Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing

Ramon Gonzalez, Mirko Fiacchini, Karl Iagnemma

Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing

Ramon Gonzalez[1,3,*], Mirko Fiacchini[2], Karl Iagnemma[3]

[1] *robonity: tech consulting, Calle Extremadura, no. 5, 04740, Almeria, Spain*
[2] *Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France*
[3] *Massachusetts Institute of Technology, 77 Massachusetts Av., 02139 Cambridge, MA, USA*

Abstract

This paper presents a new approach for predicting slippage associated with individual wheels in off-road mobile robots. More specifically, machine learning regression algorithms are trained considering proprioceptive sensing. This contribution is validated by using the MIT single-wheel testbed equipped with an MSL spare wheel. The combination of IMU-related and torque-related features outperforms the torque-related features only. Gaussian process regression results in a proper trade-off between accuracy and computation time. Another advantage of this algorithm is that it returns the variance associated with each prediction, which might be used for future route planning and control tasks. The paper also provides a comparison between machine learning regression and classification algorithms.

Keywords: Gaussian Process Regression, Inertial Measurement Unit (IMU), Machine Learning Regression, Mars Science Laboratory (MSL) wheel, slip.

1. Introduction

Mobile robots operating in off-road conditions must face two key hazards in order to achieve a successful navigation. One of those hazards concerns with the detection and avoidance of geometrical obstacles. This problem is well

[*] Corresponding author.
Email addresses: ramon@robonity.com (R. Gonzalez), mirko.fiacchini@gipsa-lab.grenoble-inp.fr (M. Fiacchini), karl.iagnemma@gmail.com (K. Iagnemma)

understood and accounts for a broad body of research (LaValle, 2006; Maimone et al., 2006; Manduchi et al., 2005; Matthies et al., 2007). Another key challenge related to the mobility of mobile robots in off-road conditions comprises non-geometrical hazards. These hazards depend on the interaction between the robot's wheels and the terrain. For example, a robot traversing loose, sloped sand might experience poor mobility, whereas a robot traversing flat, firm clay might experience excellent mobility (Iagnemma & Dubowsky, 2004).

A fundamental phenomenon derived from the wheel-soil interaction is slippage (Angelova et al., 2007; Gonzalez & Iagnemma, 2017; Wong, 2001). Slippage is a measure of the lack of progress of a wheeled ground vehicle while driving on certain terrains (e.g. sandy slopes, ripples, and low-cohesion soils). Though slippage does not necessarily mean loss of traction. The problem is that excessive slip might cause a loss of tractive effort and rover speed, and eventually robot entrapment (Gonzalez & Iagnemma, 2017). For that reason, it is critical for the success of the robot operation to estimate as accurately as possible the level of slippage of the robot's wheels. The worst situation related to slippage was experienced by the MER Spirit rover which got trapped in a sand dune in 2009. After numerous attempts to free the rover, the mission was declared concluded on May 24, 2011 (Webster & Brown).

Off-road applications also require a robot to depend on simple, on-board sensors to perceive the environment. For example, planetary exploration rovers account for a limited sensor suite mainly composed of one inertial measurement unit (IMU), motor current sensors and encoders on the wheels, and visual cameras. Such sensors generally contain significant uncertainty and error in their measurements (Iagnemma & Dubowsky, 2004).

This paper provides a methodology where machine learning regression algorithms are used for detecting slippage and its associated uncertainty. This means that slippage is understood as a random variable with a given variance. This variance is associated with the uncertainty derived from the sensors onboard the robot. In addition to that, the proposed machine learning methodology exploits the use of traditional sensors available in off-road robots such as IMU and motor current sensors. Consequently, no operational complexity is added to the rover's commanding and it is independent of lighting conditions (this is an interesting advantage for mobile robots operating in dark or shadowed areas such as mining or greenhouses). The slip derived from this methodology and its associated variance might be ultimately used for modifying the route of the robot and the control actions (Gonzalez & Iagnemma, 2016).

This paper is organized as follows. In Section 2, previous work in slip estimation and machine learning regression is reviewed. Section 3 reviews the machine learning regression algorithms considered in this work. Section 4 explains the data set (data collection) and the set of features considered for training the machine learning algorithms. Section 5 provides experimental results showing a comparison among the regression algorithms presented in this work and other algorithms previously published by the authors and based on machine learning classification (Bouguelia et al., 2017; Gonzalez et al., 2018). Finally, conclusions and future work are drawn in Section 6. The interested reader can see a video of one experiment at: `https://youtu.be/kKRSk0rAUdE`

2. Related Work

Traditionally slip estimation deals with strategies that result in a continuous value for such variable. It means that slip is given as a real number with a certain precision. For that purpose, those strategies focus on two variables: the wheel angular velocity and the forward velocity of the robot (Gonzalez & Iagnemma, 2017). One of the first methods found in the literature for estimating slip was proposed by Prof. Wong in the 1960s-70s. Slip was directly measured by comparing signals from a wheel placed in front of the robot/vehicle (Wong, 2001). A simple approach relies on comparison of wheel velocities to a robot body velocity estimate derived from integration of a linear acceleration measurement in the direction of travel (e.g. using accelerometers) (Barshan & Durrant-Whyte, 1995; Iagnemma & Ward, 2009). Another extended method for estimating rover slip is based on Visual Odometry (VO) (Angelova et al., 2007; Gonzalez et al., 2014; Matthies et al., 2007). The main limitation of those approaches is that such single number does not give any information about the uncertainty associated with the estimation and the noise in the measures and sensors.

On the other hand, a new paradigm has recently appeared in the literature and proposed by the authors of this paper (Gonzalez et al., 2018; Bouguelia et al., 2017). It estimates slip as a discrete variable and machine learning algorithms are used for solving this as a classification problem. In particular, a model is trained offline while using a set of proprioceptive measurements. Online computation is then devoted to using such model for predicting the slip in terms of new measurements obtained while the robot is moving. As described in (Gonzalez et al., 2018), slip belongs to three classes: low slip when slip is between 0 and 30 %, moderate slip when it is within the range 30 and 60 %, and high slip when slip is over 60 %. Though field tests demonstrate promising results, this approach does not give information about the uncertainty in the estimation.

