2022 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
AUTONOMY, ARTIFICIAL INTELLIGENCE & ROBOTICS TECHNICAL SESSION
AUGUST 16-18, 2022 - NOVI, MICHIGAN

GAUSSIAN PROCESS MODELING OF TERRAIN SLOPE FOR GROUND VEHICLE LOCALIZATION

Jesse Pentzer, PhD[1], **Karl Reichard, PhD**[1]

[1]Applied Research Laboratory, Penn State University, University Park, PA

ABSTRACT

This paper presents a Gaussian process model of terrain slope for use in a GPS-free localization algorithm for ground robots operating in unstructured terrain. A wheeled skid-steer robot is used to map the terrain slope within an operational area of interest. The slope data is sampled sparsely and used as training data for a Gaussian process model with a two-dimensional input. Three different covariance functions for the Gaussian process model are evaluated with hyperparameters selected through maximizing the log marginal likelihood. The resulting Gaussian process model is used in the measurement update function of a localization particle filter to generate expected slope values at particle positions. Preliminary localization testing shows sub-ten meter accuracy with no initial knowledge of position. However, the overall performance of the filter is highly dependent on the variability of the terrain that the robot traverses.

Citation: J. Pentzer, K. Reichard, "Gaussian Process Modeling of Terrain Slope for Ground Vehicle Localization," In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 16-18, 2022.

1. INTRODUCTION

Prior work has shown that terrain variations can be used for vehicle localization in both structured (roadways) and unstructured (off-road) terrains. In both instances, an a priori map of the terrain is required and significant efforts have been undertaken to model the terrain and reduce data requirements for map storage. To provide localization in such environments, we propose the use of estimation filters with terrain inclination measurements provided by Gaussian processes (GPs) trained on previously collected data points. The objectives of this research are to collect terrain variation data using a skid-steer teleoperated robot, model terrain variations using GPs, and use the models for GPS-free localization of a robot driving in the mapped area.

For vehicle road networks, particle filters and unscented Kalman filters have been shown to accurately estimate location when provided with a previously collected map of road surface inclination [1]. Within unstructured environments, terrain

variation has been utilized for localization of smaller ground robots [2]. Gaussian Process Function (GPF) models have been applied to Wi-Fi localization problems both terrestrially and on the International Space Station [3], [4], and have also been utilized to model large areas of terrain [5]. Currently there is no work combining GP models with localization for small to medium sized ground robots, which is the area of work presented in this research.

The use of GPs for terrain modeling in combination with particle filtering is a unique aspect of this work. The combination of these techniques for GPS-free localization of ground robots in unstructured environments has not been presented in the literature. Since GPS signals are prone to jamming, other methods of localization that do not rely on external inputs are of interest. In addition, this approach will allow for the evaluation of how map resolution impacts localization accuracy. This tradeoff is key when determining data collection techniques and data storage/transmission requirements.

The remainder of this paper will discuss the three phases of this research: terrain map data collection, GP terrain modeling, and particle filter localization. A discussion of conclusions and future areas of work ends the paper.

2. TERRAIN DATA COLLECTION

The first phase of this research was the collection of terrain slope data with associated position data to use as inputs to the terrain modeling algorithm. A wheeled skid-steer robot equipped with the required sensors was used to collect terrain slope data with ground truth position data. The robot, shown in Fig. 1, is a model RMP400 system produced by Segway Robotics. The robot is controlled remotely using a standard 2.4 gigahertz RC controller. On-board the robot, the Robot Operating System (ROS) is used to implement sensor drivers, orchestrate the overall software stack, and log data [6]. The attitude of the robot is measured using an Xsens MTi-G attitude and heading reference system (AHRS) with outputs of yaw, pitch, and roll at a rate of 100 hertz with a dynamic accuracy of 1 degree RMS and a resolution of 0.05 degrees. A Hemisphere A325 GPS antenna/receiver operating in Real-Time Kinematic (RTK) mode is mounted on the vehicle to provide location measurements at 20 hertz with an accuracy of 0.01 meters RMS.

Figure 1: Image of RMP400 robot equipped with sensors required for terrain data collection.

