimated vehicle position is held at "0". After several time steps, an erroneous transition is detected, and the vehicle begins to track its path. This type of error is common in the presence of noisy data. The erroneous path is tracked for about 40 meters, until a new transition point is detected. Following this point, several more transition points are detected in quick succession that correct the assumed vehicle location to its true position.

The results from this trial are typical and illustrate the algorithm's performance in noise. While the initial estimate of location is incorrect, continuous observation of the road data leads to a correction of the vehicle path. This correction time varies according to the underlying map and depends on the regular and fine segmentation of the road.

V. Conclusions and Future Work

This paper presents a novel algorithm for the localization of road vehicles based on the segmentation of the road pitch data using linear dynamical models. In the on-line phase, the models are sequentially eliminated until a segment transition is detected and only the vehicle's path remains viable. In addition to the algorithm some initial approaches to noise mitigation are also presented.

In contrast to previous approaches that build a probabilistic estimate of the vehicle's location, this approach can provide an exact vehicle location, an exact distance required to convergence, and information regarding whether an update for this section of the road is feasible. Furthermore, the treatment of the data is linear which allows for more straightforward treatment of process noise.

The results presented for this paper are exploratory and a more developed version of the algorithm will be presented in future publications. One immediate area of development is the reliance on transition points for localization. The significance of transition points must be reduced to increase robustness to noise. Another area of study is the relationship between the noise bound and the sensor noise description. It may be necessary to add multiple sensors to provide sufficient information to counteract poor sensor SNR.

References

[1] A. Dean, R. Martini, and S. Brennan, "Terrain-based road vehicle localization using particle filters," in *Proceedings of the American Control Conference*, 2008, seattle, WA.

[2] P. Vemulapalli, A. Dean, and S. Brennan, "Pitch based vehicle localization using time series subsequence matching with multi-scale extrema features," in *American Control Conference*, 2011.

[3] K. Jerath and S. Brennan, "Gps-free terrain-based vehicle tracking performance as a function of inertial sensor characteristics," in *Dynamics Systems and Control Conference*, 2011, arlington, VA.

[4] S. Kadetodat, P. Vemulapalli, S. Brennan, and C. Lagoa, "Terrain-aided localization using feature-based particle filtering," in *Dynamic Systems and Control Conference*, 2011.

[5] M. Dissanayke, P. Newman, S. Clark, H. Durrant-Whyte, and M.Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Control*, vol. 17, no. 3, pp. 229–241, June 2001.

[6] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, December 1986.

[7] R. Smith, M. Self, and P. Cheeseman, *Estimating uncertain spatial relationships in robotics*. Springer, 1990, ch. Autonomous Robot Vehicles.

[8] P. Moutarlier and R. Chatila, "An experimental system for incremental environment modeling by an autonomous mobile robot," in *Proceedings of The First International Symposium on Experimental Robotics*, 1990, london, UK.

[9] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, March 1986.

[10] T. Levitt and D. Lawton, "Qualitative navigation for mobile robots," *Journal of Artificial Intelligence*, vol. 44, no. 3, August 1990.

[11] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," in *Machine Learning*, vol. 31, no. 3, 1998, pp. 29–53.

[12] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.

[13] G. Shaffer, J. Gonzalez, and A. T. Stentz, "A comparison of two range-based pose estimators for a mobile robot," in *Proceedings of SPIE Symposium on Mobile Robots*, 1992, boston, MA.

[14] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *The fourth international symposium on Robotics Research*. MIT Press, 1988, pp. 467–474, cambridge, MA.

[15] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[16] M. Bosse, D. Koller, Z. Ghahramani, H. Durrant-Whyte, , and A. Ng, "Keypoint design and evaluation for place recognition in 2d lidar maps," *Journal of Robotics and Autonomous Systems*, vol. 57, no. 12, December 2009.

[17] K. Mikolajczyk and H. Uemura, "Action recognition with motion-appearance vocabulary forest," in *IEEE Conference on Computer VIsion and Pattern Recognition*, 2008, anchorage, AK.

[18] S. Thrun and R. Zlot, "Reduced sift features for image retrieval and indoor localization," in *Australian Conference on Robotics and Automation*, 2004.

[19] P. McLaughlan and D. Murray, "A unifying framework for structure and motion recovery from image sequences," in *Proceedings of IEEE International Conference on Computer Vision*, 1995, boston, MA.

[20] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Ng, "Simultaneous mapping and localization with sparse extended information filters," in *WARF*, 2002.

[21] M. Paskin, "Thin junction tree filters for simulatneous localization in mapping," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2003, san Francisco, CA.

[22] U. Frese, *Treemap: An O(log n) algorithm for simultaneous localization and mapping*. Springer Verlag, 2005, pp. 455–476.

[23] Y. Liu and S. Thrun, "Results for outdoor-slam using sparse extended information filters," in *Proceedings of IRCAs*, 2003.

[24] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps, "A sparsification approach to set membership identification of a class of affine hybrid systems," in *47th IEEE Conference on Decision and Control*, 2008, cancun, Mexico.