Bayesian Optimisation for Active Perception and Smooth Navigation

Jefferson R. Souza, Roman Marchant, Lionel Ott, Denis F. Wolf and Fabio Ramos

Abstract— **A key challenge for long-term autonomy is to enable a robot to automatically model properties of the environment while actively searching for better decisions to accomplish its task. This amounts to the problem of exploration-exploitation in the context of active perception. This paper addresses active perception and presents a technique to incrementally model the roughness of the terrain a robot navigates on while actively searching for waypoints that reduce the overall vibration experienced during travel. The approach employs Gaussian processes in conjunction with Bayesian optimisation for decision making. The algorithms are executed in real-time on the robot while it explores the environment. We present experiments with an outdoor vehicle navigating over several types of terrains demonstrating the properties and effectiveness of the approach.**

I. INTRODUCTION

Field robotics has seen significant progress over the last decade. Today, autonomous mobile robots are successfully employed in outdoor applications, such as mines, ports, agriculture, among many others. In these situations, a robot is expected to operate autonomously for long periods of time while navigating through different types of terrains. Although much has been done in designing autonomous robots that operate on specific conditions, actively learning the characteristics of the environment without human intervention can significantly improve robustness and expand the applicability of field robotics to many different areas.

In this paper we develop an active perception approach to learn the roughness of the terrain a robot traverses on. The harmful effect of vibration is an important problem for terrain robots [1], [2]. Excessive shaking can cause hardware damage, and it is well known to shorten the lifespan of an unmanned ground vehicle. Furthermore, vibration affects the quality of the information provided by the sensors. Undesirable noise is introduced to images, laser scan measurements and inertial measurement unit data, ultimately affecting the reliability in accomplishing the goals due to poor accuracy in localisation, mapping, and navigation.

Our goals are to achieve a good understanding of the environment, while keeping the robot as safe as possible. We propose an active learning approach to terrain roughness estimation and develop a planning algorithm that automatically trades off exploration and exploitation. The proposed methodology consists of three steps. First, acceleration data acquired from an *inertial measurement unit* (IMU) is preprocessed by a filtering stage. Secondly, the filtered data is used to continuously learn and update a *Gaussian Process* (GP) [3] model of the vibrations. Finally, new sampling locations are selected using a *Bayesian Optimisation* (BO) algorithm that trades off automatically between exploration and exploitation.

The main contributions of this work are:

1) A terrain roughness estimation model from acceleration data built using GP regression;
2) A planning algorithm based on BO that exploits and improves the terrain roughness model.

The remainder of this paper is organised as follows. Section II reviews the state of the art in terrain roughness estimation, spatial modelling with GPs and informative path planning algorithms. Section III presents the proposed methodology, detailing the principles behind GP regression applied to vibration estimation, providing a brief overview of the BO algorithm and describing the active learning approach based on BO for informative planning. The experimental setup, results and analysis are shown in section IV. Finally, section V provides conclusions and suggests directions for future works.

II. RELATED WORK

A. Terrain Roughness Estimation

In [4], explored areas of the terrain are classified based on vibration. Features from the data are obtained using combinations of Power Spectral Density and Fast Fourier Transform, which are used to classify terrain using a support vector machine. In [5], a vision and vibration-based approach fuses terrain predictions from image data and vibration. The robot uses images to classify the terrain, and vibration data to verify its prediction. Classification is performed using SVMs and results show 87% of correct classification. The main drawback of this approach is that it only classifies different terrains in a discrete manner, whereas in this work we are able to predict the amount of vibration expected in a continuous domain, with an associated level of uncertainty.

B. GPs in Spatial Modelling

GP regression [3], initially developed in geostatistics and known as Kriging, has been used widely for modelling spatially correlated data. GPs have been used for various robotic applications, such as the estimation of a wireless signal strength in an indoor environment [6], modelling gas distribution across space [7], [8], building a solar map [9],

Jefferson R. Souza and Denis F. Wolf are with the Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, Brazil {jrsouza; denis}@icmc.usp.br

Roman Marchant, Lionel Ott and Fabio Ramos are with the School of Information Technologies, The University of Sydney, NSW 2006, Australia {r.marchant; l.ott}@acfr.usyd.edu.au fabio.ramos@sydney.edu.au

learning occupancy maps of an outdoor environment with integral kernels [10] and modelling large-scale terrain using point-cloud information [11]. To the best of our knowledge, GPs have never been used to estimate the roughness of terrain from acceleration data.

