

Highly Configurable Low Cost Remote Laboratory with Integrated Support for Learning: Assessment of Acquisition Equipment

<http://dx.doi.org/10.3991/ijoe.v9iS3.2528>

D. A. H. Samuelsen and O. H. Graven
Buskerud University College, Kongsberg, Norway

Abstract—One of the activities when designing a remote laboratory is the action of selecting the data acquisition system. The hardware solutions available range from companies created high-quality components to self-designed solutions using discreet components in order to reduce the cost and gain full control over the specifications for the solution chosen. In the last few years the option of using open source, low-cost microcontroller board layouts based on microcontrollers have become available. The topic of this paper is the assessment of the quality of the data acquisition solution made available by the use of the internal analogue to digital converters (ADCs) of such low-cost solutions. The quality of the hardware is analyzed both in terms of measurement errors obtained through experimentation and comparison of quantitative data from the datasheets. The analysis presented serve as a basis for assessing whether the low-cost solutions are sufficient for running some or most of the remote laboratories, or if the increased cost of the budget solutions from professional acquisition modules can be defended.

Index Terms—Remote laboratory, hardware setup, reconfiguration.

I. INTRODUCTION

Laboratory exercises have always been an important part of engineering education. Many universities are in a process of transferring parts of these laboratories into remote controlled laboratories as shown in [1-4]. This has become an increasingly popular trend in engineering education, as it introduces added benefits where the most important are cost reduction in running laboratories as it reduces the wear and tear on the equipment and reduced staff costs, access for the users 24/7 in some cases, and access from anywhere in the world which facilitates distance learning. There are some drawbacks of remote laboratories like lack of hands-on experience for the users, and reduced contact between the supervisor and the users, but these are often regarded as insignificant in light of the benefits.

The remote laboratory setup can be divided into two major parts: The hardware of the experiment, and the control system. The control system consists of the control system for the experiment, data acquisition system and a webserver for allowing users remote access to the laboratory. Till now, the most popular solution for the control system has been the industrial data control and acquisition system from National Instruments: LabVIEW[5]. This system allows the institution to use

readymade acquisition solutions from NI[6] that give relatively easy access to data and a low entry level to the programming environment. There is also the built-in webserver application that together with the LabVIEW[5] programming environment which gives the system designers a low entry into the finished remote laboratory setup.

The hardware cost of these systems are however relatively high, when compared to solutions that are available today. A comparison of estimated cost of the different solutions will be given in the final paper for reference. While the NI-solution is based on the LabVIEW programming environment and the NI data acquisition hardware, examples of the low-cost solutions can be found in the Arduino[7] platform, which is based on the microcontrollers from Atmel[8] or Microchip[9], or user supplied designs. This type of solution will require a third party webserver application, in addition to the hardware system, which must be designed by the institution, possibly based on free available readymade designs. In this solution, the analogue to digital converter (ADC) built into the controller will be used as the primary or only ADC in the setup. The quality of such a solution will then in many cases be defined by the quality of the built-in ADC solution of the microcontroller. Poor quality of the acquisition system will result in poor quality of the data extracted from the remote laboratory. The important question, which is the topic of this paper, is then: How good quality can be expected from the internal ADC of the microcontrollers when used in a remote laboratory setup?

In the following sections the performance of the low cost solution will be assessed in two ways: by investigating the datasheets for the components of the different solutions, and by experiments run on the systems to verify what performance practical implementations actually have.

II. ANALYSIS OF THE ADC PERFORMANCE

The first insight into the performance of the ADCs built into the microcontrollers compared to the NI solution is available through the datasheets. A good source for information on ADC performance analysis can be found in application notes: AN693[10] from Microchip and AVR127[11] from Atmel. Some of the data for the ADC come naturally from the specifications of the ADC, such as number of bits, type of input (single ended, differential or pseudo-differential), input voltage range, among others. These specifications will be the same for similar types of

ADCs, whether found in high quality equipment or in low cost solutions as the built-in ADCs of microcontrollers. For this reason the analysis of these are only listed for reference as the available choices for these are limited to accepting what the designer of the microcontroller has chosen. In the following subsections the characteristics that will be different in the high-quality solutions and the low cost solution are described.

A. DC characteristics

The DC-specifications are important when the ADC is used for acquisition of signals that are relatively constant or slowly varying. Examples of this type of signals in the context of a remote laboratory would be inductor currents, capacitor voltages, position of mechanical structures, and such likes. The items below are collected from the application note AN693[10], where further details can be found.

