Journal of Industrial Information Integration 3 (2016) 20–30

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

Implementation of SCADA/HMI system for real-time controlling and performance monitoring of SDR based flight termination system

Amiya Ranjan Panda[a,*], Debahuti Mishra[b], Hare Krishna Ratha[a]

[a] Integrated Test Range, Defence Research and Development Organisation, Chandipur, Balasore 756025, India
[b] Department of Computer science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar 751030, India

ARTICLE INFO

Article history:
Available online 9 July 2016

Keywords:
Data acquisition system (DAQ)
Supervisory control and data acquisition (SCADA) systems
Human machine interface (HMI)
Control and monitoring

ABSTRACT

Flight termination system (FTS) is an important system to control and terminate a wayward flight vehicle under performance evaluation in test centre. Hence, the functional behavior assessment of the FTS system in real time is of utmost importance in test centre. In this paper, a Supervisory control and data acquisition (SCADA)/human machine interface (HMI) based system with integrated data acquisition (DAQ) facility has been designed, developed and implemented for controlling and monitoring of FTS remotely. The SCADA/HMI based system receives various intermediate processed data from FTS and parameters of onboard command reception system (CRS) from telemetry stations through Ethernet via FTS. It consists of HMI based control and monitoring unit, SCADA server, FTS and validated onboard receiver. All the FTS transmission, reception and onboard reception parameters are monitored and logged in real time, which is used for data analysis. Moreover, the FTS operational setting parameters are controlled by this system remotely. Hence, the developed system is a combination of SCADA/HMI as well as control and monitoring architecture based system implementation. The system architectures involved and the internal implementation of different modules are described here. The performance results are presented and the system is validated in real-time Tele-communication operation environment.

1. Introduction

FTS is a RF transceiver system used for termination of flight vehicle under test in test facility by transmitting commands to the onboard CRS of the FV [1]. It is required to destroy the flight vehicle, when FV deviates from its predefined trajectory and takes different path due to unpredictable failures of different onboard subsystems [2].

Real-time flight termination system demands a very highly reliable and ruggedized platform [3]. Hence, the FTS is implemented using software defined radio (SDR) platform in our previous work [4]. A suitable software defined radio (SDR) platform with FPGA has proven itself to be very efficient platform for implementing such versatile and reconfigurable system architecture [5,6].

FTS transmits various commands to the onboard CRS to terminate the flight vehicle. Each command has independent role (link assurance, no inadvertent arming, liquid fuel flow termination etc) during termination. Each command activates corresponding relay switch in the command destruction system (CDS) circuitry of onboard CRS. When all the commands are transmitted, all the relay switches of CDS are activated and make the FV terminated. In real-time flight termination scenario, the status of the relay switches operations should be monitored.

It is required to deploy multiple SDR based FTS during the test of long range flight vehicle to cover maximum safety flight range and to overcome the line of sight limitations. Previously FTS were used at predefined locations with independent control and monitoring unit. After development of SCADA/HMI system, it has been planned for a common remote control and monitoring unit for all SDR based FTSs to automate and synchronize all FTS activities during real-time FV testing scenario.

The SCADA/HMI system consists of a number of FTS and reliable communication system. Each FTS collects its status information as well as intermediate processing information and sends those data to the SCADA server, via the reliable communication system. The SCADA server displays the acquired data and allows the operator to perform remote control tasks through a GUI known as HMI based control and monitoring unit. The accurate and timely data allows for optimization of the flight termination operation. Other benefits include more efficient, reliable and most importantly, safer operations. This result in a lower cost of operation compared to earlier non-automated systems.

* Corresponding author.
E-mail address: amiya87@gmail.com (A.R. Panda).

http://dx.doi.org/10.1016/j.jii.2016.07.001
2452-414X/

Nomenclature

FTS	Flight Termination System
CRS	Command reception system
FV	Flight vehicle
DAQ	Data acquisition
GUI	Graphical user interface
SCADA	Supervisory control and data acquisition
HMI	Human machine interface
WSN	Wireless sensor network
WSAN	Wireless sensor and actuator network
GPU	General processing unit
PXI	PCI extensions for instrumentation
PXIe	PCI extension for instrumentation express
RTC	Real-time controller

The communications system provides the pathway for communication between the HMI based control and monitoring unit and the FTS at remote sites via SCADA server. This communication system can be wire, fiber optic, radio, telephone line, microwave and possibly even satellite. The HMI based control and monitoring unit gathers data from the various FTS and generally provides an operator interface for display of information. It also controls various setting parameters of the FTS at remote sites. FTS system gathers onboard CRS information from telemetry stations and act as a relay back to the SCADA server.

In this paper, implementing a SCADA/HMI based system, performance of FTS in different operating conditions has been investigated in real-time, and the results of the experimental tests have been illustrated. The tests were carried out under different conditions in a test centre. This paper is structured as follows: Section 2 describes the related works, Section 3 illustrates the system architecture, Section 4 describes software implementation, and Section 5V shows hardware implementation. Finally, Section 6 shows results and Section 7 concludes the paper.

2. Literature survey

Control and monitoring of various industrial systems located over wide geographical areas requires various automation schemes that can be used for controlling and monitoring of various parameters of the system at great distances [7–9]. Many newly introduced networking technologies (Industrial Ethernet, Wireless LAN, etc. [10]) have really contributed to the automation solutions of the industrial systems. Today, it is also possible to use a personal computer with Flex-RIO/ compact-RIO platform for controlling and monitoring a complex remote supervisory task of any industrial system [11,12].

The previously used complicated controlling and monitoring panels with push buttons, switches, signalling lamps etc., are being rapidly replaced by very efficient 'Virtual' interfaces, which are generally displayed on CRT or LCD monitors, through interactive GUI: such facilities come in the category of HMI systems [12].

It is possible to access data among various levels of automation units using adequate network architecture. Then these data can be used to predict the performance of the industrial automation system at real-time and also these data can be stored in databases for post processing analysis procedure. Systems exhibiting above mentioned features termed as SCADA system [13–18].

SCADA systems are commonly utilized currently to keep track of and also regulate dispersed hardware components in commercial setups such as nuclear power plant, electric power grids, and also water treatment [19]. Frequently, they are utilized in Critical infrastructure (CIs) where safety and security are important aspects. Because of this, they need to abide by rigorous governing standards [20].

Several previous works exist related to SCADA and WSAN or WSNs (e.g., [21–23]). Most of these proposals follow integration of WSAN/WSN and SCADA, replacing SCADA with a WSN. Recently, the use of WSAN technology has been acknowledged as promising for the CI monitoring field [24]. In this respect, WSANs, due to their deployment flexibility, have the potential to become an integral part of CIM [25]. Despite of the advantages of WSN, the integration between WSANs and SCADA systems still poses some challenges [26]. These challenges, mainly related with the difficulty to develop algorithms that handle and manage the network to provide security, quality of service, and management support [27]. Alcaraz et al. revise and analyze these threats associated with WSNs in critical systems in [28].

Due to the critical nature of our application context, the system has implemented using SCADA rather wireless networks, which tend to be generally susceptible to attacks, and the vulnerable nature of the wireless technology. With respect to the WSN and WSAN concept, we think that the technology is not yet mature enough to completely discard the SCADA.