C. Informative Path Planning

In [12], a greedy algorithm for sequential path planning was developed by exploiting the sub modularity of *Mutual Information* (MI). [13] selected sensing locations for temperature measurement using entropy reduction and MI. These approaches do not consider the expected value of the studied phenomenon, and are active in the sense of uncertainty reduction. Recently, [7] developed a BO algorithm [14] for choosing informative sensing locations that take the expected value of the phenomenon into account. A specific acquisition function that includes a penalty on the distance travelled to acquire consecutive observations was developed and applied to real robotics problems.

III. Methodology

Our main goal is to reduce the amount of vibration experienced by the robot while navigating on different terrains. To achieve this, we propose a method to learn a representation of the terrain roughness, ultimately favouring movement through traversable areas that generate lower vibration levels.

The method, shown in Fig. 1 consists of four steps:

- *Preprocessing*: Filters are applied to noisy measurements provided by an IMU to smooth the data as described in section III-A.
- *Vibration modelling*: GP regression is used to create a statistical model of vibration when traversing a terrain and to predict the vibration levels across space in unsampled areas. Details are presented in section III-B.
- *Planning*: Bayesian optimisation is employed to select locations to be visited, depending on the expected level of vibration and the uncertainty in the prediction. This is described in section III-D.
- *Localisation and Mapping*: Localises the robot in an unknown environment and creates a map of the area. Localisation is used by the vibration modelling step to assign vibration measurements to different parts of the terrain. We used a SLAM package built in ROS[1] named Gmapping[2].

A. Preprocessing

To prevent an excessive amount of noise being transferred to the vibration modelling module, we propose a preprocessing, filtering stage. Given the original signal $w(t)$, we evaluated the performance of two different strategies for filtering unwanted noise, producing a filtered signal $w_f(t)$.

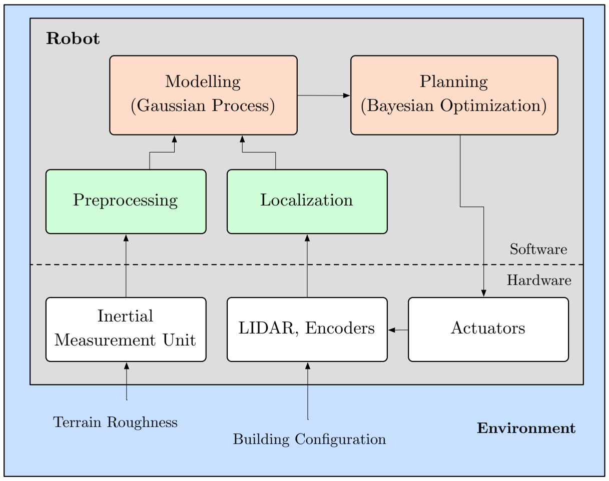


Fig. 1. Block diagram of the proposed system. Data flow is represented by arrows.

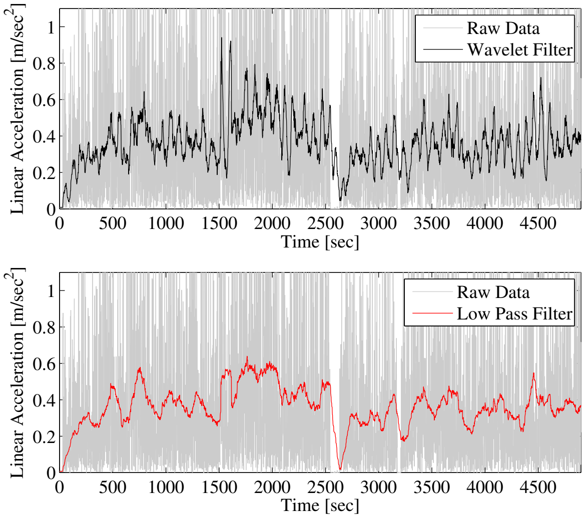


Fig. 2. Application of two digital filters on raw IMU data.

a *Wavelet Filter*: Discrete wavelet transform (DWT) [15] provides a continuous multi-resolution version of the original signal. By retaining only the fourth level of the Daubechies DWT transform coefficients, we obtain a filtered approximation of the original signal in which high frequency noise is no longer present.

b *Low Pass Filter*: A simple non-recursive digital low pass filter [16] is implemented to smooth the original signal. The output signal is given by the following expression:

$$w_f(t) = \sum_{j=0}^{N} \frac{w(t-j)}{N} \ . \tag{1}$$

Depending on the bandwidth (directly related to N) of the low pass filter, a lag or delay is observed in the output signal. By setting $N = 100$ we fix a filtering bandwidth that removes noise and keeps an upper bound on the lag.

Fig. 2 shows both filters applied to real data acquired with an accelerometer. From this plot alone, it is difficult to decide which is more effective. We return to this discussion in the experimental section and evaluate the performance of the filters in the context of the whole system to select the filtering strategy that delivers the best results.