- **Offset Error:** This is equal to an identical shift in all output values in the whole output range of the ADC. An offset error has the impact that a reading of a value equal to zero on the input, would not result in a zero value on the output, and this deviation would apply equally to all other input values. This can be corrected by simply adding or subtracting the output value by a fixed number equal to the error.
- **Gain Error:** If the slope of the linear transfer function deviates from the ideal, this is stated as a gain error, which can be corrected either at the output by multiplying the output value by a fixed number, or at the input by trimming the gain factor of the circuits feeding the input signal into the ADC.
- **Monotonicity:** If an input signal in the shape of a positive or negative ramp function is fed into the ADC, the output of the ADC should increase or decrease monotonically. There are no simple corrections for this type of error.
- **Nonlinearities:** These are often divided into two groups: Differential Nonlinearity (DNL) and Integral nonlinearity (INL). Differential nonlinearity can be visualized as the length of each of the “steps” made from the transfer function from the linear input to the quantized output. If the step length changes from one step to the next, this is described as a number between -1 and (normally) 1, where -1 means missing code. The integral nonlinearity is the maximum deviation of the transfer function from the ideal linear function, and is found from accumulating all DNL errors.
- **Absolute Error** is the combined error from all sources when no correction is made, and gives the maximum error that can be expected to occur at the output as a result of any input signal.

Table 1 and 2 show a comparison between the reference model, which is chosen to be the NI USB-6008[12] acquisition module, the Atmel ATmega1280[8], which is used in the Arduino MEGA module, and the Microchip PIC18F4550[13] which serves as an example of a user designed solution. Table 1 shows the single ended

characteristics, and table 2 shows the differential characteristics.

As can be seen from table 1 and 2, a complicating factor is that much of the information is given in different terms that make it difficult to easily compare directly the performance of the ADCs. In order to get comparable results for the analysis, practical tests are carried out, that assess the practical performance of these systems. The results of these tests are shown in table 3 and 4. The tests are carried out as described below:

- The Offset Error is measured by reading the output value of the ADC when the input is connected to analogue ground. The reading should be equally zero. As the result might be influenced by noise, an average over several readings is reported.
- The gain error is measured by performing several readings over the whole input range of the ADC, and the output values are compared to the input values, after correcting for the offset error. Then a best-fit slope is found and compared to the ideal.
- The monotonicity is analyzed by sending a ramp function into the ADC, and analyzing the output readings.
- The analyses of the nonlinear properties of the ADC are only done for the integral nonlinearity of the ADC.

The graphs of figure 1-3 shows the incremental nonlinear error (INL) found for the three devices. This test is carried out by subjecting the ADCs to a ramp input of very low frequency, and comparing the analogue input to the digital output. This was done using the full range of the ADC's input, and 10 consecutive tests were carried out for each device, and the average of the captured data were used for the analysis. A graph is then drawn for the digital output versus analogue input. On this same graph is also drawn a line that gives the ideal transfer function between the input and the output. These two graphs are then compared and their difference is plotted in figures 1-3.

TABLE I.
COMPARISON OF CHARACTERISTICS OF ADCS, SINGLE ENDED INPUT

Parameter group	Single ended input (typical values)			
	Parameter name	NI USB-6008	ATmega 1280	PIC18F 4550
Specification	Resolution	12 bits	10 bits	10 bits
	Input voltage range	-10V...10V	0V..AVcc	0V..AVcc
DC characteristics	Offset error	-	-2 bits	1 bit
	Gain error	-	2 bits	1 bit
	Monotonicity	-	-	Guaranteed
	DNL	-	0.5 bits	1 bit
	INL	-	1.25 bits	1 bit
	Absolute error	14,7mV	3 bits	-

TABLE II.
COMPARISON OF CHARACTERISTICS OF ADCS, DIFFERENTIAL INPUT

Parameter group	Differential input			
	Parameter name	NI USB-6008	ATmega 1280	PIC18F 4550
Specific-ation	Resolution	bits	8 bits	-
	Input voltage range	-10V...10V	2.7V... AVcc	-
DC characte- ristics	Offset error	-	2 bits	-
	Gain error	-	1.7%	-
	Monotonicity	-	-	-
	DNL	-	0.75 bits	-
	INL	-	2.5 bits	-
	Absolute error	14,7mV	18 bits	-

In these graphs it can be seen that the maximum errors for the ADCs of the Arduino and the Microchip devices, are less than ± 0.2 percent of the maximum reading, which gives a maximum swing of 0.4 percent of the maximum code. The Microchip ADC device lies between 0 and 0.1 percent error in the majority of the measuring range. For a 10-bit ADC this is equivalent to the least significant bit being in error. Roughly speaking, when considering the other error sources, the user is then left with 9 bits of qualified data, or 512 different values for representing the analogue signal.