3. System architecture

The developed system is a combination of two architectures i.e. SCADA/HMI and control and monitoring. As the FTS is implemented in Flex-RIO platform, it has become easy to implement the SCADA/HMI system for monitoring and controlling of FTS parameters remotely.

3.1. LabVIEW Flex-RIO software communication architecture

Flex-RIO is a sturdy, reconfigurable inlayed system consisting of three parts: a processor running a real-time OS (RTOS), a reconfigurable FPGA, and also interchangeable commercial I/O components. The real-time processor provides trustworthy, foreseeable actions and also stands out at floating-point mathematics as well as analysis, while the FPGA [29–33] excels at smaller sized jobs that call for high-speed logic as well as specific timing. Typically Flex-RIO applications integrate a HMI, which gives the operator with a GUI for keeping an eye on the system's state and also setup operating criteria.

In LabVIEW Flex-RIO Software Communication Architecture [37], a host Virtual Instrument (VI) operates on a Windows computer, a real-time VI operates on RTC and also FPGA get configured via a bit file produced based on the FPGA VI as shown in Fig. 2.

3.2. Control and monitoring software architecture

A typical design for control and monitoring applications of Flex-RIO system is depicted in Fig. 3. The host VI supplies an event-based user interface so an operator could communicate with the embedded system. The RTOS implements high-level control and the FPGA perform low-level control.

There are two network communication paths: one path for sending out commands from the user interface to the Flex-RIO equipment as well as a second path for sending out tags (current value data) from the Flex-RIO equipment to the user interface for display.

To develop a scalable application, the system must be developed as though it needs to have a single command parser activity that can be made use of to rearrange the command and translate as required. This guarantees that ongoing critical tasks are not interrupted by the arrival of the command and makes it very easy to change the code to take care of additional commands.

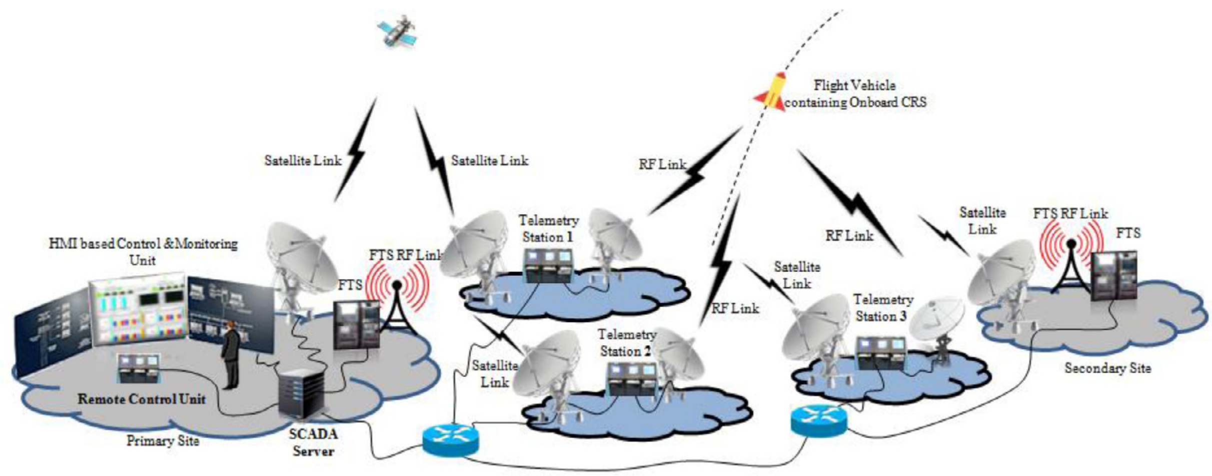

Fig. 1. System configuration.

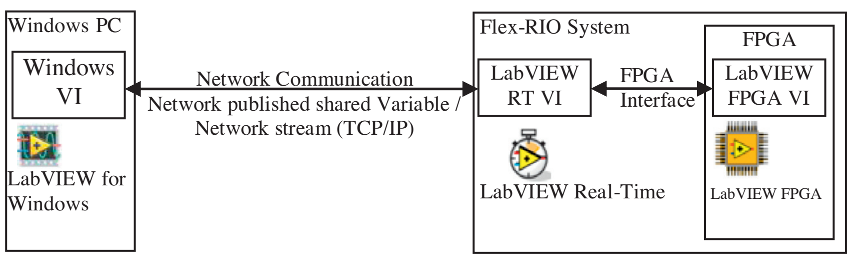

Fig. 2. Communication architecture of NI Flex-RIO system.

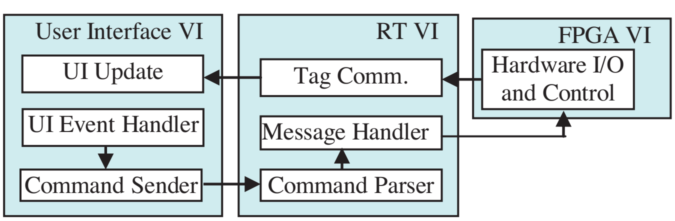

Fig. 3. Common architecture for control and monitoring application.

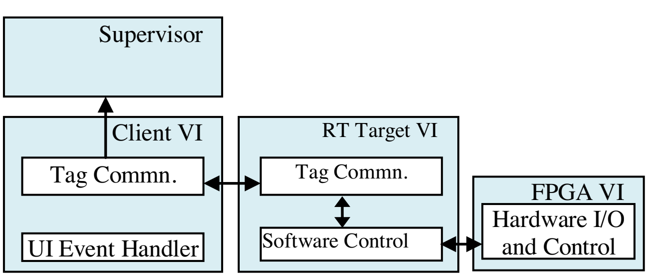

Fig. 4. Flex-RIO SCADA architecture.

3.3. SCADA/HMI software architecture

In a common SCADA application, the Flex-RIO system connects I/O channels as tags to a supervisor. Some control could be carried out on either the FPGA or RTOS. Commonly one could make use of the RIO Scan Interface to handle I/O for these types of applications since they do not require high-speed DAQ rates or customized hardware as shown in Fig. 4. With the LabVIEW data logging and supervisory control module, we could extend the software program capacities of a SCADA system to consist of the capability to log information to a historic data source as well as take care of events.

4. Software implementation

The major issue during the implementation is realization of processes and data communication path. A process is an asynchronously executing segment of code - essentially a loop. The primary step in determining the processes that determining the listing of tasks has to be achieved. Common tasks in designing our SCADA/HMI based system consist of controlling and monitoring of FTS parameters, data logging, communication to an HMI, interaction to I/O, information acquisition from FPGA, data acquisition from telemetry stations and safety logic or mistake handling. The next step is choosing just how the tasks are divided right into processes. An application with a number of processes calls for even more time to be spent on inter-process data communication. At the same time, breaking tasks into specific processes makes the program more scalable.

4.1. LabVIEW code realization for RT

When developing a real-time application, two degrees of reliabilities is taken into consideration i.e. program must run at a certain rate without interruption and factor to consider of determinism and jitter. Bounding the amount of jitter is a significant need for mission critical application. It has been accomplished by executing important tasks making use of Timed Loops in LabVIEW real-time. RTOS is much more deterministic and also reliable compared to a general-purpose OS, but it is still based on software crashes if not programmed correctly. A great deal of care has actually been taken during implementation of RT application [34].

