Industrial Application of IEC-61499 Standard with FBDK (Function Block Development Kit)

Mehul Pragjibhai Girnari[1] Prof. Vinod P. Patel[2]
[1,2]Department of Instrumentation and Control Engineering
[1,2]L.D. College of Engineering, Ahmedabad, 380015

Abstract— In today's world, intelligent automation is required for the solution of new challenges that emerges from the different disciplines of engineering. Especially in manufacturing and process industries, one of the biggest problem is to meet new market demands for flexibility and re-configurability, this field of research area attract number of researchers. An open standard IEC-61499 address this problems. This paper discuss the design of distributes industrial application with IEC-61499 standard in FBDK simulation software. This paper also discuss about designing of well-known discrete PID controller to control the level of Tank system. FBDK (Function Block Development Kit) and FBRT (Function Block Run Time) are freeware simulation software commonly used for simulation and validation of distributed control application. FBDK is compatible with IEC-61499.

Key words: IEC-61499, FBDK, Function Block, Discrete PID

I. INTRODUCTION

Automation technologies are always a subject to changes because of rapidly changing global markets and also to meet specific requirements of manufacturing industries forces researchers of different disciplines of engineering to find new approaches in order to be stay in the competitive markets. The production system currently used in the manufacturing industries provides facility for producing various products in the same plant, but this is very costly. In order to support these new requirements, the International Electro-technical Commission (IEC) launched the International Standard IEC-61499, which became official in 2005. This new standard is an extension of the well-known and used standard IEC-61131 for PLC (Programming Logic Controllers). We can say that IEC 61499 and IEC-61131-3 standards are close relatives. They share the same roots but have a different scope and purpose. So someone who is familiar with IEC-61131-3 will also cope with IEC-61499 and appreciate it.

At the point of intersection, where an old technology is slowly being replaced by a new technology, it is most challenging task for researchers of different disciples to make new technology more flexible and can address the problem faced while the use of old technology. These things creates new opportunities arise for pioneers, who look outside the box and develop ideas and upcoming solutions, resulting in a clear competitive edge.

IEC-61131 standard was based on the concept of function block, but IEC 61131 does not distinguish between the data and events. The arrival of new data (output of sensors) is itself considered an event. The continuous scan of the PLC program executes the ladder logic rungs (algorithm) associated with the data when it arrives.

In brief, IEC 61499 is event driven and designed to support distributed control. IEC 61499 standard emphasizes formal methods based on Unified Modelling Language (UML) and object oriented programming (OOP) concepts. It makes possible to distinguish between events, data, and algorithms. When particular event will arrive to the function block, it will only initiate the particular algorithm that uses the current values of data elements associated with tat event. This standard also provides flexibility in making of algorithms. We can use programming languages like instruction list (IL), ladder logic diagram (LD), and structured text (ST) as well as high-level and object oriented languages like C, java, depending on our requirements in order to create the control algorithms for the basic function blocks of IEC-61499.

This paper is organized as follows. Section-II gives some basic details about IEC-61499 standard. Section-III gives the details about various tools that can be used for designing of distributed control application with IEC-61499 standard. Section-IV will shows the basic features of FBDK and also gives idea about how one can design distributed control application in FBDK software. Finally in Section-V and Section-VI, conclusion is made and the work that can be extended in future is mentioned respectively.

II. IEC-61499 STANDARD

Because of the limitation and problem faced while using IEC-61131 standard as mentioned earlier in this paper, the International Electro-technical Commission (IEC) launched the extended version of IEC-61131 as a new Standard IEC-61499 for Industrial Process Measurement and Control Systems (IPMCS) in 2005 by combining the programmability feature of IEC-61131-3 (PLC) and distributability feature of IEC-61804 (DCS) in order to overcome the problems associated with IEC-61131 standard. The main purpose of IEC-61499 was to define a Function Block (FB) - oriented paradigm for distributed IPMCS to support distributed control.

- IEC 61499 consists of four parts which are listed below…
- Part 61499-1 "Architecture" was voted and approved as an IEC Standard in July 2004 and published in 2005.
- Part 61499-2"Software Tools Requirements" was also voted and approved as an IEC standard in July 2004 and published in 2005.
- Part 61499-3 "Application guidelines" has been approved as a technical report in 2004.
- Part 61499-4 A voting draft of 61499-4 "Rules for compliance profiles" has been approved in 2005.

