Isidro CALVO[1], Oier GARCIA DE ALBÉNIZ[1], Federico PÉREZ[1]

University of the Basque Country (UPV/EHU) (1)

A communication backbone for Substation Automation Systems based on the OMG DDS standard

***Abstract**. Today's Substation Automation Systems (SAS) are formed by diverse microprocessor-based devices that carry out several operations such as metering, monitoring, protection and actuation. Standards, such as IEC61850, aim at providing abstract component-based solutions that improve scalability and ease the replacement of the devices. This paper discusses the different types of communication traffic identified in this standard and proposes a mapping to use the recent OMG Data Distribution Service (DDS) standard as communication backbone in SAS.*

***Streszczenie.** Podstacje system automatycznego SAS są sterowane przy wykorzystaniu układów mikroprocesorowych zapewniających pomiar, zabezpieczenie i zadziałanie układów wykonawczych. Wymianę kompenentów ułatwia standard IEC61850. W artykule analizowano różne typy komunikacji w ramach tego standardu oraz zaproponowano system DDS (Data Distribution Service) jako szkielet system SAS. (**Szkielet komunikacyjny systemu SAS bazujący na standardzie OMG DDS**)*

Keywords: IEC61850, Real-Time Communications, Data Distribution Service (DDS),
Słowa kluczowe: IEC61850, komunikacja w czasie rzeczywistym.

Introduction

Both increasing competition and electric utility deregulation have introduced fundamental changes in the technologies used in the Substation Automation Systems (SAS) during the latest years. As a consequence, utilities have been forced to adopt modern technologies, such as powerful hardware platforms and data communication networks, in order to reduce costs. This scenario has been analyzed in [1] where several trends have been identified in the development of the new SAS. For example, protection relays have evolved into multifunctional universal devices, frequently known as Intelligent Electronic Devices (IEDs), which are capable of carrying out several operations such as metering, monitoring, control and automation, in addition to traditional protection functionalities. Also, integration among the components that form a SAS has become a key issue. This requirement has been solved by the adoption of several international standards, such as IEC61850 [2], that define standard data models that ease integration and information exchange. In particular, the IEC61850 standard specifies a collection of elementary services that can be performed by SAS such as breaker operation, monitoring, or reporting. Last but not least, the state-of the art IEDs include new functionalities such as remote monitoring via TCP/IP technologies [3].

As a result of these changes, the new SAS are becoming complex distributed applications in which several modular IEDs are connected with modern data communication networks. These state-of-the-art IEDs are sophisticated real-time devices that must satisfy stringent constraints. Until quite recently the vendors of these devices had to use low-level programming techniques in order to obtain the optimal performance of the hardware platform. However, the adoption of more powerful hardware platforms allows the incorporation of modern programming techniques, already used in other embedded applications domains that improve many system desired properties such as fault-tolerance, scalability and maintainability. As a matter of example, nowadays this kind of devices are adopting Real-Time Operating Systems (RTOS) or improved versions of general purpose OS (like Linux RTAI) that are better adapted to modular approach and allow getting the maximum performance of the underlying hardware.

Regarding communications, during the last years some office networks, in particular Switched Ethernet or Ethernet-modified architectures, and TCP/IP based protocols are gaining increasing acceptance for creating distributed automation applications even at the field level. There are several reasons for this shift: (1) Intelligence is being moved to individual distributed components decentralizing the applications; (2) Vertical integration is gaining higher importance; (3) Strong demands for the use of IT standards that reduce the cost of the communication hardware. An interesting analysis of the suitability of Ethernet-based IP networks, even in demanding automated applications may be found in [4].

In this context, middleware solutions simplify the design of communications and solve the problem of the heterogeneity of platforms due to the existence of multiple non-compatible vendors. In addition, most middleware technologies can run over TCP/IP networks and typically over substation LANs using high speed switched Ethernet.

Some of these middleware technologies that allow the integration of control devices are CORBA, OPC, Web Services and DDS [5]. In particular, DDS (Data Distribution Service), which has been used in the present work, is a recent middleware specification defined by the OMG which follows the publisher/subscriber paradigm. DDS provides several mechanisms to specify and manage the Quality of Service (QoS) requirements in real-time communications.

