Remote control of a robotic arm - A cross-platform study using LabVIEW and DeltaV

Hans-Petter HALVORSEN[1], Alexander JONSAAS[1], Lilly EIKEHAUG[1], Sondre MOGÅRD[1], Håkon Viumdal[1], Rune ANDERSEN[2], Saba MYLVAGANAM[1]*

University College of Southeast Norway (USN), *saba.mylvaganam@hit.no,

Emerson Process Management, Norway, Rune.Andersen@Emerson.com

Abstract

This paper is based on a collaboration between Lodz University of Technology (TUL) and University of Southeast Norway (USN) at an academic level in employing tools of ICT (Information, Communication Technology) in implementing some methods of environmental protection. As the focus was on a conceptual level, we decided to use existing ICT tools in the two University Colleges with focus on realising a plausible solution. The students and staff in TUL had a student project dealing with robots and at USN there have been extensive focus on process automation using DCS (Distributed Control Systems). With the aim of catering to the project goals, a cross platform study involving different programs was launched to control the robots in Lodz using LabVIEW of National Instruments (NI) and DeltaV of Emerson Process Management. This paper describes some main issues related to the fusion of these programs to achieve remote operation of the robot in Lodz.

The chosen client/server technology used for remote control was OPC (OLE of Process Control). Cogent OPC DataHub Tunneller was used to communicate between HSN and TUL using the internet. A DeltaV program was developed to update the robotic arm control variables. DeltaV OPC Mirror was used to map the OPC tags from the DeltaV OPC Server to a MatrikonOPC Simulation Server. The transferred OPC data is retrieved in LabVIEW at TUL. With this solution, it is possible to control the robotic arm located in TUL, from the DeltaV workstation in Norway. The LabVIEW user interface has the option of choosing between local control and remote control. The solution involving different software

platforms for remote control of robots can be modified for operation on other pertinent platforms for remote operations involving robots in firefighting, hazardous environments, deterring, damaging and destroying actions/objects of terrorists and not the least handling environmental pollutants.

Keywords

Remote control, robotics, combining various control platforms, Human Robot Interface (HRI)

1. Introduction

In the very recent past, we have seen amazing developments in robotics. Robust robots with remarkable behaviour in “mobility, agility, dexterity and speed” used in demos and real operations. Sensor fusion on board and on-board and remote controls help to execute complex operations by the robots.

Robots are used in warehouses and in handling of materials in many sectors of the industries and in big chain stores. With the use of intelligent sensors such as RFIDs, robots can locate, identify and move items very fast rarely making a mistake. In the wake of increasing demand for minimal cycle times and flawless operations, warehouses have mostly resorted to custom designed robots.

In the oil and gas industries, robots are deployed frequently for visual inspection, sample taking, test and calibration, and replacing process components. Remotely operated robots are employed in the field and are integral parts of the control system. ABB sees these robots as the remote process operators’ tools and “eyes, ears and hands” in the hazardous environments they are deployed. The operator interacts with the robot through the human-robot interface (HRI) of the control system by defining, initiating, monitoring and supervising the different tasks to be performed by the robots. According to ABB, „*In the robotized system, the human operator remains responsible for the process and its operation while the robot performs the physical tasks with sensors and tools*.” Further, according to ABB studies involving demonstrator robots „The ABB and Shell demonstrator has shown that robots can be used in oil and gas facilities to perform high precision operations in all kinds of weather”, [1], [2].