Inter-process communication plays a crucial part in developing real-time application. The data interaction mechanisms advised for data transfer in between two nondeterministic loops are usually more flexible than the mechanisms recommended for data transfer involving a time-critical or deterministic loop. Single process shared variable has been used for transfer of current value within non deterministic processes. In our application, the various parameters (command codes, data rate, mark frequency and space frequency for BFSK, carrier frequency for FM modulation, up conversion gain, down conversion gain, dwell time etc.) settings for SDR based FTS is done through single process shared variable. For deterministic process, the transfer of current value has been done through Single process shared variable with RT FIFO enabled. High throughput information transfer has actually been done via RT FIFO. In our application, Manchester encoded and decoded, BFSK modulated and demodulated, FM modulated and demodulated spectrum transfer between RT and FPGA is done using shared variable with RT FIFO enabled. Queues (non-critical) as well as RT FIFOs (critical loop) methods have been used for low-latency information transfer from one process that triggers a specific event on another process.

The state machine design pattern has been used for designing the RT application because it can be used to implement any algorithm that can be explicitly described by a state diagram or flowchart. When implementing a design pattern in LabVIEW Real-Time, one of the most essential considerations is the synchronization of various loops. In our application, the synchronization has been attained through event driven loop. The RT application runs in two modes i.e. Local mode and remote mode. The Local/Remote control is present in RCU. In local mode, command can be issued from FTS locally. It is generally used for checking of working condition of FTS. In remote mode, command can only be issued from RCU. Command can't be issued from FTS locally in this mode. It is used during flight testing scenario to avoid inadvertent arming the onboard CRS by transmitting command from remote FTS sites. In flight testing scenario, command can only be given by competent authority through RCU by monitoring the HMI based control and monitoring unit. The parameters setting of FTS is done through HMI based control and monitoring unit via RT. The initial set parameters are stored in a file in RT. On next boot of RT program, the parameters are set in FTS taking value from the stored file, unless any change in parameters has been done through HMI based control and monitoring unit. The status as well as spectrum of Manchester encoded/decoded, BFSK modulated/demodulated, FM modulated/demodulated signal are gone to HMI based control and monitoring unit via RT. The flow of the RT design is shown in the Fig. 5.

The RT also involves in reception of onboard CRS status parameters from Telemetry stations located at various geographical areas via computer data processing division using UDP protocol in reliable communication link as shown in Fig. 6. The onboard CRS parameters received from Telemetry stations, ground FTS status parameters and intermediate processed signal spectrum are transferred to SCADA server via reliable communication link.

4.2. LabVIEW code realization for FPGA

Optimization exists both in the coding of FPGA and in timing in different methods, such as keeping synchronization in reading, writing I/O and between loops, developing modular reusable sub VIs, avoiding arbitration by minimal use of common recourses and so on [35–38].

LabVIEW FPGA VI makes use of variables, memory items and FIFOs for transferring current values among loops. Variables keep the most recent information in the Flip-flops of the FPGA. An additional method for sharing the latest values is to use available memory items. LabVIEW FPGA devices have two types of memory items: target scoped and VI defined. For communicating messages or updates, or streaming data between two or more loops, FIFO data transfer method has been used. A FIFO is a data structure that holds elements in the order they are received and provides access to those elements using a first-in-first-out access policy. In our application, in transmission line, the command codes are stored in variables. Then Manchester encoding is done at 10 MHz clock, which uses memory items. The Manchester encoded signal is gone for BFSK modulation in the same clock and then transferred for FM modulation block using target scoped memory item. After FM modulation at 300 kHz, the modulated signal is transferred to digital up-conversion block using target scoped memory item. After digital up conversion done at 100 MHz clock, the modulated and unconverted signal is transferred to RT using Target to Host FIFO. Then the signal is fed to RF up conversion block (PXI-5610), to up convert at 300 MHz to 2.7 GHz frequency using I/O read write method. The RF up converted signal is fed to power amplifier for power amplification and finally transmitted through antennae. Similarly in reception line, the signal is received through preamplifier and down converted to 15 MHz FM modulated signal using RF down conversion (PXI-5600) unit. Then the signal is fed to FPGA via RT. Initially digital down conversion is done at 100 MHz clock. Then FM demodulation is done at 10 MHz clock. Then BFSK demodulation and Manchester decoding is done at 10 MHz clock rate [4]. Both in transmission and reception line, at each phase signal spectrum are sent to HMI based control and monitoring unit using various data transfer method discussed here.

Read/Write command as well as DMA FIFOs information transfer techniques have actually been utilized for inter-target information interaction in LabVIEW FPGA. The current values have been transferred through Read/Write controls and streaming data has been transferred through DMA FIFOs. DMA does not involve the host processor when reading data off the FPGA; for that reason, it is the fastest technique for moving huge quantities of information in between the FPGA target and also the host. The intermediate signal waveforms have been transferred through DMA FIFO in our application. The various parameters setting of FTS are done through Read/Write control from RT to FPGA. DMA FIFO has used for transferring intermediate spectrum of the signal from FPGA to RT as shown in Fig. 6.

4.3. LabVIEW code realization of network communication

In our project, communication with a remote client is a critical part of the project. The function of Flex-RIO system is as a 'data servers' because its primary role is to report information (status, acquired data, analyzed data, and so on) to the client. It is also usually capable of responding to commands from the client to perform application-specific activities.

Network published shared variable and Network stream are two methods used here for sharing tags and commands between Flex-RIO system and SCADA server in communication network as shown in Fig. 6. Network streams are designed and optimized for lossless, high-throughput data communication over Ethernet, and they feature enhanced connection management that automatically restores network connectivity if a disconnection occurs due to a network outage or other system failure.

Redundancy is maintained in all communication network links because of critical nature of our application context. The primary communication link used is fiber optics, which is backed up by BSNL leased line link or satellite link.

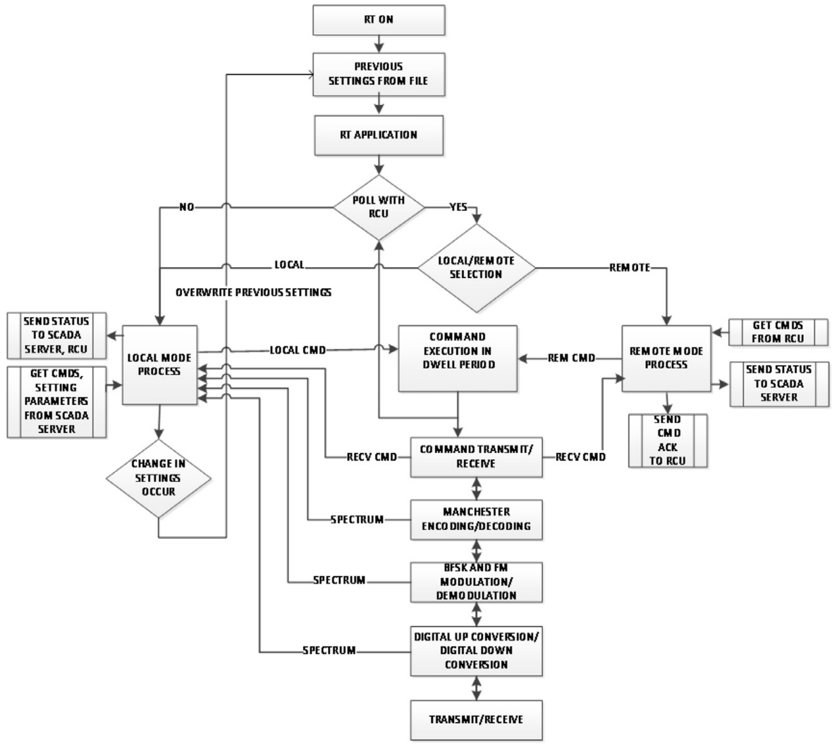

Fig. 5. Dataflow of processes in RT environment.