Currently, there are several mappings of the IEC61850 standard over CORBA [6, 7], and some work is going on to map parts of the standard to Web Services, namely maintenance and system management data flows [8]. However, the nature of the communications in SAS requires mixing different traffic patterns. Particularly, some operations require a request/response following the client/server approach such as a breaking operation, whereas in others data must be delivered simultaneously to a set of recipients adapting better to the publisher/subscriber paradigm, such as delivering substation events. Unfortunately, most middleware technologies do not mix satisfactorily both paradigms, but they follow one of them (client/server in the case of CORBA vs. publisher/subscriber in the case of DDS). In addition, process level communications in SAS require guarantees that real-time data will be delivered without being jeopardized by the interference of non-critical data sent over the same media.

Most previous works address this problem partially. They use one of these technologies, typically CORBA, for client/server operations and customized, non-standardized services for the distribution of Sampled Measured Values

(SMV) and Generic Substation Events (GSE). Other works like [9] propose combining CORBA and DDS but this approach relies on expensive products that decouple both types of information traffic.

This paper analyses the different types of traffic identified by the IEC61850 standard and maps them over DDS. In the proposed solution, the problem of coexistence of different types of data traffic is solved by the DDS middleware by adjusting properly the rich set of QoS mechanisms offered by DDS according to the guidelines provided in the paper.

The layout of the paper will be as follows. Section 2 will describe the DDS specification with special attention on QoS aspects. Section 3 will describe the main aspects of the IEC61850 standard. Section 4 will describe the proposed mapping of IEC61850 entities into DDS as well as the configuration of the DDS QoS parameters in order to guarantee the real-time requirements of SAS. The paper will finish with some conclusion remarks.

Overview of DDS

The Data Distribution Service for Real-time Systems (DDS) [5] is an emerging specification released by the Object Management Group (OMG) that provides a platform-independent middleware for Data Centric Publish/Subscribe many-to-many communications. There is an increasing number of vendors, some of them open source that allow combining different programming languages (mainly C, C++ and Java) and operating systems (such as Windows, VxWorks, QNX, Lynx and other Unix/Linux derivatives).

Instead of exchanging data in the form of messages, DDS defines a virtual Global Data Space (GDS) which is accessible by all applications. Publishers produce/write data which is consumed/read by subscribers. These data are defined in a platform independent way inherited from the CORBA specification called IDL (Interface Definition Language). Publishers and subscribers are decoupled in time, space and synchronization. For example, late joiners may obtain persistent data issued before their activation even if the application which generated the data has already finished or its host has crashed.

The DDS specification [5] defines a topic-based API that involves several entities as represented in Fig. 1:

- **Topic** – Information unit that can be produced or consumed, identified by a name. It allows anonymous and transparent communications. Topic instances are associated with a key, defined in IDL, and a set of QoS.
- **Domain** – Communication context which provides a virtual environment, encapsulating different concerns and thereby optimizing communications.
- **Domain Participant** – Entity taking part of an application in a domain.
- **Data Writer** – Entity intending to publish a Topic, providing type-safe operations to write/send data.
- **Data Reader** – Entity intending to subscribe a Topic, providing type-safe operations to read/receive data
- **Publisher** – Entity created by a Domain Participant to manage a group of Data Writers
- **Subscriber** – Entity created by a Domain Participant to manage a group of Data Readers.

This technology is being adopted in many sectors: aerospace, defense, traffic control systems, SCADA or robotics and automation. Some works propose using DDS in multimedia applications [10]; other works describe its use in critical industrial applications [11]; also a few works deal with the use of DDS in factory automation applications [9, 12].

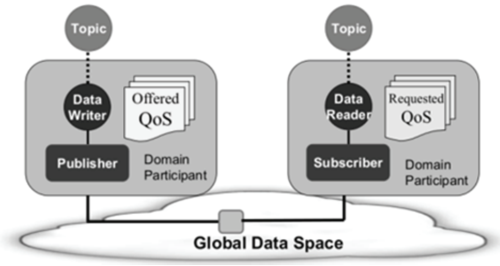

Fig.1. DDS Entities

Table 1. List of DDS Quality of Service (QoS) Policies

DDS QoS policies	
Deadline	Ownership Strength
Destination Order	Partition
Durability	Presentation
Durability Service	Reader Data Lifecycle
Entity Factory	Reliability
Group Data	Resource Limits
History	Time-Based Filter
Latency Budget	Topic Data
Lifespan	Transport Priority
Liveliness	User Data
Ownership	Writer Data Lifecycle