[1] Robot Operating System http://wiki.ros.org/

[2] Gmapping http://wiki.ros.org/gmapping

B. GP for Vibration Model

GP regression is an effective technique for modelling complex spatial processes, such as the terrain roughness in an outdoor environment. By using a set of training points, a GP can provide a prediction of the unknown function with an associated uncertainty over a continuous domain.

Formally, a GP is completely defined by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$. It learns the representation of an unknown function f in a supervised learning set up using a set S of known observations, $S = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, with $\mathbf{x}_i \in \mathbb{R}^D$ being an input in a D dimensional space and $y_i \in \mathbb{R}$ the correspondent noisy output. Given a test point $\mathbf{x}^*$ (unsampled location), the GP returns a prediction of the value of $f(\mathbf{x}^*)$ with the corresponding uncertainty. Noisy observations of the underlying function $f(\mathbf{x})$ are modelled as Gaussian distributed, $y = f(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}\left(0, \sigma_n^2\right)$.

Given a covariance function $k(\mathbf{x}, \mathbf{x}')$, the covariance matrix $K(X, X)$ can be computed by evaluating the covariance function for all observations. The predictive distribution for test locations X^* is given by

$$\mathbf{y}^* | X, \mathbf{y}, X^* \sim \mathcal{N}(\boldsymbol{\mu}^*, \Sigma^*) \,,$$

where,

$$\begin{aligned} \boldsymbol{\mu}^* &= K(X^*, X) K_X^{-1} \mathbf{y}, && (2) \\ \Sigma^* &= K(X^*, X^*) - K(X^*, X) K_X^{-1} K(X, X^*), && (3) \end{aligned}$$

with $K_X = K(X, X) + \sigma_n^2 I$.

By selecting different covariance functions, our GP vibration model can adapt more easily to complex vibration models. The particular covariance functions considered in this paper are Linear, Exponential, Squared Exponential, Matérn 3 and Matérn 5, their respective equations are given by:

$$\begin{aligned} k_{\text{LI}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \mathbf{x}\mathbf{x}', && (4) \\ k_{\text{EXP}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \exp\left(-\sqrt{r}\right), && (5) \\ k_{\text{SQEXP}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \exp\left(-\frac{r}{2}\right), && (6) \\ k_{\text{MAT3}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \left(1 + \sqrt{3r}\right) \exp\left(-\sqrt{3r}\right), && (7) \\ k_{\text{MAT5}}(\mathbf{x}, \mathbf{x}') &= \sigma_f^2 \left(1 + \sqrt{5r} + \frac{5r}{3}\right) \exp\left(-\sqrt{5r}\right) && (8) \end{aligned}$$

where $r = (\mathbf{x} - \mathbf{x}')\mathrm{L}(\mathbf{x} - \mathbf{x}')^T$, L is a diagonal matrix of size D, whose elements are $\mathrm{L}_{ii} = 1/l_i^2$, a length-scale parameter associated to each dimension of the input space. The matrix L and the factor σ_f, or signal variance, are hyper-parameters of these covariance functions. Therefore, the set of hyper-parameters that fully determine the GP is given by:

$$\boldsymbol{\theta} = \{\sigma_f, l_1, l_2, \ldots, l_D, \sigma_n\} \tag{9}$$

By tuning these hyper-parameters, the behaviour of the GP regression is adapted to achieve a good representation of the unknown function. The optimal set of hyper-parameters $\boldsymbol{\theta}^*$ can be found by maximising the *log marginal likelihood* (LML),

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\arg\max} \;\; \mathrm{LML}(\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \,, \tag{10}$$

with,

$$\mathrm{LML}(\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T K_X^{-1} \mathbf{y} - \frac{1}{2}\log|K_X| - \frac{n}{2}\log 2\pi. \tag{11}$$

Algorithm 1 Bayesian Optimisation

```
$\mathbf{x}_i$: chosen sampling point at iteration $i$.
$s$: acquisition function.
$f$: unknown function. (terrain roughness)
for $i = 1, 2, 3, \ldots$ do
    Find $\mathbf{x}_i = \arg\max_{\mathbf{x}} s(\mathbf{x})$
    Acquire a sample from $f$ at location $\mathbf{x}_i$.
    Update the GP model of $f$ with the new sample.
end for
```

We also combine Covariance Functions (CFs) to obtain new functions. The sum and product of two CFs are valid CFs [17][18]:

$$k_{\text{SUM}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{M} k_i(\mathbf{x}, \mathbf{x}') \tag{12}$$

$$k_{\text{PROD}}(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{M} k_i(\mathbf{x}, \mathbf{x}') \tag{13}$$

Depending on the number of covariance functions M, the dimensionality of the hyper-parameter vector can grow quickly, making the optimisation on this high dimensional space more costly.