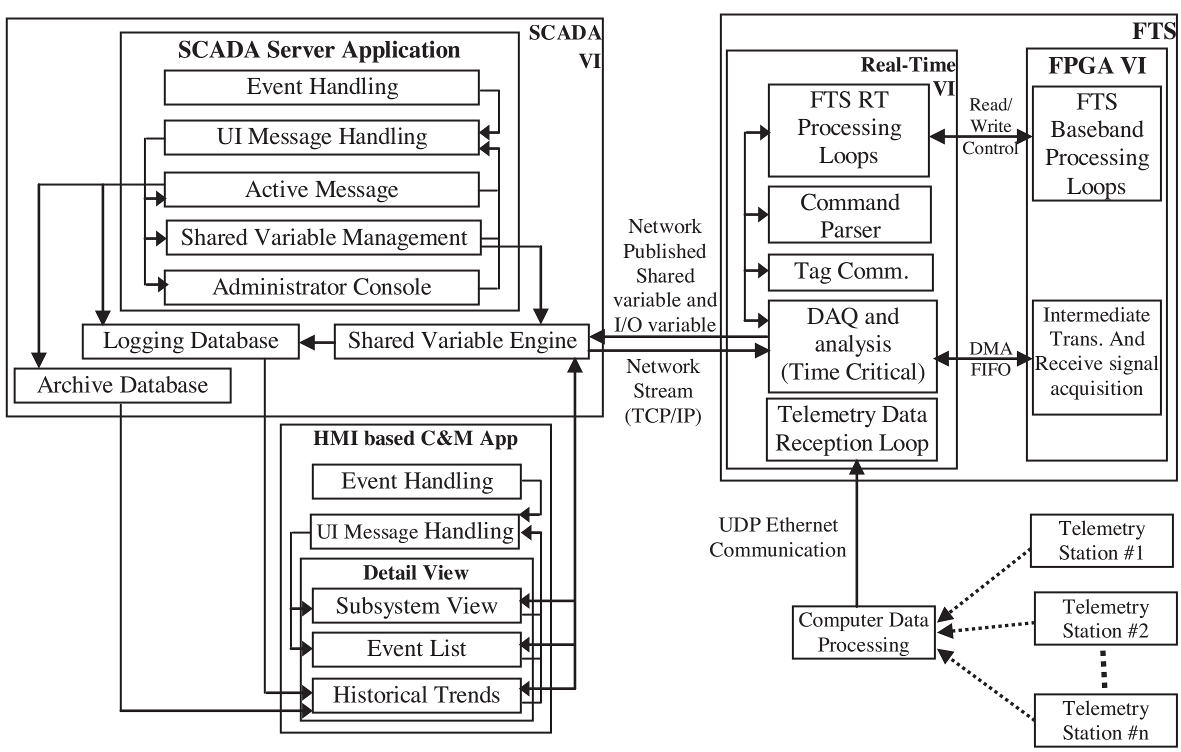

Fig. 6. Data communication in SCADA/HMI based control and monitoring unit with Flex-RIO system of FTS.

4.4. LabVIEW code realization of SCADA/HMI system

The main components of SCADA/HMI based system are SCADA server and HMI client. The main user interface of the SCADA server is responsible for managing shared variables, data logging, and data archiving. It includes the Event handling loop, Message handling loop, Shared variable management loop, Archive management loop and Timer loop running in parallel: Event Handling Loop processes user interface events and sends messages to the UI Message Loop. UI Message Loop handles UI messages from the server and transmits messages for message loops. Shared Variable Management Loop manages shared variables. Archive Management Loop handles archiving functionality. Timer Loop acts as a timer to trigger archiving. This timer loop pauses when archiving is

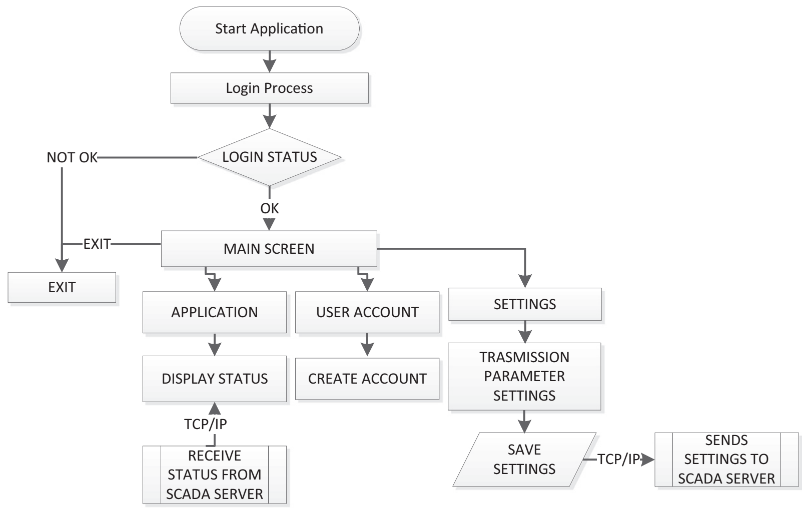

Fig. 7. Dataflow of different processes in HMI based control and monitoring unit.

disabled. Administrator Console sends configuration changes to the server. When the server receives the changes, the server applies the changes. Administrator Console includes Event handling loop and message handling loop running in parallel as shown in Fig. 6.

The client consists of the following components: Client Main displays an overview of the SCADA system and lists valid subsystems. Client includes the Event handling loop and message handling loop running in parallel. Detailed View shows detailed information of a subsystem. Detailed View includes the following loops running in parallel: Event Handling Loop processes user interface events and sends messages to the Message Handling Loop. Message Handling Loop handles messages for the Detailed View. The data flow of different processes in SCADA server is shown in Fig. 6.

The HMI based control and monitoring unit initially opens the login screen. Upon successful validation, it enters into control and monitoring application's main window. It consists of three states of the state machine implementation i.e. display application, user account and setting window as shown in Fig. 7. Hence this part is the complete implementation of event structure, state machine, control and monitoring architecture of LabVIEW. Display application part involves in display of various status parameters of FTS, spectrum of intermediate signal processing of FTS and status of communication link of whole system. User account part is responsible for management of user accounts by administrator. Setting part is responsible for setting of various parameters of FTS.