This paper comes to complete the previous approach. Here, slip is defined as a random variable, the expected value of that variable means the predicted slip and the variance is the uncertainty in such prediction. In particular, this paper solves this problem involving a random variable by using machine learning regression. It bears mentioning that the predicted slip and the uncertainty in such prediction can be certainly useful for both slip compensation Gonzalez et al. (2014) and motion planners Ordonez et al. (2017). The methodology proposed here could complement those approaches by considering routes to a target point where uncertainty in slip is minimized.

The methodology proposed here is based on Gaussian Process Regression (GPR) and slip is understood as a multivariable Gaussian distribution (Marsland, 2015; Rasmussen & Williams, 2006). GPR accounts for a broad body of research and has been used by many references, specially in the field of geostatistics as a way to generate terrain models and mobility maps. For example, in (Gonzalez et al., 2017a), a method based on GPR (i.e. Ordinary Kriging) is used for generating a mobility map accounting for measurements errors (i.e. satellite signal) and interpolation error. The path planner is formulated in such a way that routes avoid points of high uncertainty. The work (Gonzalez et al., 2017b) compare different routes using different cost functions and various performance indices. In (Karumanchi et al., 2010), GPR is used for generating a mobility map based on terrain elevation and wheel slip. Path planning then takes advantage of that map in order to improve vehicle heading and velocity in off-road slopes. For comparison purposes, this paper also takes into account the algorithms: Support Vector Regression (SVR) (Smola & Scholkopf, 2004; Vapnik, 1995), and Kernel Ridge Regression (KRR) (Murphy, 2012).

3. Machine Learning Regression

Machine learning is a branch of computer science based on the study of algorithms that can learn from and make predictions (generalize) on data. There are two main paradigms within machine learning: regression and classification. The first approach deals with taking input variables and forecasting the value of the output (dependent) variable(s). It is based on estimating the relationships among variables (independent and dependent variables) and predicting a numeric value (with an associated variance, in some cases). On the contrary, classification is related to taking input variables and deciding which of N classes they belong to, based on training from exemplars of each class (Marsland, 2015). In this sense, it is based on finding decision boundaries that can be suited to separate out the different classes.

As previously introduced, this paper aims at estimating slippage by means of a regression model derived from training data. More specifically, consider the problem of predicting the slippage, s, of a mobile robot using as input the feature vector $\mathbf{q} \in \mathbb{R}^m$ where $m \in \mathbb{Z}^+$ is the number of features. Each feature is a numerical representation of sensor data that attempts to mimic the sensory cues a human operator would exploit when attempting to detect slippery conditions (e.g. vertical acceleration). The set of features $\mathbf{q}$ is obtained after transforming the raw data coming from proprioceptive sensors onboard the robot (explained in Section 4).

3.1. Gaussian Process Regression

In this section, the regression model is formulated as a Gaussian Process Regression problem (GPR) (Marsland, 2015; Rasmussen & Williams, 2006). The objective is to find a collection of random variables with Gaussian distribution $s(\mathbf{q})$, one for every realization of the input $\mathbf{q}$, such that the joint distribution over any finite subsets of variables is also Gaussian. More specifically, our goal is to identify a functional relationship (regression model) mapping the multi-dimensional input vector, $\mathbf{q}$, on a random variable representing the slippage, s.

Definition 1 ((Rasmussen & Williams, 2006)). *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

The particularity of Gaussian processes is that they are completely specified by the mean and the covariance. In particular, then, the expected value of the slippage and the covariance function between observations are supposed given by

$$\bar{s}(\mathbf{q}) = \mathbb{E}[s(\mathbf{q})], \tag{1}$$

$$k(\mathbf{q}, \mathbf{q}') = \mathbb{E}[(s(\mathbf{q}) - \bar{s}(\mathbf{q}'))(s(\mathbf{q}) - \bar{s}(\mathbf{q}'))], \tag{2}$$

where the resulting regression function, $s(\cdot)$, is a Gaussian process denoted as follows

$$s(\mathbf{q}) = \mathcal{GP}\left(\bar{s}(\mathbf{q}), k(\mathbf{q}, \mathbf{q}')\right). \tag{3}$$

Namely, for every feature realization $\mathbf{q}$, $s(\mathbf{q})$ is a random variable with Gaussian distribution characterized by its mean and its variance. Finally, and considering that no loss of generalization is induced by considering the mean function to be zero, see (Marsland, 2015; Rasmussen & Williams, 2006) and references therein, the Gaussian process is completely determined through the definition of the covariance function $k(\mathbf{q}, \mathbf{q}')$. Among the different admissible covariance functions, we should recall the Squared Exponential (SE):

$$k(\mathbf{q}, \mathbf{q}') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}|\mathbf{q} - \mathbf{q}'|^2\right), \tag{4}$$

characterized by the two parameters σ_f and l and analogous to the Radial Basis Functions kernel for regression, see below.

Thus, given a set of N training points $\mathbf{q}_1, \ldots, \mathbf{q}_N$, a single prediction point $\mathbf{q}^*$ and $\mathbf{s} = (s_1, \ldots, s_N)$ the ground-truth slippage values at the training points, the aim is to calculate the conditional distribution of the value s^* at $\mathbf{q}^*$:

$$P(s^*|\mathbf{s}) \quad = \quad \frac{P(s^*, \mathbf{s})}{P(\mathbf{s})}. \tag{5}$$

The joint distribution $P(s^*, \mathbf{s})$ is Gaussian from the assumptions and its covariance matrix given by

$$\kappa \quad = \quad \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}, \tag{6}$$

where

$$\begin{aligned} K \quad &= \quad \begin{bmatrix} k(\mathbf{q}_1, \mathbf{q}_1) & k(\mathbf{q}_1, \mathbf{q}_1) & \ldots & k(\mathbf{q}_1, \mathbf{q}_N) \\ k(\mathbf{q}_2, \mathbf{q}_1) & k(\mathbf{q}_2, \mathbf{q}_2) & \ldots & k(\mathbf{q}_2, \mathbf{q}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{q}_N, \mathbf{q}_1) & k(\mathbf{q}_N, \mathbf{q}_2) & \ldots & k(\mathbf{q}_N, \mathbf{q}_N) \end{bmatrix}, && (7) \\ K_* \quad &= \quad [k(\mathbf{q}_*, \mathbf{q}_1) \quad k(\mathbf{q}_*, \mathbf{q}_2) \quad \ldots \quad k(\mathbf{q}_*, \mathbf{q}_N)], && (8) \\ K_{**} \quad &= \quad k(\mathbf{q}_*, \mathbf{q}_*). && (9) \end{aligned}$$

Notice that K is the covariance matrix for the training data, K_* is the covariance matrix between the test points (prediction) and the training data, and K_{**} is the covariance between the points in the test set.

