ARMADA^{CMS}

Creating the 'full picture' to enhance service productivity

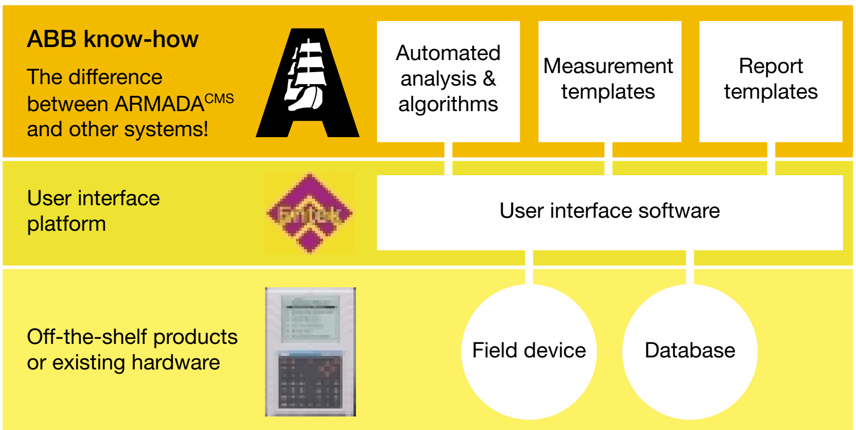
Maciej Wnek, Jukka Toukonen, Michal Orkisz, Kari Saarinen, Zbigniew Korendo, Maciej Zygmunt, Miroslaw Bistron, Cajetan Pinto

Winning a Total Motor Management contract for a large industrial plant with thousands of motors puts the onus on ABB to keep all the machines in good running order. Motor service is obviously an important factor in this – and a clear cost issue. ABB has every interest in ensuring that the service is as efficient as possible, which means getting the timing just right.

ARMADA^{CMS} was developed to take on just this kind of task. It stores the full range of machine data – currents, temperatures, drive system analyses, and much more – in one and the same database. This gives users the 'full picture', and tells them immediately when a problem has developed and how serious it is.

What is needed to efficiently handle a scenario like the one above is a preventive maintenance program that goes by the motto *not too early, not too late.* The condition of the machines has to be constantly monitored and they have to be serviced at just the 'right time'. But when is the right time? Not too early, because money is then spent unnecessarily on machines with some useful life still in them; and certainly not too late, because the consequences of equipment failure could be disastrous.

Also, how do you keep track of thousands of machines? Just cataloging them is a daunting enough task, let alone collecting the condition data.



1 ARMADA^{CMS} architecture

What, why and when – answered!

ARMADA^{CMS} is a software system designed to help in managing this task. Features include templates for different motor types indicating what to measure and when, and automated analysis for diagnosing the most common problems without human intervention. It also uses a common data format that allows an especially difficult case to be sent for evaluation to an expert anywhere in the world.

And that's not all ARMADA^{CMS} does. For example, it combines the handling of vibration data with measurement, storage and analysis of other aspects of rotating machines – eg, supply current spectra and insulation measurements. In addition, there is a report module for producing infinitely customizable reports in RTF format, which can be read by most word processors, like MS Word.

ARMADA^{CMS} is not an independent software system 1, but a set of algorithms and templates for use with standard software which provides the user interface, graphic functions and data handling and storage capability needed for communication with data collectors, database, plots, etc. This approach allowed ABB to focus on adding value to already existing systems, rather than try to 'reinvent the wheel'.

One of the main benefits of implementing ARMADA^{CMS} is that it helps field specialists work more productively, for example by speeding up data analysis and reporting processes – tasks that can take up as much as 40% of their working day.

In order to gage the productivity benefits of ARMADA^{CMS}, comparative tests were performed using ARMADA^{CMS} and a conventional system from a competitor. These tests were carried out at a soft tissue paper mill in Finland. The measurement route consisted of four squirrel cage motors, whose data were measured using the same SKF CMVA10 data collector; these data were then loaded into a computer on which both software systems were run. The tests indicated a 38% increase in speed when ARMADA^{CMS} is used to set-up, perform and report vibration tests. However, ARMADA^{CMS} is capable of much more than this test showed, so it could not be said to be a truly comprehensive comparison.

Condition monitoring with ARMADA^{CMS}

ARMADA^{CMS} has several unique features that simplify the way condition monitoring is carried out. These include some novel algorithms that are used to detect and classify faults.