The IEC61850 standard

The IEC61850 standard [2, 13, 14], which is promoted by the International Electrotechnical Commission (IEC), addresses status of the art devices in SAS. In these devices the amount of data has increased dramatically, and several requirements must be guaranteed such as [15]:

- High-speed IED to IED communication
- Connectivity throughout the utility enterprise
- High-availability
- Guaranteed delivery times
- Standards based
- Multi-vendor interoperability
- Support for periodic voltage and current samples
- Support for File Transfer
- Configuration tools (auto-configuration)

Basically, this standard specifies a collection of elementary services that can be performed within a SAS: breaker operation, monitoring, reporting, etc. These services which may be implemented using different types of technologies are distributed in a collection of independent processors (RTUs, IEDs, station computers, etc.) The global behavior of the system is achieved by means of gathering a collection of such services in global system functions (Fig. 2 shows two examples of composition of a SAS).

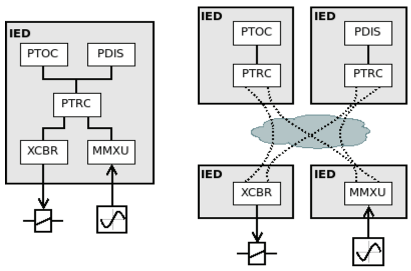

Fig. 2. Composition of a SAS from IEC61850 entities

The standard eases the interoperation of the devices on the basis of the *Abstract Communication Service Interface* (ACSI), defined as IEC61850 Part 7-2, which is a common abstract interface that provides a common set of communication functions such as data access, data reporting, data logging and control functions, which are found in most real-time utility field devices. Even though

ACSI allows discrete devices to share data and services, it is only an abstract application layer protocol without any real procedure for sending and receiving data. It can only be usable when it is mapped to standardized communication protocols that provide the real substrate for the implementation.

Other parts of the IEC 61850 standard (mainly 7-3 and 7-4) provide a hierarchical structure with several information models that represent in an abstract way the data that may be exchanged between the devices (Fig. 3). This approach provides an independent representation of the data not linked to a physical location or storage medium inside the device. These models are:

- **Server:** Represents the external visible behavior of a device. All other models inherit their structure from the Server model. The server may consist of one or more Logical Devices.
- **Logical Device:** It is basically a set of domain specific application functions. It is the way ACSI represents the capabilities of a real device. Every Logical Device contains typically one Logical Node, but occasionally there may contain more than one.
- **Logical Node:** They are used to represent idealized functional models of real devices. For example, a circuit breaker (XCBR). A Logical Node is a composition of Data entities.
- **Data:** It specifies typed information, for example, the position of a switch contained in a circuit breaker.

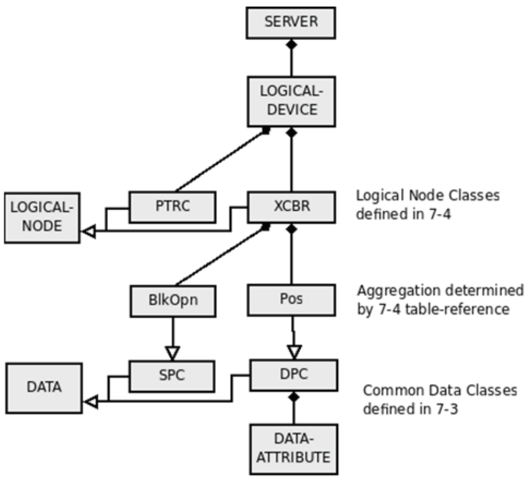

Fig. 3. Hierarchical structure of the IEC61850 standard

This object oriented modeling approach allows the adoption of object oriented programming languages like C++ for implementation purposes.

The Specific Communication Service Mapping (SCSM) describes the implementation details of services and models using a specific communication stack [14]. These services may be classified into three main groups:

- **Client / Server services**: Services that provide data under demand, i.e. one node (the server) provides the information requested by other nodes (the clients). Typically, these services do not require time-critical information exchange. Some of these services may require a response whereas others are one way.
- **Distribution of Substation Events:** These services follow a publisher / subscriber model, so when a certain event occurs in one node it must be propagated to one or more nodes. These services are time-critical and must detect any loss of data.
- **Periodic Services:** The aim of these services, which also follow a publisher / subscriber model, is the periodical distribution of sample data from where they are produced to one or more nodes.

