A Statistical Approach for Improving the Performance of a Testing Methodology for Measurement Software

Giovanni Betta, *Senior Member, IEEE*, Domenico Capriglione, *Member, IEEE*, Antonio Pietrosanto, *Member, IEEE*, and Paolo Sommella

***Abstract*—This paper describes the significant enhancements brought to an original methodology designed for testing measurement software. In a previous paper, the authors proposed a black-box seven-step procedure that allows the functional verification of complex instrument software to be performed. The main features of the procedure are concerned with the following: 1) the ability of reproducing actual correlations among the software inputs and 2) the need for a limited number of test cases. Making use of innovative statistical techniques, the methodology performance and reliability have been enhanced. Two further steps have been added with the aim of improving the correlation coefficient assessments and providing the estimations with a confidence level. Finally, a new strategy has been studied to optimize the number of test cases. The effects of the new solutions on the performance of the methodology are evaluated by applying the procedure to a complex software module employed in an automotive system. A comparison with the previous methodology version is also reported.**

***Index Terms*—Black-box approach, instrument software, reliability, software engineering, test methodology.**

I. Introduction

SOFTWARE has become a critical core industry. It is a fundamental part of systems in the most important fields of today's society, from transportation and communication to financial and medical applications. Automotive, for example, is a typical context where software procedures (e.g., antilock braking system and electronic stability program) executed by suitable microelectronic circuitry are able to grant passenger safety and comfort, but other daily life applications, such as modern cardiac pacemakers, in which approximately one-half megabyte of code helps control the pulse rate of patients with heart disorders can be implemented. For all these systems, fundamental issues, such as efficient process operation, safety, and fault tolerance, are then assured by software architecture, whose quality is assuming a growing importance in the industrial point of view. Thus, great interest is focused on software engineering [1], i.e., the application of a systematic disciplined quantifiable approach to the development, operation, and maintenance of software. In particular, among the different steps that constitute the life cycle of the software, these efforts on improving quality are also concerned with verification and validation (V&V) phases. V&V processes determine whether products of a given activity conform to the requirements of that activity and whether the software satisfies its intended use and user needs [2]. Moreover, the aim of software testing is the detection of possible errors in program execution and, in the absence of faults, the estimation of the performance of the system under test. Although this activity does not imply giving up improvements in the overall software development process, it proves to be crucial because of the efforts employed on it. It is estimated that the total cost of the testing phase is approximately 20%–33% of the total software budget for software development [3]. Furthermore, software testing is very time consuming since the time for testing is typically greater than that for coding. Thus, efforts to reduce the costs and improve the effectiveness of testing can yield substantial gains in software quality and productivity.

These features are also becoming more prevalent in the field of instrumentation and measurement technology. In fact, a modern measurement system is generally a software-based station whose software is used to perform different tasks. As an example, digital oscilloscopes are provided with embedded software that is able to perform some signal analysis and measurements in both the time (period, frequency, root-mean-square value, etc.) and frequency (particularly power spectrum) domains [4]–[6]. Consequently, to verify the accuracy of the measurement results produced through the corresponding tasks, software has to be carefully tested.

Several consolidated methodologies for testing general-purpose software exist in the literature [7], all following either a *structural* or a *functional* approach. The former (also known as the *white-box* approach) is based on a detailed knowledge of code and is usually used for checking the program in the early stages of the life cycle. Generally, the developers of the code carry out this kind of test. According to the latter approach (also known as the *black-box* approach), the internal implementation of the software being executed does not need to be known by testers. Only the outputs of a program, given certain inputs, are checked so that they conform to the functional specification of

Manuscript received July 15, 2006; revised November 16, 2007.

G. Betta and D. Capriglione are with the Department of Automation, Electromagnetism, Information Engineering, and Industrial Mathematics (DAEIMI), University of Cassino, 03043 Cassino, Italy (e-mail: betta@unicas.it; capriglione@unicas.it).

A. Pietrosanto and P. Sommella are with the Department of Information and Electrical Engineering (DIIIE), University of Salerno, 84084 Fisciano, Italy (e-mail: apietrosanto@unisa.it; psommella@unisa.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2007.915143

the system under test. The black-box approach is usually applied in the latest stages of software testing, such as integration and system testing, where greater attention is focused on the user's perspective.

As for the black-box approaches, partition testing is one of the most common methods in software quality assurance. It is designed to minimize the number of test cases by dividing tests in such a way that the system is expected to act the same way for all tests of each equivalence partition. Test inputs would be selected from each partition; in particular, equivalence partitions are designed so that every possible input belongs to one and only one equivalence partition [8]. Nevertheless, this technique has some drawbacks. It does not test all input values, and it is heuristic based because well-defined guidelines have not yet been stated for choosing inputs.

For this reason, equivalence partitioning is often applied in conjunction with other black-box techniques, such as boundary value analysis [9]. It has been widely recognized that input values at the extreme ends of, and just outside of, input domains tend to cause errors in system functionality. In the boundary value analysis, values at and just beyond the boundaries of the input domain are used to generate test cases to ensure proper functionality of the system. The boundary value analysis is an excellent way to catch common user input errors that can disrupt proper program functionality. Although the boundary value analysis can be revealed to be very good at exposing potential user interface/user input problems and allows one to generate a small set of test cases following very clear guidelines, this technique also does not test all possible inputs, and above all, it does not take into account the dependences between combinations of inputs.

With the aim of overcoming the limits of the already existing methods, in the past few years, further studies, particularly in statistical sciences, have been carried out. As a result, some methods based on experimental design and probability theory were proposed and adopted [10]–[15], which, because they are designed for automatic testing, allow costs and time reduction in software development. For example, a technique known as the orthogonal array testing system (OATS) was derived by Taguchi's robust design methodology, which is widely used in many modern areas of engineering. The OATS can be very useful in selecting the combination of test parameters that provide maximum coverage from test procedures while using a minimum number of test cases. The assumption is that the test that maximizes the interaction between parameters will find more faults. Nevertheless, the OATS is not always the right solution since it can also lead one to consider an invalid combination of parameters, particularly when attention was focused on the wrong area of application. In addition, the OATS, as well as other experimental design techniques, is mainly concerned with Boolean or a few level discrete variables and/or disregards the correlations existing among inputs. Thus, none of them fit well to the case of the instrument software. Indeed, in this particular context, input data are typically representative of measured quantities that are often continuously time varying in wide ranges and correlated with each other in some complex ways. These considerations drove the authors to the conclusion that a suitable test methodology had to be developed, which was able to carry out the V&V phases of instrument software through a good representative number of input–output real cases.

