Estimating Jumping Heights of a Small Legged Robot based on Terrain Properties, Control Efforts, and Tactile Sensor Measurements

Tae Myung Huh[1], Amy Kyungwon Han[1], and Lawrence H. Kim[1]

[1]*Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA*

Abstract

Jumping of legged mobile robots has been a highly motivated research area. When a running robot encounters obstacles comparable to its body height, jumping is one of the most effective ways to overcome them. Also, if the robots can jump over gaps or crevices, the mobility of robots in a wild field would be enhanced drastically. The jumping performance is dependent on the terrain properties as well as its jumping mechanism. We focus on low and high friction surfaces that result in higher and lower jumping heights, respectively. In this project, we classify terrain type, high versus low friction surfaces, and estimate jumping heights. Training data include stride frequency, sensor data, and jumping height. For both terrain classification and jumping height estimation, SMO regression with PUK kernel has the best performance with mean test error of 0.09 and 1.5 mm, respectively. Greedy stepwise algorithm is used for feature selection, and when trained with five most influential features, comparable accuracies are obtained. From the results, several interesting attributes of the jumping robot are also found.

1 Introduction

Jumping of legged mobile robots has been a highly motivated research area. When a running robot encounters obstacles which are comparable to its body height, jumping is one of the more effective ways to overcome them. Also, if the legged robots can jump over gaps or crevices, the mobility of robots in a wild field would be enhanced drastically.

Previous researches on jumping robots are focused on jumping mechanisms. For example, the 6-inch tall SandFlea robot from Boston Dynamics[1] is able to jump up to 8 meters. MIT researchers invented a novel leg design for their cheetah robot[2] which succeeded in jumping over hurdles of various heights.

However, the jumping performance is dependent on the terrain properties as well as its jumping mechanism. For example, jumping on a low friction surface dissipates less energy than jumping on a high friction surface. This implies that more energy can be transferred to potential energy, in former case, resulting in higher jumping height. Therefore, the jumping height varies on different surfaces even with the same jumping mechanism, and we need to understand the ground characteristics in order to control it more accurately. A similar research on terrain classification is done using accelerometers attached to the body[3] or vision[4]. Using either method is easier to implement, however, they are indirect ways to measure the ground characteristics during impact. In this project, we use five force sensors directly attached to the leg to measure the actual ground reaction forces on each segment of the leg. Based on the force sensor data, terrain characteristics, and control efforts, we classify terrain type, and estimate the jumping height.

2 Experimental Setup and Data Acquisition

Similar to previous tethered runners that run in a circle, a smaller two-legged runner is built to measure the jumping height and ground reaction forces on various terrain surfaces. The mass and dimensions of the runner are selected to mimic the structure of other small legged robots such as the RoACH class of robots, which employ curved legs for locomotion.

Two C-shaped legs are attached to the motor

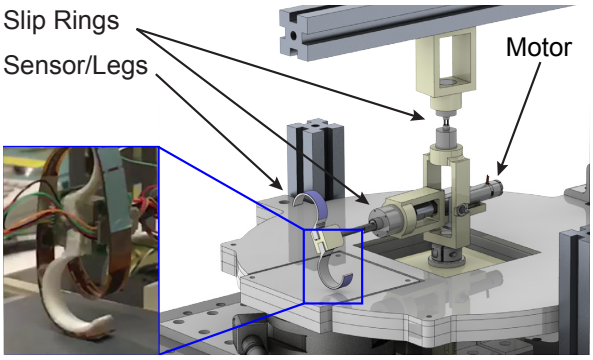

Figure 1: Experimental Setup of Two-legged Robot

Feature Index	Description
1	Robot stride frequency
2	Sensor force (net) peak amplitude
3	Sensor force (net) duration
4	Sensor force (net) area under curve
5	Sensor force (net) average amplitude
6-15	Individual taxel (1-10) force peak amplitude
16-25	Individual taxel (1-10) area under curve
26-35	Individual taxel (1-10) average amplitude
36	Sensor force peak to average
37	Height

Table 1: Machine Learning Feature Set

shaft (Maxon DCX10L EB KL 12V, 16:1 planetary gearhead), each on the other side of the shaft. The motor is mounted on a 3D printed one-axis gimbal structure that allows the motor and leg assembly to move freely about the vertical axis. The setup is shown in Figure1.

To solve the wiring problems, we employed slip-rings. Wires from rotating leg sensors are connected to the computer via a through bore slip-ring (Orbex Group, 503-0600). A microprocessor (Texas Instruments TM4C123GH6PM) is used to change stride frequency with a closed-loop PID control.