5. Hardware realization

The program for the Flex-RIO RTC, realized for performing real-time data acquisition, controlling and monitoring various FTS parameters has two levels comprising the one portion related with the FPGA and other portion related with the RTC. Direct Memory Access enables synchronization of both of the controller parts, and at the same time safeguards the effectiveness and momentum of data transfer. The RCU is connected to the RTC through serial communication link via data terminal unit (DTU), used to send trigger for command generation and transmission. Then the trigger for command generation is transferred to RTC through reliable communication link. RTC transfers the triggers to FPGA. The respective commands are generated, Manchester encoded, BFSK modulated, FM modulated, digital up converted and transferred to RF up conversion module (PXI-5610) through I/O interface of transceiver (PXI-5781). After RF up conversion, the modulated signal is fed to power amplifier for power amplification and then transmitted to air through antennae. Similarly in reception line, the signal is received through antennae and fed to preamplifier (PXI-5690). Then pre-amplified signal is fed to RF down conversion unit (PXI-5600). The down converted signal at 15 MHz is fed to FPGA via I/O interface of transceiver (PXI-5781). At FPGA, digital down conversion, FM demodulation, BFSK demodulation, Manchester decoding is done sequentially.

The SCADA server is implemented in host PC of the Flex-RIO system. The HMI based control and monitoring unit, implemented in windows PC is connected to the SCADA server through communication network via reliable communication link. The HMI based C&M unit and SCADA server implemented in host PC of the controller takes the responsibility for providing the transmission and reception analyzed signal monitoring and setting the parameters for the total system remotely.

In this scheme, the SCADA portion is implemented as server and HMI portion is implemented as client. The SCADA server manages various intermediate processed parameters of FTS, logs data in database, and regularly archives data from the logging database to the archive database. The administrator console allows the system administrator to configure and manage the SCADA server implemented in host PC. The HMI based C&M client allows the operators to monitor the system status and various intermediate processed parameters of FTS. The HMI based control and monitoring operator can view the values of the intermediate processed parameters of FTS, intermediate waveforms of FTS, acquired telemetry parameters and historical trends in a common GUI.

The solution presented here is adequate for implementation of the SCADA/ HMI based system, but more complicated task arises when the system is used for real-time flight termination operation. The FTS operational parameters are to be set from HMI based control and monitoring unit, placed at remote location. In such cases, the HMI based control and monitoring unit is connected to the SCADA server via reliable communication link and one additional loop is necessary for implementation. Therefore, in the described solution, total three sub routines are working in the same time: first one for RTC support, second one for HMI client support, third one for SCADA server processing. LabVIEW supports few tools for synchronization and the most important areas covered are occurrences, interrupts and loops executed in sequential manner. With the help of these tools, we can inhibit some part of the code while realizing the other parts. In addition, it facilitates enabling an order of the data transfer while executing loops.

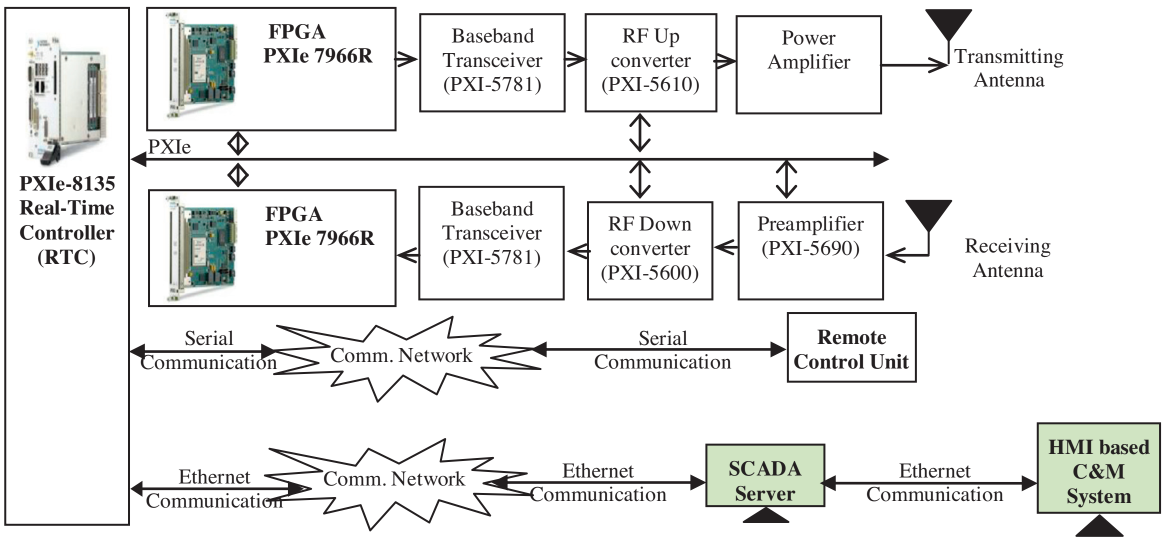

Fig. 8. Hardware realization of SCADA/HMI based integrated DAQ system in Flex-RIO platform.

Table 1
Hardware resources utilization in fts application.

Device utilization	Used	Total	Percentage usage
Total slices	13,107	14,720	89.0
Slice registers	25,991	58,880	44.1
Slice LUTs	38,430	58,880	65.3
DSP48s	420	640	65.6
Block RAMs	72	244	29.5

The SCADA/HMI based system includes FPGA, RTOS and GPUs. The intermediate signal processing status, waveform acquisition and other baseband processing status acquisitions are implemented in FPGA (Fig. 8). Both the telemetry data acquisition and FPGA data acquisition are implemented in RTOS of PXIe-8135 RTC. The SCADA server and HMI systems are implemented in GPUs. Hence, the SCADA/HMI based system of FTS with integrated DAQ facilities were implemented using high level programming language (LabVIEW) and were realized by NI PXI-7966R with Xilinx Virtex 5, SXT, FPGA [39,40].

The maximum used clock frequency of the FPGA is 100 MHz. Whole application (both FTS as well as SCADA/HMI based integrated DAQ system) needs 89.0% of SLICEs from hardware resources, whereas our previous work (SDR based FTS) needs 86% of SLICEs from hardware resources. This application needs more resources than previous work [4] due to use of many data communication methods to send the intermediate signal processing status to the HMI based control and monitoring unit. Complete resource utilization is summarized in Table 1.

6. Experimental setup and result

6.1. Experimental setup

The experiment has been done by keeping onboard receiver with decoder inside flight vehicle and received the status through telemetry stations. Then the onboard status parameters received by telemetry as well as intermediate FTS parameters are logged and monitored in SCADA/HMI based integrated DAQ system.

The experimental setup is composed of one onboard receiver with antenna and command decoder. Initially different command codes and frequency of operation are set in the FTS through HMI based control and monitoring unit. Then the commands are initiated from RCU remotely as shown in Fig. 9. Please note that this system was not created in a computer simulation, but comprised physical instruments in a test range.

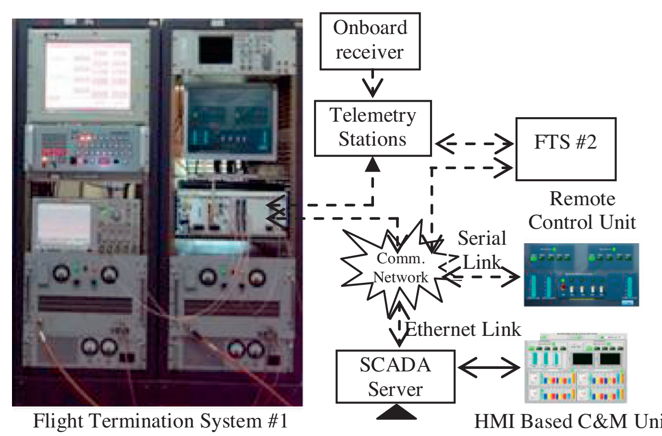

Fig. 9. Experimental Setup.