At this point, we can calculate the conditional probability in Eq. (5) as

$$P(s^*|\mathbf{s}) \sim \mathcal{N}(K_* K^{-1}\mathbf{s}, K_{**} - K_* K^{-1} K_*^T), \tag{10}$$

where $\mathcal{N}(m, \Sigma)$ denotes a Gaussian distribution with mean m and covariance Σ.

The best estimate for s^* is the mean of this distribution

$$\bar{s}^* \quad = \quad K_* K^{-1} \mathbf{s}, \tag{11}$$

and the uncertainty in our estimate is captured by its variance

$$var(s^*) \quad = \quad K_{**} - K_* K^{-1} K_*^T. \tag{12}$$

It is important to remark that in general training data (and testing data) are going to be subject to noise. The usual way to add noise into any GPR is to assume that it is an independent, identically distributed Gaussian noise with zero mean and variance σ_n^2. This implies that the covariance matrix K is now replaced by $K = K + \sigma_n^2 I$, where I is the $N \times N$ identity matrix. The conditional probability is now determined as

$$P(s^*|\mathbf{s}) \sim \mathcal{N}(K_* \left(K + \sigma_n^2 I\right)^{-1} \mathbf{s},\ K_{**} - K_* \left(K + \sigma_n^2 I\right)^{-1} K_*^T). \tag{13}$$

After considering the noisy observations, the best estimate for s^* is now obtained as

$$\bar{s}^* \quad = \quad K_* \left(K + \sigma_n^2 I\right)^{-1} \mathbf{s}, \tag{14}$$

and the variance is now given by

$$var(s^*) \quad = \quad K_{**} - K_* \left(K + \sigma_n^2 I\right)^{-1} K_*^T. \tag{15}$$

3.2. Support Vector Regression and Kernel Ridge Regression

Both Support Vector Regression (SVR) and Kernel Ridge Regression (KRR) have the objective of providing the best approximation of the function defined over the input space, in our case $s^* = f(\mathbf{q})$. They differ, substantially, on the cost functions to be minimized and then correspond to different optimality criteria of the solution. Both are based on applying the kernel trick to linear approximation and distance-based optimization problem, to introduce modelling richness but avoiding most of the computational complexity induced by nonlinearity. Therefore, it is useful to introduce the linear approximation problems and then extend them to the case of nonlinear ones, by recalling the basis of the kernel trick, as in (Marsland, 2015; Smola & Scholkopf, 2004).

The objective of SVR in the linear context, is to compute $w \in \mathbb{R}^m$ and $b \in \mathbb{R}$ such that

$$f(\mathbf{q}) = w^T \mathbf{q} + b, \tag{16}$$

is a good approximation of the relation between the measures $\mathbf{q}$ and the slippage s. To evaluate the different possible linear approximation functions, SVR considers the norm of w that is related to the flatness and to shrinkage, see (Smola & Scholkopf, 2004) and references therein, and the distance of the training values $\mathbf{s}$ from a band of amplitude ϵ around the function $f(\mathbf{q})$. This means that the optimization problem is

$$\begin{aligned} \min_{w,b,d^+,d^-} \quad & |w|^2 + C \sum_{i=1}^{N} \left(d_i^+ + d_i^- \right) \\ \text{s.t.} \quad & s_i - w^T \mathbf{q}_i - b \leq \epsilon + d_i^+, \quad \forall i = 1, \ldots, N, \\ & w^T \mathbf{q}_i + b - s_i \leq \epsilon + d_i^-, \quad \forall i = 1, \ldots, N, \\ & d_i^+ \geq 0, \quad d_i^- \geq 0 \quad \forall i = 1, \ldots, N, \end{aligned} \tag{17}$$

with C and ϵ positive constants and where the cost penalizes, besides $|w|^2$, the distances d_i^+ and d_i^- between s_i and its approximation $f(\mathbf{q}_i)$ plus and minus ϵ. By posing and solving the dual problem of (17), the expression of the optimal values of $\bar{w}$ and $\bar{b}$ can be computed as functions of the Lagrange multipliers by using the Karush-Kuhn-Tucker optimality conditions, see (Smola & Scholkopf, 2004). Computing the prediction at $\mathbf{q}^*$ reduces to evaluate

$$s^* = \bar{w}^T \mathbf{q} + \bar{b} = \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) \mathbf{q}_i^T \mathbf{q} + \bar{b},$$

with α_i^+ and α_i^+ are the optimal Lagrange multipliers that also determine the value of b, see (Smola & Scholkopf, 2004). The kernel trick consists in practice in replacing the origin feature vector $\mathbf{q}$ in (16) with a vector of functions of the features $\Phi(\mathbf{q})$, as polynomial ones e.g. $\Phi(x) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2\right]$ for $x \in \mathbb{R}^2$ for instance, for which the term $\Phi(\mathbf{q})^T \Phi(\mathbf{q}')$ is easily computable.

In fact, only the explicit expression of $k(\mathbf{q}, \mathbf{q}') = \Phi(\mathbf{q})^T \Phi(\mathbf{q}')$ is necessary, the prediction reducing to

$$s^* = \bar{w}^T \Phi(\mathbf{q}) + \bar{b} = \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) \Phi(\mathbf{q}')^T \Phi(\mathbf{q}) + \bar{b} = \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) k(\mathbf{q}, \mathbf{q}') + \bar{b},$$

that is a linear function on the extended space defined by the functions determining $\Phi(\mathbf{q})$. Common choices are the Radial Basis Function kernel:

$$k(\mathbf{q}, \mathbf{q}') = \exp\left(-\frac{1}{2\sigma^2} |\mathbf{q} - \mathbf{q}'|^2\right), \tag{18}$$

or the polynomial kernel of degree d:

$$k(\mathbf{q}, \mathbf{q}') = (1 + \mathbf{q}^T \mathbf{q}')^d. \tag{19}$$

Analogous reasoning holds for KRR (Murphy, 2012), whose objective is to find the linear function $f(\mathbf{q}) = w^T \mathbf{q}$ that minimizes a cost composed by the squared norm of w and the sum of the squared distances between $f(\mathbf{q}_i)$ and s_i, that is

$$\min_w \ \lambda |w|^2 + \sum_{i=1}^{N} \left(s_i - w^T \mathbf{q}_i\right)^2 = \min_w \ \lambda |w|^2 + (\mathbf{S} - \mathbf{Q}w)^T (\mathbf{S} - \mathbf{Q}w), \tag{20}$$

where $\mathbf{S} = [s_1 \ \ldots s_N]$ and $\mathbf{Q} = [\mathbf{q}_1 \ \ldots \mathbf{q}_N]^T$. Hence, in practice, the ridge regression aims at computing the linear function minimizing the squared prediction errors, besides the norm of w. In this case, the optimal value of the parameter vector w is analytically given by

$$\bar{w} = \left(\mathbf{Q}^T \mathbf{Q} + \lambda I_m\right)^{-1} \mathbf{Q}^T \mathbf{S}. \tag{21}$$