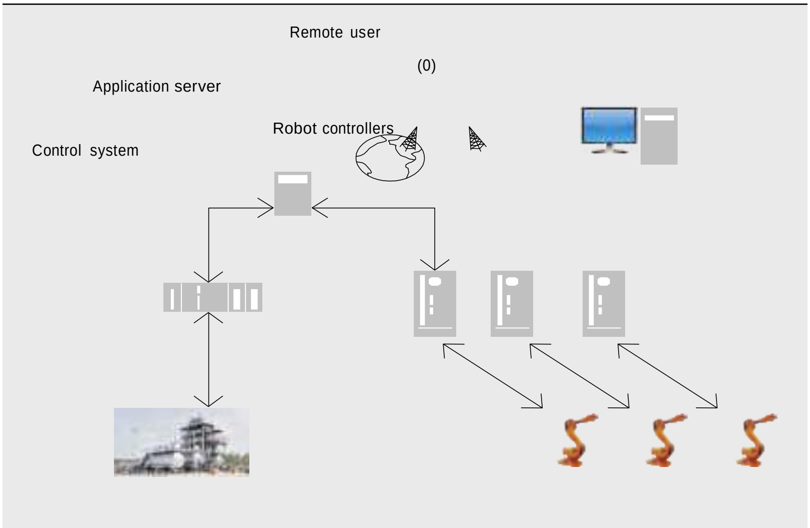

Figure 1. Robotics system architecture as conceived by ABB. HRI (Human Robot Interface) involving controllers and remote user features, [1], [2].

Already in the 1970s, NASA initiated R&D efforts in using remote-controlled robots to operate on astronauts. NASA and the US Army have invested in many R&D efforts in developing reliable robots (*robonauts)* for remote operations on people. *Robonaut* is a name coined by NASA for robots doing the work of astronauts in space. NASA uses advanced robots for robotic reconnaissance, involving a "*planetary rover under remote control to scout planned sorties prior to astronaut extra-vehicular activity (EVA)*", Long-distance surgery with surgeons remotely operating robots is mature enough to be used. Surgeons have started already from 2001 to control scalpel-wielding robots in real-time remote operations, [3].

Robots have been used in harsh environments for quite some time. In addition to applications in space as described above in the NASA study, robots have been used in subsea and nuclear power plants for decades. The versatility of robots in the aftermath of the Fukushima nuclear plant disaster, related to their dexterity in remote operations with high degree of autonomous behaviour has been presented in the media extensively.

Drones as another *avathar* of robots have been extensively used alone or in swarms in various operations, the recent being the deterring, damaging and destroying operations carried out against the *al shabaab* insurgents in Somalia.

As a preamble to our efforts, we present the software plan used by NASA in their remotely operated robonauts in Fig. 2. In our study, the data server in TUL will be in the "Raspberry Pi" unit and in the DeltaV server in USN.

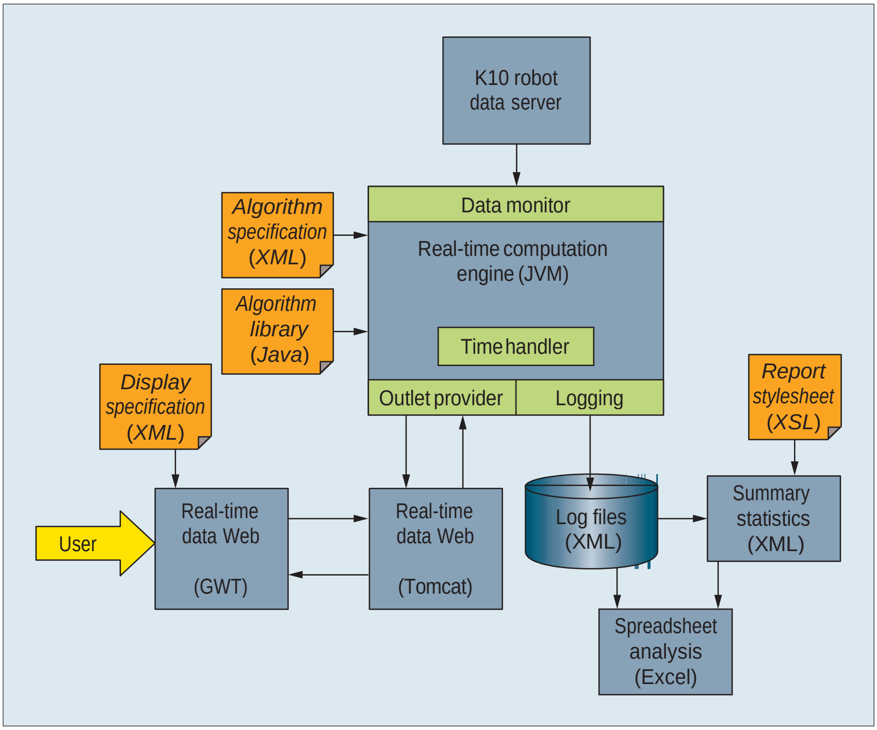

Figure 2. Software structure for prgraming the robonaut. K10 is the name given by NASA to one of its planetary robots. Robonaut is a name coined by NASA for robots doing the work of astronauts in space. This software structure is meant for a certain degree of autonomy. In our case, the operation is based on remote control, [5].

