Active monitoring of the state of motion in two-wheeled vehicles in absence of a valid GPS/GNSS signal

Simone Gelmini [*] **Silvia Strada** [*,***] **Mara Tanelli** [*]
Sergio Savaresi [*] **Claudio De Tommasi** [**]

[*] *Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo da Vinci 32, 20133 Milan, Italy (e-mail: simone.gelmini, silvia.strada, mara.tanelli, sergio.savaresi@polimi.it)*
[**] *AlfaEvolution Technology S.p.A., Via Salingrado 37, 40128 Bologna, Italy (e-mail: claudio.detommasi@alfaevolutiontechnology.it)*
[***] *Corresponding author*

Abstract: In insurance telematics, the information about the motion of a vehicle is mostly derived by a combination of inertial signals measured from e-Boxes installed integral to the vehicle. However, one must cope with poor reliability and possible lack of continuity of the GPS/GNNS signals. In this work, an inertial-based classification method that discriminates whether a two-wheeled vehicle is in motion is presented. This binary detection can be very helpful in circumstances where the GPS/GNNS signal is not sufficiently reliable and consequently the speed. With respect to what dead-reckoning algorithms do, the present contribution aims to recognize when the vehicle is moving without estimating the vehicle speed, but rather by correctly interpreting the intensity of the measured inertial signals. The approach has been extensively tested on experimental data, proving its suitability for practical applications.

Keywords: Two-Wheeled Vehicles; Crash Detection; Detection Algorithms; Insurance Telematics; Telematic e-Box.

1. INTRODUCTION

Traditional telematic devices and related service providers generally use standard GPS/GNSS technology to track vehicles position and velocity. However, GPS/GNSS sensors often suffers from signal loss, and multipath can occur due to “urban canyons”, tunnels and parking garages, skyscrapers, or tall structures blocking or weakening the connection from the satellites to the device. In fact, for generating an output signal, a GPS/GNSS device needs to connect to at least four satellite signals, out of thirty-one, in order to calibrate its exact location. Otherwise, it is not able to get an accurate reading, causing GPS/GNSS devices to become ineffective until they regain a reliable signal Kao (1991); Toledo-Moreo et al. (2009b).

Inertial low-cost navigation systems, made of Inertial Measurement Units (IMUs), integrating accelerometer and gyroscopes, generally installed integral to the vehicle, can be used for greater reliability of the vehicle motion monitoring tasks, Groves (2013). They are usually employed to continuously compute the vehicle position, orientation and velocity without the need for external references, Toledo-Moreo et al. (2009a). However, all inertial navigation systems suffer from integration errors due to mounting inaccuracy, and sensors’ bias and drifts, which induce progressively larger errors in the estimated quantities. Therefore, these measures must be periodically corrected, for example by resorting to a GPS/GNSS unit as reference for resetting the integration tasks, Bevly and Parkinson (2007).

Non-GPS/GNSS-based navigation and tracking technologies are currently being developed based also on precision time transfer using optics and chip-scale atomic clocks. Additionally, expensive vision-based navigation systems, which rely on cameras and signal processing algorithms, offer high precision through complex processing methods. However, for tracking and for the realization of telematic services, the common approach is mainly based on obtaining inertial sensors measurements at high sampling rates and fusing them to obtain position, velocity and orientation information. With these approaches, accuracy is limited only over a short time window, while drops, as mentioned before, from integration drift over longer time scales, Achtelik et al. (2009); Nagatani et al. (2000).

In this work, we are interested in establishing whether a ground vehicle (in particular a two-wheeled one) is in motion or not. This condition is crucial to be detected in certain accident-related arbitrations, see, *e.g.*, Gelmini et al. (2019), and also when the GPS/GNSS sensors are not working. In particular, we set the problem as that of learning the *in-motion* or *not in-motion* status of the vehicle by processing and classify in real time the inertial signals, without resorting to direct integration. This approach has the merit of overcoming the integration-related limitations discussed above, and to yield a repeatable and accurate method in all working conditions. The effectiveness of the approach is tested over a varied experimental data set.