The IEC 61499 standard is a hierarchical model of various models such as the application model, the system model, the device model, the resource model, and the Function Block (FB) model. The control engineer can use these models and can develop distributed control system by appropriate combination of these models. The graphical representation of such type of DCS system by using different models is shown below [5].

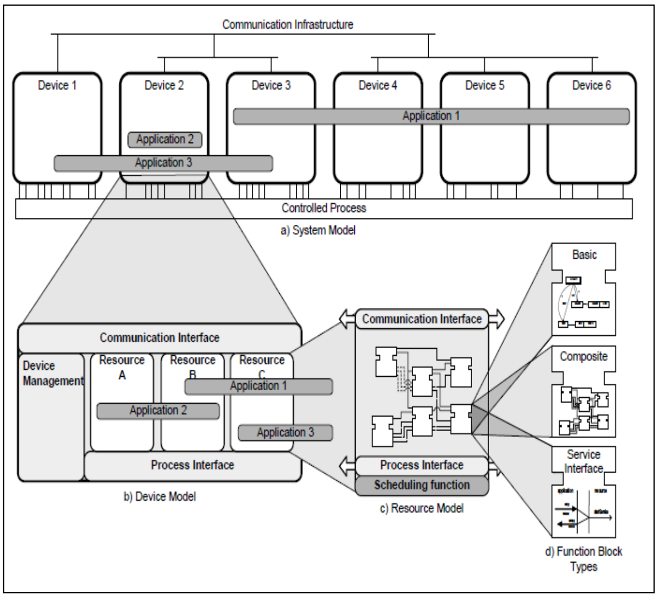

Fig. 1: Overview of the Main Models of IEC 61499

As shown in the figure-1 we can say main model say Application model of IEC-61499 standard can be made of number of System models, these System models can be made of different Device models, again Device models can be made of different Resource models, these Resource models can be considered as group of different types of Function blocks (i.e. Basic FB, Composite FB, SIFB etc.), and also Communication interface in order to communicate between various models. These all models together form a hierarchical model of main model of IEC-61499. The brief details of these all component is mentioned below [6],

1) Application: It can be defined as a network of interconnected function blocks, linked by event and data flows.
2) System model: It defines the relationship between communicating devices and applications. System model can be considered as a group of interconnected device model.
3) Device model: It is mode of resources, process interface and communication interface. A device is able to support one or more resources
4) Resource model: The resource provides facilities and services to support the execution of one or more function block application fragments. A resource can be loaded, configured, started and stopped without affecting other resources in the same device or network.so, it supports independent operation.
5) Function Block: The function block is a smallest functional unit of IEC-61499 standard. Function block consists of ECC (execution Control Chart) which is finite state machine or transition diagram and used for state transition when particular event will come. ECC uses Boolean expression concept in order to change the state. Also FB has facility that we can write algorithm to execute desired task by using different programming languages like ST, LD, JAVA, XML etc.

There are mainly three types of function blocks…

1) Basic function block
2) Composite function block
3) Service interface function block

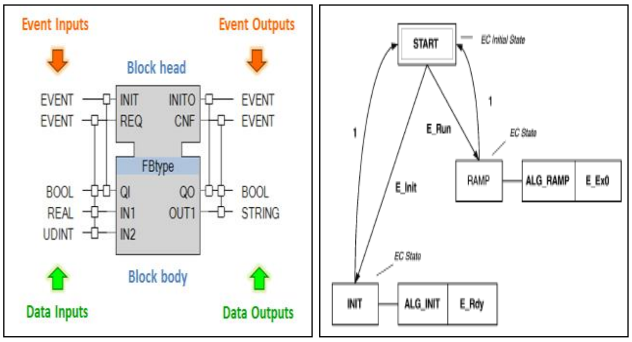

Fig. 2: a, b) Function block and ECC in IEC 61499

The function block will work as like follows, on the arrival of event at input, it will trigger the algorithm which uses the data only associated with that event and after execution of algorithm FB will update the output data and also generate output event which will again trigger another FB and so on. In this way model formed by interconnection of various FB will work.

6) Communication interface: There is also a communications interface that provides communications services for resources to exchange data via external networks with resources in remote devices.

III. SOFTWARE TOOLS FOR IEC-61499 STANDARD

The research work on the IEC 61499 implementation began around 1996 in several research centres worldwide and has resulted in a number of software tools supporting system engineering with IEC-61499[4]. Some of the most popular software tools which are compatible with IEC-61499 standard are mentioned here,

FBDK (Function Block Development Kit): This software tool has the oldest history and until now provides the widest spectrum of services-from function block development to their execution. All the design work mentioned in this paper was made in this freeware FBDK tool.