2. Brief note on LabVIEW and DeltaV

The graphical programming language LabVIEW from National Instruments (NI) gives the user a dynamic development environment leading to customizable systems with response times even in in microseconds. LabVIEW is a developmental platform useful in visualizing, creation and coding of engineering systems. LabVIEW is an application platform used in many R&D environments and can be interfaced to other systems such as Delta V, MySQL/MSQL, MATLAB, iOS, Android, etc. LabVIEW used on PCs running Windows the response time may be slow. LabVIEW-Real Time on a PC or PXI can cater to loop rates even at MHz range. The fastest platform for LabVIEW- Real Time is probably the Compact-RIO on FPGA (Field Programmable Gate Array). USN has a long-standing collaboration with NI. USN interacts closely with NI experts in Norway, Europe and the USA, in conjunction with academic and industrial R&D projects.

DCS (Distributed Control System) manages multiple process I/O signals, implements control, and supervisory routines in the whole factory. DeltaV of Emersons Process Management is a DCS for factory wide control. It is very common to find a single DCS or a concatenation of many DCSs, from a single or many vendors, in most of the modern industrial processes. DCS helps to optimize process performances at different stages and thus its usage leads to competitive advantage and enhanced overall performance without undue expenditures on person-hours in diverse operations related to supervision, control and maintenance.

DeltaV is used in many industries involved in processing of oil & gas, minerals, fertilizer etc. In USN, Campus Porsgrunn, there has been close collaboration with Emersons Process Management in testing and commissioning different new DeltaV products, recently in close collaboration with YARA, a leading international actor in the manufacturing of fertilizers. The essential nature of DeltaV DCS is presented in Fig. 3.

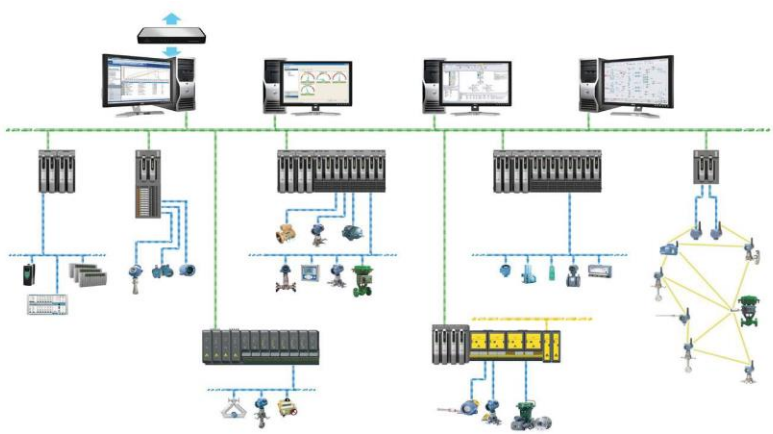

Figure 3. Distributed Control System with DelatV showing its topology from plant to control room, [6].

It should be noted, that in LabVIEW and DeltaV platforms, any operations involving writing to files, plotting and activating diverse displays would decrease loop speeds and hence slow down the overall response time considerably.

3. Usage of DeltaV in robot based warehousing

According to Kellet, 2011, “In the not so distance future, we will increasingly find large, centrally-located, fully automated and computer controlled distribution centers, in which robots busily unload incoming pallets from factories, after which other robots hauling the pallets zip across the floor to storage. At the same time, still other robots retrieve products from storage and transport them to picking stations, where stationary, gantry robots or robotic arms select products and build pallets to order for individual retail stores. Once built, these pallets are taken by robotic tuggers or automatic pallet drivers to shipping.

The warehouse of the not so distant future will thus hum with the computer-orchestrated movements of robots performing essential warehousing functions. Robotic, machine vision and motion control companies will realize this future and profit accordingly”, [4]. Human Robot Interaction and autonomous robots have become part of many industrial sectors, [7].