Terrain data was collected on a large, grass covered slope near the author's research office. An overlay of the position trajectory of the robot during data collection is shown in Fig. 2. The robot was manually driven parallel to the long dimension of the test area with approximately 5 meter spacing between trajectories. This trajectory was chosen to provide good coverage of the area that could be sub-sampled in later stages of this research to model the terrain slope. In Fig. 2, and all figures showing position in this paper, the North and East position in meters is relative to a local datum chosen to be convenient for displaying the results within this operational area.

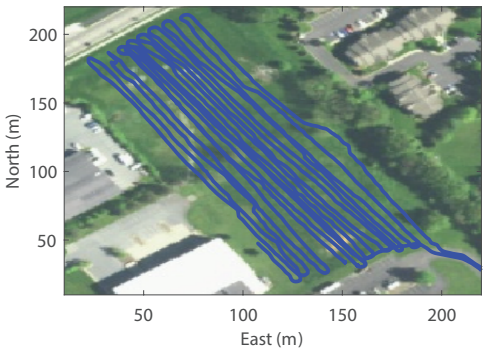

Figure 2: Trajectory of RMP400 during data collection overlaid on satellite image of operational area.

The key terrain property utilized for localization in this work is the total slope of the terrain at a given location. The total slope is a convenient metric because the vehicle will measure the same total slope at a single location regardless of the heading of the vehicle. The MTi-G unit on the robot produces an Euler angle attitude estimate consisting of roll, pitch, and yaw angles. In this work, the robot body frame is defined such that it aligns with a North-East-Down frame centered at the robot position when the attitude is zero. The first step of converting the Euler roll and pitch angles to total slope is creating a quaternion using the following equations where q_N^B is the quaternion representing the rotation from the North-East plane to the body X-Y plane and (ϕ_{NB}, θ_{NB}) are the roll and pitch angles representing the same rotation [7]. The quaternion conversion equations have been simplified using the assumption that heading is zero. The q_{N0}^B component is a function only of the rotation magnitude, and the q_{N1}^B-q_{N3}^B components are functions of the rotation magnitude and axis of rotation.

$$q_{N0}^B = \cos\left(\frac{\phi_{NB}}{2}\right)\cos\left(\frac{\theta_{NB}}{2}\right) \tag{1}$$

$$q_{N1}^B = \sin\left(\frac{\phi_{NB}}{2}\right)\cos\left(\frac{\theta_{NB}}{2}\right) \tag{2}$$

$$q_{N2}^B = \cos\left(\frac{\phi_{NB}}{2}\right)\sin\left(\frac{\theta_{NB}}{2}\right) \tag{3}$$

$$q_{N3}^B = -\sin\left(\frac{\phi_{NB}}{2}\right)\sin\left(\frac{\theta_{NB}}{2}\right) \tag{4}$$

The rotation vector can now be computed from the quaternion components using

$$\boldsymbol{\rho}_{NB} = \frac{2\arccos(q_{B0}^N)}{\sqrt{1 - {q_{B0}^N}^2}}\boldsymbol{q}_{N1:3}^B \tag{5}$$

where $\boldsymbol{\rho}_{NB}$ is the rotation vector and $\boldsymbol{q}_{N1:3}^B$ is a 3x1 vector formed by components one through three of the quaternion. The total slope in radians is then the magnitude of the rotation vector,

$$\gamma = |\boldsymbol{\rho}_{NB}|. \tag{6}$$

While conversions directly from Euler angles to rotation vectors exist, they are complicated and prone to programming errors, so the intermediate quaternion conversion is used. Calculating the total slope for the trajectory data gathered using the test robot results in Fig. 3. The test terrain has three major slope features that cross the terrain perpendicular to the data collection paths. As will be discussed in the localization section, the flat areas between terrain features present challenges for localization.

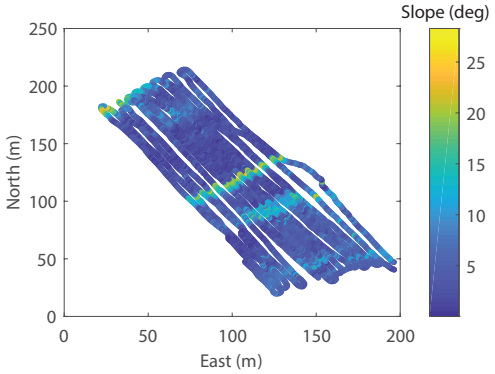

Figure 3: Total terrain slope measured during RMP400 data collection run.

