IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 18, 2018, accepted February 12, 2018, date of publication March 26, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808538

Overcoming the Loss of Performance in Unmanned Ground Vehicles Due to the Terrain Variability

JAVIER PRADO[1], FRANCISCO YANDUN[1], MIGUEL TORRES TORRITI[2], AND FERNANDO AUAT CHEEIN[1]

[1]Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaiso 2340000, Chile
[2]Department of Electrical Engineering, Pontificia Universidad Catolica de Chile, Santiago 7820436, Chile

Corresponding author: Fernando Auat Cheein (fernando.auat@usm.cl)

This work was supported by the National Commission for Science and Technology Research of Chile (Conicyt) under Grant Fondecyt 1171431 and Grant Basal FB0008, CONICYT-PCHA/Doctorado Nacional/2015-21151095.

ABSTRACT Performance in autonomous driven vehicles is susceptible of degradation when traversing different terrains, thus needing motion controllers to be tuned for different terrain profiles. Such tuning stage is a time consuming process for the programmer or operator, and it is often based on intuition or heuristic approaches, and once tuned, the performance of the vehicle varies according to the terrain nature. In this context, we provide a visual based approach to identify terrain variability and its transitions, while observing and learning the performance of the vehicle using machine learning techniques. Based on the identified terrain and the knowledge regarding the performance of the vehicle, our system self-tunes the motion controller, in real time, to enhance its performance. In particular, the trajectory tracking errors are reduced, the control input effort is decreased, and the effects of the wheel-terrain interaction are mitigated preserving the system robustness. The tests were carried out by simulation and experimentation using a robotized commercial platform. Finally, implementation details and results are included in this paper, showing an enhancement in the motion performance up to 92.4% when the highest accuracy of the terrain classifier was 84.3%.

INDEX TERMS Motion controller, computer vision, terrain identification.

I. INTRODUCTION

In industrial scenarios, the performance of motion controllers in robotic vehicles is affected by the nature of the terrain and its changes, being also susceptible to slippage and skidding situations. The development of motion controllers for wheeled mobile robots usually ensure optimal (or acceptable) performance only under nominal conditions (neglecting ground-wheel contact), where the kinematic/dynamic constraints imposed on pure rolling motion of the vehicle are strongly satisfied [1]–[4]. Other solutions attempt to avoid certain navigation surfaces by deviating the vehicle from the trajectory to be tracked [5], or re-scheming the reference route [6], [7]. However, in real-life robotic tasks, such is the case of mining or agricultural applications, some of these considerations are hardly met due to the complexity in the vehicle dynamics, large variability of the terrain and constrained workspaces [8]–[11].

Several solutions for the trajectory tracking problem in ground robotic vehicles interpret the terra-mechanical effects as disturbances [12], [13], causing con trol performance degradation in the sense of decreasing the motion accuracy as well as increasing the control input effort of the actuators [14], [15]. In addition, Klancar and Krjanc [16] provide a robust motion controller based on model, in which the control policy against disturbances was obtained by minimizing a cost function subject to speed and acceleration constraints, with the aim of avoiding slipping. Similarly, in [17], predictive controllers in cascade configuration provided capabilities of controlling speed, yaw-rate and side-slip angle for low and then extended to high vehicle's speeds. In addition, an acceptable performance of the robot and a significant reduction of the effects of disturbances were achieved in [18]–[20], but further improvements can be reached if not only relying on the inherent robustness of the feedback loop.

Industrial tasks can be characterized by models of high repeatability, thus forming systematic circumstances under which original control strategies can be made more robust against uncertainties. For instance, in [21]–[23], the control design allowed to improve the vehicle performance through predicted control actions where additive bounded uncertainties are estimated using recurrent learning processes, with the aim of reducing real and analytical model discrepancies. On the other hand, in [24], the controllers are provided with adaptation capabilities through changing models in order to compensate repetitive variations in a family of linear systems.

The behaviour of a motion controller may be conditioned by the vehicle's model and its operation conditions. A fixed control structure –or constant control parameters– does not properly solve the tracking problem without an impact on the vehicle's performance [25], [26]. In this context, the use of adaptive control techniques seems to be appropriate to re-adjust motion controllers subject to the multiple variables and conditions, as reported by [27]–[29]. For example, [30] employs an adaptive steering control strategy to compensate undue heading oscillations –caused by overload and terrain conditions– by means of time-varying proportional controllers to enhance the tracking accuracy of an automated farm tractor. In [31], it is shown that motion controllers can guarantee acceptable performance if an appropriate set of gains is scheduled to avoid large longitudinal and lateral tracking errors regardless of the vehicle's model.

As long as more details are considered in the vehicle model dynamics, the complexity of the control system might increase and a further model analysis would be necessary. Such as in the case shown in [32], where a speed control system for large-scale vehicles is described by a high non-linear model and decomposed in local linear models in order to switch the controller among different operating regimes. The robust control scheme based on gain scheduling is designed to track a full speed range and responds to disturbances characterized by terrain slopes and rolling resistance. On the other hand, [33] proposes a motion controller for speed regulation through an adaptive throttle and break control system with variable model parameters. Smooth changes in the control parameters experiences non-abrupt manoeuvring unless external disturbances make the vehicle deviates from its reference. In [34], a disturbance rejection algorithm detects slip occurrence with a vision system in order to re-orienting an autonomous excavator to a desired position. In this scenario, if certain terrain characteristics are identified to determine the type of navigation surface (e.g., texture and variability), it becomes possible to improve the performance of the motion controller by changing the controller's parameters.

One of the issues still to be solved lies in the reliability of autonomous navigation systems for ground robotic vehicles subject to terrain constraints, where the robot's resources may be compromised due to effects of the wheel-terrain interaction. A suited compensation of these effects would allow to a more accurate trajectory tracking system with a reduced energy requirement, which is crucial in the industrial field. If one considers different navigation surfaces under which the vehicle travels, the performance of the motion controller could be adjusted to those terrain conditions by means of variable controller parameters.

The main contribution of this paper lies in finding the most suitable set of gains for trajectory tracking controllers using probabilistic approaches. Such approaches allow to self-tune the system and thus to improve performance through the automatic change of gains depending on the type of terrain on which a vehicle navigates. The search for the most suitable set of gains is based on the minimization of a combined cost criteria concerning trajectory tracking errors, control input effort of the controllers and eventual terra-mechanical effects. The set of gains are obtained per terrain type during initial navigation scenarios, and then tested when terrain transitions appear. The type of navigation surface is detected using a low cost time of flight (ToF) camera which provides infrared (IR), depth and color information. These data sources are employed to detect five types of common field terrains, namely: grass, silty, stony, plowed and pavement.

Then, we test and assess available probabilistic approaches generally employed in the machine learning field, but here used to estimate the best set of gains for the trajectory tracking controllers. We cover the problem from two probabilistic points of view: implementing an Expectation Maximization approach to find the tuning parameters, and using a Gaussian Mean Shifting approach. The two methods differ in the handling of data: the first considers a probabilistic distribution for all estimates of the controller parameters, i.e., the controller parameters are treated as random variables with uniform probability distribution; the second approach identifies the behaviour of the controller parameters after several iterations to formulate the best estimate, i.e., the controller parameters are also treated as random variables but evaluated in a mobile range. Moreover, the automatic tuning of the controller parameters is based on two key points: (i) visual identification of terrain characteristics, i.e., variability and texture; (ii) dynamic variation of the controller parameters using an occurrence matrix and stability criterion.

The presented approach is implemented and tested on three motion controllers chosen from the scientific literature: [35]–[37], although others can be used instead. Thus, for each detected terrain type, our method is able to select the optimal set of gains which allow to improve the efficiency of the motion controllers in terms of the pre-defined assessment metrics. Design considerations included in the paper are followed by numerical simulations and results from extensive experimental trials on real agricultural environments using a robotised ground vehicle.

The paper is organized as follows: Section II includes a brief description of the system and its methodology. Section III explains the algorithm of terrain type classification. Section IV shows the implemented trajectory tracking controllers, metrics used to evaluate the performance of these controllers and a detailed description about the methodology of controller parameters selection. Section V shows simulation results. Section VI describes the initial considerations to carry out the trials, and it discusses the achieved results. The paper ends in Section VII with the conclusions of our work.

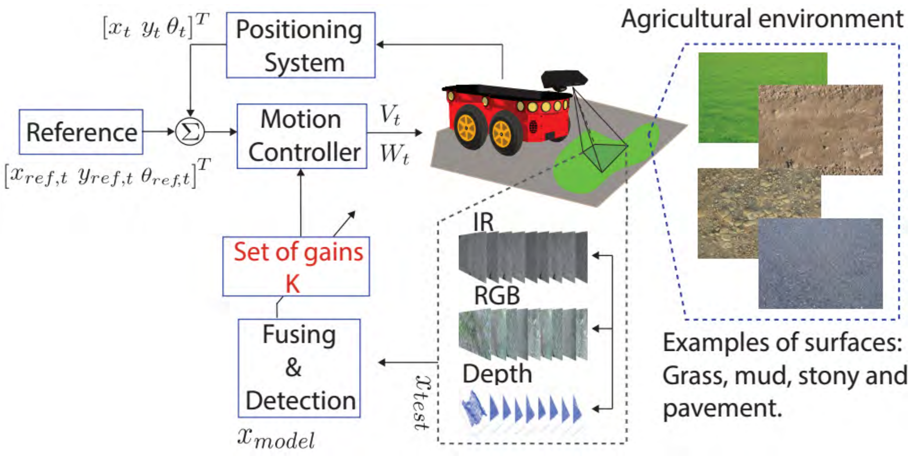


FIGURE 1. Layout of the robotic system architecture.

II. SYSTEM OVERVIEW

The scheme of the proposed solution is depicted in Fig. 3. The system is composed by an unmanned ground vehicle and a remote control point (both systems are connected through a wireless link), thus the solution has two processing units: an on-board computer capable of driving the positioning system, the control system and the terrain surface detection (standard PC with an Intel Core i5-5200U, 2.2GHz processor and 2GB RAM of memory), whereas the second unit (external) is used to switch two operation modes: manual and automatic (mini PC with Intel Atom N550 , 1.5GHz processor and 2GB RAM of memory). The manual mode consists in driving the vehicle towards a close initial position of the reference trajectory, whereas the automatic mode enables the vehicle to navigate autonomously along the trajectory. In addition, a switching function allows to recover the vehicle from automatic to manual mode for repeated operation manoeuvres.