Plant configuration

First, a database has to be created and the data of all the machines of interest entered in it. Most available systems, including Entek's Odyssey, upon which ARMADA^{CMS} is built 1, provide a hierarchical structure that reflects the plant layout, eg plant /building/floor/ assembly/machine. This helps the user to group the machines logically. While most systems let the user create templates to help with configuring the database, ARMADA^{CMS} goes a step further: It provides a set of templates not only for the motors and other rotating machines such as pumps, fans and gearboxes, but also for the electrical mains and equipment like frequency converters. What is more, these templates cover a wider range of data – from mechanical measurements and alarm levels to electrical values, temperatures, insulation data, and so on. With these templates, users can be sure of a significant productivity gain already during the system configuration.

Route definition

The next step is to define the routes. These are sets of machines and locations scheduled to be visited and inspected on a given day. Their creation is largely standardized today. In ARMADA^{CMS} (where they are defined by the Entek platform) the routes form a hierarchy based on physical locations, and reflect the structure found in the plant database. The route is then transferred from the computer to a data collector (eg, SKF CMVA). ARMADA^{CMS}, through Odyssey, supports most commercially available collectors.

Measurements

The route is loaded into a data collector, which then prompts the technician to perform various pre-programmed measurements. Descriptive messages provided by ARMADA^{CMS} templates help avoid sensor misplacement. The route conveys not only sensor locations but also all measurement settings, such as frequency ranges, resolution and so on. Vibration measurements are performed with an accelerometer connected directly to the collector's input. Similarly, supply current measurements are performed using a current clamp. Although collected data can be reviewed in the collector on-site, it is best to transfer them back to the plant database. The Entek environment allocates the data to the appropriate database records so that the current spectrum is not confused with the high-frequency time-waveform recorded at the drive-side bearing. In the next step, which is where the advanced condition analysis really begins, ARMADA^{CMS} algorithms are called from Odyssey's menu.

Analysis

Automatic analysis was one of the main goals during development of ARMADA^{CMS}. Traditionally, machine conditions are analyzed by human interpretation of the spectral data.

Different frequency-based signals (spectra) are compared with diagnostic charts. (The latter can have a variety of sources, for example the software vendor, technical literature, consultants, or personal experience.) An approach of this kind is time-consuming and obviously relies heavily on human expertise.

Unlike many software systems available on the market, ARMADA^{CMS} automatically applies a series of algorithms to both spectral and time-series data.

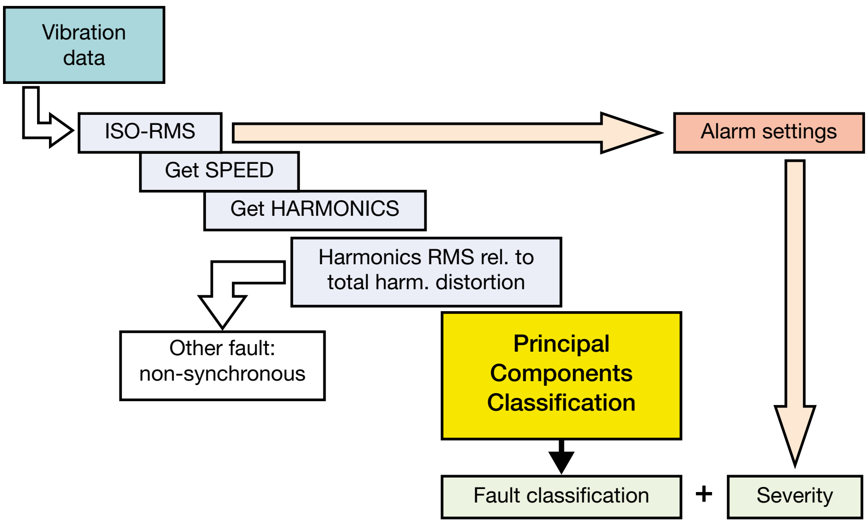
Since cases exist where not even experts with many years' experience can reliably diagnose the problem, it is unreasonable to expect an automated system to be able to resolve every diagnostic problem. Just the same, a diagnosis of even 70% of the most common problems is still an enormous help, and when several thousand machines are involved it will bring about a major productivity gain. The comparative test carried out at the soft paper tissue mill in Finland indicates that ARMADA^{CMS} reduces by about 60% the time a moderately experienced technician needs to carry out an analysis using traditional software.