3. TERRAIN MODELING

Gaussian processes [8] (GPs) are a non-parametric learning technique that have been shown to produce accurate continuous domain representations of terrain data [5]. These representations are well suited for use with particle filter estimators. At each particle location, the GP terrain map may be interrogated to provide terrain feature information. The GP representation of the terrain also incorporates uncertainty in a statistically sound way based on the data used to train the GP model. Uncertainty of the GP representation will be higher in areas that lack data points in the training set. This uncertainty is key in properly weighting particle estimates, and may also be used to plan traversal paths along areas with low uncertainty. In this work, the GP model will be trained to represent the maximum inclination of the terrain instead of the terrain elevation itself. Vehicle inclination, which is highly correlated with terrain inclination for slow-moving vehicles, may be estimated using MEMs accelerometers and gyroscopes. The following subsections will provide an overview of GP regression and then a discussion of selecting a covariance function and setting its characteristic parameters.

3.1 Gaussian Process Regression

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution, and is completely specified by a mean function and a covariance function. A process $f(x)$ is defined by the mean $m(x)$ and covariance $k(x, x')$ as

$$m(x) = \mathbb{E}[f(x)], \tag{7}$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))], \tag{8}$$

and can be written as

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{9}$$

The random variables represent the value of the function $f(x)$ at a location x. In this work, x is the location of the robot in the operational area and the function is the total terrain slope. The training data used to condition the GP joint distribution consists of the set of locations, $\boldsymbol{X}$, and the total slope measured at each location, $\boldsymbol{f}$. The joint distribution of the observation data and the desired test outputs, $\boldsymbol{f_*}$, is

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f_*} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X_*}) \\ \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X_*}) \end{bmatrix}\right) \tag{10}$$

where $\boldsymbol{K}$ is a covariance matrix to be defined and $\boldsymbol{X_*}$ are the test locations. The measurements of slope from the Xsens attitude reference system will include noise such that

$$y = f(x) + \epsilon \tag{11}$$

where y is the measured slope, $f(x)$ is the true slope, and ϵ is Gaussian noise independent with each sample and with variance σ_n^2. The joint distribution is modified to account for the noisy function values at the test locations, and the result is

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f_*} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X_*}) \\ \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X_*}) \end{bmatrix}\right) \tag{12}$$

where $\boldsymbol{I}$ is an identity matrix of the appropriate size. From [8] the predictive equations for GP regression, derived from the conditional distribution, are

$$\boldsymbol{f_*} | \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{X_*} \sim \mathcal{N}(\bar{\boldsymbol{f}}_*, \mathrm{cov}(\boldsymbol{f_*})) \tag{13}$$

where

$$\bar{\boldsymbol{f}}_* = \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}]^{-1} \boldsymbol{y}, \tag{14}$$

$$\mathrm{cov}(\boldsymbol{f_*}) = \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X_*}) - \boldsymbol{K}(\boldsymbol{X_*}, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}]^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X_*}). \tag{15}$$

If there is a single test point x_*, the GP regression can be written as

$$\bar{f}_* = \boldsymbol{k_*}^\top [\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}]^{-1} \boldsymbol{y}, \tag{16}$$

$$\mathbb{V}[f_*] = k(x_*, x_*) - \boldsymbol{k_*}^\top (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{k_*} \tag{17}$$

where the substitutions $\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ and $\boldsymbol{k_*} = \boldsymbol{K}(\boldsymbol{X}, x_*)$ have been made to simplify the notation. To completely define the regression, the covariance function $\boldsymbol{K}$ must be selected. This is the topic of the next subsection.

3.2. Covariance Function Selection

The selection of the covariance function and associated model parameters is an important step when creating a Gaussian process regression or classification system. Even for a single covariance function, there is a large variety of model parameters that may be adjusted. In addition, covariance functions of the same or different types may be combined to improve the model fit. With such a large number of free variables, a systematic approach to model selection is beneficial. In this work, the Bayesian approach of computing the probability of a model given the data is utilized to select hyperparameters for three different covariance functions [8]. This is not the only method of model section in the literature, and this is an important area of future work to improve terrain modeling.