The robot is adapted and equipped with a set of sensors at the Advance Center of Electrical and Electronic Engineering (AC3E), specifically, in the Industrial and Autonomous Robotics Research Group, from Federico Santa María Technical University, Chile. The system includes positioning sensors such as GPS with an incorporated IMU from factory (from Vector Navigation VN-2000), which is mounted aligning with the longitudinal axis of the robot. Also, a LiDAR sensor (from Hokuyo UTM-30LX) was mounted in front of the vehicle to acquire range information, and inner encoders to strength the localization system. Odometry localization information constitutes the main responsible of positioning the robotised vehicle. Regarding to the surface detection system, it relies on IR, color and depth information acquired from a Kinect for Windows $V2^{TM}$ (from Microsoft Corporation, USA) device. According to data provided by the manufacturer, the color camera has a resolution of 1920 × 1080 pixels. The IR camera (used also to estimate depth) has a resolution of 512 × 424 pixels with a field of view of 70.6 × 60 degrees and a lateral and longitudinal view range of 1 meter. The sensor was mounted on the robot pointing forward and down, with a pitch inclination of −36 degrees, as shown in Fig. 3. This configuration allows to obtain a view of the terrain in a range of 0.15 to 0.9 meters in front of the robot chassis avoiding direct sun light. Additionally, the system is supplied with a 12v battery bank and a power inverter connected to the Kinect sensor.

III. TERRAIN SURFACE DETECTION

To recognize the type of navigation surface where the vehicle traversed, we employed a supervised learning technique based on a number of exemplar frames for each surface class. In order to provide variety, these frames were acquired under different illumination conditions: with sunlight, partially and fully shadowed. Our detection system is capable of recognizing grass, silty, stony, plowed and pavement. Examples of training images used in this work are shown in Fig. 2.

The implemented methodology was first evaluated offline, and then incorporated to the control system. For the testing process, data was acquired from several trials in an agricultural field that contained the five classes under study. Later, each class was labelled using our algorithms, and the overall accuracy was calculated. To achieve this aim, each frame was handily labelled to provide the ground truth of the experiments. Once validated, the detection system was capable of providing the identified labels corresponding to the detected soil surface in such a way the control system switches its gains under request for each sample time. It is noteworthy that data acquisition and processing was implemented in C++ and Matlab (MathWorks, USA), in a shared memory framework. The detecting stage can be divided in three steps: raw data pre-processing, feature extraction, and classification as described below.

A. RAW DATA PRE-PROCESSING

Incoming data from Kinect V2 needed normalization, de-noising and fitting procedures to apply the feature extraction and classification, addressed as follows:

- The sensor was not placed parallel to the ground, obtaining a rotated depth measurement. This distortion was

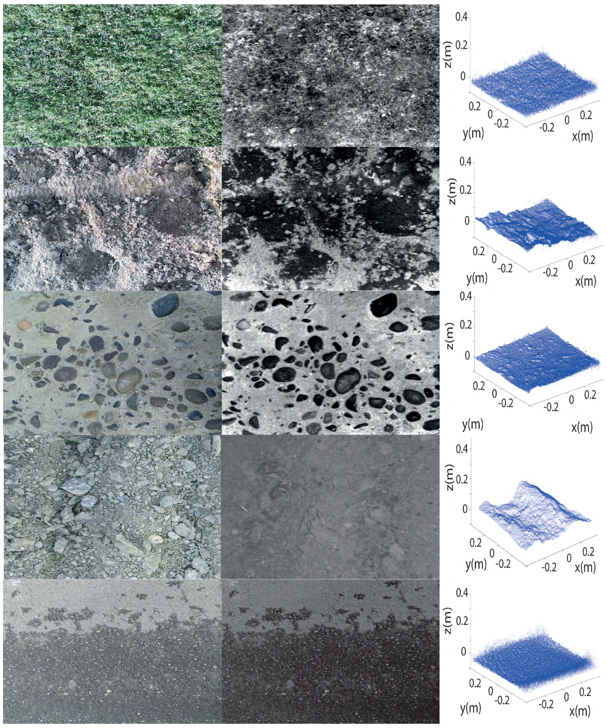


FIGURE 2. Examples of terrain classes detected in this work. From top to bottom: grass, silty, stony, plowed and pavement. The first and second columns show RGB and false color IR images. The third column depicts the point cloud used to obtain depth information.

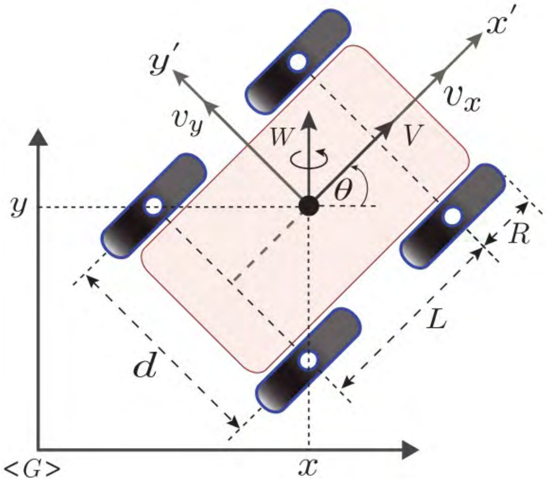


FIGURE 3. Graphical description of the kinematic model used in this work.

first corrected by using the orientation of the IMU sensor. In addition, depth measurements of flat surfaces from the Kinect kept a slightly distortion (caused by the modulation process in the measurement), which was characterized and subsequently removed following the guidelines presented in [38].

- Color and IR cameras have different resolution and field of view. Thus, RGB, IR, and depth images were registered using the intrinsic parameters of the cameras and the focal distance among them. This step allowed to obtain a coloured point cloud of the terrain.
- Some points of the IR and depth images were noisy measurements, specially when working under direct sunlight. In this scenario, pixels beyond a threshold were removed and replaced with an average intensity value of their neighbours.

B. FEATURE EXTRACTION

The terrain classification relies on characterizing the texture of the acquired IR images using the approach proposed by [39]. This technique consists on convolving the image with a filter bank (constituted of N filters), in order to produce a series of filter responses (i.e., an N-dimensional vector for each pixel). Subsequently, c exemplar filter responses (i.e., textons) for each class are obtained using a standard clustering algorithm. Once all training images have been processed, these textons are grouped in order to obtain a texton dictionary (TD), which is used to create the learning models. Following, each pixel filter response is labelled with the index of the closest texton in the dictionary, which allows to create a frequency histogram for each training image. Finally, these histograms correspond to the features used to train the classifier. In order to assign a label ℓ, these features are also obtained for the test images, and used as inputs to the supervised classifier previously trained.

In this work, we used a MR8 filter bank, which consists of a set of 38 filters including: a Gaussian, a Laplace of Gaussian, an edge and bar filters. The last two at 3 scales and 6 orientations per scale. However, only 8 filter responses are obtained by keeping the maximum value of the edge and bar filters across the orientations (i.e., an 8 dimensional vector per pixel). According to [40], this operation allows to obtain rotational invariance. In order to create the TD, we used the well known supervising technique K-means ([41]), with $c = 15$ clusters per class, obtaining a total of 75 textons in the dictionary. Later, the histograms were obtained using the Euclidean distance as metric to measure the closeness between each element of the TD and a pixel filter response. The histograms were normalized in order to compare images of different sizes, if necessary.

C. CLASSIFICATION

The K-nearest neighbour algorithm was implemented for this stage. It works by comparing the test histogram x_{test} with the models x_{model} and assigning to the test histogram the label of the N_n nearest neighbours. Hence, we compared the similarity of histograms by using the Chi-squared statistic [42], defined in Eq. 1.

$$\chi^2 = 0.5 \sum_{m=1}^{M} \frac{[x_{model}(m) - x_{test}(m)]^2}{x_{model}(m) + x_{test}(m)} \tag{1}$$

where x_{model} and x_{test} are the model and testing histograms, respectively; M is the length of the TD (i.e., the number of bins of the histograms), and m represents a single bin. The value of the N_n parameter of the classifier used in this work is five, which provided the best detection performance in preliminary tests.

Additionally, because of low reliability in color information for the presence of variable lighting conditions, this feature was only used as complementary cue for this classifier. Depth information was used in the same sense, since it allowed to characterize only one class. In the training process, the distinctive green and standard deviation of grass and plowed soil, respectively, were used to characterize these classes and to stablish thresholds for these two parameters. Later, in the classification stage, the result of the χ^2 metric was penalized when such predefined thresholds were exceeded. This heuristic method was used to increase the classification accuracy.

IV. MOTION STRATEGIES

The motion model associated with the robotic vehicle kinematics corresponds to the unicycle configuration, shown in Eqs. 2 and 3 and defined as follows:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \frac{\Delta_t R}{2} \begin{bmatrix} \left(W_{r,t} + W_{l,t}\right) cos\left(\theta_t\right) \\ \left(W_{r,t} + W_{l,t}\right) sin\left(\theta_t\right) \\ 2\left(W_{r,t} - W_{l,t}\right)/d \end{bmatrix} \tag{2}$$

$$W_{r,t} = \frac{2V_t + W_t L}{2R}, \quad W_{l,t} = \frac{2V_t - W_t L}{2R} \tag{3}$$

where the robot control commands are the traction velocity V_t and rotational velocity W_t. Each pair of side wheels are commanded by the right and left rotational velocities $W_{r,t}$ and $W_{l,t}$, respectively. R is the effective wheel radius (distance from the wheel axle to the contact point of the surface), L is the distance between the front and rear wheel axle, and d is the azimuth length among wheels. The motion controllers implemented in this brief, which are compatible with the robotic vehicle kinematics, are based on a closed-loop control system where a reference trajectory Ω is previously defined, and the vehicle's pose $[x_t\ y_t\ \theta_t]^T$ is estimated at each sample time Δ_t by the localization system. Although our methodology can be applied to several other motion controllers, we selected three controllers reported in the literature to validate our hypothesis: (i) the first controller is based on algebraic approaches, [35] (C1); (ii) the second controller uses performance criteria, [36] (C2); (iii) the third controller employs Lyapunov formalisms, [37] (C3). Table 1 describes the formulation of the controllers. These controllers are framed in a fixed control structure with a set of gains as tuning parameters. The set of gains $\mathbf{K} = [k_x\ k_y\ k_\theta]^T$ of the first motion controller is bounded in a close interval where it is ensured the stability of the controller (i.e., $k_x,\ k_y,\ k_\theta\ \in\ [0\ 1]$), whereas the second and third controller only require positive controller parameters. These controllers use the incremental position and the trajectory tracking errors (i.e., $[\Delta_x\ \Delta_y]$ and $[\Delta x_e\ \Delta y_e]$ for C1 and C2-C3 respectively) in order to obtain the control inputs. The criteria presented here selects the best set of gains $\hat{\mathbf{K}}$ based on assessment metrics in which a combined cost function is minimized subject to kinematic constrains. More details about implementation and description issues for these controllers can be found in [35]–[37], whereas the selection methods of the best set of gains and performance metrics are described following.

TABLE 1. Trajectory tracking controllers.