3 Machine Learning Method

3.1 Features

- Stride frequency : The stride frequency of the robot leg determines the kinetic energy of the robot, which is transferred to potential energy. The potential energy is directly related to the jumping height.
- Peak amplitude and intergral of sensor force : Normal ground reaction force during collision is related to the change of vertical linear momentum. Thus, the amplitude and integral of sensor force are good indicators for the jumping height.
- Sensor force duration: The duration varies for different terrains. The ground reaction force on the hard surfaces may have sharper impulse-like peak. On the other hand, force profile on the soft surfaces (i.e. thick carpet) will exhibit more widely spread peak with respect to time. Thus, sensor force duration is a key feature for terrain classification.
- Terrain type : The robot leg has different jumping performance among various terrain surfaces. Therefore, the terrain type has a significant effect on the jumping height.

3.2 Algorithm

For the terrain classification, several different classifiers from WEKA, a machine learning software workbench from University of Waikato, are deployed: Support Vector Machine (SMO with PUK kernel), Gaussian process regression, logistic regression, and Multilayer perceptron.[5] After the classification, each classifier goes through a 10-fold cross-validation to test its performance. The classifier with highest kappa statistic and lowest test mean absolute error is chosen and the corresponding confusion matrix is calculated. Then, we use Greedy Stepwise algorithm with 10-fold cross validation for feature selection. The five most influential features are selected and performance of the classifiers when trained with all features and just five features are compared by computing their respective confusion matrix and computation time.

Using this classified terrain type along with sensor data and control efforts listed in Table 1, several classifiers are implemented to estimate the jumping height. Again, WEKA library is used to implement three different regression models (linear, SMO, Gaussian Process) with three varying kernel (RBF, PUK, Poly). The functions of the kernels are shown in Table 2. Same as before, a 10-fold cross validation is used to calculate the test error for each regression model/kernel set and the following result is used to determine the most suitable one. For feature selection, same method is used as described in previous paragraph.

Type of Kernel	Function
Radial Basis Function (RBF)	$K_r(x^i, y^j) = exp(-\gamma\|x^i - x^j\|^2$
Pearson VII Function-based Universal (PUK)	$K_r(x^i, y^j) = 1/[1 + 2\sqrt{\|x^i - x^j\|^2}]^\omega$
Polynomial (Poly)	$K_r(x^i, y^j) = (x^i_T x^j + 1)^P$

Table 2: Kernel Functions

Classifiers	Kappa Statistics	Mean Absolute Error	Root Mean Squared Error	Computation Time [s]
SVM-SMO PUK Kernel	0.98	0.09	0.095	35.9
Gaussian	0.94	0.035	0.168	3.6
Logistic Regression	0.92	0.068	0.181	2.63
Multilayer Perceptron	0.97	0.020	0.118	166.7

Table 3: Terrain Classifier Error Table

	Low Friction Surface (Teflon)	**High Friction Surface** (Rubber)
Low Friction Surface	99.04	0.96
High Friction Surface	0.83	99.17

Table 4: Confusion Matrix for SMO Regression with PUK Kernel

In order to determine the minimum required size for training set, learning curve is plotted. As the number of training data increases, the train and test errors are plotted to show convergence after certain amount of training samples.

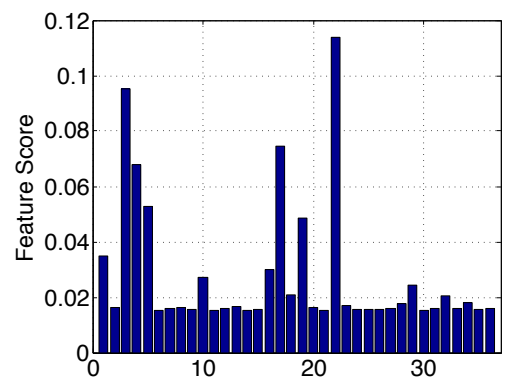

Figure 2: Feature Scores for Terrain Classification. Top five features are (1) Taxel.7 area under curve, (2) net force duration, (3) Taxel.2 area under curve, (4) net force area under curve, (5) net force average amplitude

4 Results

4.1 Terrain Classification

4.1.1 Model Selection

As shown in Table 3, the SVM with SMO and PUK kernel has the highest Kappa statistic and mean absolute error. Thus, this is selected to be our terrain classifier and the corresponding confusion matrix is calculated after a 10-fold cross validation. As shown in Table 4, the confusion matrix shows a high accuracy of approximately 99% for the SVM classifier.

4.1.2 Feature Selection

Figure 2 shows the scores for each of the features used in terrain classification when using SVM. From this, five most influential features are selected and the confusion matrix is computed after training using only these five features as shown in Table 5. The computation time is 29.5 seconds.