The NI USB-6008 device use a 12 bit ADC, and the graph for the incremental nonlinear error shows that the error lies within ± 0.1 percent of the maximum reading, which in this case gives a maximum swing of 0.2 percent

of the maximum code. The major part of the code from this ADC's error graph also lies within the 0 - 0.1 percent band. As this is a 12-bit ADC, this is equivalent to the three least significant bits being in error. Also in this case the user is left with 9 bits of qualified data or 512 different values.

Considering the practical use of the data collected by these ADCs is important. Most oscilloscopes use 8 bit when digitizing analogue measurements. The argument for using such a low resolution is the signal being displayed on a screen with relatively low resolution. An increase in the number of bits would not give the user more usable information. If measuring a signal with voltage levels from 0 to 15V, 9 bit resolution corresponds to the voltage being measured in steps of around 0.03V. If both positive and negative power supply is used, a measuring range from -15V to +15V would give a step size of around 0.06V. For comparison, consider a simple voltage divider circuit, with two equal 10kΩ resistors in series. The voltage at the middle point of this voltage divider is measured. Consider then the resistors having a tolerance of 1%, and one resistor being 1% over the rated value, while the other resistor have a real resistance equal to the rated value. When connecting this voltage divider to a 0 – 15V power supply, the voltage difference between the ideal case and the real case is 0.037V. This is above the smallest step size measurable using the qualified bits of the ADC solution presented, and hence even such small deviances from the ideal circuit can be detected using this solution.

The gain error was found by performing 10 complete readings over the whole input range of the ADCs. A best-fit transfer function was then made on the basis of correspondence between the input values and the output