To show how ARMADA^{CMS} works, a look is taken in the following at a set of algorithms dedicated to the classification of some typical defects experienced with motors. Statistically, the most important of these are misalignment, unbalance, and mechanical looseness. All of these faults can be classified with ARMADA^{CMS}, which can also detect whether there is another type of fault present or, for example, if the fault happens to be unrelated to speed harmonics. The described analysis is based on vibration measurements.

Fault classification platform

Every kind of classification system depends on high-quality, reliable and representative data for accurate results. Obtaining a good representation of a machine's state for diagnostic purposes, however, often requires large numbers of input signals (eg, current and vibration). Since thousands of measurement samples may be taken for just one motor, it is necessary to reduce the amount of input data to make it more manageable and rid it of content irrelevant for the diagnosis.

This transformation provides a more compact data set while retaining all the distinctive features needed for the



2 The vibration analysis module of ARMADA^{CMS} performs condition assessments and identifies faults responsible for the motor problems that usually manifest themselves in excessive vibration.

Analysis follows two paths: fault classification and condition severity assessment. The latter is based on an overall level of vibration velocity (RMS).

classification. Quite often, these distinctive features (in this case fault symptoms) are not even known *a priori*. The signal space transformation helps to extract this information from the reference data pool.

ARMADA^{CMS} uses Principal Components Analysis (PCA) [1] to perform the signal space transformation. The actual signal projection is preceded by non-linear preprocessing and the formation of multidimensional feature vectors.

The vibration analysis module (VAM) of ARMADA^{CMS} 2 performs condition assessments and recognizes those types of fault responsible for the motor problems that usually manifest themselves in excessive vibration. The tool is designed to diagnose motors with power ratings between 15 kW and 2 MW and speeds above 300 rpm. Analysis follows two paths: *fault classification* and *condition severity assessment*. The severity assessment, which is carried out independently of the fault recognition, is based on an overall level of vibration velocity (RMS).

If the vibration exceeds the set limits the fault classification routine is started. Otherwise, the motor condition is considered to be normal. (Bearing tests are nevertheless run, irrespective of the overall vibration level.) Fault recognition is based on vibration pattern analysis using a method known as 'Principal Components Clustering'. Prior to classification, the shaft speed is calculated from the vibration data. This involves calculating the amplitudes of vibration velocity at fundamental frequency (ie, the shaft speed) and its higher harmonics. Later, a test is performed to check whether the overall vibration originates from forces induced by shaft rotation (so-called 'synchronous faults', such as misalignment or unbalance) or is caused by other system components, eg bearings, structural resonance or a belt drive (known as 'non-synchronous faults'). It is possible in severe cases for harmonic components to be the sole cause of excessive vibration, even though the non-synchronous components contribute most to the overall vibration. The VAM will also detect this and start the fault classification routine.

Vibration measurements carried out on a faulty rotating machine yield complex signals with periodic and aperiodic components that depend on the machine's structure and on the type of fault. The faults are caused by forces which are induced by shaft rotation, and therefore depend on the shaft speed.

The orthogonal expansion of sinusoidal signals (ie, a Fourier series) is commonly used to represent the periodic signals $x^j[n],\ \ j \in \{a,h,v,nd\}$ and is described by the formula:

$$x^j[n] \approx \sum_{m=1}^{M} A_m^j \cos(2\pi m f_r n + ø_m^j),\ j \in \{a,h,v,nd\}$$

where n is the sample index and f_r, A_m^j, $ø_m^j$ are the rotation frequency, amplitude and phase, respectively. The superscripts $\{a,h,v,nd\}$ denote measurement locations: a for axial, h for horizontal, v for vertical and nd for the non-driven end radial direction.

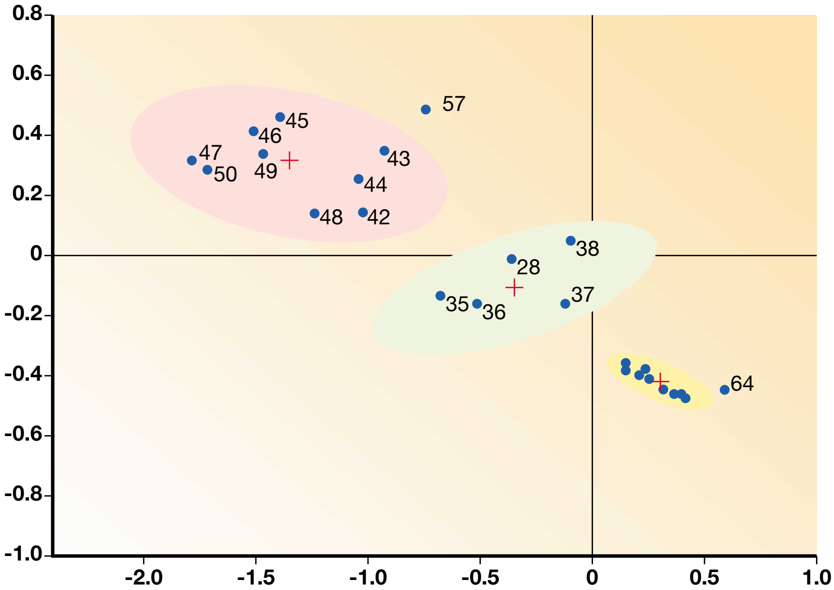
Low computational complexity and a lack of *a priori* assumptions about the signal are factors that speak for use of the Fourier transform-based method (periodogram) for orthogonal expansion parameter estimation. Its major drawback is the high variance and low resolution, which slows down signal collection considerably.