In a previous paper [16], the authors suggested a methodology based on a black-box approach and the adoption of suitable statistical techniques, which can be used for the validation of complex instrument software to also obtain efficient and realistic test results when the software inputs are correlated. The proposed test procedure was applied to diagnostic software employed in an automotive environment, and the obtained results were very promising.

However, in-depth analyses were successively carried out with the aim of investigating the reliability of the testing procedure.

As a consequence of these analyses, the identification of the correlations existing among inputs to the system under test proved to be a very critical issue, particularly when it was based on experimentally acquired data. Indeed, the Spearman rank coefficient, which is the statistic from which the input correlations are derived, can be hardly influenced by frequent occurrences of specific observations and by noise effects, which can characterize an inaccurately planned measurement campaign. In addition, a quantitative evaluation of the effectiveness of the estimated coefficients in representing actual correlations also has to be provided, with the aim of overcoming the dependence of correlation identification from a particular experimental acquisition.

In this paper, a suitable filtering strategy is proposed to enhance the estimates of the input correlations, thus avoiding the adoption of both polarized and noise-corrupted rank coefficients. In addition, a method to determine the confidence intervals of the Spearman coefficients is suggested, and the suitable modifications to the previously introduced steps are defined to take into account this further information. Moreover, to verify the general applicability of the proposed methodology, the test procedure has been applied on a different and larger experimental data set than the one used in [16] and [17].

In the following, after a brief recall of the main steps of the test methodology, the proposed improvements are reported, together with detailed considerations about the experimental results.

II. Test Methodology

The proposed methodology falls in the class of black-box approaches. This means that the software under test must be verified with a suitably studied set of inputs (the so-called *test set*), whose corresponding expected outputs are known only on the basis of the functional specifications. Since the number of input combinations can be quite unlimited in real applications, the real problem of any black-box approach is to design a test set well representative of the field behavior of the system, avoiding redundant and wrong input–output cases. To solve this main problem with reference to the measurement software testing, a seven-step procedure was defined in [16].

a) *Acquisition and statistical characterization of experimental data:* Since each input quantity is considered as a random variable, this step aims to determine the probability distributions of all the inputs.

b) *Evaluation of the experimental correlation matrix:* The existing relationships among the input variables are taken into account by the correlation matrix C, whose elements are estimated by means of the Spearman coefficients [18]. Thus, a kind of rank correlation is adopted, which is particularly appropriate when the input variables have a nonnormal data distribution. Moreover, unlike other types of statistics (such as the Pearson coefficient), the rank correlation well represents the relatedness of two variables that are monotonically but nonlinearly related.

c) *Smoothing of the input distributions:* Generally, the input probability distributions estimated in step a) exhibit a stepwise shape that could compromise the next steps (particularly Latin hypercube sampling (LHS) and induction processes). Therefore, a suitable smoothing of the input cumulative distribution function (cdf) is carried out [19]. As far as the smoothing process is concerned, two approaches could be considered: parametric and nonparametric. The former is generally applied when the initial empirical distribution is close to a "standard" distribution (e.g., Gaussian, triangle, and uniform), whereas the latter is preferred when this condition is not satisfied. In particular, among the nonparametric approaches, the one based on a kernel function method, which requires choosing the kernel (e.g., Gaussian, triangle, uniform, and Epanechnikov) and the bandwidth, can be adopted.

d) *LHS:* This stratified random procedure provides an efficient way of sampling variables from their distributions [20]. It is designed to accurately recreate the input distribution with a reduced number of samples compared with the Monte Carlo method. Given N input variables, dividing each cdf into M intervals of equal probability, the output of the LHS step is a matrix L, whose columns (M size) contain the samples of the corresponding input variable, which is in agreement with its marginal distribution, whereas rows (N size) represent the samples of the starting multivariate distribution.

e) *Correlation induction*: Whenever input variables are correlated to some extent, the random pairing of the variables (result of LHS) could generate wrong and/or impossible combinations. Therefore, a suitable correlation induction method should be employed. To this aim, one of the more widespread and known methods, i.e., the *restricted pairing algorithm* by Iman and Conover, has been adopted [21], thus obtaining the reordered matrix L^*.

f) *Correlation refinement*: To minimize the distance between target correlations and those obtained by applying an induction algorithm, an iterative refinement procedure is required [18], which is able to reduce the mean absolute deviation (MAD) between the corresponding off-diagonal elements of the target matrix C and the rank correlation matrix L^*. The result of this phase is a new rank correlation matrix R.

g) *Test case generation*: The final result of the previous step is a matrix whose rows are the M combinations of N inputs that should be submitted to the software under test. Increasing M improves the procedure performance (the MAD index is closer to 0) but increases the time required to test the software. Then, before submitting the test cases to the software, the value of M should be chosen on the basis of this tradeoff and by fixing a maximum value for the MAD index that is acceptable for the particular application considered.

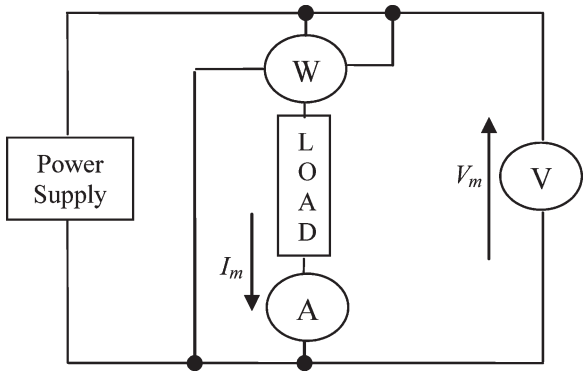


Fig. 1. Schematic of a measurement system for the power wasted by a load.

As described, the proposed methodology requires a coarse knowledge about the domains of the input quantities of the software under test. In particular, this knowledge can be derived from *a priori* available information about the system model or, more reasonably, from experimental measurements carried out without the need for a suitably designed measurement campaign.

The goal of the seven-step procedure is the identification of a test set, i.e., a proper set of values assigned to the input quantities for the validation of the software under analysis. In particular, the achieved test set aims to be a plausible generalization of the experimental combinations previously measured into the overall input domain of interest. The proposed testing methodology can be considered to be a useful tool for estimating the true performance of the developed software product in satisfying the assigned functional requirements.