In a presentation to the American Society for Engineering Education, [8], a robot based warehousing system is presented using DeltaV PLC (Programmable Logic Controller),

Fig. 4 shows the drive system with DeltaV PLC with the relevant signal from PLC to the port motor and the starboard motor. The visual and IR cameras as shown in Fig. 5 helps in the navigation of the robots. Fig. 6 shows the sensor fusion with RFID for warehouse object identification and ultrasonic sensors used for object avoidance during the movement of the robot. Fig. 7 shows the essential details of the data flow with internet/intranet using DeltaV wireless TCP/IP to the mobile robot.

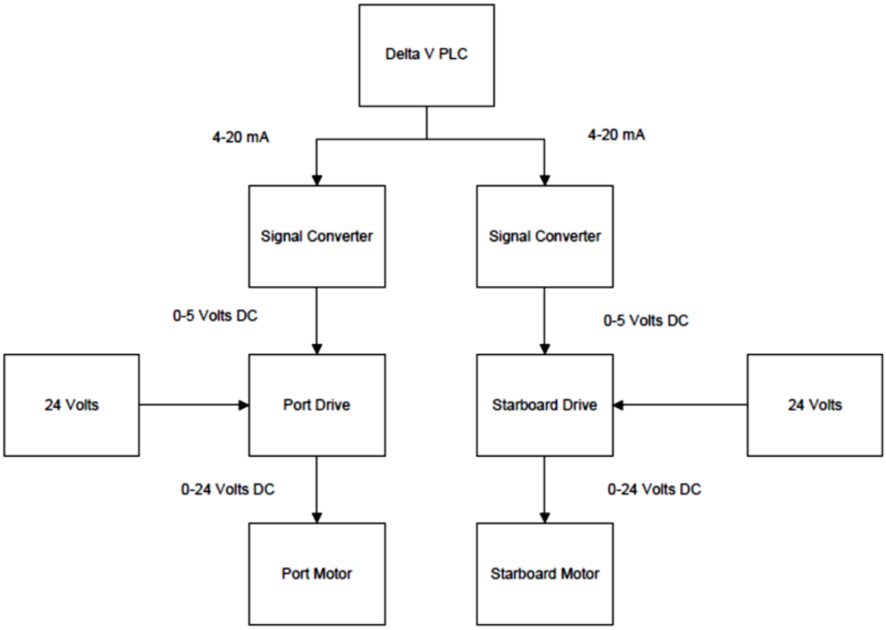

Figure 4. Self-locomotive inventory management drive system, {9].

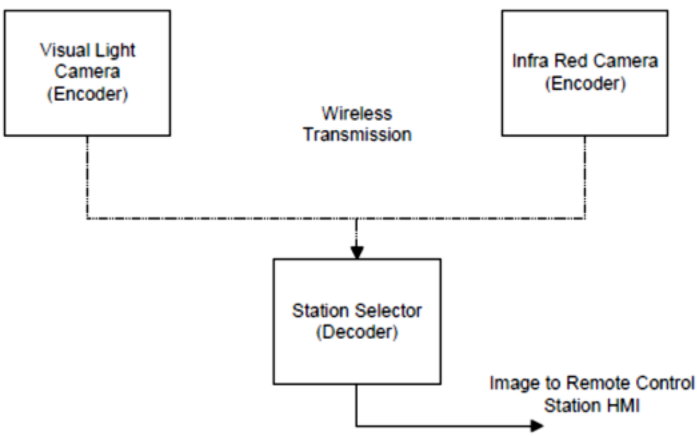

Figure 5. Navigation system using wireless transmission and visual and IR cameras,[9].