C. Bayesian Optimisation

The Bayesian Optimisation algorithm (Algorithm 1) [14], was designed to find the extreme of an unknown, noisy, and costly to evaluate function f, i.e. to find the maximum

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\arg\max} \, f(\mathbf{x}) \tag{14}$$

or minimum

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\arg\min} \, f(\mathbf{x}) = \underset{\mathbf{x}}{\arg\max} \left(-f(\mathbf{x})\right) \;. \tag{15}$$

BO is completely defined by two elements, the prior and the acquisition function. We use a GP prior as the statistical model of the unknown function. This model uses acquired samples $\mathbf{x}_i$ to build a representation of the unknown function and perform regression over the un-sampled space. Equations 2 and 3 detail the predicted mean μ and variance Σ respectively. The second fundamental element in BO is the acquisition function, which encodes the utility of sampling at location $\mathbf{x}$. The combination of these two elements (GP prior and acquisition function) can provide a smart trade off between exploration and exploitation.

At each iteration, a new sampling location is selected by reasoning over previously acquired information. By maximising the acquisition function s, we select the best location to sample the unknown function of the terrain roughness.

D. BO for Informative Planning

Choosing an appropriate model to represent the studied phenomenon is necessary but not sufficient to achieve good results. The location of the observations plays a fundamental role in the quality of the resulting model and its prediction accuracy. The most popular approach to choose sensing locations is by maximising the mutual information between the sampled and un-sampled domain [19]. However, [7] has shown that BO can be used as a suitable alternative strategy that not only considers the uncertainty in the model, but also uses the predicted value to choose sampling locations, automatically trading off between exploration and exploitation. We use this strategy to learn the roughness of the terrain, while at the same time using this roughness to select waypoints for navigating in areas with less vibration.

We use an active sampling procedure based on BO (see section III-C). By maximising the acquisition function s, we select the best location to sample the unknown function. For terrain roughness estimation we use *Distance Upper Confidence Bound* ($DUCB$) as the acquisition function s (Algorithm 1), given by

$$DUCB(\mathbf{x}|\mathbf{x}^-) = \mu(\mathbf{x}) + \kappa \cdot \sigma(\mathbf{x}) + \gamma \cdot d(\mathbf{x}, \mathbf{x}^-), \quad (16)$$

where $d(\mathbf{x}, \mathbf{x}^-)$ is the Euclidian distance between the last sampled location $\mathbf{x}^-$ and the candidate location $\mathbf{x}$, and $\gamma < 0$ ensures that goal locations are close to each other, with the purpose of reducing the cost of traversing the environment.

The predicted values $\mu(\mathbf{x})$ given by the GP model are large for higher vibration levels, and by minimising f (Equation 15), we reduce the total vibration experienced by the robot while traversing the environment. The term $\sigma(\mathbf{x})$ represents the variance of the GP model. When maximising Equation 16, note that the first term favours areas of lower vibration and the second term favours areas with higher uncertainty. In brief, the robot will tend to favour routes with lower vibration while visiting areas that reduce the uncertainty in its predictions.

Initially, the uncertainty will be high over the entire domain, i.e. the second term of Equation 16 will prevail, favouring the exploration of unknown terrain. As the robot senses unknown locations the uncertainty reduces. After some time, the predicted values of the phenomenon (first term of Equation 16) become more important to plan sensing locations, i.e. favouring exploitation. The smoothness of the transition between exploration and exploitation depends on the smoothness of the covariance function and the parameter κ.

IV. Experimental Results

To evaluate the capabilities and performance of the proposed methodology, we implemented the described system and tested its performance on a wheeled robot, Husky, by Clearpath Robotics (Fig. 3). The vehicle is equipped with a laser scanner, an inertial measurement unit and odometry, and was deployed outdoors in an area with different terrain characteristics. We assumed that vibration is independent of the orientation of the vehicle during navigation, and speed was kept constant throughout the entire trajectory. These assumptions simplify the model significantly, as we can now learn a mapping function from a spacial location to vibration directly, at a small cost in accuracy for the testing area – the GP automatically increases the noise level σ_n^2 to accommodate these simplifications. Learning more complex models that map from location, speed and orientation to vibration is possible but require substantially longer exploration trajectories as the problem's input dimensionality increases.