The most widely used covariance function within machine learning is likely to be the squared exponential function, which is defined as

$$k_{SE}(r) = e^{\left(-\frac{r^2}{2l^2}\right)} \tag{18}$$

where r is a measure of the distance between points for which a covariance is being found, and l is a parameter called the characteristic length-scale. The characteristic length-scale is tuned based on knowledge of the underlying data the GP is modeling. In this work, the characteristic length-scale represents how quickly the terrain slope is changing over spatial distances. The adjustable parameters within covariance functions are commonly referred to as hyperparameters. An example of a function with a two dimensional input drawn from a squared exponential Gaussian process with a characteristic length of 0.5 is shown in Fig. 4 and with a characteristic length of 1.5 in Fig. 5. As expected, the smaller characteristic length produces a function with significantly more variation than the larger characteristic length.

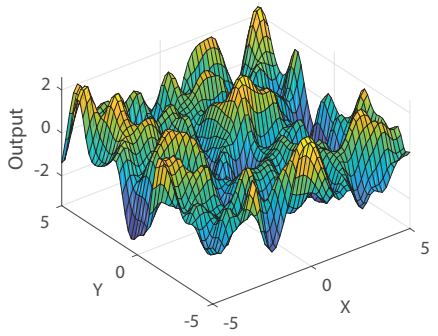

Figure 4: Function with two dimensional input drawn from squared exponential Gaussian process with characteristic length of 0.5.

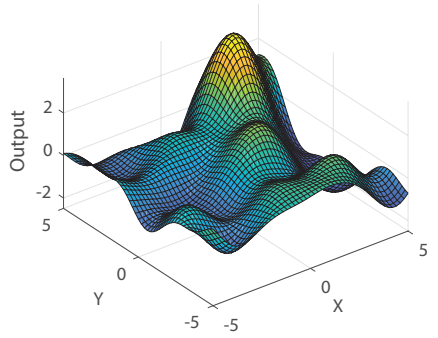

Figure 5: Function with two dimensional input drawn from squared exponential Gaussian process with characteristic length of 1.5.

The position of the robot within the operational area is expressed with a two-dimensional vector of East-North position in a local frame, (E, N), which results in a distance calculation between two points i and j of

$$r_{ij} = \sqrt{(E_i - E_j)^2 + (N_i - N_j)^2}. \tag{19}$$

In addition to the squared exponential covariance function, two additional covariance functions were evaluated for use in this work. The first is one of

the Matérn class of functions. Setting one of the hyperparameters within this class of functions to $\nu = 3/2$ yields

$$k_M(r) = \left(1 + \frac{\sqrt{3}r}{l}\right) e^{\left(-\frac{\sqrt{3}r}{l}\right)}. \tag{20}$$

The second additional function evaluated is the rational quadratic covariance function, defined as

$$k_{RQ}(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^2. \tag{21}$$

The squared exponential and Matérn functions have one hyperparameter, l, and the rational quadratic function has two, l and α. In actuality, additional hyperparameters can be included in these covariance function definitions, but they have been removed in this preliminary work to simplify the task of hyperparameter selection. The hyperparameter values are selected by adjusting them to maximize the log marginal likelihood function as discussed in the next subsection.

3.3. Selecting Hyperparameters

The hyperparameters were selected by following the process outlined in [8], which requires maximizing the log marginal likelihood (LML) function. The LML is defined to be

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} - \frac{1}{2}\log|\boldsymbol{K}| - \frac{n}{2}\log 2\pi \tag{22}$$

where $\boldsymbol{\theta}$ is the set of hyperparameters. Within the LML, the first term $-\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y}$ represents the quality of fit. Small length scales are able to fit the training data very closely at the expense of large variations and uncertainty in fit between training points. The second term, $-\frac{1}{2}\log|\boldsymbol{K}|$, represents the complexity. Small length scales are more complex and cause this term to be smaller, therefore adding less to the LML. The final term is a normalization constant. Maximizing the LML balances the quality of fit to the training data with the complexity of the model.

The first step in selecting the hyperparameters is generating the input training data set. Fig. 3 shows the slope data generated by the robot sensors at a high rate of 100 Hz. This data was sampled such that slope measurements approximately four meters from one another were used as the training set. Fig. 6 shows the position of each slope data point used in training the GP models.

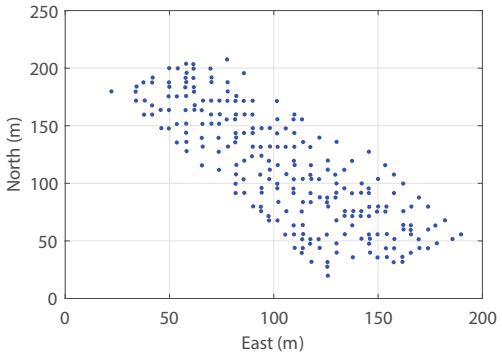

Figure 6: Training point data collection positions within the operational area.