Mapping ASCI over DDS

The different types of communication described above may be mapped into different paradigms. Traditionally, ad hoc point to point configured links have been used to communicate controllers in critical operations. Nowadays, middleware technologies that implement either client/server (C/S) or publisher/subscriber (P/S) paradigms are being adopted.

Thus, P/S paradigms, providing that certain QoS constraints are met, may be adequate to distribute both periodic and aperiodic process messages since they frequently require one-to-many or many-to-many distribution. A typical example is the simultaneous delivery of an event to a set of recipients. On the contrary, most non-real time operations are better mapped into the C/S paradigm: either in synchronous blocking operations or in asynchronous ones (without a response from the server).

In order to make different types of traffic coexist two acceptable alternatives seem to be possible: (1) adapting different technologies to allow them to coexist as proposed in [6, 7, 9] or (2) mapping the C/S traffic into the P/S paradigm. In the second approach, which has been followed in this work, efficiency will be lost but only on the request/response operations, which are frequently less critical.

The use of DDS as communication backbone for ACSI services provides certain benefits: (1) A unique middleware specification is used; (2) DDS provides a broad set of QoS parameters that, if properly configured, may be used to indicate the middleware how to deal with the different types of traffic; (3) there exist high-performance products available when compared with other middleware specifications and (4) it is a platform, language and vendor independent solution. However, this approach requires using mechanisms to execute request/response and request/non-response operations over a P/S middleware.

For example, DDS manages priorities according to the communication needs of the applications. Thus, non-Real-Time communication data will be mapped into topics with low priority values improving the traditional best-effort paradigm. Other types of traffic, such as the Sampled Measured Values, will be distributed by using a medium priority value thereby guaranteeing the basic update times. Finally, Generic Substation Events will use the highest priority levels allowed by the basic architecture upon which the communications stack is.

Since DDS is a P/S technology, using it as a communication backbone for all types of services described in the previous section (Client/Server, Distribution of Substation Events and Periodic Services) requires mapping them in terms of DDS. The proposed mapping is briefly described below:

- **Generic Substation Events (GSE)**: these events are directly mapped into the many-to-many P/S paradigm provided by DDS. Each event (or a fixed set of them) is mapped into a topic depending on the desired granularity of the system. On the other hand, each node interested in the reception of one event must subscribe to the topic that contains it. Regarding the most relevant QoS parameters for this kind of events, the *Transport priority* must be set to the highest level in the backbone and *Reliability* must be set to *reliable* to ensure that events are properly delivered.
- **Sampled Measured Values (SMV):** as in the previous case, these values are better adapted to the many-to-many P/S paradigm. Each of the monitored variables (or a fixed set of them) is mapped into a topic, so

the producer IED write the data into the topic in order to publish the information whereas other IEDs subscribe to the topics according to the application needs. Due to the nature of most substation automation systems, generally, only the last sample available is relevant. Thus, the *History window* parameter can be kept to a minimum of one sample and the *Reliability* parameter can be set to *best effort*. The priority of each topic must be set according to that of the measurement variable in the range of medium to high priorities. Due to the cyclic nature of this kind of traffic, the *Deadline* parameter can be enforced with a value marginally higher than the sampling period.

- **Report and Logging Services (RL)**: the DDS standard provides an efficient mechanism called reporting for applications to track changes to the subscribed system objects. Instead of polling the data attribute values periodically, applications can group the interesting data attributes into a data set, and require the logical node hosting this data set report any changes to the members of this data set. It typically follows a many-to-one P/S paradigm with low priority.
- **Request / Non response Services (RNR)**: this kind of communication follows a Client/Server paradigm, in the sense that a single client requests a service from a provider (the server). This is achieved using one topic per service and identifying the server in the body of the topic. Since no response is to be send, *Reliability* must be set to *Reliable*. The priority of these services is set to low since most services of this class are frequently non-critical.
- **Request / Response Services (RR)**: unlike Request / Non Response Services, these are synchronous blocking C/S invocations (i.e. the client remains blocked until a response from the server is received). This behavior is achieved by using a pair of topics: one for the request and another one for the response. The client writes the request into the request-topic and performs a blocking read until the response-topic is received. The priority of this traffic is set to lowest, since frequently these services are used for non-critical operations.

Table 2 summarizes the mapping of the most typical generic services found in substation automation systems into DDS topics and types of services. Note that due that DDS follows essentially the P/S paradigm, the C/S behavior is obtained by configuring properly the mechanisms provided by DDS.