As an example, let us consider a software developed for the diagnosis of the instruments adopted for measuring the DC power on a variable load in an electric circuitry. In particular, three instruments are adopted for respectively measuring the voltage, the current, and the power wasted by the load (see Fig. 1).

An analytical relationship, i.e., a strong correlation, exists among the measured quantities, i.e.,

$$P_m = V_m \cdot I_m \tag{1}$$

where P_m, V_m, and I_m are the quantities measured by the wattmeter (in watts), the voltmeter (in volts), and the current probe (in amperes), respectively.

Independently on the particular instrument fault detection and identification technique implemented by the diagnostic software (adoption of analytical redundancy rather than of a suitably trained neural network), a testing phase has to be carried out to verify its ability in revealing the status of the measurement instruments under multiple operating conditions (due to the different power supply and/or load variations).

The suggested methodology can be easily adopted to evaluate an important performance index of the diagnostic software, i.e., the false alarm percentage.

To this aim, the collection of V_m, I_m, and P_m during a faulty-free instrument status and in several operating ranges is required. Starting from this data set, the probability distributions and rank correlations between the input quantities V_m, I_m, and P_m are estimated and successively used to generate a new test set to be submitted to the diagnostic algorithm. It will be constituted by a finite number of statistically significant plausible combinations (triples of V_m, I_m, and P_m), satisfying the existing correlations [i.e., (1)]. Since this new test set trains the diagnostic software with input combinations corresponding to instrument faulty-free conditions, the eventual detection of a fault would reveal a wrong behavior of the diagnostic software (false alarms).

It has to be noted that even if an *a priori* evaluation of the input quantity probability distribution and rank correlations could be derived by eventually available theoretical models, an experimental measurement campaign is preferable to also take into account random effects present in practice, such as noise and disturbance superimposed on the electrical signals measured by the three instruments. This way, a more realistic and effective testing phase is carried out.

III. Toward a Substantial Enhancement

The aforementioned procedure was applied to test the instrument diagnostic software of a commercial car engine. The generation of a test case representative of the sensor faulty-free conditions allowed the robustness of the diagnostic software to be evaluated through the detection of eventual false alarms [16]. The obtained results have particularly highlighted the correctness of steps c)–g). Increasing the test set size M, the input values chosen as test cases are more closely distributed to the starting cdfs, and the minimum value of MAD (the index introduced to evaluate the efficacy of the correlation induction process) decreases, therefore evidencing the procedure capability of reproducing the real input probability distributions.

On the other hand, the evaluation of the experimental correlations (step b) is a very critical activity, particularly when the variables considered are strictly related. Indeed, the estimated correlation could strongly depend on the particular data measurement campaign performed (step a), particularly when some input regions present high occurrences and nonnegligible noise. Such conditions can invalidate the rank correlation. As an example, given two linearly related variables [as depicted in Fig. 2(a)], the evaluation on a time interval, during which the variables are held constant with the only variation due to noise, gives a Spearman coefficient that is quite far from the expected unitary value because of possible wrong associations among ranks [as highlighted in Fig. 2(b)].

Since the goal of the research activity was to realize a test procedure that is as general as possible, a detailed planning of the measurement campaign for the acquisition of the experimental data (step a), which are indispensable to estimate input distributions and correlations, is not supposed to be a user requirement. Thus, to avoid the aforementioned problems on

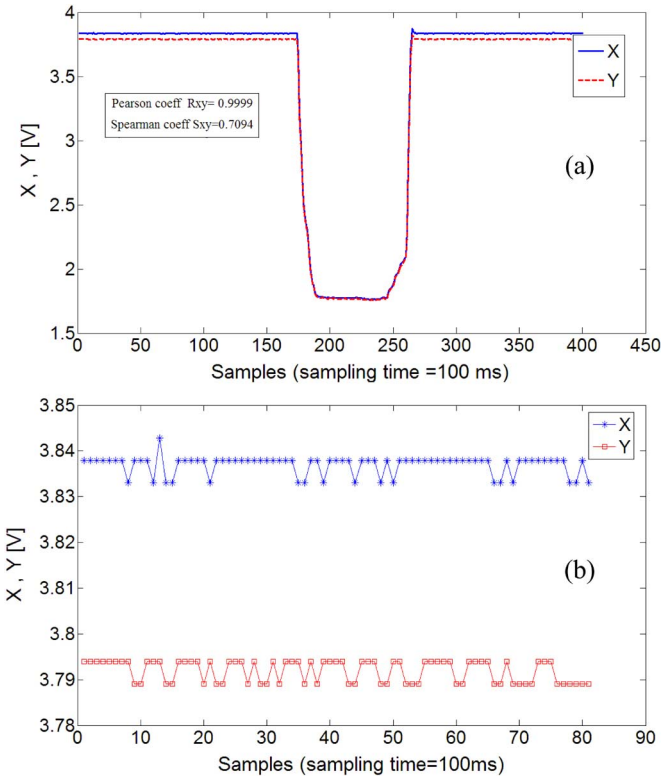


Fig. 2. Example of the two variables X and Y linearly related. (a) Evolution of X and Y versus time. (b) Magnification of a zone critical for the estimation of the rank correlations.

the correct estimation of the rank correlation, an improvement of step b) in the methodology has been studied.

In particular, two further intermediate steps are proposed in this paper: 1) filtering of the experimental data set and 2) new estimation of the target matrix.

A. Filtering of the Experimental Data Set (Step b1)

Starting from the correlation matrix obtained from the whole experimental data set (step b), the proposed guideline introduces a further step aimed to achieve a suitable data subset for estimating an accurate target, which will be adopted in the correlation induction process (steps e–f).

This subset is obtained using the procedure that follows.

1) generation of the class histograms (e.g., using the Scott's rule employed in a Matlab environment) of the two most strongly correlated variables $v1$ and $v2$, thus obtaining N_{c1} and N_{c2} classes, respectively;
2) evaluation of the minimum absolute occurrence for both variables, i.e., $O_{\min 1}$ and $O_{\min 2}$, respectively;
3) choice of $O_{\min} = \min(O_{\min 1}, O_{\min 2})$;
4) randomly selecting $O_{\min}$ values from each class of the variable corresponding to $O_{\min}$;
5) for each of the other variables and from the whole data set, selecting the values at the same time instant of those selected in item 4.