With the training set chosen, the simplex search method was used to maximize the LML of each candidate covariance function by adjusting the hyperparameter(s). The results are summarized in Table 1 and show that the Matérn and rational quadratic functions produce LML values much higher than the squared exponential. The Matérn covariance function was chosen for modeling the terrain for localization because it produced the highest overall LML.

Table 1: Results of selecting hyperparameters by maximizing the log marginal likelihood.

Covariance Function	Hyperparameters	LML
Squared Exponential	l=11	98
Matern	l=27	140
Rational Quadratic	l=16 α=0.02	136

Fig. 7 shows the result of modeling the terrain slope data using the Matérn covariance function with a characteristic length of 27 meters. The surface in the plot is created by using a 4x4 meter grid of test points, and the color of the surface is the standard deviation of the model fit at that position. The results show that the standard deviation is low within the region that was mapped by the robot and increases in regions away from the training points. This is a significant benefit of modeling the terrain with a GP. The standard deviation of the fit is automatically adjusted based on the distance from the nearest training data. This provides proper weighting of the terrain data for localization. With the GP model of the terrain slope complete, the next phase is using the model for localization of the robot.

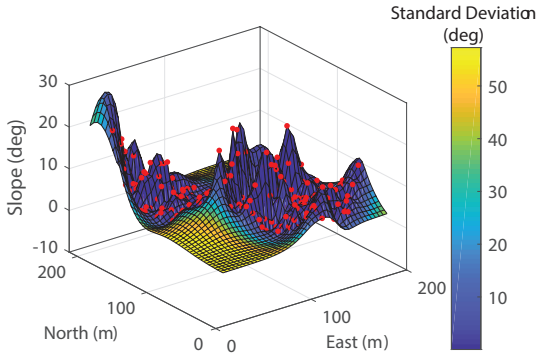

Figure 7: Mean of GP function modeling terrain with Matérn covariance function. Red markers indicate training data position and slopes. The surface is created by using a 4x4 meter grid of test points covering operational area. The color of the surface represents the standard deviation of the model fit at that position.

4. LOCALIZATION

The localization algorithm used in this research is a particle filter [9]. Particle filters are fairly standard within the robotics community, so the discussion in this paper will focus on features of the filter developed specifically for this research: the skid-steer motion model and GP measurement update. The localization algorithm begins with initialization where the particle filter is initialized with 750 particles spread evenly throughout the operational area. The next phase of the algorithm proceeds to loop through the stages of motion update, measurement and particle weighting, and particle resampling.

4.1. Motion Update

The robot used for localization experimentation is the same wheeled skid-steer robot used for map data collection. The instantaneous center of rotation (ICR) kinematic model of skid-steer motion is used to calculate vehicle velocity based on measurements of left and right side wheel speeds [10]. The velocity of a skid-steer vehicle is described by

$$v_x = \frac{V_x^r y_{ICRl} - V_x^l y_{ICRr}}{y_{ICRl} - y_{ICRr}} \tag{23}$$

$$v_y = \frac{(V_x^l - V_x^r) x_{ICRv}}{y_{ICRl} - y_{ICRr}} \tag{24}$$

$$w_z = -\frac{V_x^l - V_x^r}{y_{ICRl} - y_{ICRr}} \tag{25}$$

where v_x is the longitudinal velocity, v_y is the lateral velocity, w_z is the angular velocity, V_x^l is the left track velocity, V_x^r is the right track velocity, y_{ICRr} is the right ICR lateral position, y_{ICRl} is the left ICR lateral position, and x_{ICRv} is the longitudinal position of both the left and right ICRs. If the ICR locations are known, the velocity of the vehicle can be calculated from the track speeds. Prior work with this robot has shown that the ICR positions are $y_{ICRr} = 0.6$ meters, $y_{ICRl} = 0.6$ meters, and $x_{ICRv} = 0.0$ meters when operating on grassy terrains.

The vehicle velocity and angular rate are inputs to the velocity motion model provided in [9]. This motion model predicts the particle state forward in time and includes randomness in the velocity magnitude and direction to provide particle diversity. Wheel velocity updates are used to run the motion update at 100 hertz.