By means of further manipulations and kernelizing, see (Murphy, 2012), we obtain the optimal prediction

$$s^* = f(\mathbf{q}) = \bar{w}^T \Phi(\mathbf{q}) = \sum_{i=1}^{N} \beta_i k(\mathbf{q}, \mathbf{q}_i) \quad \text{with} \quad \beta = (K + \lambda I_N)^{-1} \mathbf{S}, \tag{22}$$

where $K \in \mathbb{R}^{N \times N}$ is the Gram matrix, i.e. $K_{i,j} = k(\mathbf{q}_i, \mathbf{q}_j)$ for all $i, j = 1, \ldots, N$. 180

4. Data Collection and Feature Selection

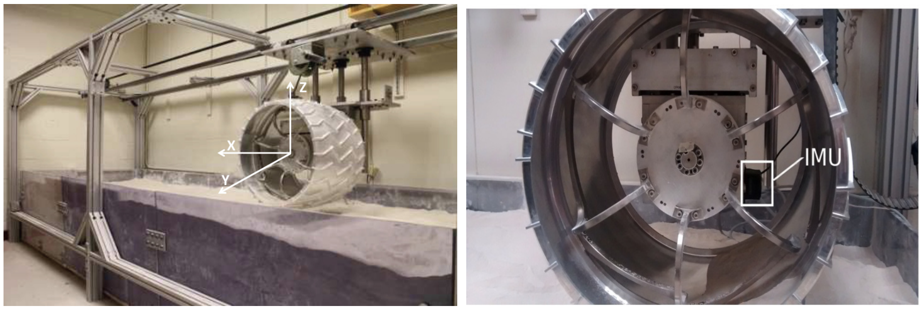
The generality of the methodology proposed in this work is validated by using one data set. This dataset comprises the lab conditions encountered in the MIT single-wheel testbed. These conditions included small soil ripples in the path of the wheel to create soil compaction resistance in a manner similar to what is currently being experienced on Mars by MSL.

4.1. MIT single-wheel testbed

In order to evaluate the performance of the proposed methodology, various physical experiments were conducted using a single-wheel testbed developed by the Robotic Mobility Group (RMG) at MIT. The system limited the wheels movement primarily to its longitudinal direction. By driving the wheel and carriage at different rates, variable slip ratios can be imposed (Figure 1a). The bin dimensions are 3.14 [m] length, 1.2 [m] wide, and 0.5 [m] depth.

The wheel in use for the experimentation was a Mars Science Laboratory (MSL) flight spare wheel. The sensing system of the testbed consists of: an IMU (MicroStrain, 3DM-GX2), a torque sensor (Futek, FSH03207), and a displacement sensor (Micro-epsilon, MK88). Data was recorded at 100 [Hz] in an external computer. A detail of the placement of the IMU sensor can be seen in Figure 1b. The soil used during testing was a Mars regolith simulant developed at MIT to replicate conditions being experienced by the MSL rover on Mars. Numerous experiments were carried out inducing wheel slip under various operation conditions (i.e., ripple geometries, wheel and pulley velocity rates) and loading conditions of the carriage pulley.

Figure 2 shows an experiment where the wheel experiences low slip ($<$ 30 [%]) at the beginning of the test and high slip ($\sim$ 90 [%]) at the end of the test. Notice the "JPL" pattern in the first tracks (used by the MSL mission planners to estimate slip), and the sinkage effect in the second figure, the wheel is stuck. The interested reader can see a video of one experiment at: `https://youtu.be/kKRSkOrAUdE`



(a) MSL flight spare wheel in MIT's testbed

(b) Position of the IMU sensor on the MSL wheel

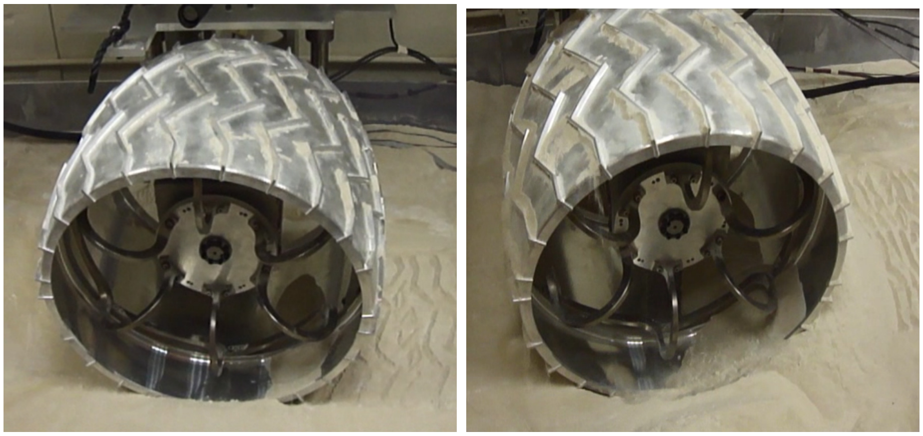
Figure 1: Single-wheel testbed developed by RMG-MIT and used for collecting experimental data. The IMU constitutes the primary sensor in the proposed methodology

For ground-truth purposes, slip was estimated measuring the angular velocity of the wheel and the angular velocity of the carriage pulley. Notice that the dataset considered in this work comprises a series of ten experiments resulting in a traverse of approximately 20 [m]. In those experiments the single wheel moved at a fixed velocity of approximately 0.15 [m/s].