6.2. Graphical user interface

The SCADA/HMI based system parameters include command transmitting chain status (chain 1/chain 2), link status of HMI based control and monitoring unit with SCADA server of FTS, link status of telemetry stations with DAQ system, Indian standard time (IST) display unit synchronized with network time protocol (NTP) server, transmitted command status of redundant chain of FTS, received command status of redundant chain of FTS, logging control status, RF parameters status of FTS (up converter gain, down converter gain, preamplifier gain), Transmitting (encoded, modulated) and receiving (decoded, demodulated) spectrum of command status. Onboard receiver parameters include command continuity, signal strength, four relays (relay 1, relay 2, relay 3, relay 4) switch status of onboard receiver, command destruction system battery status.

All the above parameters are logged in integrated DAQ implemented in RTC and monitored in HMI based control and monitoring unit as shown in Fig. 10. The onboard parameters are converted to count value (numeric), then transmitted and displayed in HMI

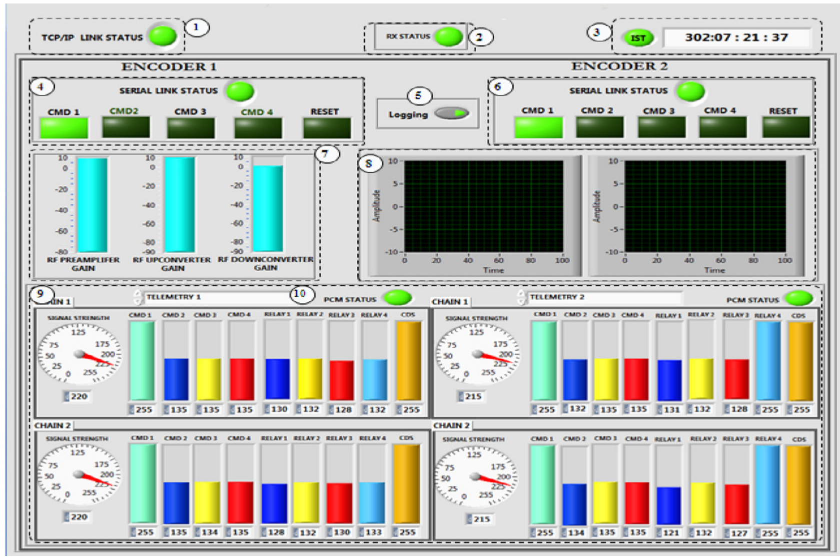

Fig. 10. GUI showing HMI based C&M unit of FTS. The numbers indicates the main units: (1) Link status of SCADA server with FTS; (2) Link status of telemetry stations with RTC; (3) IST time display unit synchronized with NTP server; [(4) & (6)] Transmitted command status of redundant chain of FTS; (5) Logging control; (6) (7) RF parameters status of FTS; (8) Transmitting and receiving spectrum of FTS; (9) Onboard parameter status via telemetry stations; (10) Telemetry station selection.

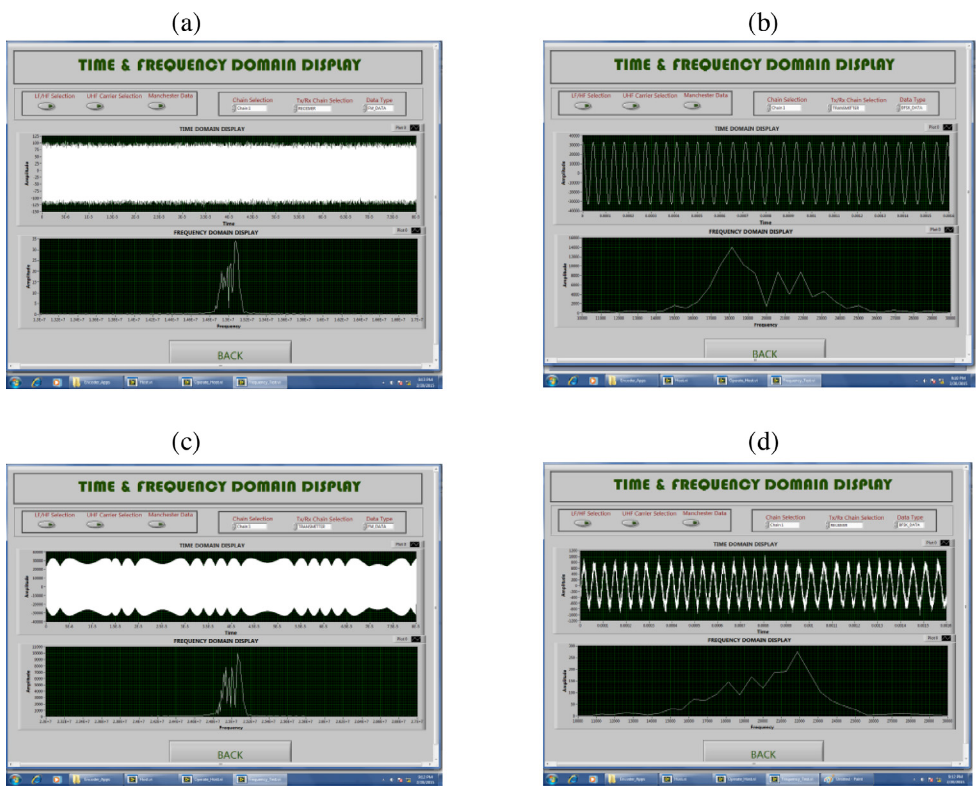

Fig. 11. GUI showing various intermediate processed waveforms of FTS signal in transmission and reception line in HMI based C&M unit of FTS. (a) FM data in reception line, (b) BFSK data in transmission line, (c) FM data in transmission line, (d) BFSK data in reception line.

based control and monitoring integrated display unit for optimization in coding as well as better visibility.

The intermediate waveforms (Manchester encoding, BFSK modulation, FM modulation) in both transmitted and received line of FTS are monitored in HMI based C&M unit of FTS as shown in Fig. 11.

Fig. 12(a) shows the FTS parameters setting panel of HMI based C&M unit and Fig. 12(b) Shows the RCU implemented in LabVIEW for commanding FTS remotely.

The HMI based C&M unit is composed of a total of four screens i.e. application display screen, user account management screen, FTS parameters setting screen, intermediate signal processing spectrum display screen. Security at this point is controlled in three different ways. First, only users with accounts in the system can access the SCADA. There are different types of accounts, depending on the privileges needed: administrator, operator and guest.

Fig. 12. (a) FTS Parameters setting panel in remote HMI unit and (b) RCU front panel.

Table 2
Delay evaluation.