Table 2: Summary of the mapping into topics and services

		Paradigm	Topics (per variable)	Distribution	Content Filtered
ACSI Services	Generic Substation Events	Publish / Subscribe	1	Many to many	Yes
	Sampled Measured Values		1	Many to many	Yes
	Report / Logging		1	Many to one	No
	Request / Response	Client / Server	2	One to one	Yes
	Request / Non Response		1	One to one	Yes

DDS uses a subset of IDL to describe the topics of the applications. Table 3 provides sample IDL interfaces that illustrate how the data of the ACSI services should be defined in order to distribute them among the nodes of the distributed application for the above described services. It is to be noted that Request/Response and Request/Non Response services share the same type for their requests.

Table 3: Sample Interface Definition

	IDL

Generic Substation Events	`struct GSE {`
	`string LN;`
	`string EVENT;`
	`string VALUE;`
	`};`
	`#pragma keylist GSE EVENT`

Sampled Measured Values	`struct SMV {`
	`string LN;`
	`string DATA;`
	`string VALUE;`
	`};`
	`#pragma keylist SMV`
	`LN DATA // two keys`

Report / Logging	`struct Log {`
	`string LN;`
	`string LOGNAME;`
	`string VALUE;`
	`};`
	`#pragma keylist Log LN`

Request	`struct Request {`
	`string DEST_LN;`
	`string SERVICE;`
	`};`
	`#pragma keylist Request`
	`DEST_LN SERVICE // two keys`

Response	`struct Response {`
	`string LN;`
	`string SERVICE;`
	`string VALUE;`
	`};`
	`#pragma keylist Response // no key`

Following there is a brief discussion about the recommended configuration of the DDS QoS parameters to be used in Substation Automation Systems (SAS). Table 4 summarizes these values.

The *Deadline* parameter is only used in Sample Measured Values (SMV) with a value related to the period of the variable sampling since it represents the maximum separation between two topic updates.

Durability is set to *Persistent* in Generic Substation Events (GSE) since they are typically alarms or other events that must be distributed to the nodes even if they are disconnected from the application. For the remaining topics this parameter is not considered relevant and, consequently, is set as *Volatile*.

History is typically configured in SAS so that the DDS middleware keep only the last sample for SMV, whereas the remaining types of ACSI services (GSE, RL, RNR, RR) keep a value of N depending of the application. In the case of C/S operations, the value of N determines the length of the processing queues of the servers.

Latency budget, which specifies the maximum acceptable delivery delay from the Data Writer to the Data Reader, was set to a relatively low value for the GSE meaning that they should be delivered very rapidly and at a value related to the period for SMV.

The *Reliability* parameter is set to *best effort* for the SMV variables, since an efficient delivery is more important than resending a lost message. Generally, the current value of a SMV variable has no meaning after a specific interval of time has elapsed and it is more important to distribute the most recent value as soon as possible. For the remaining variables this parameter was set to *reliable* since the message could be resent if necessary.

The *Transport priority* is set to the maximum value for the GSE messages because the distribution of these messages is considered the most critical, whereas Request-Response and Reporting messages are given the lowest priority.

Table 4: Summary of the used QoS

	ACSI Services				
	GSE	**SMV**	**RL**	**RNR**	**RR**
Deadline	-	Maximum process time	-	-	-
Durability	Persistent	Volatile	Volatile	Volatile	Volatile
History	Keep N	Keep last	Keep N	Keep N	Keep N
Latency Budget	Estimated urgency	33-50% of Period	-	-	-
Ownership	Shared	Shared	Exclusive	Exclusive	Exclusive
Reliability	Reliable	Best effort	Reliable	Reliable	Reliable
Transport Priority	Highest	High	Lowest	Low	Lowest

Conclusions

Modern SAS are complex distributed systems in which several modular IEDs are connected with modern data communication networks. They are required to satisfy stringent constraints such as fault-tolerance, scalability and maintainability. This evolution is producing a technological shift which promotes the adoption of new technologies related to modern OS and communication networks.

Middleware technologies are one example of these new technologies that are being adopted in SAS since they ease the creation of new applications by promoting modular components concepts.

In particular, the OMG DDS is an efficient middleware specification that is gaining acceptation in critical systems. This work proposes using DDS as communication backbone for all operations in a SAS. It also provides a mapping of the IEC61850 standard over DDS with special emphasis on certain issues such as mixing different traffic patterns and configuring the DDS QoS parameters in order to ensure that the communication requirements are met for all the types of traffic.