4.2. Measurement Update and Resampling

The measurement update phase is triggered every four seconds while the robot is moving. The Xsens inertial sensor on the robot provides an attitude estimate that is converted to total slope, y_k. Next, the set of robot locations stored in the particle set is used as the test input array $\boldsymbol{X}_*$ and (14) and (15) are used to find the GP model mean and covariance for each input. Note that the inverse term in (14) and (15) depends only on training data, and in this work it is precomputed and stored as part of the GP terrain model. This reduces the computational requirements to run the filter significantly. Given the slope measurement y_k, the GP model mean for the i^{th} particle, $\bar{f}_{*i}$, and the GP model variance for the i^{th} particle, σ^2_{*i}, the weight of particle i is calculated using

$$w_i = e^{-\frac{1}{2}\frac{(y_k-\bar{f}_{*i})^2}{\sigma^2_{*i}}}. \tag{26}$$

The particle weights are then normalized with the sum of all the particle weights. The low variance resampling method is used to resample the weighted particle set [9]. The estimated position of the vehicle is the mean of the particle positions. After the update and resampling have finished, the motion update function is called until it is time for a new measurement update.

4.3. Localization Results

While the time evolution of a particle filter is best viewed in a video, a series of plots at key times in the localization test are shown here to provide insight into the behavior of the filter. Fig. 8 shows the particle positions overlaid on a contour plot of the terrain slope four seconds into the localization test. The true position of the robot at this step is shown by a black star, and the estimated position from the filter is shown by the green star. At this time in the test, only a single measurement update has been input to the filter, so the particles are mostly spread uniformly across the operational area. It can be seen that particles have been removed from areas of large slope because the measurement of slope at the current position is small.

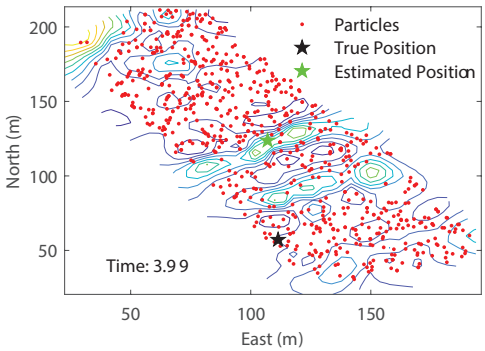

Figure 8: Particle set 4 seconds into the localization run. Only a single measurement has been used to update the filter.

Fig. 9 shows the particle positions 64 seconds into the localization test run. At this point, the robot has traveled over enough unique features to cluster the particle estimates, but the overall particle position average is still far from the true location.

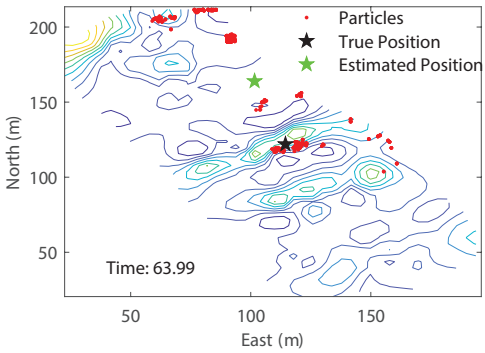

Figure 9: Particle set 64 seconds into the localization run.

Fig. 10 shows the particle positions 100 seconds into the localization run and after the robot has traversed the large slope feature that runs diagonally across the operational area. At this point the particle set is clustered around the true position. It can be seen in the particle distribution that there is more uncertainty in the position estimate parallel

to the slope feature. Because the feature runs across the operational area, the particle filter cannot discriminate between positions on a line parallel to the slope feature.

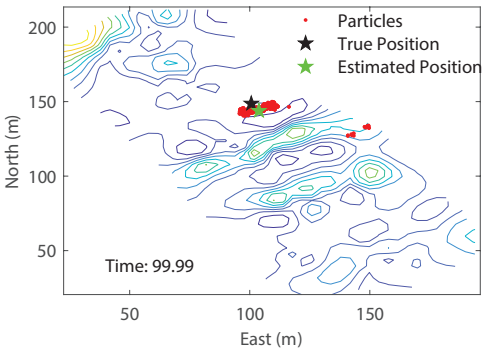

Figure 10: Particle set 100 seconds into the localization run.