Controller	Formulation
Auat Cheein & Scaglia. –C1:	$\begin{cases} V_t = \frac{1}{\Delta_t}(\Delta_x \cos\theta_{ez,t} + \Delta_y \sin\theta_{ez,t}) \\ W_t = \left(\frac{\theta_{ez,t+1} - k_\theta(\theta_{ez,t} - \theta_t) - \theta_t}{\Delta_t}\right) \\ with: \\ \Delta_x = x_{ref,t+1} - k_x(x_{ref,t} - x_t) - x_t \\ \Delta_y = y_{ref,t+1} - k_y(y_{ref,t} - y_t) - y_t \\ 0 < k_x,\ k_y,\ k_\theta < 1 \end{cases}$
Yi Guo et al. –C2:	$\begin{cases} V_t = k_x\xi(\omega_r, v_r)\Delta x_e + v_r cos(\theta_e) \\ W_t = \omega_r + \xi(\omega_r, v_r)v_r \frac{sin(\theta_t)}{\theta_t}\Delta y_e + k_\theta\theta_e \\ \xi(\omega_r, v_r) = 2k_y\sqrt{\omega_r^2 + k_\theta v_r^2} \\ \Delta x_e = e_x cos(\theta_t) - e_y sin(\theta_t) \\ \Delta y_e = -e_x sin(\theta_t) + e_y cos(\theta_t) \\ e_x = (x_{ref,t} - x_{e,t}), e_y = (y_{ref,t} - y_{e,t}) \\ with: \\ k_x,\ k_y,\ k_\theta > 0 \end{cases}$
Kanayama et al. –C3:	$\begin{cases} V_t = v_r \cos\theta_e + k_x\Delta x_e \\ W_t = \omega_r + v_r(k_y\Delta y_e + k_\theta \sin\theta_e) \\ \Delta x_e = e_x cos(\theta_t) - e_y sin(\theta_t) \\ \Delta y_e = -e_x sin(\theta_t) + e_y cos(\theta_t) \\ e_x = (x_{ref,t} - x_{e,t}), e_y = (y_{ref,t} - y_{e,t}) \\ with: \\ k_x,\ k_y,\ k_\theta > 0 \end{cases}$

A. ASSESSMENT METRICS

To evaluate the performance of the trajectory tracking controllers under different gain settings and to obtain such gains, we used the metrics suggested in [43]. Briefly:

- The first metric is the *cumulative tracking error* ($C^{\Omega}_{x,y}$) which represents the total squared errors between the given trajectory waypoints $\#\Omega$ and the vehicle's position $[x_t\ y_t]^T$. This metric also describes a degree of accuracy in tracking the reference trajectory $\Omega = [x_{ref,t}\ y_{ref,t}]^T$, as shown in Eq. 4.
- The second metric corresponds to the *cumulative control effort* ($C^{\Omega}_{V,W}$). It is associated with the amount of kinematic energy employed by the robot while following a pre-defined trajectory, this can be seen in Eq. 5.

$$C^{\Omega}_{x,y} = \rho \sum_{t=0}^{\#\Omega} (x_{ref,t} - x_t)^2 + (y_{ref,t} - y_t)^2 \tag{4}$$

$$C^{\Omega}_{V,W} = \gamma \sum_{t=0}^{\#\Omega} V_t^2 + \eta \sum_{t=0}^{\#\Omega} W_t^2 \tag{5}$$

where the parameters ρ, γ and η are positive values to provide priority in each cost function and to give a consistent sum among the two metrics.

- Finally, the *cumulative total cost* assesses the performance of trajectory tracking controllers, defined as the sum of the metrics above mentioned, see Eq. 6.

$$C^{\Omega}_{Tot} = C^{\Omega}_{x,y} + C^{\Omega}_{V,W} \tag{6}$$

The wheel-terrain effects can be associated with the slip ratio s (with $s = (v_{x,t} - \omega_{r,t}R)/v_{x,t},\ v_{x,t} \neq 0$), which provides an idea of how much speed has decreased due to traction loss in each wheel. Also, during cornering or lateral shifting manoeuvres, side motion effects can be identified by the side slip angle β (with $\tan\beta_t = v_{y,t}/v_{x,t},\ v_{x,t} \neq 0$). After the estimation of these slip parameters, one can use them as motion constraints within the minimization of the *cumulative total cost* to obtain the best set of gains $\hat{\mathbf{K}}$ as shown in Eq. 7.

$$\begin{aligned} & \underset{\mathbf{K}\in R^{n}}{minimize}\ C^{\Omega}_{Tot} \\ & subject\ to\ s^{min} \leq s^{o}_{t} \leq s^{max}, \\ & \qquad \beta^{min} \leq \beta^{o}_{t} \leq \beta^{max}, \\ & \qquad 0 \leq v_{x,t} \leq s^{o}_{t} v^{max}_{x} + \omega_{r,t}R, \\ & \qquad 0 \leq v_{y,t} \leq v^{max}_{x} \tan\beta^{o}_{t} + \lambda\omega_{r,t}R. \end{aligned} \tag{7}$$

where n corresponds to the dimension of $\mathbf{K}$, $v_{x,t}$ and $v_{y,t}$ are the components of the linear vehicle speed. s^{o}_{t} and β^{o}_{t} are the slip ratio and side slip angle within safety ranges so that the vehicle does not slip, i.e., $[s^{min}, s^{max}]$ and $[\beta^{min}, \beta^{max}]$. In addition, $\omega_{r,t}$ is the angular wheel speed, λ is the relationship width-length of the vehicle.

B. DETERMINING CONTROLLER PARAMETERS

The self-tuning methodology relies on the purpose of obtaining dynamically the best set of gains $\hat{\mathbf{K}}$ in a bounded range where the stability of the closed-loop system is not affected. Two steps are carried out before implementing the automatic selection of the controller parameters. The first consists on generating set of gains that minimize the cost function previously defined in the Section IV-A, named hereafter as Monte Carlo Seeds E. The second step concerns clustering the total set of gains in order to select the best controller parameter for each navigation surface since certain set of gains are more suitable for one terrain type than others. This step is carried out by using two approaches: Expectation-Maximization and Gaussian Mean Shifting, as clustering and learning strategies. The best set of gains obtained after the last two steps are applied to the controllers depending on the navigation surface, and the switching rate of the gains are identified in such a way that the motion controller stability is not compromised, as will be shown in SectionIV-C.

1) MONTE CARLO SEED GENERATION

An E number of set of gains $\mathbf{K}$ are generated pseudo-randomly in a bounded range where the stability of the controller is ensured at each time step. For our proposal, the boundaries are defined as presented in Section IV and the selection criterion is depicted in Algorithm 1. Briefly:

Algorithm 1 Generation of Monte Carlo Seeds

```
1:  Let $E$ be the maximum number of Monte Carlo seeds.
2:  Let $\mathbf{K^0}$ be the initial set of gains of the trajectory tracking controllers in $\Pi$, where $\Pi$ is the domain of $\mathbf{K}$. $\mathbf{K^0}$ is uniformly distributed between $\mathbf{K_{min}}$ and $\mathbf{K_{max}}$.
3:  Let $C^{0}_{V,W}$ and $C^{0}_{x,y}$ be the maximum control effort and error cost, originally initialized at very high values. Thus, obtain $C^{0}_{Tot}$.
4:  Let $[x_t\ \ y_t]^T$ be the robot's current position and $[x_{ref,t+1}\ \ y_{ref,t+1}]^T \in \Omega$ the desired position at time instant $t+1$.
5:  Let $v_t = [v_{x,t}\ \ v_{y,t}]^T$ and $\omega_{r,t}$ be the measurements of the vehicle linear speed and angular wheel speed, respectively.
6:  if Speed constraints are fulfilled according to Eq.7 then
7:      for $j = 1$ to $E$ do
8:          Generate random parameters $\mathbf{K^j}$ following Gaussian distributions within the stable range of the controllers.
9:          Calculate the total error cost $C^{j}_{Tot}$ according to $\mathbf{K^j}$, $[x_t\ y_t]^T$, $[x_{ref,t+1}\ y_{ref,t+1}]^T$ and $[V_t\ W_t]^T$ using the trajectory tracking controllers shown in Table 1.
10:         if $C^{j}_{Tot} < C^{j-1}_{Tot} \wedge (V_t < V^{max} \wedge W_t < W^{max})$ then
11:             Save $C^{j}_{Tot}$
12:             Save $\mathbf{K}^{j}_{new}$ with $\mathbf{K}^{j}_{new} = \mathbf{K}^{j}$
13:         end if
14:     end for
15: end if
```

- Lines of code (1)-(3) show the initial conditions to generate the Monte Carlo seeds E. The number of seeds can be assumed as a design criterion, which are generated during the robot's motion at each time step. As a result, a number of effective seeds ε are obtained after the selection of the best set of gains until the vehicle reaches the total number of waypoints $\#\Omega$. Note that in line of code (2), each parameter of the controller is considered as a random variable following a Gaussian distribution. In addition, reference trajectory, positioning variables and velocities are considered to be evaluated if the motion constraints are fulfilled according to the proposed slip criteria, see lines of code (4)-(5).
- The *for-loop*, lines of code (7)-(14), is the core of the algorithm. As can be seen in lines of code (8)-(13), the Monte Carlo method selects the set of gains $\mathbf{K_{new}}$ that provide the lowest tracking error and a reduction of the control input effort while the vehicle experiments a minor slip effect. At this point, it is worth mentioning that this method also generates the set of gains that do not allow saturation in the actuators, see line of code (10).

The set of gains $\mathbf{K_{new}}$ found by the Monte Carlo algorithm are used as a training set to create clusters, where the most representative cluster constitutes the region where the best set of gains can be obtained. The criteria to identify such clusters are based on well known unsupervised learning techniques: Expectation Maximization (EM) and Gaussian Mean Shifting (GMS). Each probabilistic approach gives an estimation of the best set of gains $\hat{\mathbf{K}}$ with its corresponding covariance matrix $\mathbf{\Sigma}$. Following, the EM and GMS approaches are explained in detail in the context of our work.

2) EXPECTATION-MAXIMIZATION APPROACH (EM)

This method assumes that each set of gains follows a probabilistic distribution to group the training set in a certain number of clusters, providing an estimation of the optimum set of gains $\hat{\mathbf{K}}$ and a covariance matrix $\mathbf{\Sigma}$ (see [44] and the references therein). This method is based on a maximum a posteriori approach whose formulation can be seen in Eq. 8.

$$\hat{\mathbf{K}} = \underset{\mathbf{K}\in\mathbb{R}^n}{argmax} \left\{ log \prod_{i=1}^{\varepsilon} p(\mathbf{K}^{(i)}|\alpha, \mathbf{\Sigma}, \mu) \right\} \tag{8}$$

where $\mathbf{K}^{(i)}$ is the ith set of gains in the training set. This method accepts several probability distributions, but here the normal distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$ is considered since it is known beforehand by the proposed Monte Carlo method. In addition, the dimension of the Monte Carlo seeds is n and α is a latent membership variable associated with each distribution $p(\mathbf{K}^{(i)}|\alpha, \mathbf{\Sigma}, \mu)$. The procedure to find $\hat{\mathbf{K}}$ and $\mathbf{\Sigma}$ is shown in Algorithm 2. Briefly;

- In lines of code (1)-(5), the initial parameters of the algorithm are set. Among these parameters, the number of clusters c and Monte Carlo seeds are chosen as a design criterion. The latent variables α are initially selected with small values, which represent the latent probability of belonging to a given cluster.
- Expectation: Lines of code (6)-(10) show the procedure to determinate a matrix of weights $\mathbf{\Upsilon}$ used to modify the estimation of the set of gains per iteration.
- Maximization: The optimization problem from Eq. (8) is solved in lines of code (12)-(16) updating a mean μ and the covariance matrix $\mathbf{\Sigma}$ for each cluster. The algorithm stops when the associated log-likelihood converges according to a given threshold, see lines of code (17)-(23).