As for the size of the achieved subset, denoting N_c as the number of classes obtained for the histogram of the variable corresponding to $O_{\min}$, the coefficient estimations are made on

a number of samples given by $O_{\min} \cdot N_c$, uniformly spaced in the N_c classes. This is equivalent to reducing the number of equal valued samples with the same ranks, in which, because of noise, samples with the same ranks of the other strictly correlated variables do not correspond. Due to this filter action, the estimation of the correlation coefficients will be more robust to the previously mentioned noise effects.

A verification of the filtering effectiveness could be based on the evaluation of an additional statistic, such as the Pearson coefficient between $v1$ and $v2$. Indeed, if $v1$ and $v2$ are linearly dependent, both rank correlation and linear correlation coefficients should be very close to unity. Nevertheless, as previously mentioned, in these cases, the Spearman coefficients could underestimate the actual correlation among $v1$ and $v2$ because of noise on the data. On the contrary, the Pearson coefficients offer a more robust statistic whenever $v1$ and $v2$ are strongly linearly correlated. Consequently, the method proposed to confirm the correctness of the filtering phase is based on the verification that the Spearman coefficients, which are evaluated on the new subset (in step b2), are closer to the Pearson coefficients than those estimated on the whole data set.

B. New Estimation of the Target Matrix (Step b2)

The target rank correlation matrix will then be achieved on the data subset obtained in step b1). However, a confidence interval has to be determined for each estimated coefficient to take into account its variability. Some methods are proposed in the literature for this task [22]–[25].

In [22], let θ_0 be the estimate of a population Spearman correlation on the available data set. Assuming asymptotic normality of θ, a large-sample 100% $(1-\alpha)$ confidence interval for θ may be approximated as

$$\theta_0 \pm z_{\alpha/2} \left(\sqrt{\frac{(1+\theta_0^2/2) \cdot (1-\theta_0^2)^2}{n-3}} \right) \tag{2}$$

where n is the sample size used for estimating θ_0, and $z_{\alpha/2}$ is the point of the standard unit normal distribution exceeded with probability $\alpha/2$. Equation (2) has been proved to be very accurate in evaluating the confidence interval for Spearman correlations under bivariate normality for $\theta_0 \leq 0.95$, also with small n (a few dozen observations).

An approach suggested in [23] and [24] to estimate confidence intervals for the Spearman rank correlation coefficient for nonnormally distributed data is called the bootstrap technique, which was introduced by Efron and Tibshirani [25]. The bootstrap is a computationally intensive statistical technique that allows one to make inferences from data without making strong distributional assumptions about the statistic that is calculated and/or the data, thus revealing its usefulness in situations like the software and medical contexts, where measurements are limited, nonnormally distributed, and/or contain extreme values.

The basic idea of bootstrap is adopting resampling with replacement (also known as Monte Carlo resampling) to estimate the statistic's sampling distribution, which can be used to estimate confidence intervals for that particular statistic.

In practice, the bootstrap technique is described here.

1) By resampling with replacement the original data set of size n, create m resampled data sets (also known as bootstrap samples) that contain the same number of observations n; as a result, in each bootstrap sample, each original observation may include zero, either once or multiple times.
2) For each resampled data set, compute the descriptive statistic of choice.
3) From the collection of m values obtained from the previous step, compute a confidence interval using one of the following options: the *normal approximation method*, the *percentile method*, and the *bias-corrected (BC) method*.

In particular, the *normal method* computes an approximate standard error using the sampling distribution resulting from all the bootstrap resamples. Thus, denoting as $\mathrm{SE}(\theta_{\mathrm{sample}})$ the approximate standard error, the 100% $(1-\alpha)$ confidence interval for θ is computed as follows:

$$\theta_0 \pm z_{\alpha/2} \cdot \mathrm{SE}(\theta_{\mathrm{sample}}) \tag{3}$$

where $z_{\alpha/2}$ is the point of the standard unit normal distribution exceeded with probability $\alpha/2$.

The *percentile method* uses the frequency histogram of the m statistics computed from the bootstrap samples; thus, the 100% $(1-\alpha)$ confidence interval for θ is estimated as

$$[\theta_{\alpha/2}; \theta_{1-\alpha/2}] \tag{4}$$

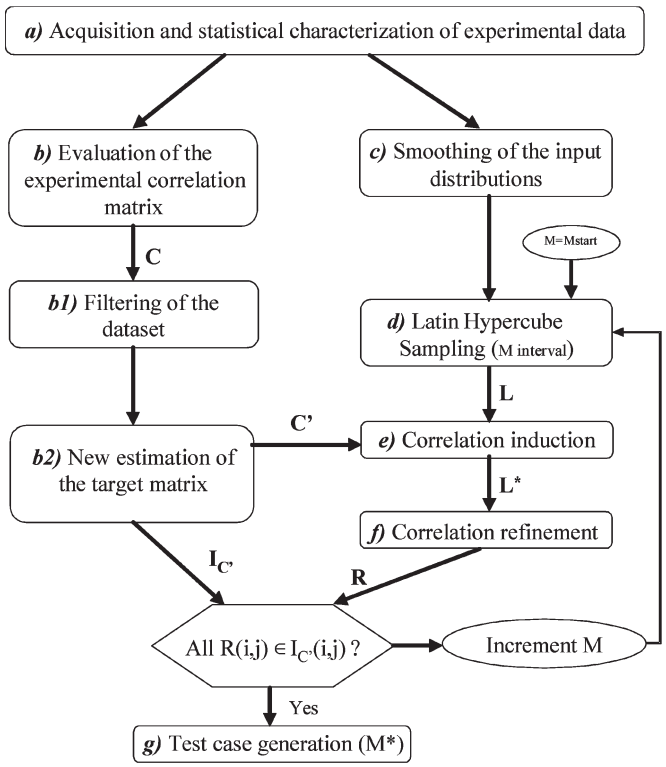
where $\theta_{\alpha/2}$ and $\theta_{1-\alpha/2}$ are the quantiles of the sampling distribution $F_m(\theta)$, i.e., $F_m(\theta_{\alpha/2}) = \alpha/2$ and $F_m(\theta_{1-\alpha/2}) = 1-\alpha/2$, respectively. The application of this method is also possible if $F_m(\theta)$ is not centered on θ_0.