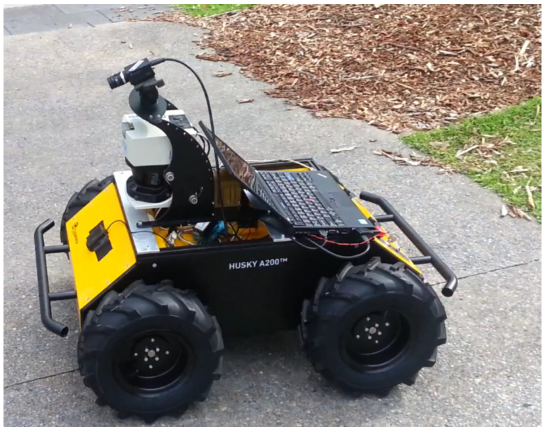


Fig. 3. Clearpath HUSKY robotic platform used in the experiments.

Our proposed method runs in real-time on a standard laptop and is implemented in C++. We use ROS to interface with the robot and our method, and to provide localisation.

A. Model

In the first experiment, our focus is to find a suitable model that can represent the environment properly. To this end we need to choose a filtering strategy for preprocessing the data and a covariance function that yields good prediction results. We compare the predictive performance of our model by combining a low pass filter or a wavelet filter with the covariance functions detailed in section III-B.

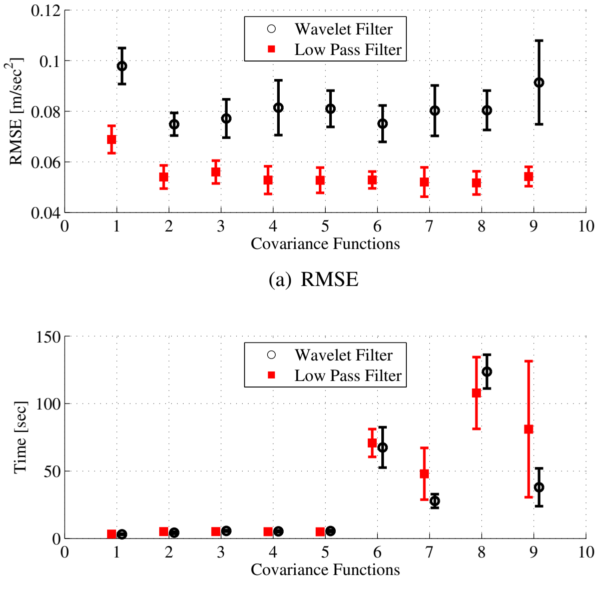
The robot was driven randomly over the environment shown in Fig. 5(a), only with the purpose of gathering data. The linear acceleration data for the vertical axis of the robot was collected at a frequency of 10Hz and each measurement was spatially referenced using the localisation of the robot. This dataset, consisting of 1000 measurements, was partitioned into 70% for training and 30% for testing purposes. For each combination of filters and covariance functions we ran the experiments 10 times, comparing their performance using two indicators with their respective mean and standard deviation: 1) the time required to learn the representation in seconds and 2) *Root Mean Squared Error* (RMSE) over the test set in m/s^2. These results are shown in Table I and Fig. 4.

Overall, the low pass filter has smaller error than the wavelet filter for every covariance function. There is clear evidence that sums or products of covariance functions take much longer to train than individual covariance functions,

TABLE I

RESULTS FOR PREPROCESSING FILTERS AND COVARIANCE FUNCTION. RMSE OBTAINED FROM 30% OF THE TEST DATA, AND TRAINING TIME FOR MAXIMISING LML WITH 700 OBSERVATIONS.

		Low Pass Filter		**Wavelet Filter**	
Index	**Covariance function**	**RMSE** $(10^{-2}m/s^2)$	**Time** (s)	**RMSE** $(10^{-2}m/s^2)$	**Time** (s)
1	Linear (LI)	6.8836 ± 0.5393	3.2836 ± 0.3272	9.7869 ± 0.7096	3.1382 ± 0.4614
2	Exponential (EP)	5.4013 ± 0.4583	5.1260 ± 0.7097	7.4870 ± 0.4479	4.3600 ± 0.7276
3	Squared EP (SE)	5.5994 ± 0.4510	5.1191 ± 0.5014	7.7127 ± 0.7563	5.6706 ± 0.5639
4	Matérn 3 (M3)	5.2796 ± 0.5485	5.0507 ± 0.5947	8.1409 ± 1.0812	5.3240 ± 0.6323
5	**Matérn 5 (M5)**	$\mathbf{5.2741 \pm 0.4984}$	$\mathbf{4.9580 \pm 0.6434}$	8.0989 ± 0.7176	5.5355 ± 0.6924
6	Sum (SE, M3, M5)	5.2848 ± 0.3319	70.806 ± 10.283	7.5104 ± 0.7193	67.500 ± 14.940
7	Prod (SE, M3, M5)	5.2042 ± 0.5799	47.945 ± 19.168	8.0247 ± 0.9952	27.813 ± 5.0636
8	Sum (LI, EP, SE, M3, M5)	5.1685 ± 0.4569	107.84 ± 26.586	8.0398 ± 0.7777	123.64 ± 12.516
9	Prod (LI, EP, SE, M3, M5)	5.4207 ± 0.3824	80.988 ± 50.401	9.1381 ± 1.6532	37.976 ± 14.008