CORFU Engineering Support System: This tool was developed by the group of Professor Cleanthes Thramboulidis at the University of Patras (Greece). CORFU ESS is an IEC 61499- compliant Engineering Support System which adopts a hybrid approach for the development of industrial process measurement and control systems that integrates UML with the Function Block concept.

ISaGRAF v.5.0: Previously, this tool was a popular workbench for programmable logic controllers compliant with IEC-61131-1, but, version 5.0 developers included the support of IEC 61499 function blocks, thus becoming the world’s first automation software to be compliant with both the IEC-61131 and the IEC 61499 industrial standards.

0^3neida Workbench: It is an open source project conceived by 0^3neida. The code base of the Workbench was created in JAVA by Dr. James Christensen. The Workbench is intended to provide a collaborative environment for function block enthusiasts worldwide and will accommodate engineering concepts extending IEC 61499, for example, to include the concept of Automation Object.

IV. SYSTEM IN FBDK

This section discuss about the designing of distributed control system in FBDK tool with the example of Tank Level control (controlled by discrete incremental PID controller) example by taking three cases of application.

- Case-1: Single tank level control system
- Case-2: Two tank level controller with two controller
- Case-3: Two tank level balancing with single controller

As we have already mentioned in section-II that this overall system model is made up of different Device model. The system model of single tank level control system with discrete PID controller is shown in the figure-3. For all devices we need to include resources and other function blocks necessary for algorithm execution (i.e. Basic FB and Composite FB) and also for communication between different devices and resources (i.e. SIFB-Publisher and Subscriber block).

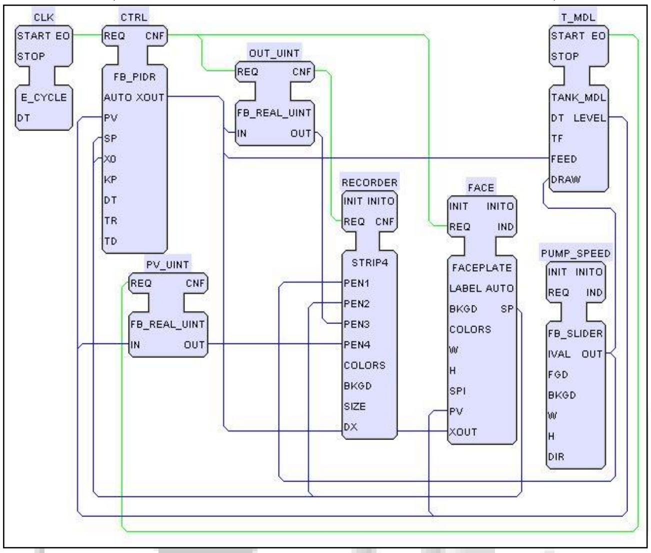

Fig. 3: Overall system model in FBDK for single tank system

A. *PID Controller*

Proportional-Integral-Derivative (PID) control is still widely used in industries because of its simplicity. No need for a plant model. No design to be performed. The user just installs a controller and adjusts 3 gains to get the best achievable performance. Most PID controllers nowadays are digital. In this work discrete incremental PID implementation on in FBDK is done by using ST (Structured Text) and JAVA programming languages.

Defining u(t) as the controller output, the final form of the PID algorithm is:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t)$$

Where,

K_p : Proportional gain, a tuning parameter

K_i : Integral gain, a tuning parameter

K_d : Derivative gain, a tuning parameter

e : Error = SP - PV

t : Time or instantaneous time (the present)

τ : Variable of integration; takes on values from time 0 to the present t.

Another form of PID that used here is sometimes called a parallel form, can be represented in the equation form as.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dT + K_d \frac{d}{dt} e(t)$$

The Laplace transform of above equation can be written as

$$G(s) = K_p + \frac{K_i}{s} + s\, K_d$$

We can easily convert the parameters from one form to another by noting that

$$K_p = K$$

$$K_i = \frac{K}{T_i}$$

$$K_d = K\, T_d$$

For digital implementation of PID controller, we are more interested in a Z-transform of Laplace of above equation, which can be written as follows…

$$U(z) = \left[K_p + \frac{K_i}{1-z^{-1}} + K_d(1-z^{-1})\right] E(z)$$

Rearranging this equation will give...

$$U(z) = \frac{\left[(K_p + K_i + K_d) + (-K_p - 2K_d)z^{-1} + K_d\, z^{-2}\right]}{1 - z^{-1}} E(z)$$