The estimated set of gains $\hat{\mathbf{K}}_j$ obtained in Algorithm 2 represents the centroids from each cluster, and the covariance matrices $\mathbf{\Sigma_j}$ provides an idea of the regions that enclose such gains. Finally, each set of gain from the training set has a correspondence to each cluster.

Algorithm 2 Expectation-Maximization Approach ([44]–[46])

```
1: Let $c$ be the number of clusters.
2: Let $\varepsilon$ be the number of samples –Monte Carlo seeds.
3: Initialize a matrix of weights $\mathbf{\Upsilon}$ and $\mathbf{\Sigma} \in S^n_{++}$.
4: Select $c$ initial $\mu_j$s $\in \mathbb{R}^n$ from any set of gains obtained from the training set.
5: Let $\alpha_\mathbf{j} \in \mathbb{R}^n$ be the vector of independent and identically distributed latent variables, where $j = 1, ..., c$.
6: for $j = 1$ to $c$ do
7:     for $i = 1$ to $\varepsilon$ do
8:         Calculate the weights $\mathbf{\Upsilon}^i_j$ as the quotient between the Gaussian distribution: $\mathcal{N}(\mathbf{K}^{(i)}, \mu_j, \mathbf{\Sigma}_j)$ and the sum of weighted Gaussian distributions $\sum_{j=1}^{c} \alpha_j \mathcal{N}(\mathbf{K}^{(i)}, \mu_j, \mathbf{\Sigma}_j)$.
9:     end for
10: end for
11: for $j = 1$ to $c$ do
12:     for $i = 1$ to $\varepsilon$ do
13:         Calculate the latent vector $\alpha_\mathbf{j}$ as the weights $\mathbf{\Upsilon}^i_j$ normalized for $\varepsilon$.
14:         Calculate the estimation of the set of gains $\boldsymbol{\mu}_\mathbf{j}$ weighting them with $\mathbf{\Upsilon}^i_j$.
15:         Update the covariance matrix $\mathbf{\Sigma_j}$ with $\sum_{i=1}^{\varepsilon} \mathbf{\Upsilon}^{(i)}_j (\mathbf{K}^{(i)} - \mu_j)(\mathbf{K}^{(i)} - \mu_j)^T$.
16:     end for
17:     if $log \quad l(\mathbf{K}^{(i)}_j, \alpha, \mathbf{\Sigma}, \mu)$ converges $\forall i = 1, ..., \varepsilon$ and $j = 1, ..., c$ then
18:         Assign the final means as: $\hat{\mathbf{K}}_j = \mu_\mathbf{j}, \ \forall j = 1, ..., c$.
19:         Save the covariance matrix $\mathbf{\Sigma}_j, \ \forall j = 1, ..., c$.
20:         Break.
21:     else
22:         Continue with the process.
23:     end if
24: end for
```

3) GAUSSIAN MEAN SHIFTING APPROACH (GMS)

Although this method can use different probability distribution kernels, we considered that the training set follows a Gaussian distribution following Section IV-B1. Additionally, these gains are weighted with the distribution κ and evaluated with a vector of mean-shift $f(\mathbf{K})$ with the aim of introducing a type of dissociation among the set of gains and the cluster's centroids $\mu^{(i)}$ when clustering, see Eqs. 9-10. This method assumes a mobile-window around each gain in the training set, and thus it is shifted in the direction of the gradient where the concentration density of the set of gains in each cluster is the highest. In addition, the gradient of the distribution provides a vector of shifting weight $m(\mathbf{K})$ which points towards the best estimation $\hat{\mathbf{K}}$ within the most probable cluster, see Eqs. (11)-(12).

$$\kappa(\mathcal{D}) = exp\left(-\frac{1}{2}\mathcal{D}\right) \tag{9}$$

$$f(\mathbf{K}) = \frac{\sum_{i=1}^{\varepsilon} \pi_j |2\pi\,\Sigma_i|^{-1/2} \kappa'(\mathcal{D}(K, \mu^{(i)}; \Sigma_i))\Sigma_i^{-1}\mu^{(i)}}{\sum_{i=1}^{\varepsilon} \pi_j |2\pi\,\Sigma_i|^{-1/2} \kappa'(\mathcal{D}(K, \mu^{(i)}; \Sigma_i))\Sigma_i^{-1}} \tag{10}$$

$$m(\mathbf{K}) = \frac{\pi_j \kappa'(\mathcal{D}(\mathbf{K}, \mu^{(i)}; \Sigma_i))}{\sum_{i'=1}^{\varepsilon} \pi_{i'} \kappa'(\mathcal{D}(\mathbf{K}, \mu^{(i)}; \Sigma_i))} \tag{11}$$

$$\mathbf{K} = \sum_{i=1}^{\varepsilon} m(\mathbf{K})\mu^{(i)} \tag{12}$$

where κ is the kernel associated with the probability distribution, $f(\mathbf{K})$ is the function which maps the shifted set of gains $\mathbf{K}$, $m(\mathbf{K})$ is the vector of shifting weight. π is the weight or mixing proportion of the point j, the argument

Algorithm 3 Gaussian Mean Shifting Approach [48]

```
1:  Initialize a matrix $\mathbf{Q} \in S^n_{++}$.
2:  Initialize a matrix $\mathbf{\Sigma_i} \in S^n_{++}$.
3:  Select a set of gains in the training set as a centroid $\mu^{(i)}$.
4:  Select randomly a set of gains $\mathbf{K}$ among the training set.
5:  for $i = 1$ to $\varepsilon$ do
6:      repeat
7:          Calculate the Mahalanobis distance $\mathcal{D}(\cdot)$ between the set of gains $\mathbf{K}$ in the training set and cluster's centroids $\mu^{(i)}$.
8:          Calculate the vector of mean-shift $f(\mathbf{K})$.
9:          Calculate the vector of shifting weight $m(\mathbf{K})$.
10:         Calculate $\mathbf{K}$ with Eq 12.
11:         Calculate $\mathbf{\Phi} \in S^n_{++}$ with $\mathbf{\Phi} = \nabla f(\mathbf{K})$.
12:         Calculate the evolution of the covariance matrix $\mathbf{\Sigma}_{i+1} = \mathbf{\Phi}\mathbf{\Sigma}_i\mathbf{\Phi} + \mathbf{Q}$.
13:     until $\mathbf{K}'s$ update $< \epsilon$ (threshold)
14:     Save $\hat{\mathbf{K}}$ as $\hat{\mathbf{K}} = \mathbf{K_i}$, and $\mathbf{\Sigma} = \mathbf{\Sigma_i}$.
15: end for
16: Verify the connected components based on a minimum distance. For instance, the Mahalanobis distance.
```

$\mathcal{D}$ represents the Mahalanobis distance whose advantage relies on considering the correlation among set of gains, i.e., $\mathcal{D}(\mathbf{K}, \mu^{(i)}; \mathbf{\Sigma}) = (\mathbf{K} - \mu^{(i)})^T \mathbf{\Sigma}_i^{-1}(\mathbf{K} - \mu^{(i)})$. The procedure to obtain the best set of gains $\hat{\mathbf{K}}$ is shown in Algorithm 3. Briefly;

- The heading –lines of code (1)-(4)– resembles to the previous algorithm, but this case considers a process covariance matrix $\mathbf{Q}$. The initial set of gains and an auxiliary centroid $\mu^{(i)}$ are selected from the training set.
- The core of the algorithm relies on lines of code (5)-(15). For each point in the training set, the set of gains are evaluated by a level of membership in a given cluster with the distance $\mathcal{D}$, whereas the means shift with the vector $m(\mathbf{K})$, see lines of code (7)-(9). The temporary estimation $\mathbf{K}$ is updated with the vector $m(\mathbf{K})$ until it converges. The covariance matrix $\mathbf{\Sigma}$ is updated as in the estimation process of the optimal set of gains; however, this matrix is weighted with the gradient of the distribution $f(\mathbf{K})$. The resulting set of gains are compared with the previous estimations of $\mathbf{K}$ through the distance $\mathcal{D}$ in order to check if the estimation process has converged, see line of code (13). The estimations $\hat{\mathbf{K}}$ and $\mathbf{\Sigma}$ are then saved–line of code (14).
- Finally, the process is repeated according to the number of Monte Carlo samples ε, and a last step verifies the connected components of the estimations in order to identify if any estimation is repeated, as shown in line of code (16). More detailed information regarding to this method can be found in [47] and [48].

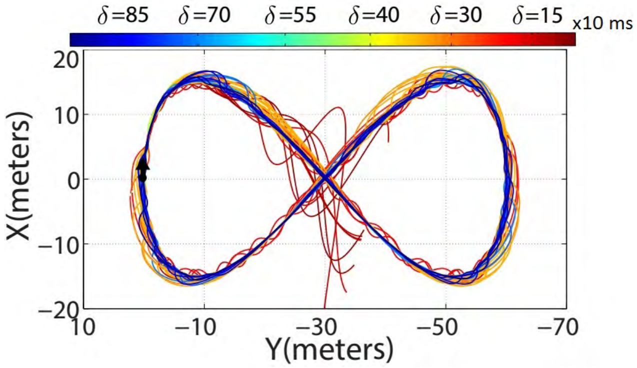


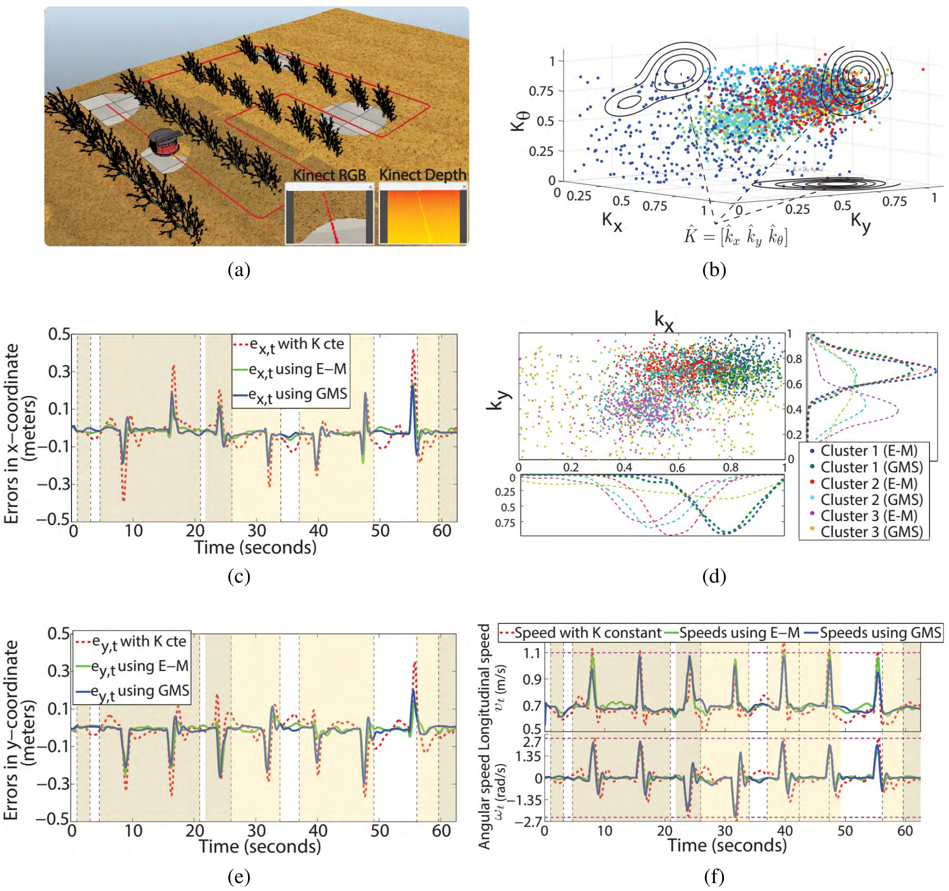
FIGURE 4. Trajectory tracking responses when switching the control parameters between their minimum and maximum values. The color bar represents the range of tests for the different switching times, being red the switching time closer to the instability and blue the safety switching time. As can bee seen, the error starts to grow unbounded while switching faster.