(a) RMSE

(b) Time

Fig. 4. Results for preprocessing filters and Covariance Functions (CFs). RMSE obtained from 30% of the test data, and training time for maximising LML with 700 observations. The index of the CFs are shown in Table I.

which is an expected result given the increase in the dimensionality of the hyper-parameter search space. The Linear covariance function presents the fastest training time, however, the error is too large to be acceptable. Considering the rest of the relatively fast covariance functions, Matérn 5 is the one that produces the smallest error without compromising training time.

The best set of hyper-parameters for Matérn 5, found by maximising Equation 11, are:

$$\{\sigma_f, l_{posx}, l_{posy}, \sigma_n\} = \{0.104, 1.016, 0.990, 0.028\} \quad (17)$$

Fig. 5(b) shows the mean and Fig. 5(c) the variance of the learnt GP model using a low pass filter and Matérn 5 covariance function with optimal hyper parameters given by Equation 17. The mean of the vibration estimate shows a clear distinction between two explored terrains, grass and asphalt.

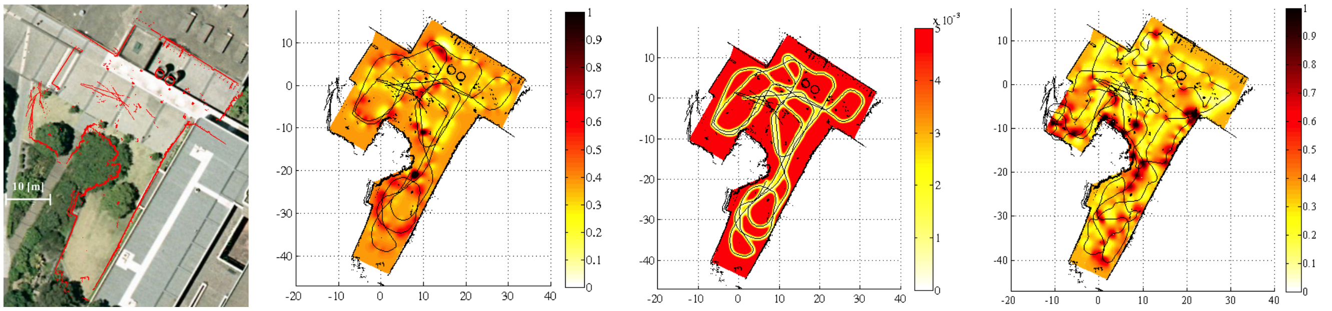
B. Path Planning

The second experiment consists of autonomously selecting sampling locations using the BO algorithm described in section III-D. The selected covariance function (Matérn 5) in section IV-A and the optimal hyper parameters were determined from Equation 17. We use a low pass filter to remove high-frequency noise from the IMU due to its simplicity and good results obtained in the previous section. The parameters of the acquisition function DUCB, $\{\kappa, \gamma\}$, are set to $\{1.25, 0.0125\}$. These were manually tuned resulting in the right amount of exploration (κ) while choosing consecutive sensing locations near each other (γ). However, a strategy such as [20] could be used for optimising these parameters automatically.

First, we check the explorative behaviour of our method. The robot is programmed to autonomously select sampling locations using BO over the large environment shown in Fig. 5(a), with approximately $1000m^2$. The trajectory followed by the robot (including waypoints) and the estimated vibration across the domain are shown in Fig. 5(d). It can be seen that the robot covers the whole area, and achieves this in about 17 minutes. Over time, the robot will explore areas of low vibration more often than areas with high vibration. As a result, low vibration areas will be modelled with higher accuracy. Note that this is not a problem, as the objective is not only to learn the best model of vibration over the entire region, but also to minimise excessive vibration during navigation. The predicted values for vibration agree with the expected results, considering that the locations with obstacles are identified as peaks in Fig. 5(d) and the asphalt area has a lower vibration level on average.

Finally, we compare our sampling method to an entropy based planner [21], where sampling locations are selected based only on entropy reduction. The comparison is conducted in two smaller scenarios where the robot is forced to stay in a restricted area (Fig. 6(a) and Fig. 6(f)). These scenes contain obstacles that produce sharp changes in vibration levels. Fig. 6 shows the path followed by the robot, the selected sampling locations, and the vibration prediction over the domain for each scenario.