An averaged periodogram method can be used to improve the statistical properties of the periodogram, but this would require even longer measurement times.

A drawback of low-frequency measurements for shaft speed estimation is that they take a long time, during which the rotating frequency can change significantly. This affects the accuracy of the Fourier method. To solve the problem, high-resolution parametric methods were developed [2, 3], one of them – the maximum-likelihood method – being considered as an option.

As an alternative to the traditional Fourier-based approach, ABB applies the time-domain maximum-likelihood method [4] to obtain accurate speed estimations from short-time measurements. The amplitudes A_m^j and phases $ø_m^j$ are calculated with the estimated rotating frequency, using the least squares method [2].

The actual fault recognition starts with a statistical orthogonal expansion, followed by PCA and subsequent cluster analysis [5]. The reference data set was created from a comprehensive program of laboratory experiments and carefully controlled factory floor data. Through extensive numerical analysis it has been



3 Fault clustering with the vibration analysis module of ARMADA[CMS]

Red cluster *Misalignment*

Green cluster *Looseness*

Yellow cluster *Unbalance*

<Client>	<$custname>
<Address>	<$custaddr.1> <$custaddr.2> <$custaddr.3>
<Postal_Code>	<$custpost>
<Contact>	<$custcontact>
<Telephone>	<$custphone>
<Fax>	<$custfax>

ABB

<Machine_Name>	<_name>	**<IF vibrms><Vibration_RMS**	<vibrms>
<Description>	<_descript>		
<Area>	<_grandpt>		
<Position>	<_parent>		
<Criticality>	<_criticality>		
<Last_modification>	<_modtime>		

<CHARACTERISTICS>	<$objtype>		
<name.i>	<val.i>	<name.(i+n)>	<val.(i+n)>
<name.i>	<val.i>	<name.(i+n)>	<val.(i+n)>
<BEARINGS_SHAFT>			
<name.i>	<val.i>	<name.(i+n)>	<val.(i+n)>
<OTHER>			
<name.i>	<val.i>	<name.(i+n)>	<val.(i+n)>

4 A typical template used to report vibration data

possible to substantially reduce the measured data input for fault classification from a 43-element feature vector to a two-coordinate representation which can be plotted on an *x*-*y* graph 3. Each dot on this graph represents a single test case (machine). A fault is indicated if a dot corresponding to the examined machine falls within any of the fault clusters.