C. REAL-TIME CHANGE OF THE SET OF GAINS

Before the training process, the robotic platform follows pre-defined trajectories with the three trajectory tracking controllers previously described, while a number of Monte Carlo seeds are collected for each terrain type according to Section IV-B1. The training procedure is performed over five types of terrain surfaces: grass, muddy terrain, stonny terrain, plowed soil and pavement. Each training stage is carried out attempting not to match the reference trajectory with transitions among the different types of terrains (i.e., remaining a single type of terrain) in order to obtain the best set of gains $\hat{\mathbf{K}}$ for each terrain type and motion controller. The best set of gains are selected as a result of the highest performance of the motion controllers according to Section IV-A. Additionally, as the vehicle traverses, information from the Kinect sensor (IR, color and depth) is also acquired under the different environmental constrains (e.g. different terrain types, lit and shady scenarios) to obtain the model histograms x_{model} of the different terrain types as described in Section III.

Once the best set of gains $\hat{\mathbf{K}}$ and the histograms x_{model} are estimated after the training process, the parameters of the trajectory tracking controllers are switched among the best set of gains when a new transition of terrain appears according to the terrain surface detection system. Therefore, the best set of gains are applied when estimations of terrain transitions occur. These estimations are based on the velocity of the robot and distance between the focal axis of the Kinect and geometry center of the robot. In addition, we considered that if the control parameters change as fast as a terrain transition is detected or when a false detection is triggered, the stability of the controller may be compromised. For this reason, we studied the minimum and maximum frequency in which the set of gains can change without affecting the stability of the controllers.

The minimum switching time (or maximum switching frequency) is a concerning point for each controller. Since our research focus relies on providing a new methodology to find the best set of gains for motion controllers, the later would require a particular theoretical analysis to define the allowable frequency boundaries to switch the gains and guarantee



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 5. Simulation results. Figure 5a shows the simulation environment for tracking trajectories. In Fig. 5b, the effective seeds are shown in color dots, where the red dots represent the gains with the total cost lower than the depicted in blue. The level curves for each plane represent the area where the probability to find the lowest cost is maximum. In Fig. 5d, a gain projection in the plane $k_x - k_y$ shows the distribution of the gains according to each cluster (three clusters) and selection method (i.e., EM and GMS). The best set of gains are the centroids of the distributions with the the lowest standard deviation. Figures 5c and 5e-5f show tracking errors and speeds when using constant parameters and changing gains for each methodology. The shaded areas represent the identified terrain surface, while their transitions appears in the flanks of vertical dotted lines.

stability on each controller, which is not the aim of our work. However, we prepared an empirical trial to study the stability of the proposed controllers under changes in $\mathbf{K}$, as explained below:

- We defined the bounded gain values that ensure the controller stability, described as $\mathbf{K_{min}}$ and $\mathbf{K_{max}}$.
- The sampling time of the robot and the minimum switching time were equally set to Δ_t.
- We planned an ∞-shaped trajectory kinematically compatible with all implemented controllers.
- For each motion controller, we performed the following test:

(a) We set the switching time $\Delta_{switch} = \delta \times \Delta_t$ (with $\delta = 1$ initially), and changed between $\mathbf{K_{min}}$ and $\mathbf{K_{max}}$ at each switching time while tracking the given trajectory.

(b) We evaluated the trajectory tracking errors and the control effort. If the errors grew unbounded, thus the system was unstable for such Δ_{switch}.

(c) We repeated the experiment increasing $\delta = \delta + 1$, until we found that the tracking errors remained bounded during the throughout trajectory, and therefore the controller was stable to such minimum switching time Δ^{min}_{switch}.

With the aim of finding the minimum switching time in which the set of gains are able to be changed without affecting the system stability, we considered switching all the set of gains simultaneously between its minimum and maximum admissible values, at different switching times. The analysis was carried out by simulation, tracking one hundred curly trajectories (i.e., ∞-shaped trajectories) for each motion controller, and the switching frequency was increased proportionally with the sampling time. After we performed the stability tests, the minimum switching time Δ^{min}_{switch} that ensures stability in the closed-loop system was 800ms, 700ms and 500ms for the controllers C1, C2, and C3, respectively. Thus, we selected the highest switching time (800ms) to change the best set of gains via either simulation or field test.

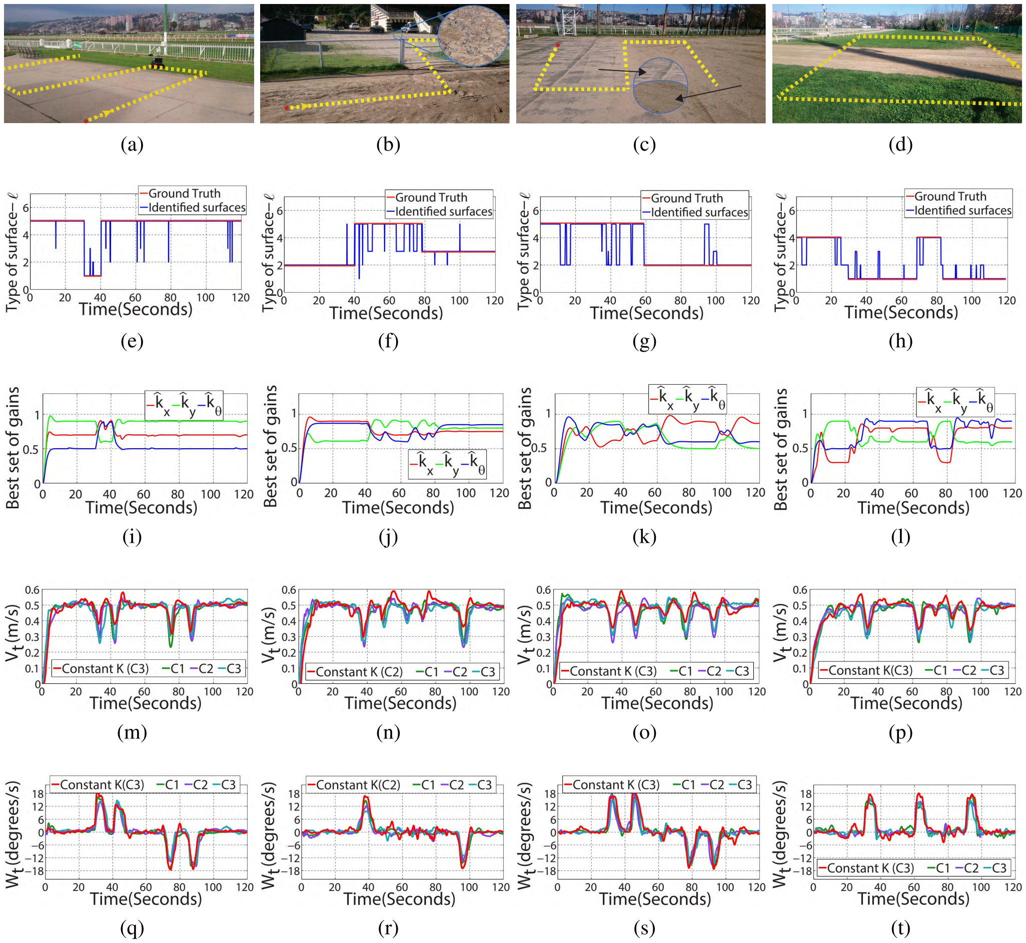


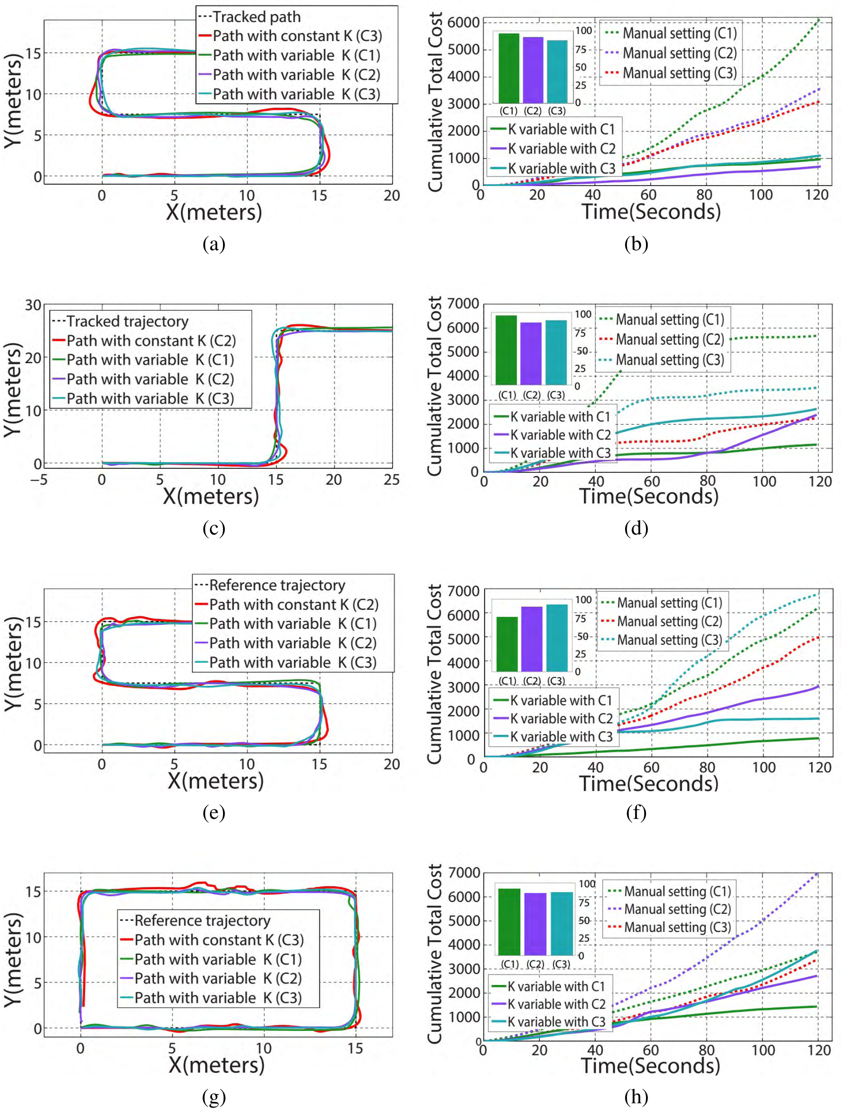
FIGURE 6. Field tests. Each column represents one of the four trials in field. Figures 6a- 6d shows terrain transitions and trajectories used in each experimentation. Figures 6e-6h show an example of the identified terrain types in each trial. The labels ℓ numbered from one to five represent grass, silty terrain, stony terrain, plowed soil and pavement, respectively. In solid red line we show the corresponding terrain type. Figures 6i-6l show the best set of gains applied to the controller obtained as a function of the terrain transitions. Figures 6m-6t show the control input commands when using constant and tuned set of gains for each controller and trial.