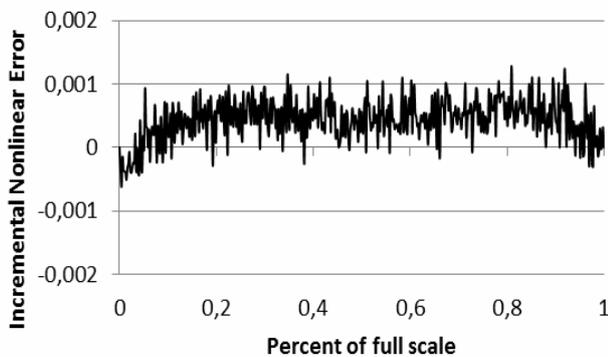


Figure 1. Nonlinearity over measurement range PIC18F4550

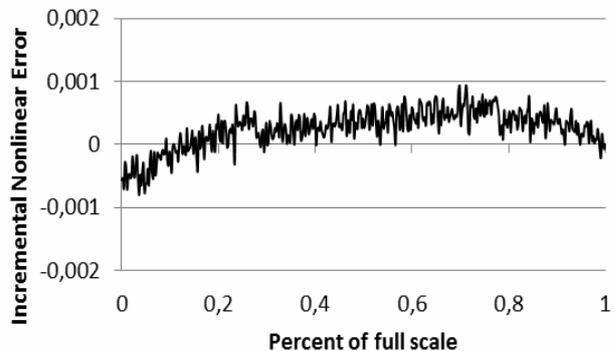


Figure 3. Nonlinearity over measurement range NI USB-6008

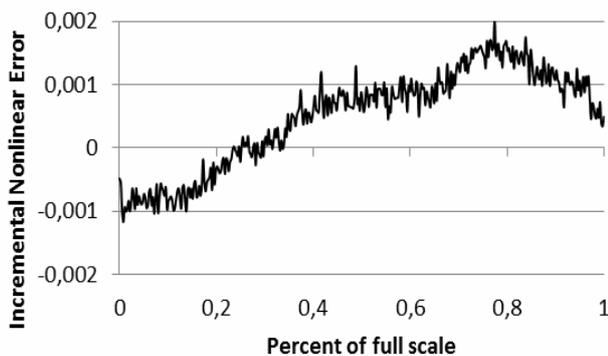


Figure 2. Nonlinearity over measurement range Arduino

TABLE III.
COMPARISON OF RESULTS OF DC CHARACTERISTICS ANALYSES,
SINGLE ENDED INPUT

Parameter name	Single ended input		
	NI USB-6008	ATmega 1280	PIC18F 4550
Offset error	<0.01% of full scale	<0.1% of full scale	<0.1% of full scale
Gain error	0,001	0,002	-0,0003
Monotonicity	No missing codes	No missing codes	No missing codes
INL	< $\pm 0.1\%$ of full scale	< $\pm 0.2\%$ of full scale	< $\pm 0.2\%$ of full scale

code. Then the slope of this was compared to the slope of an ideal transfer function, and the differences between these two are reported in table III for each device. Here it can be seen that the gain error is small for all devices, except for a relatively high value for the NI-device, while the Microchip device had an unexpected small error.

These findings, together with results from the other tests are summarized in table III. Here it can be seen that the offset error is smaller than the analogue value represented by the least significant bit of the ADCs.

According to this analysis, depending on the application, the industry standard NI USB-6008 can very well be replaced by the PIC18 series microcontrollers from Microchip, despite the higher resolution of the ADC in the NI-unit. This is mainly due to the relatively higher error level of the NI-device, when taking into account the number of bits. The Arduino device exhibit slightly larger errors in its readings, although these may not be excluding factors on their own.

B. AC characteristics

The AC characteristics are important for applications where the measured signals vary significantly within a short time frame or for AC signals in general. The datasheets for the microcontrollers contain little or no specifications for the AC characteristics. This is possibly a result of the AC characteristics being highly influenced by the analog circuitry and PCB layout between the signal source and the analogue input of the microcontroller. This is beyond the control of the microcontroller designer and left to the PCB designer to decide.

Practical tests are carried out to analyze the performance of the ADCs in the different systems. These AC performance tests are done using a standard signal generator set to produce a sinusoidal signal of 1kHz. This signal is then measured using all three devices: NI USB-6008, Arduino ATmega 1280, and the Microchip PIC18F4550. All three devices were set to do samples with a sample rate of 10 kHz. The sampling interval allowed the devices to capture 20 periods of the sinusoidal signal, and then transfer the data to a computer. The data were then analyzed, first by performing an FFT transform. Figures 4-6 show these FFT transforms for each of the three devices. As the number representation for the different devices was not equal, the measured values were normalized. This normalization also gave comparable results from the FFT analysis, so that each of the graphs have been scaled from 0 to 100.

As only the NI-device was capable of measuring negative voltages, all the three measurements were done using a voltage level shifter between the signal source and the devices. This circuit also included a buffer circuit in the form of a voltage follower connected op-amp, in order to reduce the output impedance of the signal source. This is in accordance with the instructions for the use of the ADC given in the datasheets for all three devices. This part of the circuit was identical for all three devices.

C. Practical implementation

The practical implementation was quite different for the three devices. The NI USB-6008 requires that the NI DAQmx software/driver solution is installed on the computer. In addition LabVIEW or a program utilizing the NI DAQmx-software library must be used to create a soft-