If we take K_1, K_2, K_3 as following,

$$K_2 = -K_p - 2K_d$$

$$K_3 = K_d$$

So, above equation can then be rewritten as

$$U(z) - z^{-1}\, U(z) = [\, K_1 + K_2 z^{-1} + K_3\, z^{-2}]E(z)$$

By taking inverse Z-transform above equation can be converted back to difference equation as

$$u[k] = u[k-1] + K_1 e[k] + K_2 e[k-1] + K_3 e[k-2]$$

This form will be suitable for implementation. This equation clearly indicates that the future controller output will be the sum of the previous control action and the weighted sum of present and previous errors. Here, the equation derived for discrete incremental PID controller can be implementation with two different languages, namely ST (Structured Text) and JAVA programming languages in FBDK tool in order to control level of the tank (for all three cases). Also, the implementation of the different controller like P, PI, and PD can be also made by making particular coefficients to zero.

i.e. for P-Controller → TR=TD=0

PI-Controller → TD=0

PD-Controller → TR=0. Etc.

1) *Case-1: Single Tank Level Control System*

When we mapped (download) all the resources and devices into system model in FBDK (Function Block Development Kit) platform that will create a distributed system. When we compile it and run it in FBDK Editor, the system will be generated which will contain three panels as shown in the figure-4. Also, for same system the controller response for disturbance (changing pump value) is shown in the figure-5.

2) *Case-2: Two Tank Level Controller With Two Controllers*

For two tank level control system, in order to control the level of both tank at desired set point irrespective of disturbance we need to use two controllers (one for each tank).

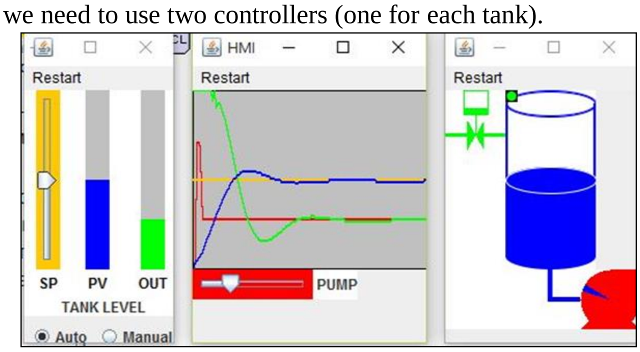

Fig. 4: Single Tank System Simulation in FBDK

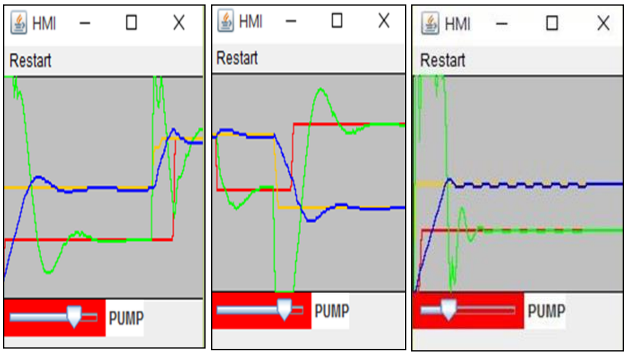

Fig. 5: PID Controller Response for Single tank System with Disturbance

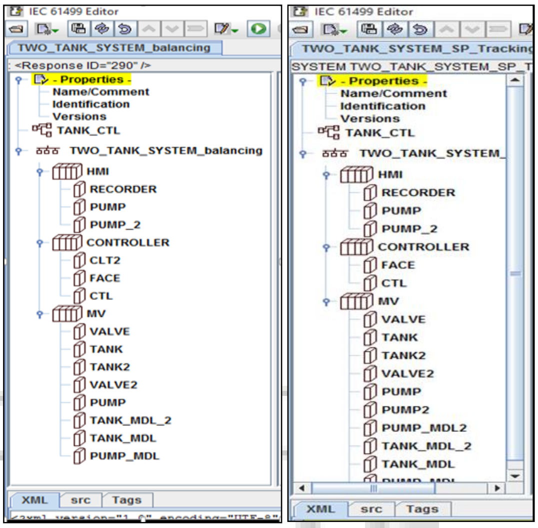

Fig. 6: a, b) Hierarchical model in FBDK for two tank system 6-a) with two controllers 6-b) with single controller