Figures 4 shows several responses obtained to test the stability in the controller C3.

Once obtained the minimum switching time in which the set of gains can be applied without compromising the stability of the controllers, the gains are tuned in such a way that time constraints are not violated. To do so, a time delay τ is introduced in the dynamic change of the control parameters under which the best set of gains $\hat{\mathbf{K}} = [\hat{k}_x \; \hat{k}_y \; \hat{k}_\theta]$ corresponding to the new terrain condition are reached within the identified time limit.

V. SIMULATION RESULTS

Before testing in field, simulations trials were carried out using the controller C2 with the best set of gains obtained from the two proposed methodologies (EM and GMS). The simulations were performed using a robotic vehicle subject to terrain constraints in the V-Rep simulation framework linked with Matlab. The given trajectory was approximately 62.5 m long, the distance between any two consecutive waypoints within the trajectory was 0.07 m and the sampling time of the controller was set to 0.1 s. The trajectory shape has squared corners as suggested by [49]. Part of the trajectory sections were matched with the three different terrain types to incorporate a more realistic simulation (i.e., high friction, low friction and default surface according to the V-Rep software). In addition, after tracking the given trajectory, the number of effective Monte Carlo seeds to be employed in EM and GMS was determined to be $\varepsilon = 3125$, whereas the total number of waypoints were $\#\Omega = 625$. The control parameters were bounded within the stable interval $[\mathbf{K_{min}} \; \mathbf{K_{max}}]$, and the set of gains changed according to the minimum and maximum switching time defined in the stability test. Figure 5 shows the simulation environment, clusters of the set of gains, trajectory tracking errors and vehicle speeds. As can bee seen in Figs. 5c and 5e, the longitudinal and lateral tracking errors are smaller when using either two methodologies than the responses obtained with constant control parameters. In particular, this can be seen when the vehicle



(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 7. **Results obtained in the experimental test. From top to bottom, each row shows the results obtained for each trial. The first column represents the trajectory tracking results for each controller using the set of tuned gains in contrast to the manual tuning with the lowest total cost, whereas the second column shows the *cumulative total costs* acquired with the manual setting of the control parameters (dotted lines) and with the proposed approach (solid lines). The accuracy of the classifier is represented by bar plots inside the total cost plots, and their colors are associated with each controller. The red dotted lines represent the lowest cost obtained when using constant set of gains. Note that the accuracy showed in the bar plots is only associated with the performance obtained using K variable.**

has to turn or experiences terrain transitions on the road (i.e., shaded areas). Due to the fact that similar simulation results were obtained for both probabilistic approaches, we only considered the set of gains obtained in the EM approach in the field experimentations.

VI. FIELD RESULTS

To validate our hypothesis, we performed three experimental tests. The first case studies the response of our methodology to different trajectories and terrains due to wheel-terrain interaction using the visual terrain identification; the second case deals with the level of reaction of the set of gains according to different speed profiles; and the third case verifies the consistence of the methodology in systematic tests.

A. FIRST TEST

Before evaluating our approach, some implementation issues regarding the terrain detection method and motion controllers required additional consideration. Briefly;

- To avoid spurious readings in case of the IR camera get saturated, depth and IR measurements beyond a threshold of 95% and 80% of their maximum values are set to be zero, respectively.

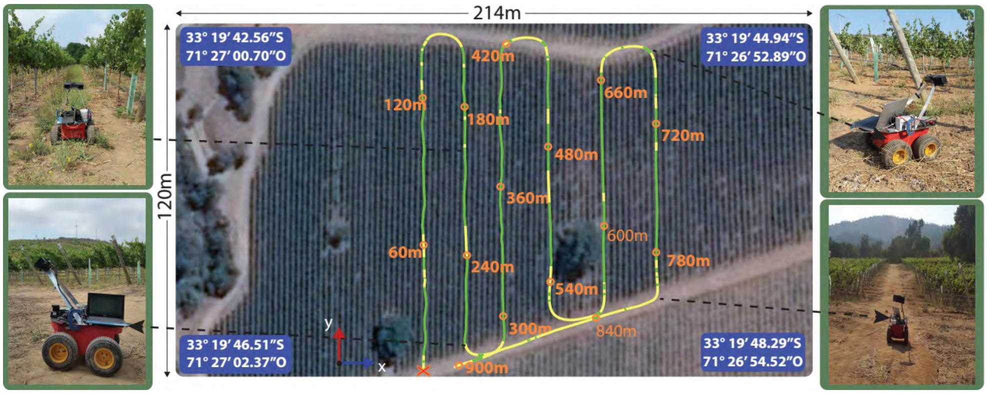
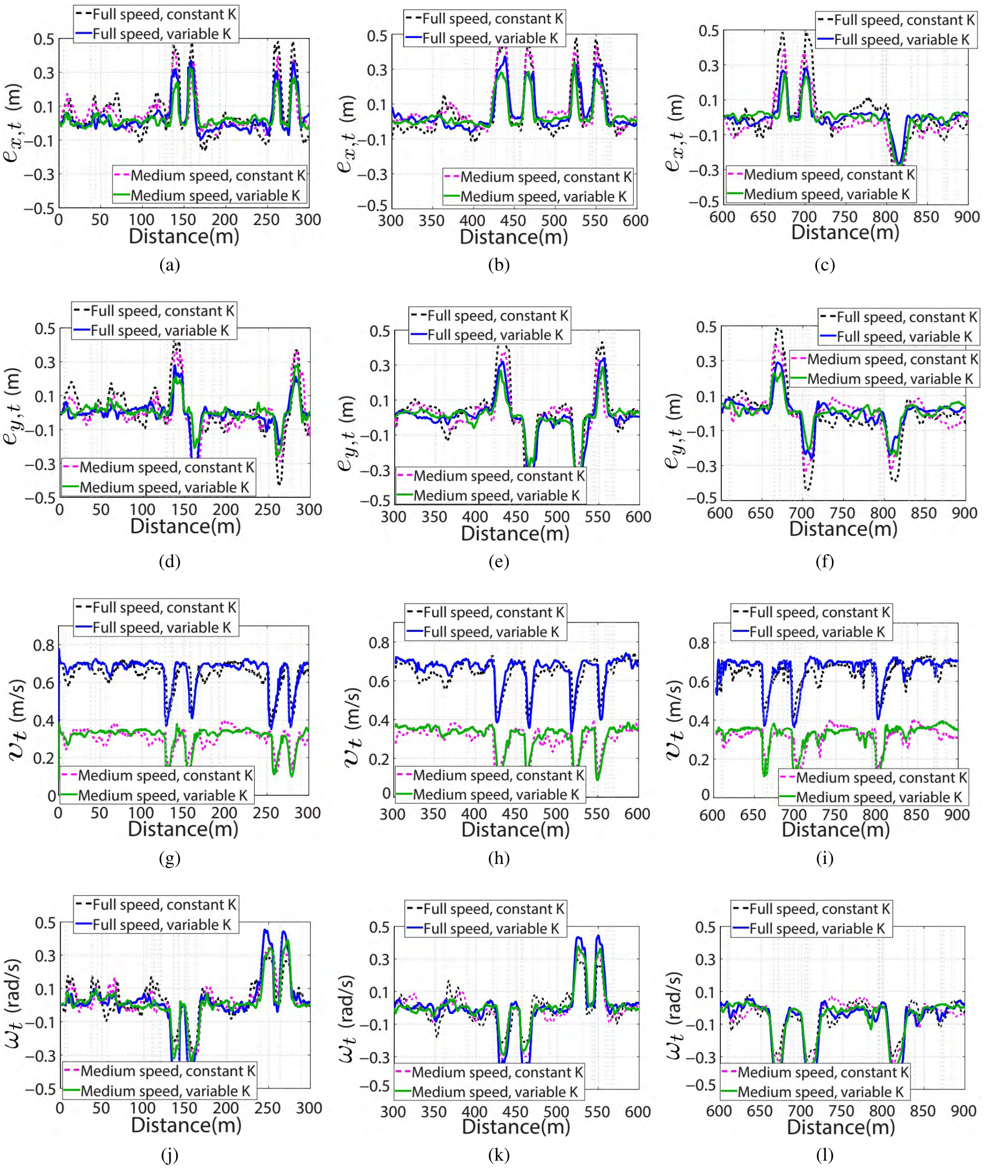


FIGURE 8. Long range test environment. The center of the picture shows a satelital image of the agricultural environment where the experiments were carried out –Vineyard of Casa Blanca, Chile. The side images show snapshots of the vehicle in-situ when traversed the pointed places. The trajectory of approximately 900m-long traversed two types of surfaces: grass and plowed terrain, as shown respectively in the green and yellow solid lines of the reconstruction of the travelled trajectory by the robot within a Cartesian coordinate reference system.

- The maximum traction velocity of the robot is set to $V_{max} = 0.5\ m/s$, and the maximum rotational velocity is $W_{max} = 140°\ rad/s$.
- The initial pose of the vehicle is initially aligned to the reference trajectory and always set to $[x_0\ y_0\ \theta_0]^T = [0\ 0\ 0]^T$.
- The reference trajectory Ω is planned according to the experimentation field before the vehicle starts to navigate, in such a way that some sections match the pre-identified terrain types. Two shapes are considered in this work: squared and S-shaped trajectories. The number of waypoints $\#\Omega$ is relative to the trajectory's length. Hence, the trajectory contains uniformly spaced waypoints in a distance step of 250 mm.