2. PROBLEM STATEMENT AND EXPERIMENTAL SETUP

The motion classification approach presented in this work is designed, trained, and validated against experimental data recorded with a telematic e-Box, which is placed to be approximately integral with the motorcycle center of mass (Fig. 1). The devices is equipped with:

- An IMU, recording both the triaxial acceleration ($a = [a_x, a_y, a_z]'$) and angular rate ($\omega = [\omega_x, \omega_y, \omega_z]'$) vectors, with a sampling frequency of 400 Hz.
- A GPS/GNSS receiver, the sampling frequency of which is 10 Hz, measuring the vehicle speed. As illustrated later in this contribution, speed is only used to label the data and build the training and validations sets, as it is supposed to be non-available when the algorithm is working.

In particular, the vehicle is considered *in-motion* when its speed is estimated to be larger than a certain threshold, while it is considered to be *not in-motion* otherwise. The threshold is set to 3 km/h, which corresponds to the maximum value of the GPS/GNSS receiver's noise when the vehicle is not moving.

In what follows, it is assumed that the input signals are perfectly aligned with the vehicle standard reference frame. However, since the e-Boxe can be freely installed on the vehicle, the measurements are affected by its mounting orientation. Therefore, to ensure that the measured quantities are aligned as expected, axes are virtually rotated through a self-calibration algorithm, as described in Gelmini et al. (2018a).

The experimental data presented herein are collected on a 500 cm^3 Piaggio MP3. Tests are conducted under mixed traffic conditions, both on urban and extra-urban roads.

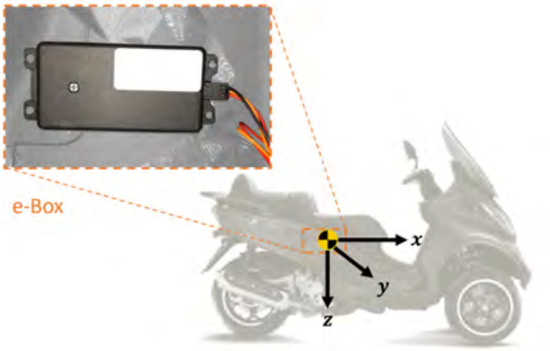

Fig. 1. The experimental setup: a Piaggio MP3 and the e-Box mounted underneath the seat.

3. FEATURES ENGINEERING

As previously described, the proposed algorithm is designed to detect whether a two-wheeled vehicle is in motion or not without relying on the presence of a valid GPS/GNSS signal. In this section, the features engineering phases used to train the classification algorithm are discussed.

3.1 Preliminary data analysis

In this preliminary phase, four scenarios are considered:

- *Standing Engine OFF*: The motorcycle is powered off and the vehicle is standing still.
- *Standing Engine On Low RPM*: The motorcycle is standing still, powered on, with the engine idling.
- *Moving Low Speed*: The vehicle is driven stationary at low speed (up to 10 km/h).
- *Moving High Speed*: The vehicle is driven at constant speed, at different velocities, all larger than 10 km/h.

Fig. 2 plots pairs of signals in the four vehicle conditions indicated above. As shown, signals exhibit significant differences in terms of their spanned range. In fact, the size of the clusters grows with increasing levels of vehicle vibrations. In fact, when the vehicle is powered off, the range of the signals is approximately zero. Instead, when powered on, the engine-induced vibrations have increasing intensity, which further grows when the vehicle is in motion. These differences can be modeled considering the variance of the signals.

The ratio of the variance of each signal in *Moving Low speed* and *Standing Engine On Low RPM*

$$\frac{\mathrm{Var}_{LS}(j)}{\mathrm{Var}_{SE_{ON}}(j)}, \tag{1}$$

with $j = \{a_x,\ a_y,\ a_z,\ \omega_x,\ \omega_y,\ \omega_z\}$, is analyzed and shown in Fig. 3. This analysis proves that $\omega_z,\ \omega_y,\ a_x$ have a much more discriminating power than $\omega_x,\ a_z,\ a_y$. This result can also be explained intuitively: At low speed, it is difficult (or even impossible) to lean rapidly (*i.e.*, large values of ω_x), the vehicle lateral slip is minimal (*i.e.*, low a_y), and the heave movement is only subject to road imperfections (*i.e.*, a_z is limited). For this reason, to reduce the problem dimensionality, only these three most informative signals are used from now on for classification purposes.