This work has been financed by the University of the Basque Country (UPV/EHU) through grant EHU09/29 and by the Basque Government through the FACTWARE project (grant SAIOTEK-PE11UN061).

REFERENCES

[1] Kezunovic, M., "Future trends in protective relaying, substation automation, testing and related standardization" Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Vol. 1, 6-10 Oct. 2002 pp. 598 - 602

[2] IEC TC-57. Communication Networks and Systems in Substations – Part 7-1: Basic communication structure for substation and feeder equipment – Principles and models. International Standard 61850-7-1. IEC 2003

[3] Calvo, I., Marcos, M., Orive, D., Huidobro, A., "Object-oriented based architecture for accessing remotely electrical protection relays", Proc. of the IEEE ETFA03 Conference, 2003, Volume 2, 16-19 Sept, 2003, pp. 614 – 621

[4] Skeie, T., Johannessen, S., Holmeide, O., "Timeliness of real-time IP communication in switched industrial Ethernet networks", IEEE Transactions on Industrial Informatics, Vol. 2, No. 1., pp. 25-39 Feb. 2006

[5] OMG, Object Management Group, "Data Distribution Service for Real-time Systems", v1.2, June 2007.

[6] Ozansoy, C.R., Zayegh, A., Kalam, "Interoperable CORBA Middleware Design for Substation Communication Systems", 8th IEE International Conference on Developments in Power System Protection, April 2004, pp. 705-708.

[7] Sanz, R., "Embedding Interoperable Objects in Automation Systems", Proc. of the IEEE 28th Annual Conference of the Industrial Electronics Society (IECON02), Nov. 2002, pp. 2261-2265

[8] Hossenlopp, L., "Engineering Perspectives on IEC 61850", IEEE Power & Energy Magazine, May/June 2007, pp. 45-50

[9] Calvo, I., Garcia de Albéniz, O., Noguero A., Pérez, F., "Towards a Modular and Scalable Design for the Communications of Electrical Protection Relays", Proc. of the IEEE 35th Annual Conference of the Industrial Electronics Society (IECON09), Nov. 2009, pp. 2511 – 2516.

[10] J. Lopez-Vega, J. Povedano-Molina, J. Sanchez-Monedero, and J.M. Lopez-Soler, "Políticas de QoS en una Plataforma de Trabajo Colaborativo sobre Middleware DDS", XIII Jornadas de Tiempo Real JTR2010, February 2010.

[11] G. Pardo-Castellote, B. Farabaugh, and R. Warren, "An introduction to DDS and Data-Centric Communications", 2005 Available at http://www.omg.org/news/whitepapers/Intro_To_DDS.pdf.

[12] I. Calvo, F. Pérez, I. Etxeberria, and G. Morán, "Control Communications with DDS using IEC61499 Service Interface Function Blocks", IEEE Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1 – 4, Bilbao, Spain, 13-16 Sept. 2010. DOI: 10.1109/ETFA.2010.5641207.

[13] IEC TC-57. Communication networks and systems in substations – Part 7.2: Basic communication structure for substation and feeder equipment – Abstract Communication Service Interface (ACSI). International Standard 61850-7-, IEC 2003

[14] IEC TC57. Communication Networks and Systems in Substations- Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8892-3, International Standard 61850-8-2. IEC 2004

[15] Mackiewicz, R., "Overview of IEC 61850 and Benefits", Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES, May, 2006, Dallas, USA, 21-24 May, pp. 376-383

***Authors**: Dr. Isidro Calvo, University College of Engineering of Vitoria-Gasteiz (UPV/EHU), C/Nieves Cano, 12, 01006, Vitoria-Gasteiz, SPAIN, E-mail: isidro.calvo@ehu.es; Ing. Oier García de Albéniz, Dept. Systems Engineering and Automatic Control, E.T.S.I. de Bilbao (UPV/EHU), Alda Urquijo, s/n, 48013, Bilbao, SPAIN, E-mail: g.albeniz@gmail.com; Dr. Federico Pérez, Dept. Systems Engineering and Automatic Control, E.T.S.I. de Bilbao (UPV/EHU), Alda Urquijo, s/n, 48013, Bilbao, SPAIN, E-mail: federico.perez@ehu.es.*