To identify the advantages of the methodology, we considered two scenarios in this test. The first consists in tracking the reference trajectory with a manual setting of the set of gains regardless of the terrain type (i.e., preserving constant the control parameters while navigating). The parameters **K** for the manual setting are heuristically obtained according to previous trials and evaluated for an acceptable performance with respect to the *cumulative total cost* as mentioned in Section IV-A. Only the responses of the controllers with manual tuning and lowest *cumulative total costs* are shown for comparison purposes. The second scenario considers the three proposed motion controllers tuned with the time-varying control gains according to the different navigation surfaces, as described in Section IV-B. In both scenarios, the test is performed under the same terrain conditions, where clear transitions among surface types were intentionally met to assess the capabilities of our method. Also, we considered four trials, where the reference trajectories traversed four types of terrain transitions: pavement–grass, muddy terrain–stony surface, pavement–muddy terrain, and grass–plowed soil. Some of these transitions included more than a pair of terrain types due to the non-uniformity of the terrain, then all the model histograms (x_{model}) obtained in the training process were enabled to be compared within the classifier. Some snap-shots of the testing environment, identified surface transitions, control commands, and trajectory tracking results along with the *cumulative total cost* and accuracy of image-texture classifier are shown in Figs. 6-7. An example of how the best set of gains changes according to the identified surface is also shown in Figs. 6e-6l. As can be seen, although some false identifications were triggered by the image-texture classifier, the control parameters were automatically adjusted smoothly in order to preserve the system stability until the terrain change is consistent. In Figs. 6m-6t, the experiments also show that the algorithm, as formulated, was capable of delivering suited control input commands for any of the three controllers when using our methodology compared with the control inputs with constant set of gains and low *cumulative total cost*. At this point, it is worth mentioning that the aim of this work is not to compare responses among controllers, but analyse how they are enhanced with our methodology.

Following, in the trajectory tracking results of Fig.7, the black dotted lines represent the reference trajectory. The red solid lines are the tracked trajectories with the manual setting of **K** corresponding to the controller with the lowest *cumulative total cost*, whereas the another tracked trajectories are when using our approach under the three motion controllers: C1, C2 and C3. Although a certain level of improvement in the trajectory tracking errors can be seen by inspection for all tests (the robot is closer to the reference –see first column of Fig. 7), deeper comparisons and analysis are required between the terrain detection and controller performance. To do so, the *cumulative total costs* and the accuracy of the classifier for each trial and controller are shown in the second column of Fig. 7. As can be seen, most of the lowest *cumulative total costs* for each motion controller are achieved when tuning the control parameters as a function of the terrain type. In fact, the best performance



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

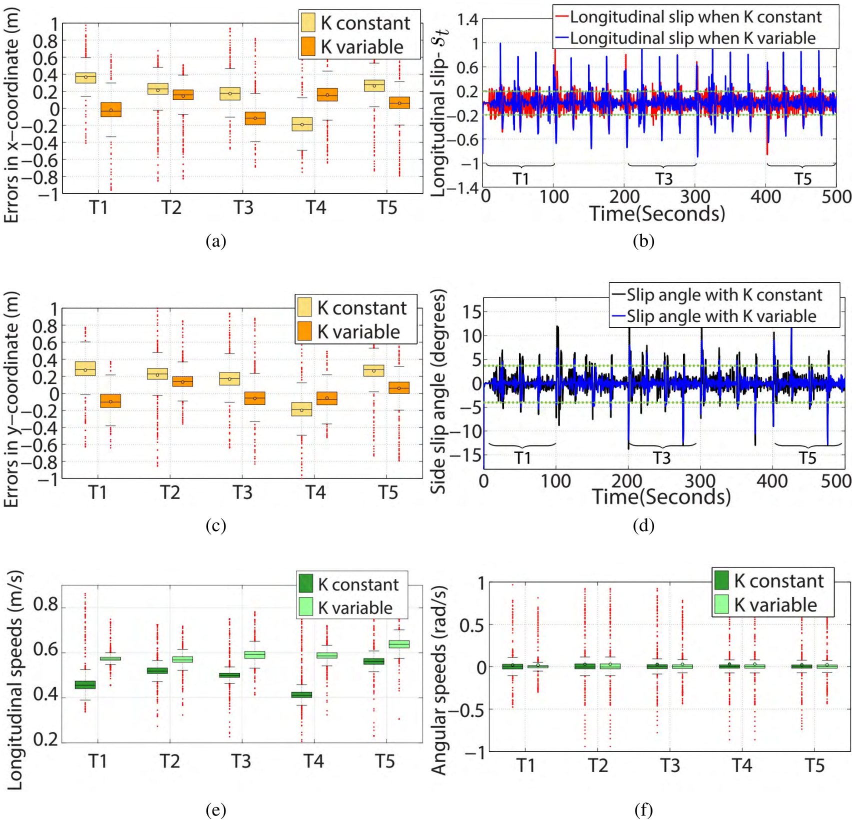
(j) (k) (l)

FIGURE 9. Field results for four different trials. The first and second trial correspond to the responses at full vehicle speed with constant and variable set of gains –dotted black and solid blue lines respectively, whereas the third and fourth trial for medium speed with unchanged and switched gains –magenta dotted and green solid lines respectively. Figures 9a- 9f show the longitudinal and lateral tracking errors vs. traversed distance, where the total trajectory distance is composed by the three sections of 300m long. Figures 9g-9i show the linear vehicle speed and Figs. 9j-9l show the angular speed. The shaded vertical lines represent points where the set of gains changed according to the identified terrain type.

of the controllers arises when the image-texture classifier shows the highest accuracy. On the other hand, the lowest controller performance is obtained when the most inaccurate detection of the terrain surface occurs, although it is higher than that obtained when using constant parameters for all trials. For instance, the cumulative total costs for the first trial were $C^{\Omega}_{Tot} = 968.2$, $C^{\Omega}_{Tot} = 700.8$ and $C^{\Omega}_{Tot} = 1103.1$ for the controllers C1, C2, and C3 respectively when using our methodology, whereas for constant tuning the total cost reached up to $C^{\Omega}_{Tot} = 6168.2$, $C^{\Omega}_{Tot} = 3564.8$ and $C^{\Omega}_{Tot} = 3100.1$. Further, the accuracy of the terrain detection when using each controller was: 92.4%, 90.1% and 87.6%. Thus, the total cost saved by our methodology with respect to the manual setting reached up to 84.3%, 80.3% and 64.4% for each controller. The total cost obtained in the remaining trials result similar, thus achieving the aim of this work.

B. SECOND TEST

In this second experiment, we tested the ability of our approach to change the set of gains while maintaining stability within the full range speeds. To do so, two vehicle speed profiles are considered: medium and full speed –0.35m/s and



(a) (b) (c) (d) (e) (f)

FIGURE 10. Field results for the first test in five turns (i.e., T1-T5), and comparing the automatic tuning of the set of gains with manual setting. Figs. 10b-10d show tracking errors for each repeated trajectory. Figs. 10f-10e show the vehicle speeds for each turn. The coloured boxes show the mean values, whereas the red dotted points represent the measurements of the corresponding variables. Figs. 10a-10c show the slip ratio and side slip angle respectively. The green dotted lines represent the slip constraints.

0.7m/s respectively. A longer range experiment with respect to the first test is carried out in an agricultural environment, where two terrain types appear in the course of the trials: grass and plowed terrain, see Fig. 8. The robot autonomously tracked a trajectory of approximately 900 m of total length by using our approach and constant control parameters for comparison purposes. The trajectory was initially planned in such a way that the vehicle navigates along eight alleys between the furrows of a vineyard plantation, without colliding or going through furrows of the environment. In addition, for this test, we considered the motion controller C2 with the best set of gains $\hat{\mathbf{K}}$ for each terrain type as well as the initial conditions obtained in the first test.

Figure 9 shows the longitudinal, lateral tracking errors $(e_{x,t}, e_{y,t})$, and vehicle speeds (v_t, ω_t) for each tracked trajectory section of 300 m long. For the medium speed profile, the lateral and longitudinal tracking errors showed smaller values when using our approach than those obtained with constant control parameters. Although a small difference between these errors can be seen due to a relative low vehicle speed, almost zero lateral and longitudinal errors can be achieved by the controller under our methodology even in the presence of the effects of the terrain transitions and slip. In addition, a major performance improvement of the controller can be seen after each terrain transition appears, resulting in a scenario of more reliable control parameter setting for motion controllers under restricted spaces of navigability and different types of terrain. Furthermore, a similar behaviour of the controller can be seen in the case of full vehicle speed in the sense of reducing tracking errors when using our approach, but the errors were increased with respect to those obtained in medium speeds due to different kinematic constraints and slip conditions. Moreover, the best set of gains $\hat{\mathbf{K}}$ consistently varies with the terrain changes and within the maximum switching time Δ^{min}_{switch} either for the medium or maximum speed profiles, in such a way that the trajectory tracking errors $(e_{x,t}, e_{y,t})$ remain bounded as well as the vehicle speeds (v_t, ω_t), preserving the stability of the motion controller despite the presence of disturbances coming from frequent terrain transitions.

C. THIRD TEST

In this test, the vehicle tracks a five turn square trajectory (as suggested by [49]) with the motion controller C3. A trial with constant set of gains is also performed with the aim of comparing recurrent consistency of the methodology for each turn, i.e., same terrain transitions. In this test, the vehicle navigates through two types of surface: pavement and stony terrain, in which the transitions occur midway. In order to verify the robustness of the controllers, the best set of gains were obtained in such a way that the slip ratio for each wheel was set to $s_t^o \in \pm 20\%$ and side slip angle $\beta_t^o \in \pm 4(deg)$. Also, the control input commands were constrained by software to $V_{max} = 0.7m/s$ and $W_{max} = 1rad/s$, whereas the vehicle speed profile was set to 0.6 m/s.

The trajectory tracking errors, speeds and slip variables can be seen in Fig. 10. As shown in the statistical representation of Figs. 10b and 10d, for each turn, the trajectory tracking errors (i.e., longitudinal and lateral errors) show reduced values with respect to the manual setting, and these errors reach close to zero repetitively even under the presence of considerable slip conditions. Large errors appear when the vehicle turns close to each corner and increase because of the presence of relevant slip ratio/side slip angle (see [49]). These terra-mechanical effects are partially mitigated with the appropriate selection of the set of gains and satisfactory constraint fulfilment. In Fig. 10f, the means of longitudinal speeds show consistent values for each turn reaching the vehicle speed profile previously stated. Similar results can be seen in Fig. 10e for the angular speed. Additionally, the slip constrains are satisfied by inspection according to Figs.10a and 10c. The longitudinal slip is reduced throughout all test, particularly where a new terrain transition appears. However, during cornering and lateral shifting manoeuvres the slip phenomena remains a relevant percentage of slip ratio and side slip angle. To sum up, the performance of the motion controller with the automatic tuning of the set of gains is improved without further intervention in the controller during the recurrent test.