In terms of the quality of the model produced by each



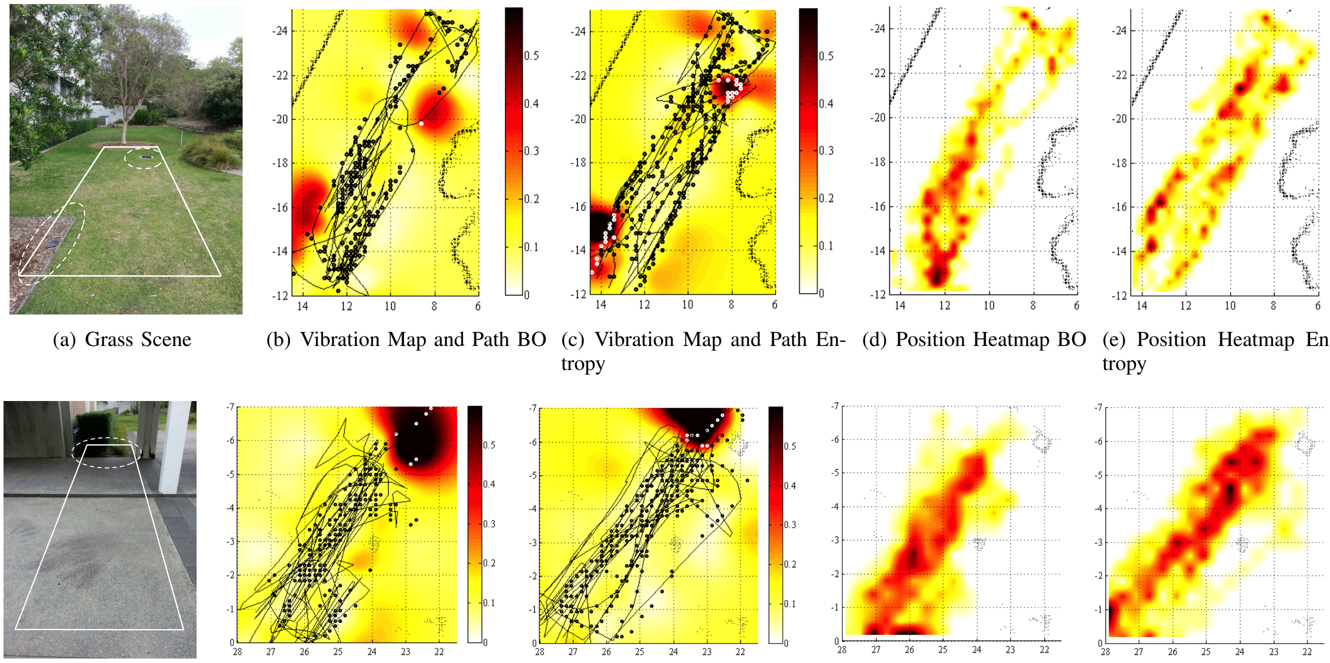
(a) Map matching

(b) Mean

(c) Variance

(d) Predicted mean after autonomous navigation

Fig. 5. (a) The scenario used for the first experiment bounded by the map generated by the robot. (b) Mean of the predicted vibration. (c) Variance for the prediction. (d) Predicted mean after autonomous navigation. Distance is in metres and prediction is in m/s^2.



(a) Grass Scene

(b) Vibration Map and Path BO

(c) Vibration Map and Path Entropy

(d) Position Heatmap BO

(e) Position Heatmap Entropy

(f) Asphalt Scene

(g) Asphalt BO

(h) Asphalt Entropy

(i) Position Heatmap BO

(j) Position Heatmap Entropy

Fig. 6. Comparison between BO and Entropy planning strategies. (a) and (f) show experiment area as a rectangle and obstacles that generate vibration are marked with circles. (b),(c),(g) and (h) show a surface for the prediction of vertical acceleration in m/s^2 and axis in metres. (d), (e), (i) and (j) show a heat map representing the time spent at each location by the robot for the duration of the experiment.

strategy, both show correctly the area(s) of high vibration. However, the most interesting and expected result is that the BO algorithm avoids areas of high vibration after identifying them during exploration whereas the entropy based planner chooses sampling locations only based on uncertainty, disregarding the amount of expected vibration. Position-based heat-maps are generated for the path followed by each robot. It can be seen that for BO the robot covers the whole area but concentrates in locations with lower vibration (Fig. 6(d) and Fig. 6(i)). In contrast, the entropy based sampling approach (Fig. 6(e) and Fig. 6(j)) is spread over the whole sampling domain homogeneously.

Table II shows a numeric comparison between the total vertical acceleration (vibration) experienced by the robot on each scenario and presents a quantitative indicator. The BO planner clearly results in less vibration over the trajectory, in terms of mean and standard deviation, but achieves an equally accurate model of the environment.