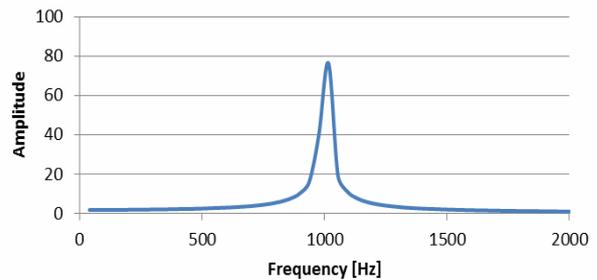


Figure 4. FFT transform for the NI USB-6008

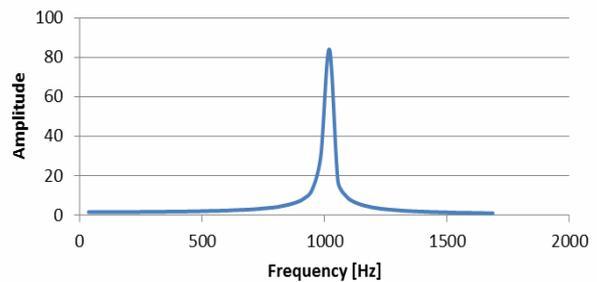


Figure 5. FFT transform for the Arduino ATmega 1280

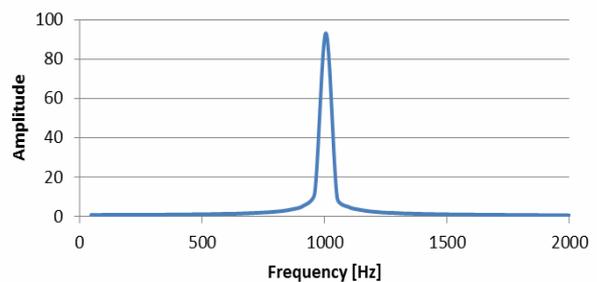


Figure 6. FFT transform for the Microchip PIC18F4550

ware program that performs the reading of the data and presents this in a file. The USB device itself has only 512 bytes of internal memory available for buffering data. An immediate result of this is that the sampling rate is limited by the transfer rate of the USB protocol. NI has chosen to further limit this to only 10kS/s for all channels in sum, meaning that if two channels should be read simultaneously, the effective sampling rate will be limited to half of that: 5kS/s. In a typical remote laboratory setting, more values should be read at the same time. Consider for instance a remote laboratory for a BJT amplifier. In this case at least all voltages around the transistor should be measured, in addition to the input voltage and the output voltage. This gives a total of 5 different measurement points, giving a maximum sampling rate of just 2kS/s. This sampling rate is on the border of the Nyquist criteria, when assuming an input signal of 1kHz. Using the NI USB-6009 device would allow the sampling rate to be increased to nearly 5 times this, and would be more acceptable.

The Arduino solution with its Atmel ATmega1280 device use a completely different programming environment, with a host monitor program installed with the user supplied code, both running simultaneously in the microcontroller. This host program consume some of the

processing time, limiting the effective processing capability of the solution, so that a maximum sampling rate equal to the NI-device was achievable, but not more. This results in lower sampling rate of more channels are used, as was the case for the NI-device.

The Microchip device runs the user supplied code alone, if debugging mode is deactivated. This increase the available sampling rate to near 100kS/s if one channel is used, assuming the device runs on 40MHz system clock frequency. Lower sampling rate will be the case if more signals are to be sampled simultaneously, but 8 channels at 10kS/s is easily achievable.

The main difference between the NI-solution and the two others is the use of larger internal buffers to store data during sampling. This increases the sampling rate of the devices to a much higher rate than is achievable if data should be transferred during sampling, because of the increased processor load of transferring data to another host system.

III. SYSTEM COST

The NI-solution is considered the expensive in this paper, although we are utilizing the low-cost devices from NI. The NI USB-6008 with connectors cost around 200 U.S. Dollars, while the NI USB-6009 has a cost of around 300 U.S. Dollars. The LabVIEW software must be bought in addition, in order to use the devices if that is not available from the start. The minimum development system is priced at around 1000 U.S. Dollars. Such a system is then ready to start development of a remote laboratory, except for the remote laboratory hardware.

The Arduino solution has proven to be somewhat limited, but the cost of this solution is extremely low due to the solution being open source. In an effort to remove the limitations, the "high-cost" solution from Digilent[14], the "chipKIT Max32" can be bought for around 50 U.S. Dollars. This includes a Microchip PIC32MX795F512 running at 80MHz with 128k RAM. A readymade add-on board for Ethernet connection can be bought, also for around 50 U.S. Dollars. The development software is freeware, and no other costs add to the minimum system needed for development of a remote laboratory, except for the remote laboratory hardware. There are other suppliers of Arduino systems that can offer the original boards at a significantly lower cost than the prices listed here.

The discreet Microchip solution has extremely low component cost. A large range of PIC18 devices with system clock frequency ranging from 48 to 64 MHz are available for less than 2 U.S. Dollars. These devices typically have from 3 to 4kbytes of RAM. This can be increased by adding external memory. 1Mbit serial SRAM devices cost around 1.5 U.S. Dollars. The sampling rate can be further increased by adding external ADCs, controlled by the PIC18 device. These have a cost typically of 1-2 U.S. Dollars. Added to this comes the increased development cost of the printed circuit board for the microcontroller and possibly the external devices. A solution for transferring data to a host system or the internet must also be implemented. However, when seeing this in conjunction with the development of the remote laboratory hardware, the relative increase in system cost will be relatively low.