VII. CONCLUSIONS

A methodology to enhance the performance of trajectory tracking controllers based on real-time visual terrain classification has been presented in this brief. The aim of this work was focused on a proper selection of control parameters using an image-texture classifier, using intensity, color and depth information from a Kinect V2, although other sensors could be included to refine the characterization of the terrain profiling. For our proposal, three trajectory tracking controllers previously published were chosen in order to implement and test our approach. Several tests were carried out in simulation, and the hypothesis validation was carried out through field experimentations under different shaped trajectories. The trials were performed considering typical agricultural terrains: grass, pavement, stony, muddy terrain and plowed soils. For assessing the performance of the motion controllers, metrics related to *tracking errors* and *cumulative control effort* were considered in this brief. The aim of our method was accomplished showing that the performance of the trajectory tracking controllers was improved up to 92.4% considering that the highest accuracy of the classifier was around 84.3%. The latter has direct impact on the energy and resource management of a machinery, thus leading the future research of the authors.

REFERENCES

[1] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, "Approximating kinematics for tracked mobile robots," *Int. J. Robot. Res.*, vol. 24, pp. 867–878, Oct. 2005.

[2] Y. Zhang, J. H. Park, and K. T. Chong, "Model algorithm control for path tracking of wheeled mobile robots," *Int. J. Precis. Eng. Manuf.*, vol. 11, no. 5, pp. 705–714, 2010.

[3] K. Shinomoto, T. Higuchi, and D. Toratani, "Online trajectory optimization using receding horizon guidance control for rovers," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2015, pp. 977–982.

[4] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[5] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2335–2340.

[6] Y. P. Li, T. Zielinska, M. H. Ang, and W. Lin, "Wheel-ground interaction modelling and torque distribution for a redundant mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 3362–3367.

[7] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *Int. J. Robot. Res.*, vol. 26, no. 2, pp. 141–166, 2007.

[8] F. Lange and G. Hirzinger, "Predictive visual tracking of lines by industrial robots," *Int. J. Robot. Res.*, vol. 22, pp. 889–903, Oct. 2003.

[9] H. F. Durrant-Whyte, "An autonomous guided vehicle for cargo handling applications," *Int. J. Robot. Res.*, vol. 15, no. 5, pp. 407–440, 1996.

[10] S. Lacroix *et al.*, "Autonomous rover navigation on unknown terrains: Functions and integration," *Int. J. Robot. Res.*, vol. 21, pp. 917–942, Oct. 2002.

[11] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "High accuracy path tracking for vehicles in presence of sliding: Application to farm vehicle automatic guidance for agricultural tasks," *Auto. Robots*, vol. 21, no. 1, pp. 79–97, 2006.

[12] V. Subramanian, T. F. Burks, and A. A. Arroyo, "Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation," *Comput. Electron. Agricult.*, vol. 53, no. 2, pp. 130–143, 2006.

[13] J.-C. Ryu and S. K. Agrawal, "Differential flatness-based robust control of mobile robots in the presence of slip," *Int. J. Robot. Res.*, vol. 30, pp. 463–475, Dec. 2011.

[14] J. P. Gray and V. V. Vantsevich, "Multi-vehicle convoy mobility in severe terrain conditions: Factor impact analysis, estimation and control strategy," *J. Terramechanics*, vol. 61, pp. 43–61, Oct. 2015.

[15] J. P. Gray, V. V. Vantsevich, and J. Paldan, "Agile tire slippage dynamics for radical enhancement of vehicle mobility," *J. Terramechanics*, vol. 65, pp. 14–37, Jun. 2016.

[16] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robot. Auto. Syst.*, vol. 55, no. 6, pp. 460–469, 2007.

[17] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 92–102, Mar. 2009.

[18] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive tracking control for robots with unknown kinematic and dynamic properties," *Int. J. Robot. Res.*, vol. 25, pp. 283–296, Mar. 2006.

[19] J. Backman, T. Oksanen, and A. Visala, "Navigation system for agricultural machines: Nonlinear model predictive path tracking," *Comput. Electron. Agriculture*, vol. 82, pp. 32–43, Mar. 2012.

[20] S. Wei, K. Uthaichana, M. Žefran, and R. DeCarlo, "Hybrid model predictive control for the stabilization of wheeled mobile robots subject to wheel slippage," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2181–2193, Nov. 2013.

[21] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez, "Adaptive control for a mobile robot under slip conditions using an LMI-based approach," in *Proc. Eur. Control Conf. (ECC)*, Aug. 2009, pp. 1251–1256.

[22] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez, "Adaptive control for a mobile robot under slip conditions using an LMI-based approach," *Eur. J. Control*, vol. 16, no. 2, pp. 144–155, 2010.

[23] E. Kayacan, H. Ramon, and W. Saeys, "Robust trajectory tracking error model-based predictive control for unmanned ground vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 2, pp. 806–814, Apr. 2016.

[24] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodriguez, "Online robust tube-based MPC for time-varying systems: A practical approach," *Int. J. Control*, vol. 84, no. 6, pp. 1157–1170, 2011.

[25] C. Cariou, R. Lenain, B. Thuilot, and P. Martinet, "Adaptive control of four-wheel-steering off-road mobile robots: Application to path tracking and heading control in presence of sliding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 1759–1764.

[26] Y. Zhang, G. Liu, and B. Luo, "Finite-time cascaded tracking control approach for mobile robots," *Inf. Sci.*, vol. 284, pp. 31–43, Nov. 2014.

[27] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[28] Y.-G. Ha, J.-C. Sohn, Y.-J. Cho, and H. Yoon, "A robotic service framework supporting automated integration of ubiquitous sensors and devices," *Inf. Sci.*, vol. 177, no. 3, pp. 657–679, 2007.

[29] R. Socas, S. Dormido, and R. Dormido, "Optimal threshold setting for event-based control strategies," *IEEE Access*, vol. 5, pp. 2880–2893, 2017.

[30] J. B. Derrick and D. M. Bevly, "Adaptive steering control of a farm tractor with varying yaw rate properties," *J. Field Robot.*, vol. 26, nos. 6–7, pp. 519–536, 2009.

[31] C. Jung, H. Kim, Y. Son, K. Lee, and K. Yi, "Parameter adaptive steering control for autonomous driving," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1462–1467.

[32] K. J. Hunt, T. A. Johansen, J. Kalkkuhl, H. Fritz, and T. Gottsche, "Speed control design for an experimental vehicle using a generalized gain scheduling approach," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 381–395, May 2000.

[33] H. Kim, D. Kim, K. Yi, and I. Shu, "Time-varying parameter adaptive vehicle speed control," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 581–588, Feb. 2016.

[34] P. Saeedi *et al.*, "An autonomous excavator with vision-based track-slippage control," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 1, pp. 67–84, Jan. 2005.

[35] F. A. Cheein and G. Scaglia, "Trajectory tracking controller design for unmanned vehicles: A new methodology," *J. Field Robot.*, vol. 31, no. 6, pp. 861–887, 2014.

[36] Y. Guo, L. E. Parker, D. Jung, and Z. Dong, "Performance-based rough terrain navigation for nonholonomic mobile robots," in *Proc. 29th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2003, pp. 2811–2816.

[37] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, pp. 384–389.

[38] E. Lachat, H. Macher, M.-A. Mittet, T. Landes, and P. Grussenmeyer, "First experiences with Kinect v2 sensor for close range 3D modelling," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 40, no. 5, pp. 93–100, 2015.

[39] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput. Vis.*, vol. 62, pp. 61–81, Apr. 2005.

[40] M. Varma and A. Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," in *Computer Vision—ECCV*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Berlin, Germany: Springer, 2002, pp. 255–271.

[41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2000.

[42] W. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical recipes in C: The art of scientific computing," *Math. Gazette*, vol. 73, no. 464, pp. 167–170, 2002.

[43] S. Blažič, "A novel trajectory-tracking control law for wheeled mobile robots," *Robot. Auton. Syst.*, vol. 59, no. 11, pp. 1001–1007, 2011.

[44] G. Xuan, W. Zhang, and P. Chai, "EM algorithms of Gaussian mixture model and hidden Markov model," in *Proc. Int. Conf. Image Process.*, vol. 1. Oct. 2001, pp. 145–148.

[45] A. Maligo and S. Lacroix, "Classification of outdoor 3D lidar data based on unsupervised Gaussian mixture models," in *Proc. IEEE Int. Symp. Safety, Secur., Rescue Robot.*, Oct. 2015, pp. 1–7.

[46] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2012.

[47] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[48] M. A. Carreira-Perpinan, "Gaussian mean-shift is an EM algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 767–776, May 2007.

[49] S. A. Roth and P. Batavia, "Evaluating path tracker performance for outdoor mobile robots," in *Proc. Autom. Technol. Off-Road Equip.*, 2002, p. 388.

JAVIER PRADO received the B.S. degree in electronics and control engineering from Escuela Politécnica Nacional, Quito, Ecuador, in 2011. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. His Ph.D. thesis is on predictive control techniques for agricultural machinery, under the presence of terrain and environment constraints. In the past year, he has collaborated in academic tasks serving as an instructor to under-graduate courses of mobile robotics. His current research interests include agriculture robotics, autonomous navigation, robust control, predictive control, and machine learning applications.

FRANCISCO YANDUN received the B.S. degree in electronics and control engineering from Escuela Politécnica Nacional, Quito, Ecuador, in 2013. Since 2014, he is currently pursuing the Ph.D. degree from the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. His Ph.D. thesis is about mobile robot perception in agricultural environments. In the last year, he has involved with teaching by serving as an Instructor to undergraduate courses of mobile robotics. His research interests include agriculture robotics, sensor fusion, estimation theory, and machine learning applications.

MIGUEL TORRES TORRITI received the B.Sc. and M.Sc. degrees in electrical engineering from the Pontificia Universidad Catolica de Chile (PUC Chile), in 1996 and 1998, respectively, and the Ph.D. degree from McGill University, Montreal, Canada, in 2003. From 2004 to 2005, he was a Senior Applications Engineer with General Electric Chile in the implementation of the multivariable process controllers for the ENAP Bío Bío and Magallanes refineries. In 2005, he joined the Department of Electrical Engineering, School of Engineering, PUC Chile, where he is currently an Associate Professor, a Director of the undergraduate program in autonomous and robotic systems, and an Associate Dean of social responsibility. His fields of experience comprise systems modeling and control, estimation, robot dynamics, and sensors and perception. His current research projects include the development of navigation strategies for mining and agricultural machines considering interactions with the environment, data fusion of satellite imagery and wireless sensor networks for early warning of flash-floods, and the development of assistance devices for persons with motor disabilities.

FERNANDO AUAT CHEEIN received the B.S. degree in electronics engineering from the Universidad Nacional de Tucuman, Argentina, in 2002, and the M.S. and Ph.D. degrees in control systems from the Universidad Nacional de San Juan, Argentina, in 2005 and 2009, respectively. Since 2017, he has been an Associate Professor with the Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile. He is a funder of Autonomous and Industrial Robotics Research Group and part of the board of the Advanced Center for Electrical and Electronic Engineering, Chile. His research interests include autonomous navigation, agricultural robotics, and estimation theory.