Fig. 11 provides a plot of the error between the particle filter position estimate and the true vehicle position versus time during a localization run. As expected for this type of localization algorithm, the accuracy of the estimate depends almost entirely on the variety of features in the environment. Near the beginning of the run, the robot is driving over areas of with significant terrain slope change, resulting in low position estimate error. In the latter half of the test run, the robot is driving primarily on flat areas of the map and the error is much larger. The filter algorithm tracks the average particle weight from recent updates and compares it with the long-term average particle weights. When the average recent particle weight is much lower than the long-term average weight, none of the particles are at positions that match current measurements and the particle set is reinitialized uniformly across the operational area. An instance of this occurring is labeled in Fig. 11. Overall, the results show that the GP localization filter provides an accurate estimate of position when there is sufficient terrain variability.

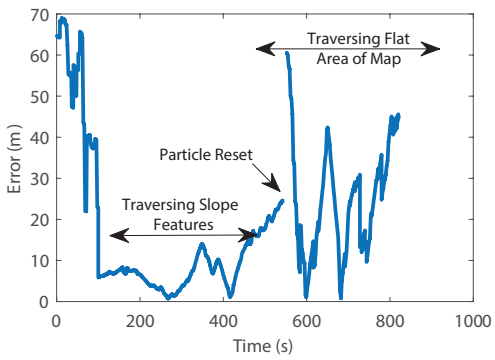

Figure 11: Error between particle filter position estimate and true vehicle position during 800 second localization run.

5. CONCLUSIONS

This paper has presented an algorithm for modeling terrain slope variations using Gaussian processes. When coupled with a particle filter and an accurate model of robot motion, the GP model of terrain slope can be used in the measurement update function to provide accurate position estimates in GPS denied environments. The performance of the particle filter over time is very dependent on the variability of the terrain traversed by the robot. During testing, the particle filter algorithm provided sub-10 meter accuracy when the robot traversed major slope features in the operational area. Ultimately, this localization approach should be coupled with other sources of information that complement one another. For example, a visual odometry algorithm could be used to estimate motion over time, and when the robot traverses terrain features, this localization system could update that visual odometry estimate to remove drift.

This research provides the groundwork for many areas of future exploration. The relationship between the density of points in the training data and the accuracy of the localization estimate should be explored. If lower density training data can be used, this reduces the memory required to store and process the GP terrain model. In addition, the

candidate covariance functions evaluated in this work can be expanded with additional hyperparameters to better model terrain data. Covariance functions can also be combined, the sum and product being examples, to improve model fit. An example of this could be two squared exponential functions with different characteristic lengths. This model selection would perform better over terrain with short, bumpy variations on top of large scale changes in elevation. Overall, this preliminary work has shown that GP models of terrain slope are a viable method of implementing the measurement update phase of a localization particle filter, but additional work is needed to fully develop the terrain model and compare localization performance with other feature-based algorithms.

6. ACKNOWLEDGEMENT

This work was supported by the Internal Science and Technology Program, administered by the Applied Research Laboratory, The Pennsylvania State University. No official endorsement should be inferred.

7. REFERENCES

[1] A. J. Dean, J. W. Langelaan, and S. N. Brennan, "Improvements in terrain-based road vehicle localization by initializing an unscented kalman filter using particle filters," in *Proceedings of the American Control Conference*, 2010.

[2] X. Zhu, C. Qiu, and M. A. Minor, "Terrain-inclination-based three-dimensional localization for mobile robots in outdoor environments," *Journal of Field Robotics*, vol. 31, no. 3, pp. 477–492, 2014.

[3] B. F. D. Hähnel and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceedings of Robotics: Science and Systems*, 2006.

[4] Y. Jongwoon, T. Kim, C. Provencher, and T. Fong, "WiFi localization on the international space station," in *Proceedings of IEEE Symposium on Intelligent Embedded Systems*, 2014.

[5] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair, "Gaussian process modeling of large scale terrain," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.

[6] Open Source Robotics Foundation, *ROS: Robot Operating System*, https://www.ros.org/.

[7] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.

[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[10] J. Pentzer, S. Brennan, and K. Reichard, "Model-based prediction of skid-steer robot kinematics using on-line estimation of track instantaneous centers of rotation," *Journal of Field Robotics*, vol. 32, no. 3, pp. 455–476, 2014.