As for the *BC method*, it is a slight modification of the percentile method that introduces a bias correction z_0 that allows the centering of $F_m(\theta)$ on θ_0 (this condition is also known as nondistorted $F_m(\theta)$). The 100% $(1-\alpha)$ confidence interval for θ is estimated as

$$\left[\theta^*_{\alpha/2}; \theta^*_{1-\alpha/2}\right] \tag{5}$$

where $\theta^*_{\alpha/2}$ and $\theta^*_{1-\alpha/2}$ are the "corrected" quantiles of the sampling distribution $F_m(\theta)$, i.e., $F_m(\theta^*_{\alpha/2}) = \Phi(2z_0 + z_{\alpha/2})$ and $F_m(\theta^*_{1-\alpha/2}) = \Phi(2z_0 + z_{1-\alpha/2})$, where $\Phi(x)$ is the cdf of the normal standard distribution, and $z_0 = \Phi^{-1}(F_m(\theta_0))$.

It has to be noted that whatever approach is adopted, once a confidence level is fixed, a variability interval is defined for each Spearman coefficient. In fact, the method adopted to estimate this interval leads one to choose the more appropriate statistic to be used as the target value. Moreover, if the interval estimation is carried out by means of (2), (3), or (5), the correlation coefficient evaluated on the subset must be considered as the target value, i.e., $C'(i,j)$. On the other hand, the use of (4) leads one to consider the median of the bootstrap sampling distribution as the target value. The adoption of a particular



Legend

C = correlation matrix (NxN) on the experimental data set

C' = correlation matrix (NxN) on the subset (target matrix)

$I_{C'}$ (i,j) = variability interval of C'(i,j) with i≠j

M = Number of LHS samples

L = matrix (MxN) of LHS samples

L* = correlation matrix (MxN) induced

R = correlation matrix (MxN) on L* samples

M* = Number of test case generated

Fig. 3. Flowchart of the new test procedure.

bootstrapping method does not seem to be a limitation but, rather, a choice that is dependent on hazard considerations related to the experimental campaign (i.e., how much you can guess on the goodness of the statistic Θ_o in representing the actual correlation).

Independently from the adopted method, the variability interval could be used to choose the minimum number of test cases needed to achieve an induced correlation matrix approaching the target value.

The flowchart of the new procedure, including steps b1) and b2), is reported in Fig. 3.

Once a starting M value, i.e., M_{start}, is defined, the correlation induction and correlation refinement algorithms reorder the LHS sample with the aim of approaching the target correlation matrix C'.

If all the elements of the inducted correlation matrix R are within the corresponding variability interval $I_{C'}(i,j)$ (with $i,j = 1,\ldots,N$, and $i \neq j$), the whole procedure is stopped; else, M is incremented, and the LHS and induction processes are repeated.

This strategy is very appropriate because it avoids the unnecessary increase in the number of test cases. In fact, the increase of M reduces the MAD value and consequently provides an inducted correlation matrix closer to the target value. Nevertheless, this accuracy level could be insignificant if compared with the estimated variability of the target coefficients, thus not improving the overall test reliability.

IV. Application Example

To verify the improvements introduced by steps b1) and b2), the procedure will be applied for testing the instrument diagnostic software previously described in [16].

This software system was developed for the diagnosis of some sensors employed in a commercial car engine: the manifold pressure sensor, the environmental pressure sensor, the intake air temperature sensor, and the two sensors used to detect the throttle valve position. Specifically, the ten software input quantities are electrical signals directly coming from the sensors (i.e., v-p1-vf, v-p2-vf, v-p-air, v-p-amb, and v-T-air) and measurement information (i.e., a-vf-d, a-cv-p, v-eng, T-eng, and pda-p) provided by other external modules, whereas the output quantities mainly refer to the sensor status (i.e., faulty or faulty-free). A new experimental data set constituted by 30 000 samples acquired in faulty-free conditions was used for the test methodology verification. It is applied to generate test cases representative of the sensor faulty-free conditions, in which the corresponding *false alarm percentage* FA% should be equal to 0%. Then, this index will give an indication of the goodness of the test methodology because in sensor faulty-free conditions, the false alarms arising in this case could only be imputable to the generation of wrong *test cases*.

All the steps except the data acquisitions were developed in a Matlab environment.

For the sake of brevity in the following, only the application of the new steps will be described in detail.

A. Filtering of the Experimental Data Set

Starting from the experimental correlation matrix C, as reported in Table I, it is possible to identify v-p1-vf and a-vf-d as the most strongly correlated variables (rank coefficient $= 0.99674$), and a high correlation also exists among v-p1-vf and v-p2-vf (rank coefficient $= -0.99299$).

Indeed, rank coefficients very close to unity are expected because v-p1-vf and v-p2-vf measure the same quantity; in the same manner, both v-p1-vf and v-p2-vf are strictly correlated to a-vf-d, which is the set point for the quantity measured by v-p1-vf and v-p2-vf. As a consequence, the steps described in Section III-A have to be executed to achieve the data subset on which to evaluate the target correlation matrix.

In particular, using Scott's rule employed in a Matlab environment, the class histograms for v-p1-vf and a-vf-d (hereinafter $v1$ and $v2$, respectively) were evaluated (Fig. 4). From their analyses, we obtained $O_{\min(v1)} = 30$, $N_{c1} = 31$, and $O_{\min(v2)} = 66$, $N_{c2} = 20$. As a consequence, $N_c = 31$ and $O_{\min} = 30$ were considered, and the size of the selected subset was $O_{\min} \cdot N_c = 930$.

TABLE I
EXPERIMENTAL RANK CORRELATION MATRIX C (10×10)

	p-pda	v-eng	T-eng	v-p1-vf	v-p2-vf	v-p-air	v-p-am	a-vf-d	a-cvcp	v-T-air
p-pda	1	-0.59836	-0.16778	-0.79239	0.79002	-0.68599	0.02559	-0.79505	-0.47535	-0.30799
v-eng	-0.59836	1	0.44330	0.58919	-0.57658	0.32014	0.04373	0.59051	-0.05504	0.56130
T-eng	-0.16778	0.44330	1	0.24719	-0.24419	0.14166	0.02850	0.24757	0.00463	0.41513
v-p1-vf	-0.79239	0.58919	0.24719	1	-0.99299	0.91956	-0.02368	0.99674	0.44623	0.43111
v-p2-vf	0.79002	-0.57658	-0.24419	-0.99299	1	-0.92215	0.03451	-0.99533	-0.45601	-0.41750
v-p-air	-0.68599	0.32014	0.14166	0.91956	-0.92215	1	-0.04013	0.92223	0.51962	0.33911
v-p-am	0.02559	0.04373	0.02850	-0.02368	0.03451	-0.04013	1	-0.02400	-0.10340	0.08330
a-vf-d	-0.79505	0.59051	0.24757	0.99674	-0.99533	0.92223	-0.02400	1	0.44777	0.43212
a-cvcp	-0.47535	-0.05504	0.00463	0.44623	-0.45601	0.51962	-0.10340	0.44777	1	0.03009
v-T-air	-0.30799	0.56130	0.41513	0.43111	-0.41750	0.33911	0.08330	0.43212	0.03009	1