4.2 Height Regression

4.2.1 Model Selection

Table 6 shows that the Gaussian Process regression with PUK kernel has the highest performance in estimating training samples whereas SMO regression with PUK kernel has the minimum test errors of 0.48 mm. On the other hand, linear regression and SMO regression with RBF kernel has the worst accuracy on each surface respectively. When trained with all features, even the highest test error (SMO, RBF Kernel) does not exceed 2.72 mm of jumping height.

	Low Friction Surface (Teflon)	**High Friction Surface** (Rubber)
Low Friction Surface	91.73	8.27
High Friction Surface	12.70	87.30

Table 5: Confusion Matrix for SMO Regression with PUK Kernel, Using Top Five Features

High Friction Surface	**Train Error Mean (mm)**		**Test Error Mean (mm)**		**Computing Time (sec)**	
	All	Top 5	All	Top 5	All	Top 5
Linear Reg.	1.01	1.90	1.04	1.91	4.5	4.0
SMO (RBF)	0.93	1.83	0.95	1.84	25.2	20.6
SMO (PUK)	0.04	0.34	0.48	0.48	101.3	84.4
SMO (Poly)	0.40	1.31	0.54	1.34	988.1	29.1
Gaussian (RBF)	0.49	1.27	0.56	1.28	107.5	110.9
Gaussian (PUK)	0.00	0.17	0.49	0.68	101.1	95.7
Gaussian (Poly)	0.33	1.96	0.58	1.97	104.7	95.0

Low Friction Surface	**Train Error Mean (mm)**		**Test Error Mean (mm)**		**Computing Time (sec)**	
	All	Top 5	All	Top 5	All	Top 5
Linear Reg.	2.53	3.33	2.63	3.36	4.3	4.0
SMO (RBF)	2.65	3.53	2.72	3.56	23.9	21.8
SMO (PUK)	0.24	1.34	1.33	1.50	147.1	35.8
SMO (Poly)	1.22	3.32	1.51	3.38	403.5	19.9
Gaussian (RBF)	1.58	2.66	1.76	2.69	107.8	104.1
Gaussian (PUK)	0.01	1.04	1.42	1.74	100.6	94.8
Gaussian (Poly)	1.04	5.18	1.97	5.30	102.0	95.4

Table 6: Error table from each model for Height Estimation.

4.2.2 Feature Selection

Figure 3 shows scores for each feature on both surfaces after applying Gaussian Process regression with PUK kernel. We selected top five highest scored features for each surface and trained only with those features. The corresponding train and test errors are listed in Table 6. When trained with top 5 features, the SMO model with PUK kernel remained the best model with lowest test error, whereas the Gaussian Process with Poly Kernel became the worst with train and test error of 5.18 mm and 5.30 mm, respectively.

4.2.3 Training set size evaluation

Figure 4 demonstrates the train and test errors of the regression with respect to the size of the training set. Train and test error start to converge around 800 samples, which means that 800 training data is an adequate size for our application.

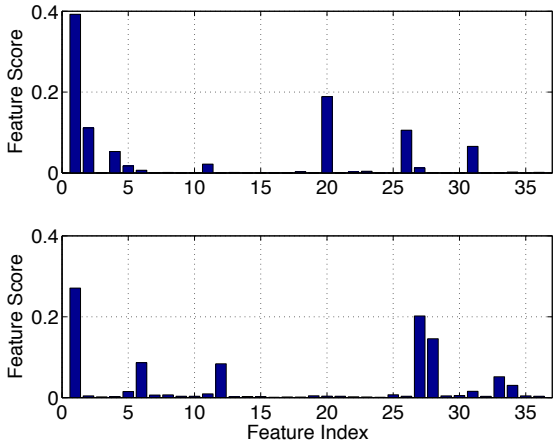

Figure 3: Feature Weight Sensor Subplot. Top five features on high friction surface : (1) robot stride frequency, (2) Taxel.5 area under curve, (3) net force peak amplitude, (4) Taxel.1 average amplitude, (5) Taxel.6 average amplitude Top five features on low friction surface : (1) robot stride frequency, (2) Taxel.2 average amplitude, (3) Taxel.3 average amplitude, (4) Taxel.1 peak amplitude, (5) Taxel.7 peak amplitude

5 Discussion

5.1 Terrain Classification

Using SMO with PUK kernel, high accuracy of about 99% is obtained as shown in Table 4. Binary nature of this terrain classification most likely had significant influence on the accuracy. In the future, more classes will be introduced and the corresponding accuracy would most likely reduce.