It is worth to mention that there exists a critical scenario that has not been introduced yet: *Standing Engine On High RPM*. In this condition, the vehicle is powered on, standing still, with the engine intentionally accelerated, so to increase its RPMs. This operation is often performed by some riders who turn the throttle handlebar while waiting to restart (*e.g.*, at traffic lights). This *non moving* scenario shows that the vibrations sensed by the IMU are comparable, or even more significant, than those experienced when the vehicle is in motion at constant low speed (Fig. 4). This condition proves that analyzing inertial data without considering their dynamics would probably lead to a non-robust detection algorithm.

To effectively detect when the vehicle is moving, inertial measurements are analyzed in the frequency domain. In this context, the differences appear to be clearer, as depicted in Fig. 5. In fact, focusing only on the two previously misleading scenarios (*i.e.*, *Standing Engine On High RPM* and *Moving Low Speed*), one may see that the related information content is spread over different harmonic ranges: At *Moving Low Speed*, the information is mainly present up to 10 Hz for all the signals, with additional peaks at 19.6, 38, 79 Hz. On the contrary, when the vehicle is standing still, the spectra are almost flat to zero up to 14 Hz, where the first peak is encountered. Thus, the influence of the vehicle dynamics, which affects the measurements only when the vehicle is in motion, is dominant at low frequencies, while engine-transmission resonances influence the harmonics above 10 Hz.

3.2 Data pre-processing

Since the goal of the proposed approach is to be vehicle-independent, the analysis in the frequency domain provided insights on how to pre-process inertial signals so to remove the effect of vehicle's resonances, retaining only the informative content related to the navigation dynamics of interest. To this end, signals are filtered with a first-order low-pass filter

$$F(s) = \frac{2\pi f_c}{s + 2\pi f_c}. \tag{2}$$

The cut-off frequency (f_c) is chosen based on a sensitivity analysis aiming to enhance the ratio between the signals variance for *Moving Low Speed* over the *Standing Engine*

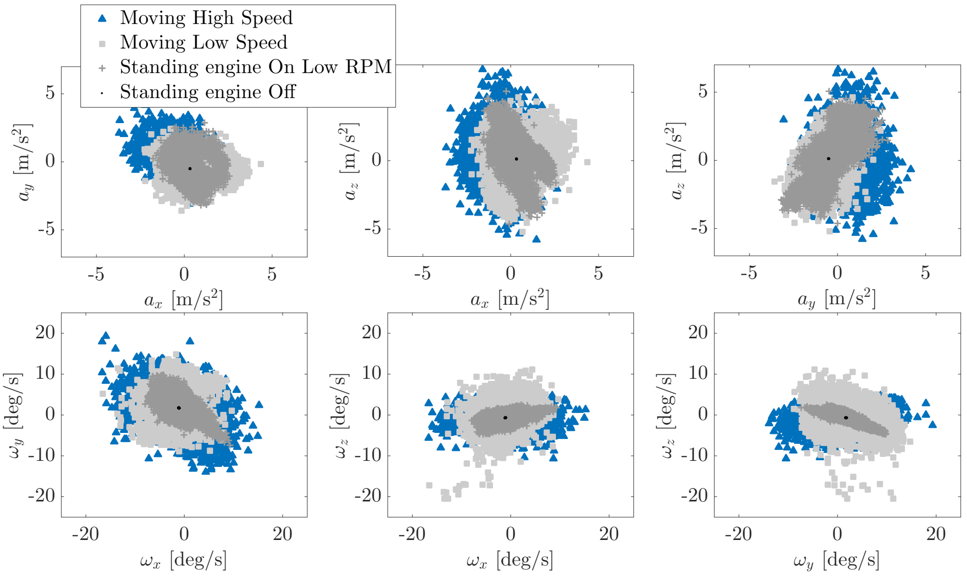

Fig. 2. Analysis of the six inertial measured signals for different vehicle conditions: Clusters are more spread for a more intense use of the vehicle. Note that a_z was debiased from the effect of gravity.