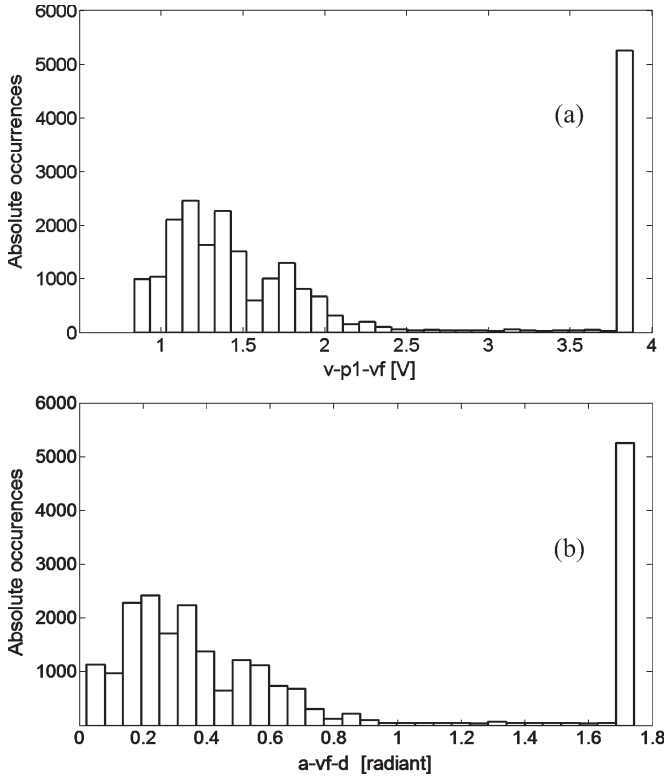


Fig. 4. Histograms of the occurrences for the two most related variables. (a) v-p1-vf. (b) a-vf-d.

B. Estimation of the Target Matrix

As an example, to avoid any hypothesis (such as asymptotic normality) on the distributions of the correlation coefficients $C'(i,j)$, among the available techniques, the *percentile method* is preferred for the evaluation of the variability intervals and the target correlation matrix. Consequently, the target matrix C' was achieved by considering the median of the sampling distribution; in this case, we evaluated $m = 500$ bootstrap samples (see Table II).

Fixing a 100% $(1-\alpha)$ confidence interval, the interval variability matrix $I_{C'}$ is obtained by applying (4). As an example, the matrix $I_{C'}$, with a fixed $\alpha = 0.05$, is reported in Table III.

As described in Section III, the verification of the filtering effectiveness is suitable. Then, the Pearson coefficients related to $v1$ and $v2$ were evaluated, obtaining $r(v1, v2) = 0.99998$ on the entire data set, thus revealing a linear relationship. As a consequence, the method suggested in Section III is applicable. Then, since $C'(v1, v2) = 0.99974$ (the Spearman coefficient evaluated on the subset) is closer to $r(v1, v2)$ than $C(v1, v2) = 0.99674$ (the Spearman coefficient evaluated on the whole data set), it is reasonable to consider that the subset achieved by the filtering phase is a better representative of the strict correlations existing between $v1$ and $v2$.

Similar considerations could be made for the other methods applicable for the estimation of the variability intervals of the correlation coefficients. Therefore, depending on the hypothesis made about the correlation coefficient statistical distributions, instead of (4), either (2), (3), or (5) could be applied to estimate the matrix $I_{C'}$ and the corresponding target matrix C'.

Whatever approach is adopted, once C' and $I_{C'}$ are estimated, the flowchart in Fig. 3 can continue with steps c)–f). The smoothing of the input distributions (step c) was performed by considering the Epanechnikov kernel and bandwidths, which are automatically selected by means of the Matlab statistical toolbox.

V. PERFORMANCE COMPARISON

To quantify the advantages offered by the proposed enhancements, the procedure performance is compared with the previous procedure version [16], which does not include steps b1) and b2).

For the fixed confidence level $\alpha = 0.05$, the whole procedure was repeated 50 times, achieving a mean value for $M^* = 72$. The generated test cases provided FA$\% = 1.25\%$ (mean value for the 50 procedure repetitions). Then, the procedure was run (50 times) without steps b1) and b2) (thus considering C as the target matrix), with $M = M^* = 72$. The mean false alarm percentage was FA$\% = 8.3\%$. This means that the new procedure actually increases the test reliability (in the ideal case, FA$\% = 0$).

By considering other values of $1-\alpha$ (i.e., 0.80, 0.5, 0.25, and 0.05), different values of M^* (i.e., 108, 294, 1288 and 4462, respectively) were found in correspondence, and similar comparisons can be carried out. The results are summarized in Table IV, where P_A denotes the procedure that includes step b1) and b2), and P_B denotes the previously proposed version [16].

As for P_A, the smaller the $1-\alpha$ value, the higher the M^*, and the better the FA%. This trend is expected since for smaller confidence levels $1-\alpha$, smaller variability intervals $I_{C'}(i,j)$ are achieved. Then, a greater number of test cases,

TABLE II
TARGET CORRELATION MATRIX C' (10×10)

	p-pda	v-eng	T-eng	v-p1-vf	v-p2-vf	v-p-air	v-p-am	a-vf-d	a-cvcp	v-T-air
p-pda	1	-0.52633	-0.20072	-0.67754	0.67744	-0.65773	0.03119	-0.67751	-0.32644	-0.39538
v-eng	-0.52633	1	0.51702	0.43974	-0.43811	0.21160	0.04478	0.43793	-0.15762	0.69521
T-eng	-0.20072	0.51702	1	0.29381	-0.29332	0.17137	0.03721	0.29309	0.02958	0.57389
v-p1-vf	-0.67754	0.43974	0.29381	1	-0.99994	0.90097	-0.02250	0.99974	0.40202	0.40686
v-p2-vf	0.67744	-0.43811	-0.29332	-0.99994	1	-0.90168	0.02317	-0.99972	-0.40348	-0.40528
v-p-air	-0.65773	0.21160	0.17137	0.90097	-0.90168	1	-0.02011	0.90179	0.47927	0.26713
v-p-am	0.03119	0.04478	0.03721	-0.02250	0.02317	-0.02011	1	-0.02229	-0.15642	0.12978
a-vf-d	-0.67751	0.43793	0.29309	0.99974	-0.99972	0.90179	-0.02229	1	0.40229	0.40493
a-cvcp	-0.32644	-0.15762	0.02958	0.40202	-0.40348	0.47927	-0.15642	0.40229	1	-0.06897
v-T-air	-0.39538	0.69521	0.57389	0.40686	-0.40528	0.26713	0.12978	0.40493	-0.06897	1