Application	Average delay	Standard deviation	Maximum delay	Delay bound requirement
Command transmitting chain status	70.2 ms	3.9 ms	90.2 ms	< 100 ms
Link status of HMI based C&M unit with SCADA server	68.6 ms	4.3 ms	85 ms	< 100 ms
Link status of Telemetry stations with FTS	72.1 ms	5.1 ms	92 ms	< 100 ms
IST Time display	100.5 ms	15.6 ms	130.3 ms	< 1 s
Transmitted command status	69.1 ms	6.5 ms	79.8 ms	< 100 ms
Received command status **RTT***	90.1 ms	3.2 ms	98 ms	< 200 ms
Up converter gain	30.3 ms	3.2 ms	47.1 ms	< 100 ms
Down converter gain	27.9 ms	4.8 ms	36 ms	< 100 ms
Preamplifier gain	28.1 ms	5.1 ms	43.1 ms	< 100 ms
Transmitting command spectrum status	170.1 ms	10.2 ms	190.5 ms	< 500 ms
Receiving command spectrum status	180.4 ms	25.1 ms	201.6 ms	< 500 ms
Signal strength status	62.1 ms	3.9 ms	80 ms	< 100 ms
Onboard Relay 1 status **RTT***	110.4 ms	20.9 ms	130.3 ms	< 200 ms
Onboard Relay 2 status **RTT***	112.5 ms	15.3 ms	140.1 ms	< 200 ms
Onboard Relay 3 status **RTT***	104.6 ms	17 ms	134.9 ms	< 200 ms
Onboard Relay 4 status **RTT***	123.7 ms	25.1 ms	143.0 ms	< 200 ms

* Round trip time

6.3. Experimental results

Tests were conducted in the Test range in order to assess the performance of the whole system over an E1 carrier of 2 Mbps optical-fiber link.

The SCADA/HMI based system spanned three sectors of the test bed, as depicted in Fig. 1. The SCADA server and the HMI based C&M unit were located in the control center of the primary substation. Within the primary substation one FTS was present. At the secondary substation another FTS was present. Both of the FTS were connected with SCADA server as well as Telemetry stations with reliable communication link.

In SCADA/HMI based system, the main performance concerns are related with message loss and delay. This was especially true for the band-width demanding spectrum transfer service, with spectrum being transmitted from secondary substation over multiple hops. All the systems in the Test range are synchronized with network time protocol server for timing synchronization. The delay of each parameter is calculated at receiver application by transmitting the Time-stamp from the sender application to receiver application. In most of the parameters the round trip time is involved in calculation of the delay, in which processing delay has been taken care of during calculation. The delay bound requirement is shown in the table is analyzed with the observed delay of the particular parameter. No message loss was detected for all the services, whereas delay was kept with acceptable delay bounds for all the services (See Table 2).

Table 3
Performance in a 3-hop topology.

Packet error rate (%)	Throughput (kbit/sec)
0	1300
5	967
10	776

The impact of system configuration was evaluated in terms of maximum achievable throughput. Experiments were carried out with a 3-hop topology and the packet size was 300 bytes. It was found that the system has performed well for a packet size of 900 bytes (see Table 3).

7. Conclusion

Suitable programming methods for FTS parameters acquisition from FPGA, RTOS and onboard parameters from telemetry stations have been chosen and implemented to ensure optimized uses of hardware. The delay between transmitted and received signal between SCADA server and FTS unit has been analyzed, which plays a vital role during performance measurement of SCADA/HMI based system during real-time operation scenario.

The SCADA/HMI based system proved to be an effective system for control and monitoring of FTS. The operational team was getting the holistic picture of entire sub-systems and was able to diagnose and solve technical faults faster. This helped them to visualize command transmission by FTS, command reception by onboard CRS, relay switch activation in FV etc at ease.

This paper shows the implementation of SCADA/HMI based system for control and monitoring of two SDR based FTS. In future multiple Flex-RIO based FTS can be controlled and monitored through this SCADA/HMI based system with slight modification in LabVIEW programming code. According to these results and considerations, we conclude that the implemented SCADA/HMI system should satisfy the requirements.

References

[1] M. Khuzadi, On health monitoring, IEEE Aerosp. Electron. Syst. Mag. 26 (12) (2011) 10–16.

[2] B. Ratkevich, S. Brierley, D. Lupia, T. Leiker, Space launch flight termination system initial development, in: IEEE Aerospace Conference, 2013, pp. 1–9.

[3] F. Corl, Phong Do, S.V. Iyengar, Reliability products for space launch vehicle safety assessments, in: Annual Reliability and Maintainability Symposium (RAMS), 2014, pp. 1–4.

[4] A.R. Panda, D. Mishra, H.K. Ratha, FPGA implementation of software defined radio based flight termination system, IEEE Trans. Ind. Informat. 11 (Feb. (1)) (2015) 74–82.

[5] T. Ulversoy, Software defined radio: challenges and opportunities, IEEE Commun. Surv. Tuts. 12 (4) (2010) 531–550.

[6] G. Baldini, et al., Security aspects in software defined radio and cognitive radio networks: a survey and a way ahead, IEEE Commun. Surv. Tuts. 14 (2) (2012) 355–379.

[7] A. Vaccaro, V. Loia, G. Formato, P. Wall, V. Terzija, A self-organizing architecture for decentralized smart microgrids synchronization, control, and monitoring, IEEE Trans. Ind. Informat. 11 (Feb. (1)) (2015) 289–298.

[8] Cheng-i Chen, A two-stage solution procedure for digital power metering according to IEEE standard 1459-2010 in single-phase system, IEEE Trans. Ind. Informat. 60 (Dec. (12)) (2013) 5550–5557 Dec.

[9] C. Buffa, A. Tocchio, G. Langfelder, A versatile instrument for the characterization of capacitive micro- and nano electromechanical systems, IEEE Trans. Instrum. Meas. 61 (Jul. (7)) (2012) 2012–2021.

[10] A.M. Grilo, J. Chen, M. Diaz, D. Garrido, A. Casaca, An integrated WSAN and SCADA system for monitoring a critical infrastructure, IEEE Trans. Ind. Informat 10 (Aug. (3)) (2014) 1755–1764.

[11] F. Marignetti, M. Minutillo, A. Perna, E. Jannelli, Assessment of fuel cell performance under different air stoichiometries and fuel composition, IEEE Trans. Ind. Electr. (June (6)) (2011) 2420–2426 58.

[12] G.R. Biswal, R.P. Maheshwari, M.L. Dewal, Modeling, control, and monitoring of S3RS –based hydrogen cooling system in thermal power plant, IEEE Trans. Ind. Electr. 59 (Jan. (1)) (2012) 562–570.

[13] C. Queiroz, A. Mahmood, Z. Tari, A probabilistic model to predict the survivability of SCADA systems, IEEE Trans. Ind. Informat. 9 (Nov. (4)) (2013) 1975–1985.

[14] N. Iriondo, E. Estevez, M. Marcos, Automatic generation of the supervisor code for industrial switched-mode systems, IEEE Trans. Ind. Informat. 9 (Nov. (4)) (2013) 1868–1878.

[15] P. Nader, P. Honeine, P. Beauseroy, l_p -norms in one-class classification for intrusion detection in SCADA systems, IEEE Trans. Ind. Informat. 10 (Nov. (4)) (2014) 2308–2317.

[16] I.N. Fovino, A. Coletta, A. Carcano, M. Masera, Critical state-based filtering system for securing SCADA network protocols, IEEE Trans. Ind. Electr. 59 (Oct. (10)) (2012) 3943–3950.