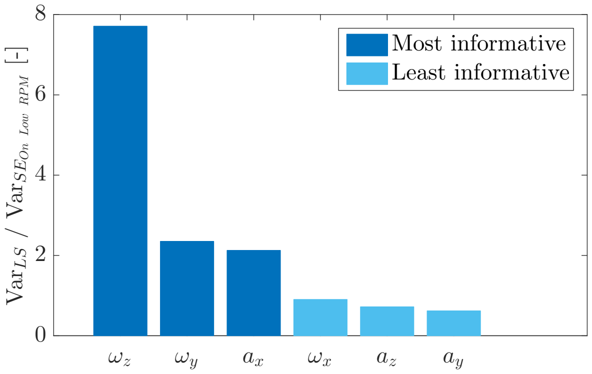

Fig. 3. Ratio of the variance computed for scenario *Moving Low Speed* (*LS*) over *Standing Engine On Low RPM* (SE_{ON}). For ω_x, a_z, a_y, the ratio is less than unity, meaning that the vibrations, when moving, are significantly less intense than when standing still.

On High RPM condition, which can be considered as a quantitative descriptor to measure how different the two clusters are. As illustrated in Fig. 6, this ratio increases for small values of f_c, though greater values of f_c provide a prompter response. It can be seen that while ω_z discriminates between the two cases even for $f_c = 10$ Hz, ω_y and a_x require a more significant filtering action. A good trade-off is found for $f_c = 0.5$ Hz, despite the non-negligible phase lag.

Once the data are filtered, as shown in Fig. 7, the size of the ranges for *Moving Low speed* and *Standing Engine On High RPM* are sufficiently different to distinguish between the two classes, even in this particularly critical scenario.

3.3 Features extraction

To extract the discriminating information, two features are derived for each signal s, with $s = \{a_x,\ \omega_y,\ \omega_z\}$, over a sliding window that buffers the last N samples:

- s_{ran}, computes the difference between the maximum and minimum values of the current values stored in the sliding window

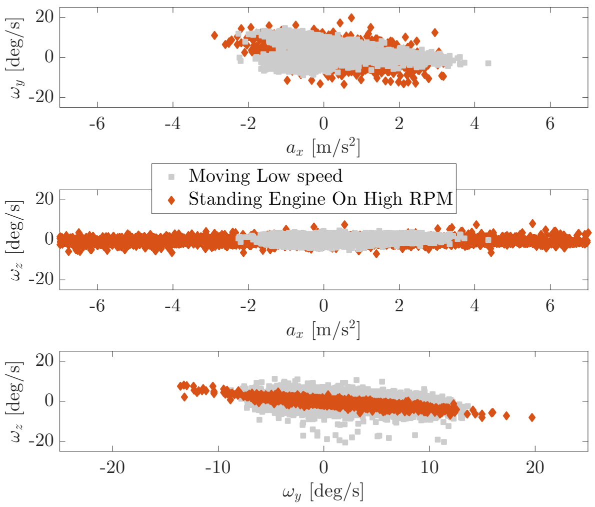

Fig. 4. Analysis of the special case *Standing Engine On High RPM* vs *Moving Low Speed* for the most informative signals. In this case, the clusters are overlapped and the *not-moving* one is larger than its *moving* counterpart.

$$\begin{aligned} s_{ran}(k) = &\max\left(s(k,\ldots,k-N_{ran})\right) \\ &-\min\left(s(k,\ldots,k-N_{ran})\right). \end{aligned} \tag{3}$$

- s_{var}, computes the variance of the buffered values

$$\begin{aligned} s_{var}(k) = \mathbb{E}\Bigg[\Big(&s(k,\ldots,k-N_{var}) \\ &-\mathbb{E}\left[s(k,\ldots,k-N_{var})\right]\Big)^2\Bigg]. \end{aligned} \tag{4}$$