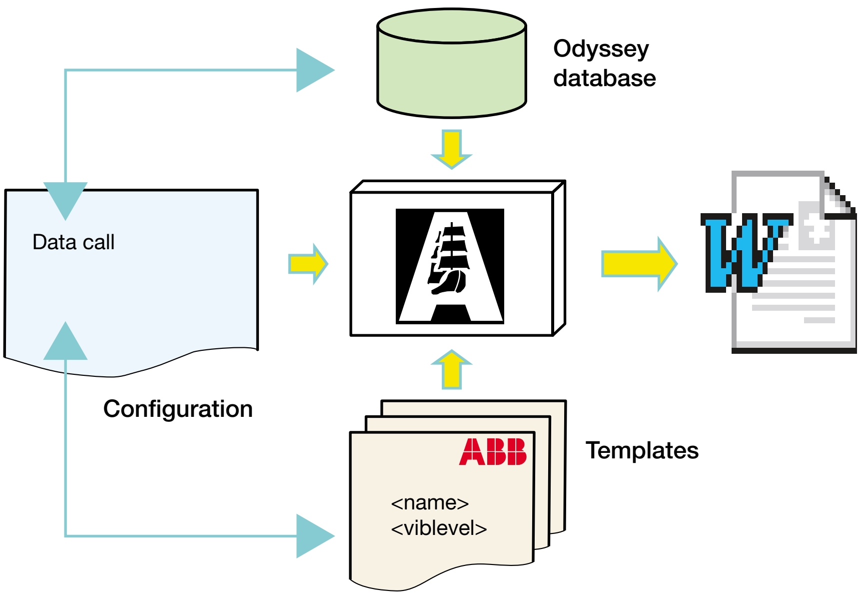
Reliability

Besides the classified fault type, the VAM provides a reliability value ranging from 30% to 100%, indicating how confident the user can be that the classification is correct. Values below 100% indicate that the classification does not lie within the fault cluster. The further away the dot is from the cluster's center, the lower the reliability will be.

For example, in test case 48 the fault is classified as a 'misalignment' with 100% reliability. In case 64, on the other hand, the classification is 'unbalance', this time with a reliability of 83%. Looking at test case 57, it can be seen that it is rated 'not categorized' – the result of its reliability being only 13.4%.

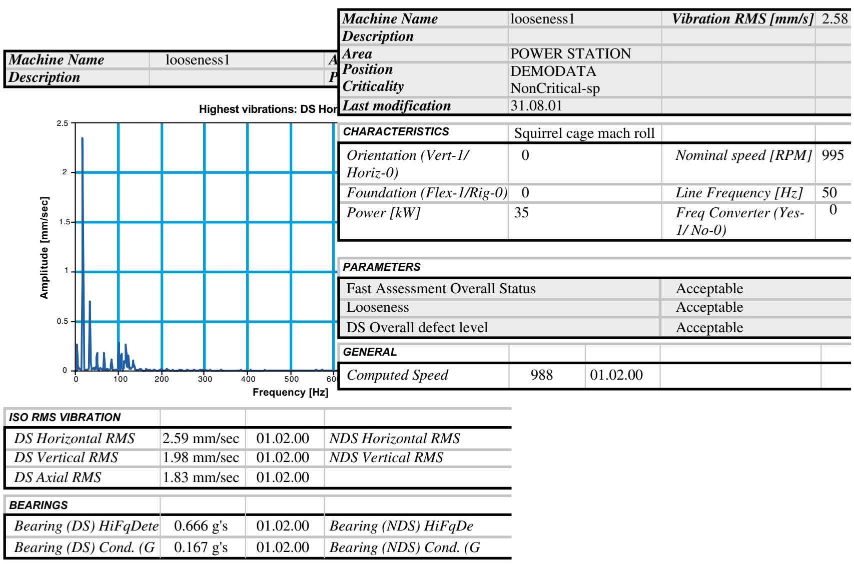
Reporting

Another key goal during development of ARMADA[CMS] was automated reporting. Most condition monitoring software providers focus more on managing information within the system than on convenient ways of exporting it to other systems. Such tools are therefore generally cumbersome. In contrast, the proprietary ARMADA[CMS] reporting system is hugely flexible. The final report is



5 One-button reporting

ARMADA^CMS retrieves all the data it uses for reporting from the database. The mechanism, which is completely transparent to the user, is based on standard SQL calls. SQL queries link variables used in templates with the true data from the database. Documents are in standard RTF format.



6 Sample reports generated by ARMADA^CMS

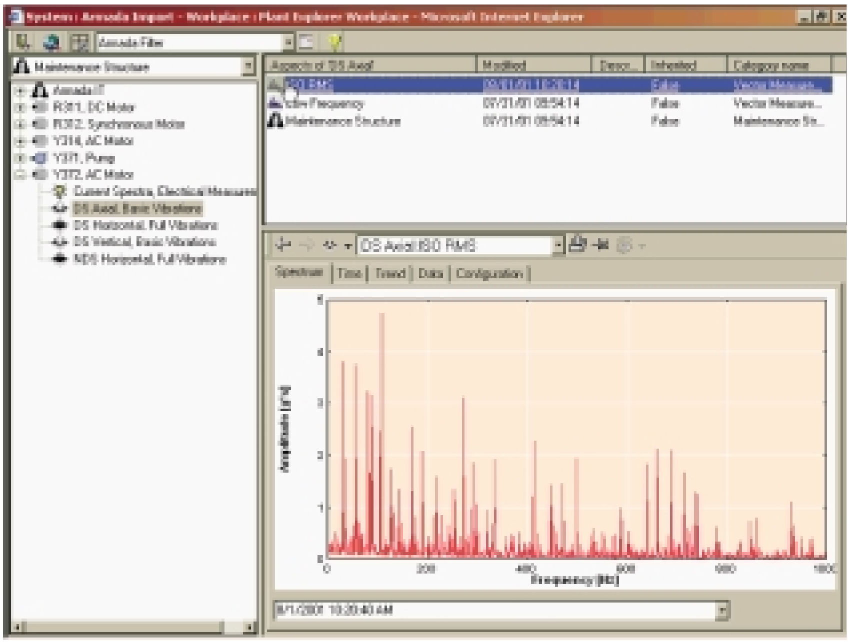
configured in an RTF (Rich Text Format) file, for example using MS Word or any other of the popular word processors.