Figure 3a shows the data set collected by the MIT single-wheel testbed and used for validating the performance of the learning algorithms. It covers a broad variety of slip rates from 0% to almost 100 %. Notice the apparent relation between the slip values and the rest of features derived from the sensors. For example, when slip increases the variance in the data from the IMU also increases. A similar situation occurs with the motor torque. Figures 3b, 3c show the two datasets used for training and testing the algorithms using the hold-out cross-validation strategy (70 % of the samples used for training, 9235) and (30 % of the samples for testing, 3958).

4.2. Feature selection

This section presents the four features that have been chosen to form the feature input vector to the learning algorithms. The reasoning behind this choice is based on our experience on this field, see for example (Gonzalez et al., 2018;



(a) Low-slip condition (< 30 %) (b) High-slip condition (~ 90 %)

Figure 2: Testing using the MSL flight spare wheel at the RMG-MIT single-wheel testbed. A better understanding of these two conditions is found in the following video: `https://www.youtube.com/watch?v=kKRSk0rAUdE`

Iagnemma & Ward, 2009). Notice that after those four features were chosen, the performance of the machine learning algorithms was extensively analyzed while considering various combinations of those features (e.g. only one feature, two features, three features, etc.). The best result was obtained when the four variables were considered, which demonstrated their independency (Gonzalez et al., 2018).

The first feature is the absolute value of the wheel torque

$$q_{i,1} = T_i, \tag{23}$$

where T_i is the i-th instance of motor torque. This motor torque is derived from a current sensor mounted on the servo motor. Notice that during normal outdoor driving, terrain unevenness leads to variations in wheel torque. This value is increased when the robot is experiencing moderate or high slip (Gonzalez et al., 2018).

The rest of features are collected by an IMU sensor. These features were chosen as the variance of the N_w element groupings i of the linear acceleration (x-axis), $\dot{x}_{i,N_w}$, the degree of pitch (y-axis), $\dot{\phi}_{i,N_w}$, and the vertical acceleration (z-axis), $\dot{z}_{i,N_w}$; see (Iagnemma & Ward, 2009) for further detail of this technique (sliding variance),

$$\begin{aligned} q_{i,2} &= var(\dot{x}_{i,N_w}) = E((\dot{x}_{i,N_w} - E(\dot{x}_{i,N_w}))^2), \qquad (24) \\ q_{i,3} &= var(\dot{\phi}_{i,N_w}) = E((\dot{\phi}_{i,N_w} - E(\dot{\phi}_{i,N_w}))^2), \qquad (25) \\ q_{i,4} &= var(\dot{z}_{i,N_w}) = E((\dot{z}_{i,N_w} - E(\dot{z}_{i,N_w}))^2). \qquad (26) \end{aligned}$$

During normal outdoor driving, terrain unevenness leads to variations in those variables. As with the wheel torque, this variation is maximized when the robot is experiencing moderate or high slip.

5. Results

This section analyzes the performance of the machine learning regression algorithms considered in this research. There are four important points to be considered: (1) the accuracy and the mean absolute error of the algorithms with respect to the ground-truth slippage; (2) the influence of the features considered by the machine learning algorithms (IMU-related and torque-related features versus torque-related features only); (3) the performance of the machine learning regression algorithms used here against a well-known machine learning classification algorithm (i.e. SVM) following the procedure previously described by the authors in (Gonzalez et al., 2018); (4) analysis of the computation time required by the learning algorithms (training time and testing time). These experiments have been run on a standard-performance computer (Intel Core i7, 3GHz, 16 GB RAM, OS X). The software has been implemented in Python by using the machine learning library scikit-learn (`http://scikit-learn.org`).

The machine learning regression algorithms SVR and KRR have been manually tuned (kernel function), and the tuning leading to the best result has been used in this analysis. The kernel used by the GPR algorithm has also been manually specified, but the algorithm automatically optimizes over the hyperparameters of the kernel.

The hold-out cross-validation method has been used for selecting the training and testing samples (70 % of the samples used for training, 9235) and (30 % of the samples for testing, 3958).

5.1. Regression algorithms

This section compares the performance of the machine learning regression algorithms (GPR, SVR, KRR). Additionally, this section also discusses the importance of considering IMU-related and torque-related features for maximizing the performance of the learning algorithms.

Figure 4 shows the models obtained for the training data. Following the machine learning regression paradigm, these models are used later for predicted new inputs.

The predicted values while considering the testing dataset are displayed in Figure 5. The three regression algorithms perform quite well and catch almost all the testing points (except some outliers). Notice that Figure 5b shows the advantageous feature of the GPR algorithm, as it also returns the variance associated with each point. This variance might be used for future route planning and control tasks (e.g. implementing a robust motion control strategy as previously published in Gonzalez et al. (2011)).

Finally, Figure 6 shows another significant contribution of this paper. Notice that the mean absolute error between the predicted values and the ground-truth slip is quite different when considering IMU-related and torque-related features and torque-related features only. In fact, in the latter case, the error is almost double than in the first case.

5.2. Regression algorithms vs classification algorithm

This section compares the performance of the machine learning regression algorithms used in this paper with the methodology previously published by the authors in (Gonzalez et al., 2018). In that reference, slippage estimation is solved as a machine learning classification problem. In this sense, slippage is understood as a discrete variable belonging to three categories or classes. That is, low slip when slip is less than 30 %, moderate slip when slip is between 30 and 60 %, and high slip when slip is higher than 60 %. In order to be able to compare both approaches (regression and classification), the predicted slip values obtained with the regression methods have been discretized according to those three ranges. The machine learning classification algorithm considered here is the Support Vector Machine (SVC) algorithm, see (Gonzalez et al., 2018) for further details.

Figure 7 shows the performance of the three machine learning regression algorithms (GRP, SVR, KRR), and the machine learning classification algorithm (SVC). According to the confusion matrices shown in Figures 7a, 7b, both GPR and SVC work well for the low-slip and high-slip classes (accuracy around 85 %). A lower accuracy is observed for the moderate-slip class (around 75 %). This lower accuracy (75 % vs 85-95%) can be due to the fact that there are less training samples in the moderate-slip range than in the other two slip levels.