Hence, six features are extracted, three variances (*i.e.*, $a_{x_{var}}$, $\omega_{y_{var}}$, $\omega_{z_{var}}$), and three max-min differences (*i.e.*, $a_{x_{ran}}$, $\omega_{y_{ran}}$, $\omega_{z_{ran}}$). In this application, the buffer size is chosen to be $N_{ran} = 500$, which allows storing the past 1.25 seconds of the signals. Alternatively, the variance can be computed with a filtering chain, as discussed in Gelmini et al. (2018b). In this case, an equivalent window N_{var} is obtained with a chain of high-pass and low-pass filters, the

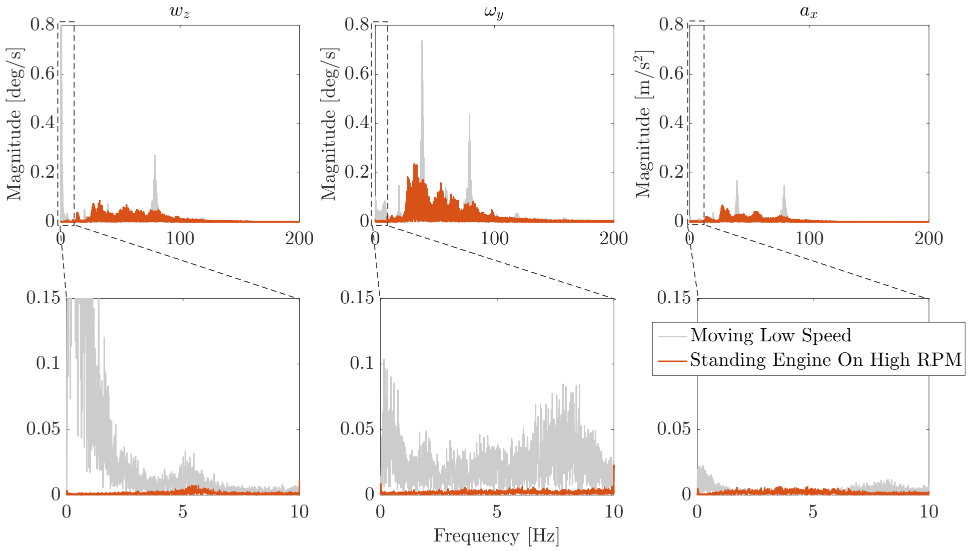

Fig. 5. Spectral analysis of a_x, ω_y, ω_z for *Standing Engine On High RPM* and *Moving Low Speed*. The two scenarios, which are hardly distinguishable in the time-domain framework, are instead well separated in the frequency domain.

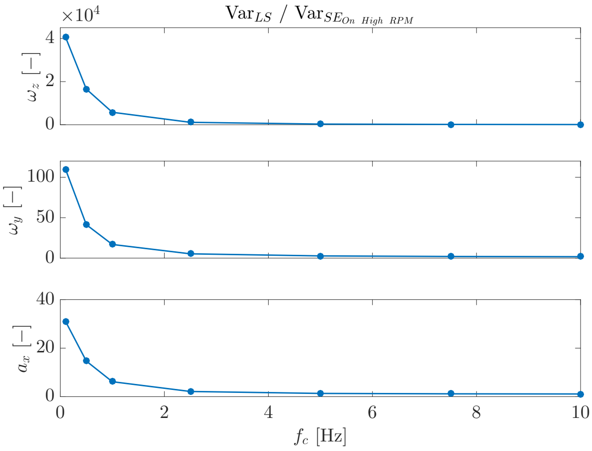

Fig. 6. Sensitivity analysis of the low-pass filter cut-off frequency: The ratio between the variance computed on the filtered signals increases for low value of f_c.

cut-off frequency of which are respectively $f_{c_{high}} = 0.35$ Hz and $f_{c_{low}} = 1.5$ Hz.