TABLE III
COEFFICIENT VARIABILITY INTERVAL MATRIX $I_{C'}$ (10×10) WITH $\alpha = 0.05$

	p-pda	v-eng	T-eng	v-p1-vf	v-p2-vf	v-p-air	v-p-am	a-vf-d	a-cvcp	v-T-air
p-pda	1	-0.56621 -0.48561	-0.25857 -0.14276	-0.71077 -0.64273	0.64264 0.71045	-0.68850 -0.62314	-0.03814 0.09275	-0.71058 -0.64269	-0.40132 -0.24335	-0.44267 -0.34808
v-eng	-0.56621 -0.48561	1	0.46838 0.56130	0.38092 0.49210	-0.49069 -0.37907	0.14040 0.27842	-0.01168 0.11836	0.37953 0.49082	-0.22397 -0.09457	0.65092 0.73578
T-eng	-0.25857 -0.14276	0.46838 0.56130	1	0.22888 0.35660	-0.35627 -0.22853	0.10206 0.23757	-0.02420 0.09171	0.22796 0.35619	-0.03536 0.09028	0.52592 0.61688
v-p1-vf	-0.71077 -0.64273	0.38092 0.49210	0.22888 0.35660	1	-0.99996 -0.99992	0.88179 0.91663	-0.08896 0.03982	0.99960 0.99982	0.32815 0.46957	0.35555 0.46290
v-p2-vf	0.64264 0.71045	-0.49069 -0.37907	-0.35627 -0.22853	-0.99996 -0.99992	1	-0.91730 -0.88240	-0.03958 0.08951	-0.99980 -0.99958	-0.47146 -0.33021	-0.46154 -0.35390
v-p-air	-0.68850 -0.62314	0.14040 0.27842	0.10206 0.23757	0.88179 0.91663	-0.91730 -0.88240	1	-0.08401 0.04477	0.88293 0.91754	0.41121 0.54251	0.20549 0.33194
v-p-am	-0.03814 0.09275	-0.01168 0.11836	-0.02420 0.09171	-0.08896 0.03982	-0.03958 0.08951	-0.08401 0.04477	1	-0.08859 0.04011	-0.21580 -0.09618	0.07233 0.19615
a-vf-d	-0.71058 -0.64269	0.37953 0.49082	0.22796 0.35619	0.99960 0.99982	-0.99980 -0.99958	0.88293 0.91754	-0.08859 0.04011	1	0.32820 0.46977	0.35309 0.46105
a-cvcp	-0.40132 -0.24335	-0.22397 -0.09457	-0.03536 0.09028	0.32815 0.46957	-0.47146 -0.33021	0.41121 0.54251	-0.21580 -0.09618	0.32820 0.46977	1	-0.13169 -0.00667
v-T-air	-0.44267 -0.34808	0.65092 0.73578	0.52592 0.61688	0.35555 0.46290	-0.46154 -0.35390	0.20549 0.33194	0.07233 0.19615	0.35309 0.46105	-0.13169 -0.00667	1

TABLE IV
PERFORMANCE COMPARISON OF P_A AND P_B

P_A			P_B		
(1-α)	*M**	*FA %*	*M*	*FA %*	*MAD*
0.95	72	1.25	72	8.70	12.1e-5
0.80	108	0.83	108	8.50	7.3e-5
0.50	294	0.58	294	8.32	2.5e-5
0.25	1288	0.49	1288	8.32	4.4e-6
0.05	4462	0.47	4462	8.45	1.2e-6

i.e., M^*, are necessary to assure that each $R(i,j)$ is internal to the corresponding $I_{C'}(i,j)$. Consequently, for smaller $1-\alpha$ values, the induced correlation matrix R is closer to the target, i.e., C', and the generated test cases are more accurate, as confirmed by the lower false alarm percentages.

As for P_B, the increase of M^* leads to the decrease in MAD, but FA% is practically constant and is always worse than P_A. These results confirm the goodness of the correlation induction stage (steps c–f) because of the MAD trend, but at the same time, they confirm the importance of steps b1) and b2) for improving the test case reliability and for optimizing the number of test cases.

Moreover, in P_B, the target matrix is C (see Table I). Therefore, referring to the highly correlated inputs, namely v-p1-vf and a-vf-d, a worse estimation of their linear dependence is carried out with respect to C' (in the ideal case, the Spearman coefficient should be very close to 1), and consequently, a greater number of wrong test cases are generated, as confirmed by the higher false alarm percentages.

VI. CONCLUSION

Improvements to an innovative instrument software test methodology have been proposed. Two intermediate stages have been added to the previous seven-step procedure. They are mainly concerned with the refinement of the estimated correlation matrix C. This goal is obtained by means of the following: 1) evaluation of the target correlation matrix C' on a subset attained by filtering the experimental data set and

2) employment of suitable statistical methods for the evaluation of the Spearman coefficient variability interval $I_{C'}(i,j)$.

In addition, a new exit condition of the iterative procedure, which is defined on the basis of the matrix $I_{C'}$, grants a minimum number of test cases without worsening the test reliability. In particular, the benefits offered by the proposed enhancements, which are verified on a complex software system developed for diagnostic applications in an automotive environment, confirm a meaningful reduction of both test cases to be considered, as well as false alarms. Further development will concern the application of the proposed test methodology to other instrumentation and measurement software.

REFERENCES

[1] "Statistical Software Engineering," *Panel on Statistical Methods in Software Engineering Committee on Applied and Theoretical Statistics Board on Mathematical Sciences*, 1996.

[2] *Standard for Software Verification and Validation,* IEEE Std. 1012-1998, 1998.

[3] W. S. Humphrey, *Managing the Software Process*. Reading, MA: Addison-Wesley, 1989.