The report template is stored as a separate 'document'. Instead of values, this document contains variable names which are subsequently replaced by diagnostic data from the database. To enhance the design of the templates ABB has created a simple scripting language to allow use of loops, 'if' statements, etc, that make the table size, number of graphs and so on variable and dependent on the database content 4. Graphs can be inserted in the templates as ActiveX MS Graph objects – something which all MS Office users will be familiar with. These allow for customization of the line thickness, point markers, axis sizes, etc.

All the data ARMADA^CMS uses in reporting is retrieved from the database. The mechanism, which is completely transparent to the user, is based on standard SQL calls. SQL queries, which are defined in a separate file, link variables used in templates with the true data from the database. Such an approach makes it easy to internationalize the reporting system. Five languages – Spanish, German, English, Portuguese and French – are currently supported.

This has resulted in a key feature of ARMADA^CMS – 'one-button' reporting 5. The documents are in standard RTF format. Since the templates are RTF files, customized reports can be easily created and added to the existing library for sharing within the organization 6. This proprietary reporting system, which uses the standard database and standardized

7 Embedding ARMADA^{CMS} in ABB's industrial^{IT} environment creates a scenario in which the machines themselves convey a plant's service and maintenance data, enabling the most effective kind of condition monitoring.



reports, reflects company requirements in the field.

Limitless possibilities

ARMADA^{CMS}, being a modular system, has virtually limitless expansion capability. For example, modules may be developed in the future for diagnosing industrial fans or gearboxes. And the scope of recognizable faults can be broadened as more reference data become available. Also, since most of the existing ARMADA^{CMS} algorithms are based on statistical observations, results from each new case can be fed back to further improve their reliability. This is made easy by ABB Group-wide data format standardization with ARMADA^{CMS} templates.

Embedding ARMADA^{CMS} into ABB's Industrial^{IT} environment is another area of application with enormous potential **7**. Through a powerful new enterprise architecture called Aspect Objects^{TM}, each component of a drive system, right down to its bearings, rotor, coupling, etc, is enabled to carry information specific to certain diagnostic tools. Examples are relevant nameplate data, measurement settings, processing algorithms, and so on. With this architecture in place, an induction motor will ignore a service engineer's request for information, eg about its 'commutator brush condition', since it 'knows' that, being an induction motor, it does not have a commutator.

In such a scenario, widely distributed items of diagnostic know-how are put together to form a truly powerful solution. Thus, through Industrial^{IT}, ARMADA^{CMS} would enter a world where the machines themselves are capable of conveying a plant's service and maintenance data – surely the key to truly effective condition monitoring.

All trademarks are the properties of their owners.

Authors

Dr. Maciej Wnek
Dr. Michal Orkisz
Dr. Zbigniew Korendo
Dr. Maciej Zygmunt
Miroslaw Bistron
ABB Corporate Research
PL-31 038 Krakow, Poland
maciej.wnek@pl.abb.com
http://www.abb.com/research

Jukka Toukonen
Kari Saarinen
ABB Corporate Research
FI-65101 Vaasa, Finland
jukka.toukonen@fi.abb.com

Cajetan Pinto
ABB Lenzohm Service Ltd
Mumbai 400 088, India
cajetan.t.pinto@in.abb.com

References

[1] **I. T. Jolliffe**: Principal Component Analysis. Publ: Springer Verlag, USA.
[2] **S. M. Kay**: Modern Spectral Estimation: Theory and Application. Publ: Prentice-Hall, UK.
[3] **S. J. Marple**: Digital Spectral Analysis with Applications. Publ: Prentice-Hall, UK.
[4] **K. Saarinen**, **M. Orkisz**: Finnish patent 20000646 (2001).
[5] **K. Saarinen, Z. Korendo**: Polish patent 338286 (2001).