4. DETECTION ALGORITHM AND EXPERIMENTAL RESULTS

The features extracted in the previous section can be used to classify the motion of the vehicle. Thus, the classification can be performed with Support Vector Machines, a machine-learning classification algorithm that optimizes the class separation finding the optimal hyperplane that maximizes the separation margin between any training point and the hyperplane itself Scholkopf and Smola (2001); Duda et al. (2012); Friedman et al. (2001). This hyperplane is obtained solving the following optimization problem

$$\begin{aligned} &\min_{w\in\mathcal{H},\ b\in\mathbb{R},\ \xi\in\mathbb{R}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ &\textit{subject to } y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i,\ \forall i = 1,\ldots,m \\ &\xi_i \geq 0, \end{aligned} \tag{5}$$

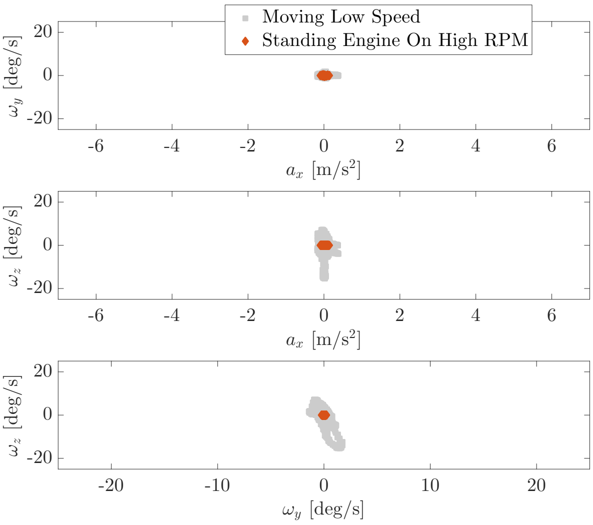

Fig. 7. Analysis of the special case *Standing Engine On High RPM* vs *Moving Low Speed* for the most informative signals (*i.e.*, a_x,ω_z, and ω_y), filtered with a low-pass filter. In this case, the clusters have different dimensions, making the classification possible.

in which w is a vector orthogonal to the hyperplane, $\mathcal{H}$ is the product space, $(x_i,\ y_i)$ are the training inputs and outputs, and $C > 0$ the penalty parameter of the error term ξ, which limits the bias if the training dataset is corrupted with outliers. Besides, SVM is preferred, among all the linear classifiers, for its relatively low computational effort during the prediction phase, crucial for running on devices with reduced computational power, such as the telematic e-Boxes herein considered.

The SVM algorithm is trained with all possible combinations of features extracted from training data. The dataset used in the learning process is composed of a hundred miles of tests recorded on both urban roads and highways, both maintaining the vehicle stationary at a given speed and reproducing a more natural driving profile, with different stops.

4.1 Validation

The trained algorithm is then evaluated and optimized against a validation dataset, containing only real riding scenarios not used in the training phase. In the upper part of Fig. 8, the classification performance is analyzed in terms of accuracy, specificity and sensitivity. As shown, most of the classifiers achieve comparable high performance for all the analyzed indexes. The SVM number 47, trained with $\omega_{y_{var}}$ and $\omega_{z_{var}}$, shows the highest accuracy (97.38%), with both sensitivity and specificity above 97.2%. Overall, it is possible to claim that the vast majority of the features combinations provide solid results.

When a dynamic system is involved in a classification, the output bounces when switching from one class to another. Thus, we can compute the average time to robustly detect when a vehicle starts moving (denoted as *rise time*), and when it stops (*fall time*). Results shown in the lower part of Fig. 8 point out that most of these classifiers are generally faster at detecting when the vehicle starts moving (0.25 seconds) than when it stops (2.5 seconds). This is mainly due to the low-pass filtering and data buffering steps, which lag the output.

In turn, this delay affects of course the classification performance indexes. Therefore, at each class variation, classification performance can be evaluated removing the effects of these transients, assuming the classifier to have settled to the new class. We can denote with *steady* the classification performance removing the data within the average rise and fall times. As further discussed in Table 1, steady-state results confirm the classifier performance: SVM 47 obtains an accuracy of 99.47%, the highest sensitivity of 99.37% and an important specificity increment, reaching now 99.49%.