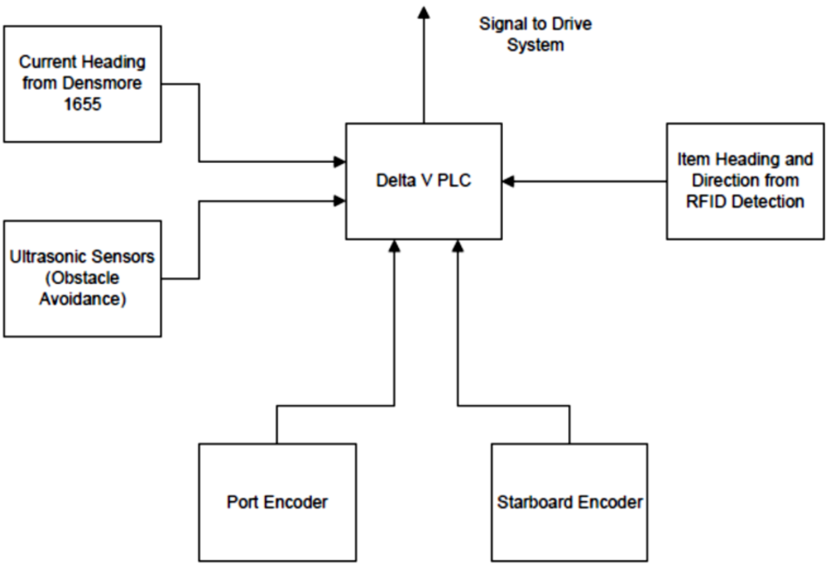

Figure 6. Vision system of the self-locomotive inventory system using RFID. Ultrasonic sensors are used for object avoidance. Sensor fusion performed on board, [9].

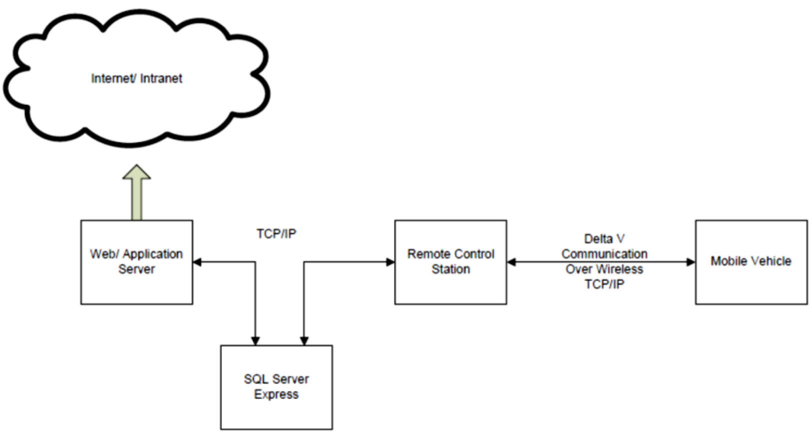

Figure 7. Data flow with internet/intranet using DeltaV wireless TCP/IP to the mobile robot, [9]

More recently, robots with dexterity and intelligence paralleling living beings like dog have been presented, [10].

2. Cross platform system description

Fig. 8 displays a schematic overview of the system, and the communication lines between the devices.

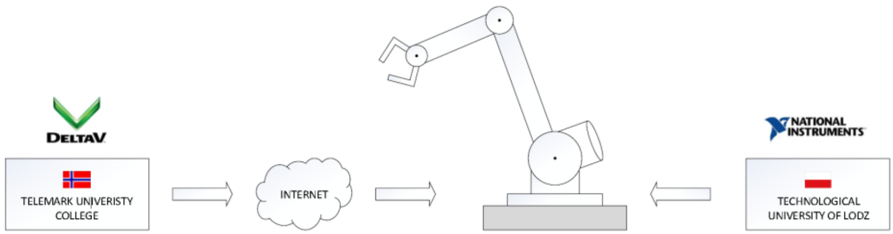

Figure 8. Schematic overview, the first draft of the project from the start-up phase.

The robotic arm (Fig. 9) used in this project is the uArm made by EVOL [11], [12]. This is an Arduino powered robotic arm, with four degrees of freedom (DOF). This robot is based industrial robots used for packing and assembling purposes such as those used in warehousing. Three digital servomotors are used for the basic movements of the arm, whereas an additional servomotor attached at the end of the arm moves and rotates the grip facilities. The input to the grip arm servomotor is a Boolean signal, whereas the other servo inputs are analog signals.