[4] A. Ferrero, "Software for personal instruments," *IEEE Trans. Instrum. Meas.*, vol. 39, no. 6, pp. 860–863, Dec. 1990.

[5] G. Betta, C. Liguori, M. D'apuzzo, and A. Pietrosanto, "An intelligent FFT-analyzer," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 5, pp. 1173–1179, Oct. 1998.

[6] M. G. D'Elia, C. Liguori, V. Paciello, and A. Pietrosanto, "Software customization to provide digital oscilloscope with enhanced period-measurement features," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 2, pp. 493–500, Apr. 2006.

[7] H. Freeman, "Software testing," *IEEE Instrum. Meas. Mag.*, vol. 5, no. 3, pp. 48–50, Sep. 2002.

[8] T. Y. Chen, M. Y. Cheng, P. L. Poon, T. H. Tse, and Y. T. Yu, "A study on input domain partitioning," in *Proc. 20th IASTED Int. Conf. Appl. Inf.*, Calgary, AB, Canada, 2002, pp. 176–181.

[9] M. Ramachandran, "Testing software components using boundary value analysis," in *Proc. 29th Euromicro Conf.*, 2003, pp. 94–98.

[10] S. M. Phadke, *Quality Engineering Using Robust Design*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[11] S. Stoica, "Robust test methods applied to functional design verification," in *Proc. Test Conf.*, Sep. 1999, pp. 848–857.

[12] D. M. Cohen, S. L. Dalal, M. L. Fredman, and G. C. Patton, "The AETG system: An approach to testing based on combinatorial design," *IEEE Trans. Softw. Eng.*, vol. 23, no. 7, pp. 437–444, Jul. 1997.

[13] M. A. Bailey, T. E. Moyers, and S. Ntafos, "An application of random software testing," in *IEEE MILCOM, Conf. Rec.*, Nov. 1995, vol. 3, pp. 1098–1202.

[14] S. C. Ntafos, "On comparisons of random, partition, and proportional partition testing," *IEEE Trans. Softw. Eng.*, vol. 27, no. 10, pp. 949–960, Oct. 2001.

[15] J. Musa, "Operational profiles in software reliability engineering," *IEEE Softw.*, vol. 10, no. 2, pp. 14–32, Mar. 1993.

[16] G. Betta, D. Capriglione, and A. Pietrosanto, "A methodology to test instrument software: An application to the diagnostic in automotive systems," in *Proc. 22th IEEE Instrum. Meas. Technol. Conf.*, May 2005, vol. 1, pp. 245–249.

[17] G. Betta, D. Capriglione, A. Pietrosanto, and P. Sommella, "A reliable and robust methodology for testing measurement software," in *Proc. 23rd IEEE Instrum. Meas. Technol. Conf.*, Apr. 2006, pp. 2101–2106.

[18] R. Dandekar, M. Cohen, and N. Kirkendall, "Applicability of Latin hypercube sampling to create multivariate synthetic micro data," in *Proc. Exchange Technol. Know-how New Techn. Technol. for Statist.*, Crete, Greece, 2001, pp. 839–847.

[19] B. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

[20] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliab. Eng. Syst. Saf.*, vol. 81, no. 1, pp. 23–69, Jul. 2003.

[21] R. L. Iman and W. J. Conover, "A distribution-free approach to inducing rank correlation among input variables," *Commun. Stat.*, vol. B11, no. 3, pp. 311–334, 1982.

[22] D. Bonnet and T. Wright, "Sample size requirements for estimating Pearson, Kendall and Spearman correlations," *Psykometrika*, vol. 65, no. 1, pp. 23–28, Mar. 2000.

[23] S. Ley and M. R. Smith, "Evaluation of several nonparametric bootstrap methods to estimate confidence intervals for software metrics," *IEEE Trans. Softw. Eng.*, vol. 29, no. 11, pp. 996–1003, Nov. 2003.

[24] J. S. Haukoos and R. J. Lewis, "Bootstrapping confidence intervals for statistics with difficult distributions," *Adv. Stat. Academy Emergency Med.*, vol. 12, no. 4, pp. 360–365, Apr. 2005.

[25] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1998.

Giovanni Betta (A'91–M'92–SM'01) was born in Napoli, Italy, in 1961. He received the M.S. degree in electrical engineering from the University of Napoli in 1984.

In 1989, he joined the Department of Computer Science, University of Napoli, as an Assistant Professor of electrical measurements. In 1992, he became an Associate Professor of electrical measurements with the University of Cassino, Cassino, Italy, where, since 1999, he has been a Full Professor of electrical and electronic measurements and where is also the Dean of the School of Engineering. His current research interests include artificial-intelligence-based measurement systems, sensor realization and characterization, measurement systems for fault detection and diagnosis, and measurement software testing.

Prof. Betta is a member of the IEEE Instrumentation and Measurement Society.

Domenico Capriglione (M'04) was born in Cava de' Tirreni, Italy, in 1975. He received the M.S. degree (with honors) in electronic engineering from the University of Salerno, Fisciano, Italy, in 2000.

In 2001, he joined the Department of Automation, Electromagnetism, Information Engineering, and Industrial Mathematics (DAEIMI), University of Cassino, Cassino, Italy, as a Researcher, where, since 2001, he has been an Assistant Professor of electrical and electronic measurements. His current research interests include intelligent measurement systems for fault detection and diagnosis, measurement of electromagnetic compatibility, development of DSP-based measurement systems, and measurement software testing.

Prof. Capriglione is a member of the IEEE Instrumentation and Measurement Society.

Antonio Pietrosanto (M'99) was born in Napoli, Italy, in 1961. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Napoli in 1986 and 1990, respectively.

In 1991, he joined the Department of Information Engineering and Applied Mathematics, University of Salerno, Fisciano, Italy, as an Assistant Professor of electrical measurements and then became an Associate Professor of electrical measurements in 1999, where, since 2001, he has been a Full Professor of electrical and electronic measurements. His scientific interests are principally concerned with sensor realization and characterization, wireless instrument interface, digital signal and image processing, and instrument fault detection and isolation.

Prof. Pietrosanto is a member of the IEEE Instrumentation and Measurement Society.

Paolo Sommella was born in Salerno, Italy, in 1979. He received the M.S. degree (with honors) in electronic engineering in 2004 from the University of Salerno, Fisciano, Italy, where he is currently working toward the Ph.D. degree in information engineering.

His current research interests include intelligent measurement systems for fault detection and diagnosis, measurement in software engineering, and image processing for medical applications.