Spikes can be defined as very short-time outliers in the prediction (*e.g.*, in this application, spikes are no longer than few samples – it is unrealistic to assume that the motorcycle is in motion for such a short amount of time only). Despite spikes do not play a significant role in the evaluation of the classification performance, it is interesting to evaluate the number of spikes for the different classifiers. As shown in the bottom right of Fig. 8, the best classifier in terms of accuracy (*i.e.*, SVM 47), is quite affected by spikes, with 13 instances in 1400 s. This result should not surprise: A prompt classifier (*i.e.*, small fall and rise time) is more subject to short inconsistent events than slower and more conservative ones.

4.2 Testing

For a fair evaluation of the performance, the same SVMs are employed against a testing dataset. Classification performance slightly drops for all the trained models. In particular, SVM 47 proves not to be the most performing classifier as it was for the validation dataset, although it is still in the top five ones. In testing, SVM 47 achieves 94.231% in accuracy, 97.748% in sensitivity, with a significant drop in specificity (93.617%), which is explained with an increment of the fall time, as opposed to all the other classifiers in which drops. By analyzing the performance of the classifier at steady-state, all the top classifiers, including SVM 47, achieve at least 99% of accuracy and specificity, and 98.5% in sensitivity, results comparable with the same obtained in validation.

To better analyze the achieved classification performance, the four best performing classifiers are compared in detail in Table 1. The following remarks are in order:

- Performance of these four classifiers is comparable and consistent in validation and testing, both in the nominal and steady-state cases, with limited differences also for the mean rise and fall time.
- All the four most performing classifiers have $\omega_{z_{var}}$ as their most relevant feature. Thus, yaw-angle variations that occur in the transition between the stand still and *in motion* conditions are the most relevant physical phenomenon in this context. The second key feature is the pitch rate, with minimum differences using either $\omega_{y_{var}}$ or $\omega_{y_{ran}}$.
- The classifiers become more conservative (yet more robust against the testing dataset) including also $a_{x_{ran}}$, the longitudinal acceleration. However, the classifiers slightly underperform when the longitudinal acceleration is added.

5. CONCLUDING REMARKS

In this work, a classification method that allows discriminating an in-motion versus a non-in-motion two-wheeled vehicle was presented. Notably, the approach employs only IMU-based inertial measurements, and thus can reliably work when GPS/GNSS sensors are not working, due to multiple reasons. The capability of discriminating among the two states is of particular interest in the process of detecting whether a crash occurred, and in possible crash-related arbitration within the insurance telematic context. The effectiveness of the approach is witnessed by an extensive experimental campaign that covered both urban and non-urban driving under many different dynamic conditions.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Fabio Martellotta with the experimental data collection activities.

REFERENCES

Achtelik, M., Bachrach, A., He, R., Prentice, S., and Roy, N. (2009). Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. *Unmanned Systems Technology XI.*

Bevly, D.M. and Parkinson, B. (2007). Cascaded kalman filters for accurate estimation of multiple biases, dead-reckoning navigation, and full state feedback control of ground vehicles. *IEEE Transactions on Control Systems Technology.*

Duda, R.O., Hart, P.E., and Stork, D.G. (2012). *Pattern classification.* John Wiley & Sons.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning.* Springer series in statistics New York.

Gelmini, S., Fouka, M., Strada, S., Tanelli, M., Savaresi, S., and De Tommasi, C. (2018a). Self-calibration algorithm for an imu in two-wheeled vehicles: design and experimental validation. *Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems.*

Gelmini, S., Strada, S., Tanelli, M., Savaresi, S., and Biase, V. (2018b). Analysis and development of a novel algorithm for the in-vehicle hand-usage of a smartphone. *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC).*

Gelmini, S., Strada, S., Tanelli, M., Savaresi, S., and De Tommasi, C. (2019). Automatic crash detection system for two-wheeled vehicles: design and experimental