Figure 7c shows the performance of the machine learning algorithms in terms of accuracy. As observed, all of them are around 80 %, which demonstrates the suitability of these strategies for estimating wheel slippage. Finally, Figure 7d addresses another important topic, the training and testing times. This could be a key factor depending on the application of the mobile robot and its computing resources. As observed, the machine learning classifier (SVC) runs much faster than the regression algorithms (almost one order of magnitude). The worst scenario is obtained with KRR when considering the training step. GPR leads to the highest testing time. It is important to remark that these computation times refer to the time required by the machine learning method for evaluating the whole dataset (more than 9000 samples for training and almost 4000 for testing). Therefore, in case of deploying these approaches in real-time they will need to evaluate just one single sample, which will be done of the order of milliseconds. In any case, the result displayed in Figure 7d overviews the method that would run faster.

6. Conclusions

This paper comes to complete the work already published by the authors in (Bouguelia et al., 2017; Gonzalez et al., 2018), where machine learning is used for estimating wheel slip by using proprioceptive sensing. The novelty of this work is that machine learning regression is used instead of machine learning classification. In this sense, the learning algorithms return a continuous value for slip and, even, the variance associated with such value. This information might be exploited by path planners in order to avoid those areas with high uncertainty / variability in the estimated slip (Gonzalez et al., 2017a; Lee et al., 2016). Additionally, motion controllers might generate robust control actions despite uncertainty in slippage (Gonzalez et al., 2014, 2011). Finally, low-level traction controllers might use this information for compensating for slip, see an example with discrete slip in (Gonzalez & Iagnemma, 2016).

Future efforts will be devoted to running physical experiments with actual off-road mobile robots under a broad set of terrains (e.g. sand, gravel, bedrock), and terrain geometries (flat, slope, ripples). In addition to that, new sensing devices will be tested (e.g. load cells, cameras). This information will be also used by the machine learning algorithms for predicting slip. The use of deep learning strategies for estimating slippage as a discrete variable will be also part of future efforts.

Acknowledgement

The research described in this publication was partially carried out at the Massachusetts Institute of Technology (Cambridge, MA, USA) under the STTR Contract NNX15CA25C funded by NASA.

References

Angelova, A., Matthies, L., Helmick, D., & Perona, P. (2007). Learning and Prediction of Slip from Visual Information. *Journal of Field Robotics*, *24*, 205–231.

Barshan, B., & Durrant-Whyte, H. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, *11*, 328–342.

Bouguelia, M., Gonzalez, R., Iagnemma, K., & Byttner, S. (2017). Unsupervised detection of soil embedding events for planetary exploration rovers. *Journal of Terramechanics*, *73*.

Gonzalez, R., Apostolopoulos, D., & Iagnemma, K. (2018). Slippage and immobilization detection for planetary exploration rovers via machine learning and proprioceptive sensing. *Journal of Field Robotics*, *35*, 231–247.

Gonzalez, R., Fiacchini, M., Guzman, J. L., Alamo, T., & Rodriguez, F. (2011). Robust Tube-based Predictive Control for Mobile Robots in Off-Road Conditions. *Robotics and Autonomous Systems*, *59*, 711–726.

Gonzalez, R., & Iagnemma, K. (2016). Soil Embedding Avoidance for Planetary Exploration Rovers. In *8th ISTVS Amercias Conference*. Detroit, MI, USA: International Society for Vehicle-Terrain Systems.

Gonzalez, R., & Iagnemma, K. (2017). Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges. *Journal of Field Robotics*, *DOI: 10.1002/rob.21761*, 1–14.

Gonzalez, R., Jayakumar, P., & Iagnemma, K. (2017a). Generation of stochastic mobility maps for large-scale route planning of ground vehicles: a case study. *Journal of Terramechanics*, *69*, 1–11.

Gonzalez, R., Jayakumar, P., & Iagnemma, K. (2017b). Stochastic Mobility Prediction of Ground Vehicles over Large Spatial Regions: A Geostatistical Approach. *Autonomous Robots*, *41*, 311–331.

Gonzalez, R., Rodriguez, F., & Guzman, J. L. (2014). *Autonomous Tracked Robots in Planar Off-Road Conditions. Modelling, Localization and Motion Control*. Series: Studies in Systems, Decision and Control. Springer, Germany.

Iagnemma, K., & Dubowsky, S. (2004). *Mobile Robots in Rough Terrain. Estimation, Motion Planning, and Control with Application to Planetary Rovers*. Springer Tracts in Advanced Robotics. Springer, Germany.

Iagnemma, K., & Ward, C. C. (2009). Classification–based Wheel Slip Detection and Detector Fusion for Mobile Robots on Outdoor Terrain. *Autonomous Robots*, *26*, 33–46.

Karumanchi, S., Allen, T., Bailey, T., & Scheding, S. (2010). Non-parametric Learning to Aid Path Planning over Slopes. *The International Journal of Robotics Research*, *29*, 997 – 1018.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, United Kingdom. Available at http://planning.cs.uiuc.edu/.

Lee, S., Gonzalez, R., & Iagnemma, K. (2016). Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain. In *IEEE Int. Conf. on Robotics and Automation (ICRA)* (pp. 2569 – 2574). Stockholm, Sweden.

Maimone, M., Biesiadecki, J., Tunstel, E., Cheng, Y., & Leger, C. (2006). Surface navigation and mobility intelligence on the Mars Exploration Rovers. In *Intelligence for Space Robotics* (pp. 45–69). TSI Press.

Manduchi, R., Castano, A., Talukder, A., & Matthies, L. (2005). Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation. *Autonomous Robots*, *18*, 81 – 102.

Marsland, S. (2015). *Machine Learning. An Algorithmic Perspective*. (Second ed.). CRC Press.

Matthies, L., Maimone, M., Johnson, A., Cheng, Y., Willson, R., Villalpando, C., Goldberg, S., & Huertas, A. (2007). Computer Vision on Mars. *International Journal of Computer Vision*, *75*, 67–92.

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press, USA.

Ordonez, C., Gupta, N., Reese, B., Seegmiller, N., Kelly, A., & Collins, E. (2017). Learning of skid-steered kinematic and dynamic models for motion planning. *Robotics and Autonomous Systems*, *95*, 207 – 221.