3) Case-3: Two Tank Level Balancing With Single Controller

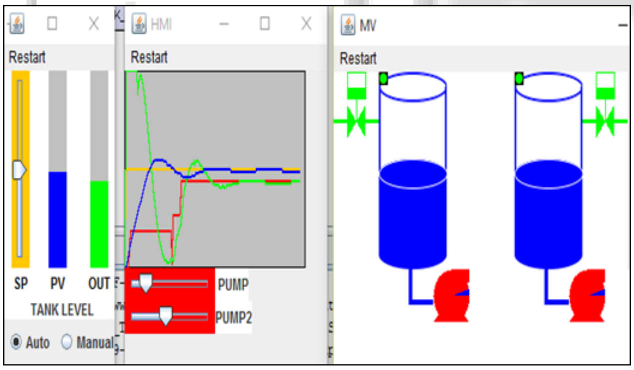

Fig. 7: Two Tank System with two Individual Controller (Case-2)

In this case only one controller is used which will control the level of both tank at a desired set point irrespective of disturbance by correcting the difference between levels of both tank continuously.

The figure-7 and figure-8 shows how the controller will work for case-2 and case-3 respectively.

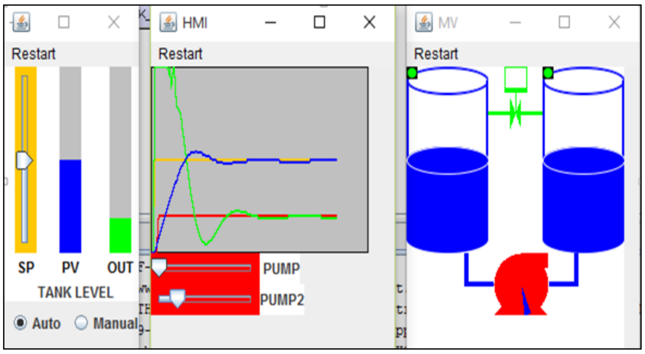

Fig. 8: Two Tank System Single Controller (Case-3)

V. CONCLUSION

IEC-61499 standard provides a standardized architecture for encapsulation, reuse and deployment of IP in automation and control system. Industrial adoption of IEC-61499 will greatly depend on availability of commercially supported software tools and runtime platforms. In FBDK when single event arrives to FB it is easy to program algorithm but, when more event occur simultaneously the programming of algorithm becomes quite complex and requires more skill to develop algorithm.

VI. FUTURE WORK

In this paper, it is shown that how we can design a distributed control system with IEC-61499 standard in FBDK by taking example of tank level control system which is SISO (Single Input Single Output). As we know to design a PID controller for single tank level control system is an easy task, but for industrial multivariable (MIMO) distributed control applications like in manufacturing industries, oils refineries, textiles industries and chemicals industries , the designing of controller will be challenging task. Also, for future we can include some advanced control strategies like Fuzzy control, Model Predictive Control (MPC) etc. these all advanced control strategies are very efficient for multivariable processes.

It should also be possible to combine the features of Matlab tool and FBDK tool by the function block transformation which will utilize the advantages of both tools. i.e. MATLAB Simulink provides an environment for modelling and simulation of control and embedded systems, while FBDK is good for designing distributed control systems.

REFERENCES

[1] http://www.iec61499.com
[2] http://www.holobloc.com/papers/iec61499/overview.htm
[3] John Fischer & Thomas O. Boucher, "Workbook for designing Distributed Control Applications Using Rockwell Automation's HOLOBLOC Prototyping Software" Working Paper No-05-017.
[4] "IEC 61499 Function Blocks For Embedded And Distributed Control Systems Design" Book By Valeriy Vyatkin, ISA 2007.
[5] "Real-time_execution_for_IEC_61499", Book by Alois Zoitl edition of 2009.

[6] "Modeling Control Systems Using IEC-61499, Applying Function Blocks To Distributed System" Book By Robert Lewis. Reprint Edition 2008.
[7] "Model-Driven Design Using IEC 61499 A Synchronous Approach for Embedded and Automation Systems ",Book By Li Hsien Yoong, Partha S. Roop, Zeeshan E. Bhatti, Matthew M.Y. Kuo. Springer Publication, 2015.
[8] "Modelling Control Systems Using IEC 61499", Book By Alois Zoitl and Robert Lewis, 2nd Edition 2014.
[9] "IEC 61131-3: Programming Industrial Automation Systems", Book by Karl-Heinz John, Michael Tiegelkamp, 2nd Edition, Springer Publication 2010.
[10] "Reconfigurable Embedded Control Systems: Applications for Flexibility and Agility", Book by Mohamed Khalgui, Hans-Michael Hanisch, 2011.