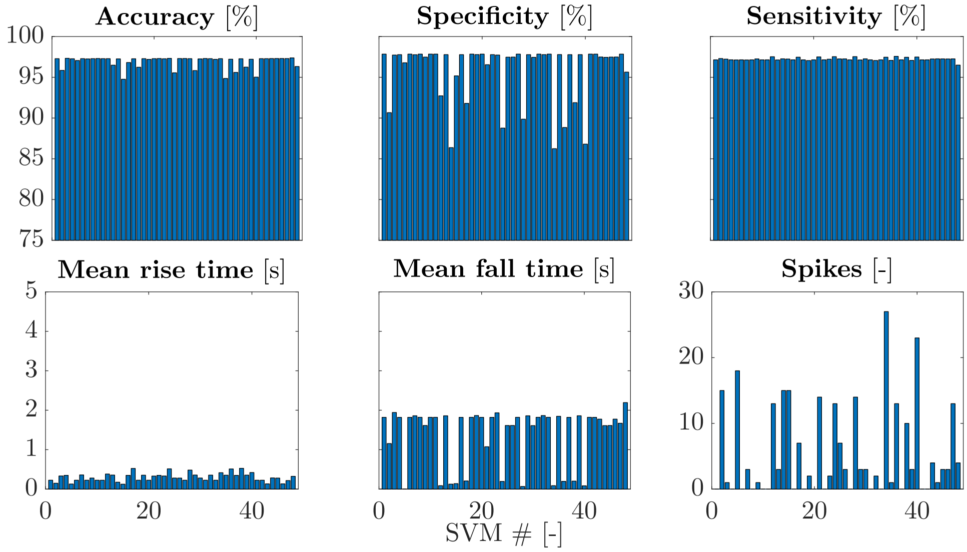

Fig. 8. Performance of the trained SVM against the validation dataset. As shown, almost all the classifiers achieve the same performance.

Dataset	Features	Classifier	Accuracy		Sensitivity		Specificity		Time fall steady [s]	Time rise steady [s]
			Nominal	Steady	Nominal	Steady	Nominal	Steady		
Validation	$\omega_{y_{var}}$ $\omega_{z_{var}}$	47	97.38	99.47	97.85	99.37	97.27	99.49	0.26	2.7
	$\omega_{y_{ran}}$ $\omega_{z_{var}}$	3	97.33	99.65	97.73	99.64	97.23	99.65	0.42	2.72
	$\omega_{y_{var}}$ $\omega_{y_{ran}}$ $\omega_{z_{var}}$	23	97.33	99.65	97.73	99.63	97.22	99.65	0.42	2.7
	$a_{x_{ran}}$ $\omega_{y_{ran}}$ $\omega_{z_{var}}$ $\omega_{z_{ran}}$	36	97.31	99.53	97.48	99.7	97.26	99.49	0.45	2.26
Testing	$\omega_{y_{var}}$ $\omega_{z_{var}}$	47	94.23	99.22	97.75	99.24	93.62	99.22	0.35	2.3
	$\omega_{y_{ran}}$ $\omega_{z_{var}}$	3	94.57	99.02	97.46	98.49	94.01	99.12	0.17	2.02
	$\omega_{y_{var}}$ $\omega_{y_{ran}}$ $\omega_{z_{var}}$	23	94.54	99.25	97.52	98.6	93.97	99.37	0.16	2.37
	$a_{x_{ran}}$ $\omega_{y_{ran}}$ $\omega_{z_{var}}$ $\omega_{z_{ran}}$	36	94.62	99.2	97.01	98.33	94.19	99.36	0.2	2.54

Table 1. Comparison of the performance of the most performing classifiers against validation and testing data.

validation. *Proceedings of the 9th IFAC International Symposium on Advances in Automotive Control (AAC).*

Groves, P.D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems.* Artech house.

Kao, W.W. (1991). Integration of gps and dead-reckoning navigation systems.

Nagatani, K., Tachibana, S., Sofne, M., and Tanaka, Y. (2000). Improvement of odometry for omnidirectional vehicle using optical flow information. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Scholkopf, B. and Smola, A.J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press.

Toledo-Moreo, R., Bétaille, D., and Peyret, F. (2009a). Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps. *IEEE Transactions on Intelligent Transportation Systems.*

Toledo-Moreo, R., Bétaille, D., Peyret, F., and Laneurit, J. (2009b). Fusing gnss, dead-reckoning, and enhanced maps for road vehicle lane-level navigation. *IEEE Journal of Selected Topics in Signal Processing.*