V. Conclusions

We propose a simple yet effective active learning method to learn the characteristics of the terrain in outdoor environments. Particularly, we model terrain roughness using a GP trained with acceleration measurements on the vertical axis of the robot. To do this efficiently and maintain the robot as safe as possible at the same time, we use a BO approach for choosing sampling locations, automatically trading off

TABLE II
AVERAGE ACCELERATION IN $[m/s^2]$ FOR BO AND ENTROPY PLANNERS.

Scenario	Method	Mean	Std
Grass	BO	0.096	0.068
Grass	Entropy	0.154	0.148
Asphalt	BO	0.103	0.122
Asphalt	Entropy	0.127	0.256

between exploration and exploitation. A representation of the roughness was learnt by selecting a covariance function and hyper parameters that best predict the vibration phenomenon across space. Additionally, we evaluated the BO method and compared it against an entropy reduction strategy. Here the robot learns a model of the terrain roughness, while at the same time reducing unnecessary exposure to excessive vibration.

We believe the contributions made in this paper are an important step towards long-term autonomy for outdoor robotics. The roughness of the terrain can now be quickly modelled while reducing the risk of potential damages to the robot in an efficient manner. The algorithms were implemented and executed in a real robot while navigating in an unstructured environment, demonstrating empirical evidence of their effectiveness for real-time applications.

As future work, we will consider more input dimensions beyond the vibration model, such as the speed, the travel direction, and features from a visual perception system. These can help building a more complete representation and understanding of the environment.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support of CNPq Foundation, through the Science without Borders program, Ministry of Science and Technology of Brazil, process 207255/2012-1.

REFERENCES

[1] P. Komma, C. Weiss, and A. Zell, “Adaptive Bayesian Filtering for Vibration-based Terrain Classification,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[2] C. Weiss, N. Fechner, M. Stark, and A. Zell, “Comparison of Different Approaches to Vibration-based Terrain Classification,” in *European Conference on Mobile Robots (ECMR)*, 2007.

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

[4] C. Weiss, H. Fröhlich, and A. Zell, “Vibration-based Terrain Classification Using Support Vector Machines,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[5] C. Weiss, H. Tamimi, and A. Zell, “A Combination of Vision and Vibration-based Terrain Classification,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[6] B. Ferris, D. Hahnel, and D. Fox, “Gaussian Processes for Signal Strength-Based Location Estimation,” in *Robotics Science and Systems (RSS)*, 2006.

[7] R. Marchant and F. Ramos, “Bayesian Optimisation for Intelligent Environmental Monitoring,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[8] C. Stachniss, C. Plagemann, A. Lilienthal, and W. Burgard, “Gas Distribution Modeling Using Sparse Gaussian Process Mixture Models,” in *Robotics Science and Systems (RSS)*, 2008.

[9] P. A. Plonski, P. Tokekar, and V. Isler, “Energy-Efficient Path Planning for Solar-Powered Mobile Robots,” in *International Symposium on Experimental Robotics (ISER)*, 2012.

[10] S. T. O'Callaghan and F. T. Ramos, “Gaussian Process Occupancy Maps,” *The International Journal of Robotics Research*, 2012.

[11] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, “Gaussian Process Modelling of Large-Scale Terrain,” *Journal of Field Robotics*, 2009.

[12] A. Krause and C. Guestrin, “Near-optimal Observation Selection using Submodular Functions,” in *Association for the Advancement of Artificial Intelligence Conference on Artificial Inteligence (AAAI)*, 2007.

[13] C. Guestrin, A. Krause, and A. P. Singh, “Near-Optimal Sensor Placements in Gaussian Processes,” in *International Conference on Machine Learning (ICML)*, 2005.

[14] E. Brochu, V. M. Cora, and N. de Freitas, “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning,” University of British Columbia, Tech. Rep. arXiv:1012.2599, 2010.

[15] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall, 1998.

[16] M. G. Siqueira and P. S. R. Diniz, *The Electrical Engineering Handbook, Chapter 2, Digital Filters*. Elsevier Academic Press, 2005.

[17] S. Brahim-Belhouari and A. Bermak, “Gaussian Process for Non-Stationary Time Series Prediction,” *Computational Statistics and Data Analysis*, 2004.

[18] B. Schöolkopf and A. Smola, *Learning with Kernels*. The MIT Press, Cambridge, Massachuset, 2002.

[19] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, “Efficient Planning of Informative Paths for Multiple Robots,” in *International Joint Conference on Artificial Inteligence (IJCAI)*, 2007.

[20] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” in *Neural Information Processing Systems (NIPS)*, 2012.

[21] A. Singh, F. Ramos, H. Whyte, and W. Kaiser, “Modeling and Decision Making in Spatio-temporal Processes for Environmental Surveillance,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.