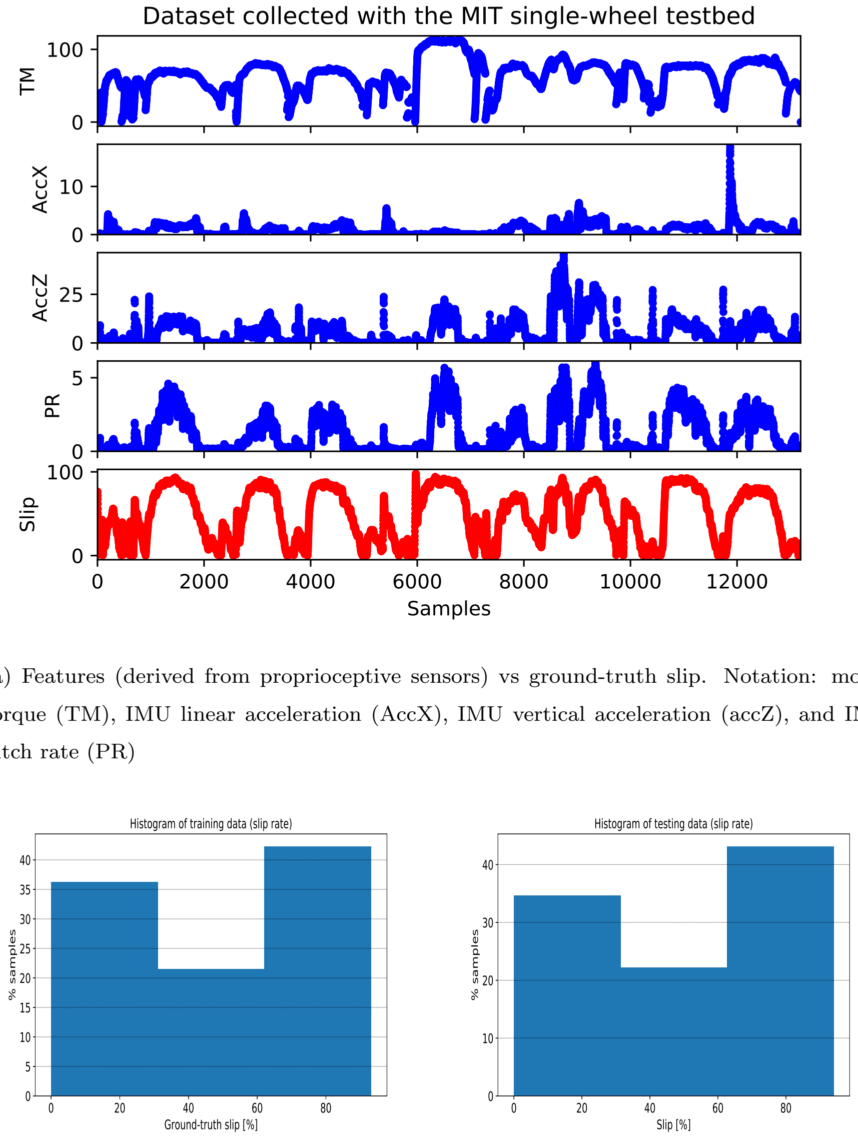
Rasmussen, C., & Williams, K. (2006). *Gaussian Processes for machine learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press.

Smola, A., & Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*, 199 – 222.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, Berlin, Germany.

Webster, G., & Brown, D. (). Now a Stationary Research Platform. NASA's Mars rover Spirit starts a new chapter in Red Planet Scientific Studies.

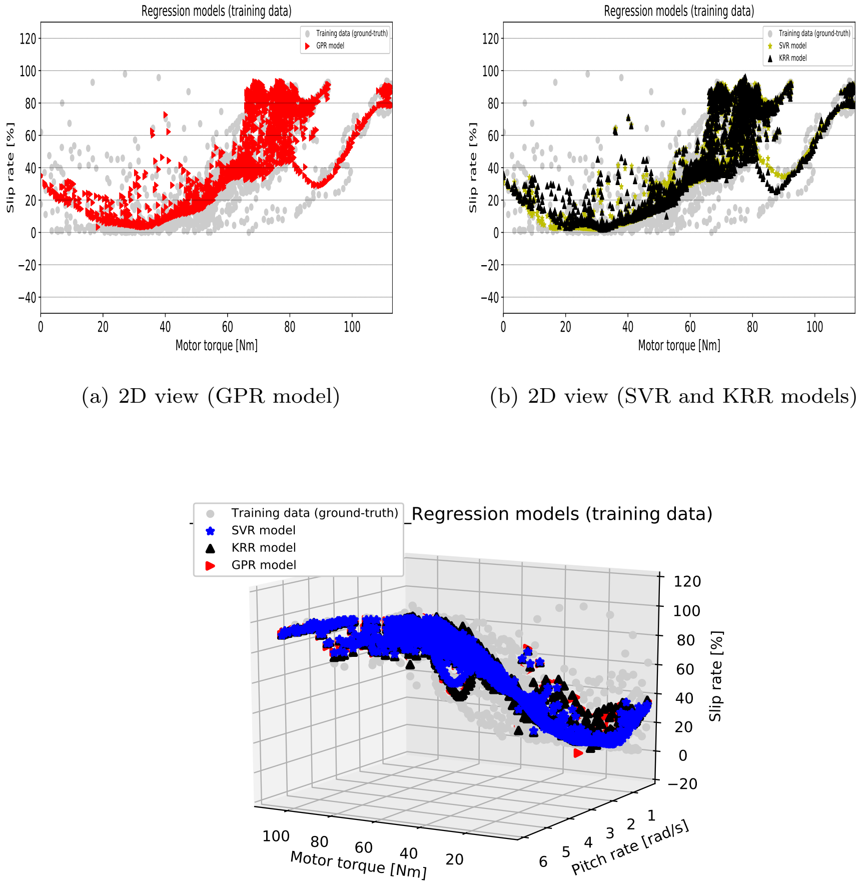
Wong, J. (2001). *Theory of Ground Vehicles*. (Third ed.). John Wiley & Sons, Inc., USA.