IV. CONCLUSIONS

The main conclusion is that the internal ADCs of microcontrollers are of good enough quality to be a realistic alternative to the low-cost data acquisition solutions from National Instruments (NI) when used for remote laboratory applications. The first alternative solution, the Arduino system offers like the NI-solution a package of integrated systems, including prototyping-ready hardware, software development environment, monitor program solution for debugging and library package for low entry-level to program development. The second alternative presented is based on the use of discreet PIC18 series microcontrollers from Microchip. Although Microchip offer a range of development boards for their microcontrollers, the second alternative solution presented in this paper offer no ready-made development boards, as this will have the lowest component cost.

After verifying that the hardware is capable of doing data acquisition, the question remains as to whether the total cost of system implementation can be lowered using the alternative solutions. There is naturally no exact answer to that question. This depends on a number of factors, but there are some key elements that should be considered: The two major differences between the NI-solution and the Arduino solution are the software development environment, and how sampled data are transferred to a host system or to the internet. It can be argued that developing from scratch a microcontroller based acquisition system is time consuming. However, when developing a remote laboratory, a whole lot of hardware needs to be created, and much effort is put into adaptation to the acquisition hardware selected. Deciding on which platform to go for is then a question of which background the staff doing the development exhibit.

REFERENCES

- [1] O. H. Graven and D. Samuelsen, "Building and configuring a power supply in a remote lab experiment," presented at the AsiaMIC Phuket, Thailand, 2010.
- [2] D. Samuelsen and O. H. Graven, "Using a remote lab for real physical measurements on an electric motor," presented at the AsiaMIC, Phuket, Thailand, 2010.
- [3] B. Pradarelli, L. Latorre, and P. Nouet, "Integrated circuits testing: Remote access to test equipment for labs and engineering," *iJOE*, vol. 5, Special Issue 1: REV2009, 2009.
- [4] J. M. M. Ferreira and D. Mueller, "Online labs and the marvel experience," *iJOE*, vol. 1, No.1, 2005.
- [5] National Instruments, National Instruments, LabVIEW, [Web page] [cited 2012.04.04] Available from: <http://www.ni.com/labview/>
- [6] NI Multifunction Data Acquisition. [Web page] [cited 2012.04.04]; Available from: <http://www.ni.com/dataacquisition/multifunction/>.
- [7] Arduino-ArduinoBoardMega, [Web page] [cited 2012.04.04] Available from: <http://arduino.cc/en/Main/ArduinoBoardMega>.
- [8] 8-Bit Atmel Microcontroller. [Electronic Document PDF] [cited 2012.04.04] Available: <http://www.atmel.com/Images/doc2549.pdf>
- [9] Microchip 8-bit Micro controllers. [Web page] [cited 2012.04.04] Available: http://www.microchip.com/pagehandler/en_us/family/8bit/.
- [10] Application note AN693. [Electronic Document PDF] [cited 2012.04.04] Available: <http://www1.microchip.com/downloads/en/appnotes/00693a.pdf>.

- [11] Application note AVR127 – Understanding ADC parameters. [Electronic document PDF] [cited 2012.04.04] Available: <http://www.atmel.com/Images/doc8456.pdf>
- [12] “Low-Cost, Bus-Powered Multifunction DAQ for USB” [Electronic document PDF] [cited 2012.04.04] Available: <http://sine.ni.com/ds/app/doc/p/id/ds-218/lang/en>
- [13] “PIC18F4550 Family Data Sheet” [Web page] [cited 2012.12.12] Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>
- [14] Digilent Inc. [Web page] [cited 2012.12.12] Available: <http://www.digilentinc.com/index.cfm>

AUTHORS

D. A. H. Samuelson is with Buskerud University College, Department for Technology, Kongsberg, Norway (e-mail: dag.samuelson@hibu.no).

O. H. Graven is with Buskerud University College, Department for Technology, Kongsberg, Norway (e-mail: olaf.hallan.graven@hibu.no).

This article is an extended and modified version of a paper presented at the International Conference on Remote Engineering & Virtual Instrumentation (REV2012), held at University of Deusto, Bilbao, Spain, July 4-6, 2012. Received 25 February 2013. Published as resubmitted by the authors 20 March 2013.