The five most influential features found from greedy stepwise algorithm are integral of channel 7 force, all channel's force duration, integral of channel 2 force, integral of all channel's force, and average net force of all channel. The stride frequency didn't seem to have a particular significance and this intuitively makes sense as terrain remains mostly the same regardless of the stride frequency. When trained with just the top 5 features, the computation time is decreased approximately 16.7% but with a sacrifice in accuracy of about 10% compared to the original results. This may be desirable if time is the more pressing constraint.

5.2 Jumping Height Regression

Generally, the test errors of various regression models are less than 2.7 mm when trained with all fea-

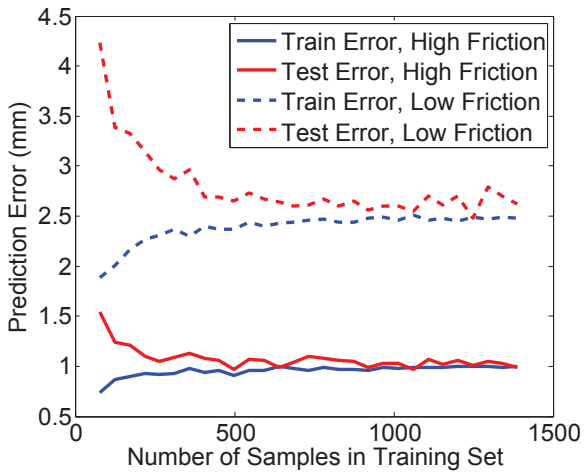

Figure 4: Error vs. Sample Size Table. Train error and test error converges after train data size of 800.

tures, a range acceptable for our application. Compared to the high friction surface, low friction surface has higher error overall. This is because more slips tend to occur on low friction surface resulting in a higher uncertainty. Linear regression does not use any dimensional changes and the resulting error suggest that the current feature space is not adequate. In terms of the kernels, PUK exhibits the best performance. RBF kernel does not perform any significantly better than a linear regression. This means that the high dimensional space made by RBF kernels does not result in the separable distribution for regression. When we use PUK or Poly kernel, the lowest training error is with the Gaussian model while the lowest test error is from the SMO model. This implies that Gaussian model is more of an overfit than SMO model with the PUK or Poly kernels. However, there is a trade-off between the error and computation time. The Gaussian Process regression has generally higher computation time than SMO and linear regression models. However, the Poly kernel worsen the efficiency of SMO regression drastically. Thus, one should carefully select the model and kernel, maintaining a balance between learning performance and computation time.

On each surface, the top 5 features are different from each other except for one feature: robot stride frequency. The running frequency is the most important feature for both cases because it determines the kinetic energy of the leg which is directly related to its jumping height. However, other four features are features of different sensor channels, located at different leg segments. This means that the dynamics of the leg is different on each surface so ground contact point is different. When we use top five features, the maximum error becomes 5.3 mm which is double the maximum error when trained with all features. However, the computation time does not reduce significantly except the SMO model with PUK, Poly kernel. Thus, depending on the model/kernel set, using only five features might not be beneficial in terms of computation time and errors.

Also, from the training set size evaluation, half of current training data has the same test and train error difference as the whole training set. Thus, we can save time and avoid overfitting by collecting only 800 training data.

6 Conclusion & Future Work

In conclusion, with the data and features obtained from the jumping robot experiments, SMO with PUK kernel demonstrate a test error of less than 1% for terrain classification, whereas for jumping height regression, SMO regression and Gaussian Process with PUK kernel both result in a model with test error mean of less than 1.5 mm. Using the greedy stepwise algorithm, five most influential features were selected and the test error means when trained with all features versus top five features are compared to validate its performance. This sometimes allow the algorithm to compute more efficiently based on the model/kernel set, useful for dynamic circumstances. In the future, the attributes of the jumping robot learned from this project will be used to build a robot that can control its jumping height.

References

[1] E. Ackerman, "Boston dynamics sand flea robot demonstrates astonishing jumping skills," *IEEE Spectrum Robotics Blog*, vol. 2, p. 1, 2012.

[2] X. Wang, M. Li, W. Guo, P. Wang, and L. Sun, "Design and development of a cheetah robot under the neural mechanism controlling the leg's muscles," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2749–2755, IEEE, 2012.

[3] F. L. Garcia Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing, "Performance analysis and terrain classification for a legged robot over rough terrain," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ*

International Conference on, pp. 513–519, IEEE, 2012.

[4] D. W. Haldane, P. Fankhauser, R. Siegwart, and R. S. Fearing, "Detection of slippery terrain with a heterogeneous team of legged robots," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 4576–4581, IEEE, 2014.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.