(a) Features (derived from proprioceptive sensors) vs ground-truth slip. Notation: motor torque (TM), IMU linear acceleration (AccX), IMU vertical acceleration (accZ), and IMU pitch rate (PR)

(b) Histogram of the samples used for training (c) Histogram of the samples used for testing

Figure 3: Dataset used for validating the proposed methodology. Notice that the leave-one-out cross-validation strategy has been used for randomly splitting the training and testing datasets (70 % training, 30 % testing). The motor torque is given in $[Nm]$. The other variables are given in terms of the variance with respect to group of signals given in $[m/s^2]$, $[m/s^2]$, and $[rad/s]$, respectively (see next section)

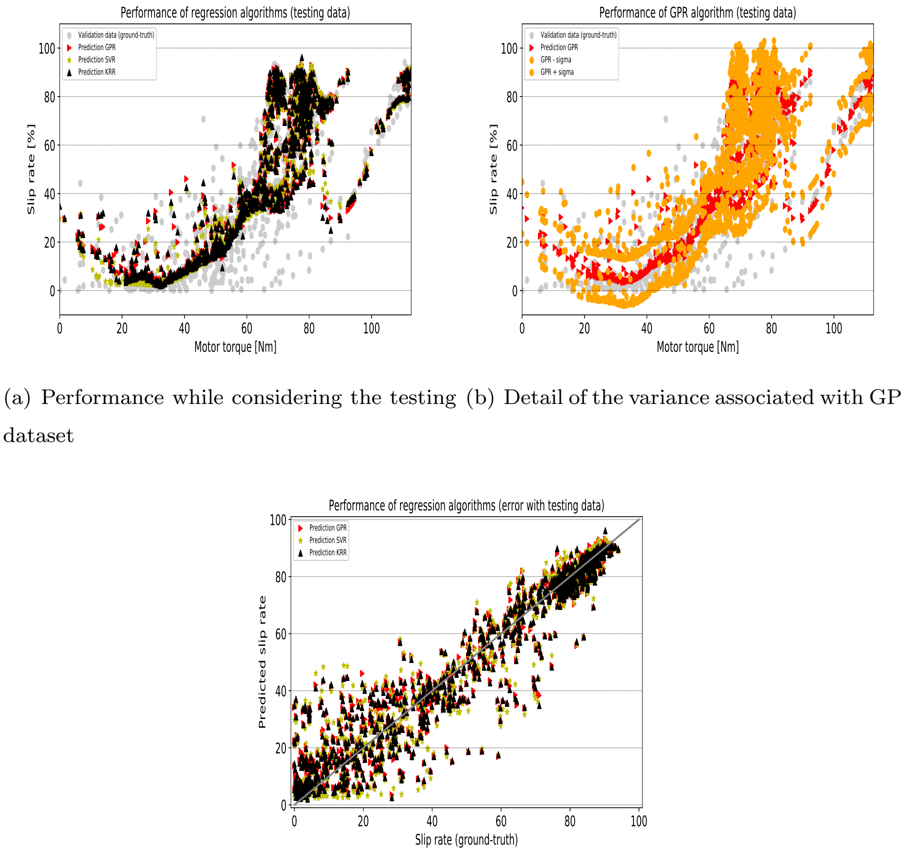


(a) 2D view (GPR model)

(b) 2D view (SVR and KRR models)

(c) 3D view (motor torque, pitch rate, and ground-truth slip)

Figure 4: Models obtained after the training step (GPR, SVR, KRR). For a better view of this figure, see the color version available online



(a) Performance while considering the testing dataset

(b) Detail of the variance associated with GPR

(c) Mean absolute error (predicted values versus ground-truth)

Figure 5: Performance of the machine learning regression algorithms (GPR, SVR, KRR)

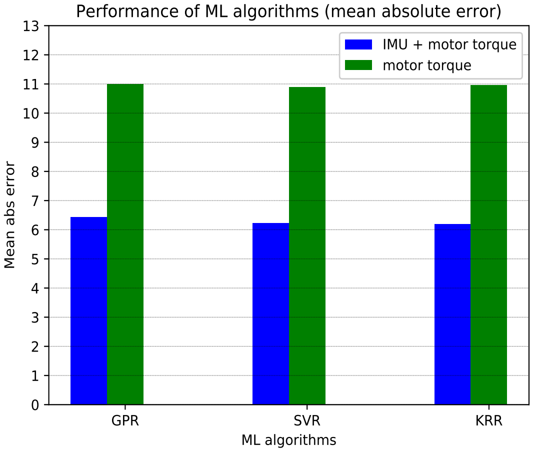
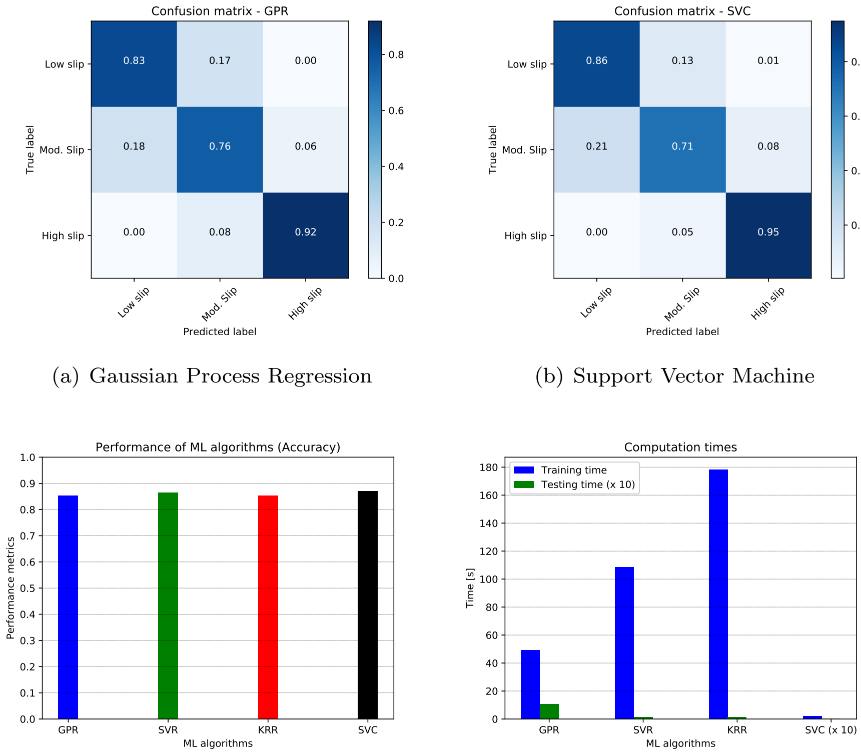


Figure 6: Mean absolute error in terms of the IMU-related and torque-related features and the torque-related features



(a) Gaussian Process Regression

(b) Support Vector Machine

(c) Accuracy

(d) Computation times

Figure 7: Performance of regression and classification algorithms