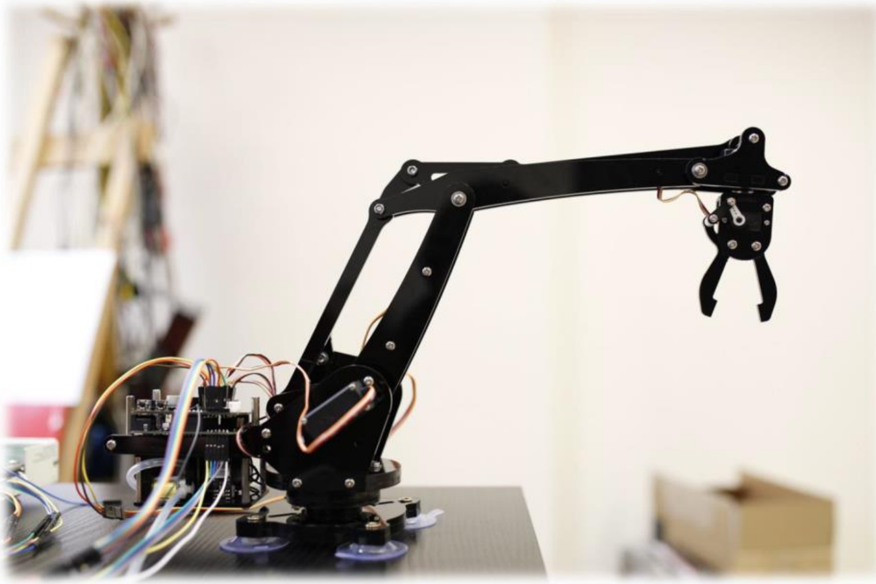

Figure 9. The robotic arm for remote control from USN. Althoug a simple device, the system was bulit form scratch enabling the groups to practise hardware software system integration with cross-platform operations.

2.1. LabVIEW

In this project the uARM is connected to the included Arduino compatible shield, which supplies the power to the servomotors of the robot. The shield is further connected to a DAQ device from National Instruments (NI-USB6259). The DAQ device is locally controlled by a LabVIEW based program (Fig. 10).

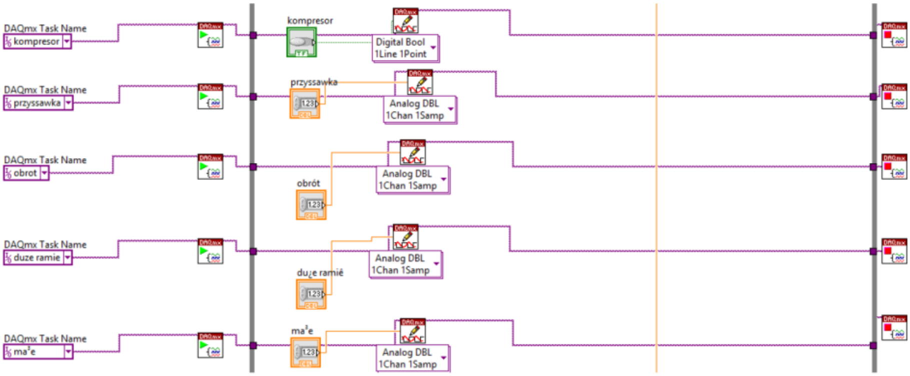

Figure 10. LabVIEW program for controlling the uArm locally.

For the remote control, Open Platform Communications (OPC) was the primary choice because of prior knowledge, and easy integration of DeltaV and LabVIEW. An OPC DataSocket test program was created for enabling reading of OPC tags. This program was implemented in the uArm program as a SubVI. The program part containing the DataSockets is shown in Fig. 11. Each Datasocket is connected to one tag in the OPC server. OPC

- is a standard that defines the communication of data between devices from different manufactures
- requires an OPC server that communicates with the OPC clients
- allows "plug-and-play", gives benefits as reduces installation time and the opportunity to choose products from different manufactures

Different standards are currently available for OPC, "Real-time" data (OPC DA), Historical data (OPC HDA), Alarm & Event data (OPC AE), etc.