[17] L.B. Ristic, B.I. Jeftenic, Implementation of fuzzy control to improve energy efficiency of variable speed bulk material transportation, IEEE Trans. Ind. Electr. 59 (Jul. (7)) (2012) 2959–2969.

[18] C. Chen, Y. Chen, Intelligent Identification of voltage variation events based on IEEE Std 1159-2009 for SCADA of distributed energy system, IEEE Trans. Ind. Electr. 62 (Apr. (4)) (2015) 2604–2611.

[19] S.A. Boyer, SCADA Supervisory Control and Data Acquisition, ISA—International Society of Automation, Englewood, CO, USA, June 2009 ISBN: 978-1-936007-09-7.

[20] P.J. May, C. Koski, Addressing public risks: extreme events and critical infrastructures, Rev. Policy Res. 30 (2013) 139–159.

[21] C. Amarawardhana, K.S. Dayananada, H. Porawagama, C. Gamage, Case study of WSN as a replacement for SCADA, in: Proc. Int. Conf. Ind. Inf. Syst. (ICIIS), Dec. 2009, pp. 49–54.

[22] S. Veleva, D. Davcev, Implementation perspectives of wireless sensor and actuator networks in smart grid, Adv. Mater. Res 433–440 (Jan. 2012) 6737.

[23] C. Alcaraz, J. Lopez, J. Zhou, R. Roman, Secure SCADA framework for the protection of energy control systems, Concurrency Comput. 23 (2011) 1431–1442.

[24] M. Merabti, M. Kennedy, W. Hurst, Critical infrastructure protection: a 21st century challenge, in: Proc. Int. Conf. Commun. Inf. Technol. (ICCIT), Mar. 2011, pp. 1–6.

[25] A. Davis, H. Chang, Airport protection using wireless sensor networks, in: Proc. IEEE Conf. Technol. Homeland Secur. (HST), Nov. 2012, pp. 36–42.

[26] C. Alcaraz, G. Fernandez, F. Carvajal, Security aspects of SCADA and DCS environments, in: Critical Infrastructure Protection, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 120–149.

[27] A.M. Grilo, J. Chen, M. Diaz, D. Garrido, A. Casaca, An integrated WSAN and SCADA system for monitoring a critical infrastructure, IEEE Trans. Ind. Informat. 10 (Apr. (3)) (Apr. 2014) 1755–1764.

[28] C. Alcaraz, J. Lopez, J. Zhou, R. Roman, Secure SCADA framework for the protection of energy control systems, Concurrency Comput. 23 (2011) 1431–1442.

[29] B. Jovanovic, R. Jevtic, C. Carreras, Binary division power models for high-level power estimation of FPGA-based DSP circuits, IEEE Trans. Ind. Informat. 10 (Feb. (1)) (2014) 393–398.

[30] Z. Hajduk, B. Trybus, J. Sadolewski, Architecture of FPGA embedded multiprocessor programmable controller, IEEE Trans. Ind. Electr. 62 (May (5)) (2015) 2952–2961.

[31] O. Jimenez, O. Lucia, I. Urriza, L.A. Barragan, D. Navarro, V. Dinavahi, Implementation of an FPGA-based online hardware-in-the-loop emulator using high-level synthesis tools for resonant power converters applied to induction heating appliances, IEEE Trans. Ind. Electr. 62 (Apr. (4)) (2015) 2006–2014.

[32] K. Myeongsu, K. Jaeyoung, K. Jong-Myon, An FPGA-based multicore system for real-time bearing fault diagnosis using ultrasampling rate AE signals, IEEE Trans. Ind. Electr. 62 (Apr. (4)) (2015) 2319–2329.

[33] N. Roshandel Tavana, V. Dinavahi, A general framework for FPGA-based real-time emulation of electrical machines for HIL applications, IEEE Trans. Ind. Electr. 62 (Apr. (4)) (2015) 2041–2053.

[34] D. Beck, H. Brand, C. Karagiannis, C. Rauth, The first approach to object oriented programming for LabVIEW real-time targets, IEEE Trans. Nuclear Sci. 53 (June (1)) (2006) 930–935.

[35] J. Druant, T. Vyncke, F. De Belie, P. Sergeant, J. Melkebeek, Adding inverter fault detection to model-based predictive control for flying-capacitor inverters, IEEE Trans. Ind. Electr. 62 (4) (2015) 2054–2063.

[36] W. Li, L. Gregoire, J. Belanger, A modular multilevel converter pulse generation and capacitor voltage balance method optimized for FPGA implementation, IEEE Trans. Ind. Electr. 62 (May (5)) (2015) 2859–2867.

[37] S. Xiong, D.B. Bogy, Hard disk drive servo system based on field-programmable gate arrays, IEEE Trans. Ind. Electr. 61 (Sept. (9)) (2014) 4878–4884.

[38] Z. Zhang, H. Xu, M. Xue, Z. Chen, T. Sun, R. Kennel, C.M. Hackl, Predictive control with novel virtual-flux estimation for back-to-back power converters, IEEE Trans. Ind. Electr. 62 (May (5)) (2015) 2823–2834.

[39] Virtex-5 Family Overview, Product Specification, DS100 (v5.0), Feb, 2009.

[40] National Instruments, NI PXI-7966R R Series Multifunction RIO With Virtex-5 SXT FPGA [Online]. Available: http://ni.com.

Amiya Ranjan Panda was born in Odisha, India on July 15, 1987. He received the B.Tech degree in Information Technology from Biju Patnaik University of Technology, Odisha, India in 2009 and M.Tech degree in computer science and engineering from Kalinga Institute of Industrial Technology, Deemed University, Odisha, India in 2012. He is currently working as research associate at Integrated Test Range, Defence Research and Development Organization (DRDO), India and worked as junior research fellow and senior research fellow from 2010 to 2015 in DRDO, India. He is currently pursuing his Ph.D. degree in Siksha 'O' Anusandhan University, Odisha, India. His research is mainly in real time data acquisition, communication, software defined radio and flight termination system. His research interests are supervisory control and data acquisition, machine learning and soft computing.

Debahuti Mishra received the B.Tech degree in computer science and engineering from Utkal University, Odisha, India in 1994 and M.Tech degree from Kalinga Institute of Industrial Technology Deemed University, Odisha, India in 2006. She did her Ph.D. in computer science and engineering from Siksha 'O' Anusandhan University, Odisha, India in the year 2011. She is an professor in the Department of Computer Science and Engineering, Institute of Technical Education & Research (ITER) under Siksha 'O' Anusandhan University, Odisha, India. Her research areas include data mining, bioinformatics, data acquisition, software engineering, soft computing and machine learning. She has contributed three books and more than 105 research level papers to many national and international journals and conferences.

Hare Krishna Ratha received the B.Tech. degree in electronics & telecommunication engineering and the M.Tech. degree in telecommunications system engineering from Sambalpur University, India and IIT Kharagpur, India in 1988 and 1996 respectively. He had been working as scientist in Defence Research & Development Organisation, India for the last 25 years in various capacities. Currently he is working as additional director at Integrated Test Range, Defence Research & Development Organisation, India. His research interests broadly include various range technologies like real time data acquisition, communication, computing, and flight termination system.