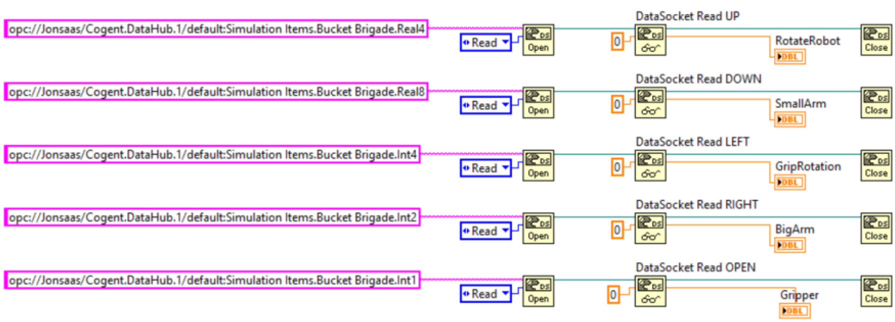

Figure 11. LabVIEW program for connecting OPC tags to numeric controllers

To enable the end-user to switch between local LabVIEW control and remote DeltaV control, a case structure was included at the local computer. Hence, the DAQmx write blocks are connected directly to the controls found in the front panel of the LabVIEW program, when operating the robot locally. The block diagram of the LabVIEW program with the OPC readings is shown in Fig. 12. MatrikonOPC Simulation Server and MatrikonOPC Explorer were used to test the program on a stand-alone computer.

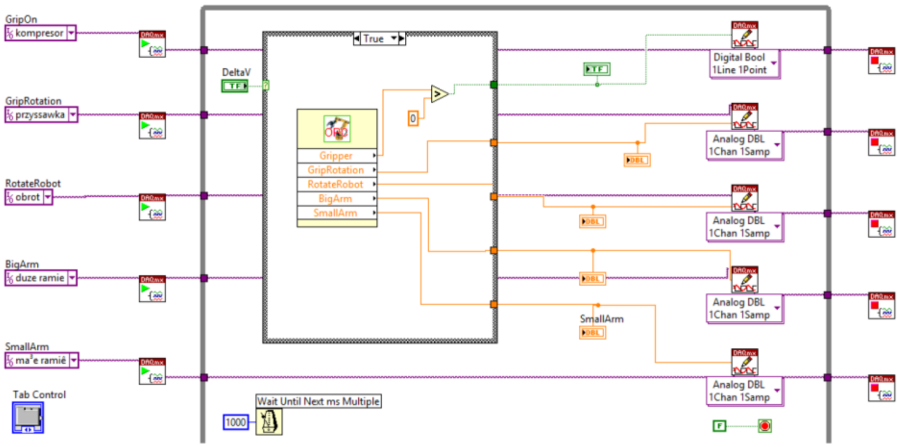

Figure12. The LabVIEW program where a case structure is used to switch between local control using LabVIEW and remote control using DeltaV. When the DeltaV button is enabled the OPC SUBVI is activated.

The selection betweeen local and remote control is made in the front panel of the LabVIEW program, using Tab Control. The different tabs are shown in Fig. 13, where remote control is activated in the upper block of the figure. When the DeltaV tab is selected, the operator can click the button to activate remote control. Clicking the button sets the Case Structure to „true", and the SubVI is activated. The uArm section on the right side of the front panel shows the voltages that are sent to the robotic arm.

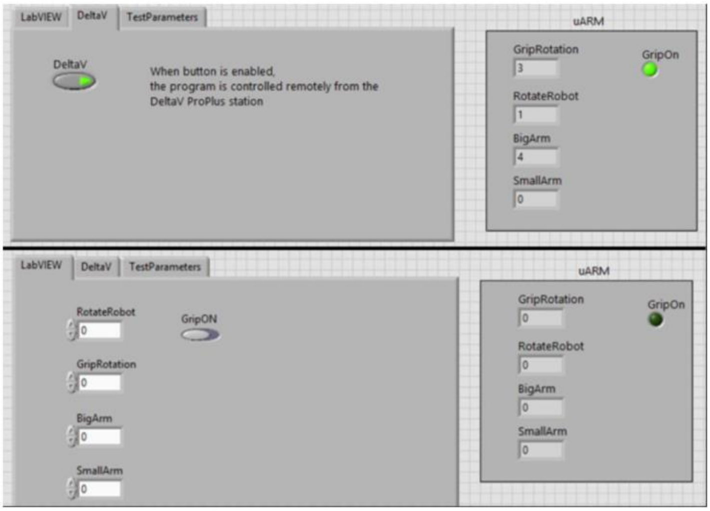

Figure 13. LabVIEW front panel for controlling the robotic arm either by LabVIEW directly, shown in the buttom part of the figure, or remotely with DeltaV, shown in top part of the figure

3.2. DeltaV based remote control

The following section describes the development of the DeltaV program used for remote control. To manually write the motor values in DeltaV, 5 DataLinks with DeltaV Data Entry Expert was added to the HMI in DeltaV Operate. Each DataLink is connected to an Output block in DeltaV Control Studio. The current values of the Output blocks are automatically added to the DeltaV OPC server. DeltaV OPC server runs in the background when the Pro Plus is powered on.

The program was tested by connecting to the DeltaV OPC server in MatrikonOPC Explorer and validating the tags of the Output blocks.

A robust remote real-time connection between the user and client, is enabeled by a „tunnel" across the internet. Hence, Distributed Component Object Model (DCOM) and Windows security problems are avoided by this peer-to-peer connection across the network [13].

As DeltaV does not have a built-in OPC tunnelling functionality, a third party software was needed to support remote transfer of the data. OPC DataHub by Cogent was the chosen software as this is free to use for one hour sessions. It was sufficient for testing, and the software may be upgraded to full version by purchasing a license. Due to unknown reasons, it was not possible to transfer the data directly from the DeltaV OPC Server when using OPC DataHub Tunnelling. Hence, DeltaV OPC Mirror was used to map the data from the DeltaV OPC server to a MatrikonOPC Simulation server. The finished DeltaV HMI in DeltaV Operate Run is shown in Fig. 14. Note that this HMI is intended only for manual update of control values during remote control.

Figure 14. The DeltaV user interface, where the operator is able to change by activating the the control variables for the uArm.

3.3. Remote Control

Local testing was done by connecting the DeltaV Pro Plus and the LabVIEW computer using an Ethernet cable. OPC DataHub was installed on the LabVIEW computer and the server side of the OPC tunnel was configured on the DeltaV Pro Plus. The IP address of the computer running the OPC Server is used for remote connection between the computers. For added security the OPC tunnel may be configured with a password [14]. The connection was working locally as the values were successfully transferred from DeltaV to LabVIEW. The finished LabVIEW uArm program with the DataSocket SubVI were sent to Poland for remote testing. OPC DataHub was installed on the LabVIEW computer in Poland and the connection was successful as the change of data from DeltaV could be seen internally in the OPC DataHub program. The LabVIEW SubVI would not run as an error appeared when opening the DataSockets. This solution did not work and was discarded.

The LabVIEW DSC Module supports shared variables that can be used for connecting to OPC Tags. In LabVIEW Project Explorer, an I/O Server was created and connected to the OPC Server1. Shared variables was connected to the specific OPC tags through the I/O Server. The old content in the OPC DataSocket SubVI was replaced with the Shared variables. The new SubVI is shown in Fig. 15, where the Shared Variables are connected to the numerical outputs of the SubVI.

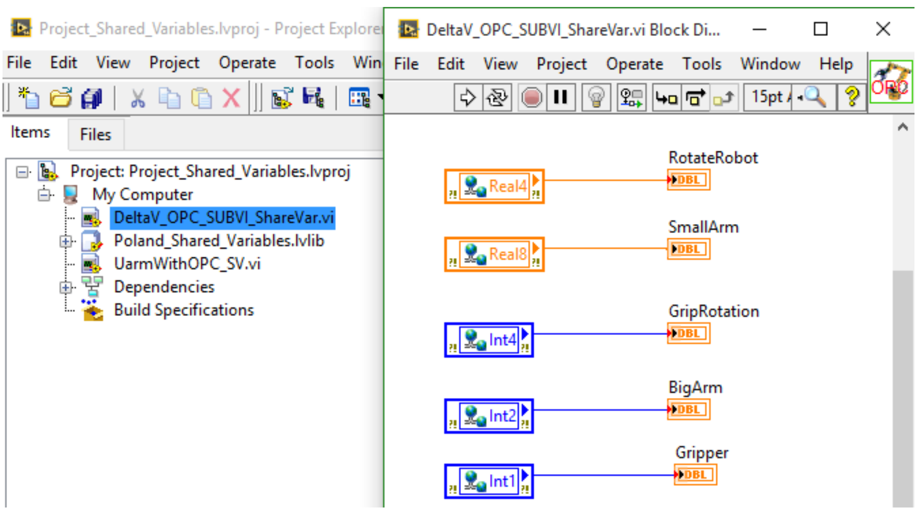

Figure 15. LabVIEW project and VI showing shared vaiables that are connected to numerical outputs

The final data flow for remote control of the robot is shown in the block diagram in Fig. 16.

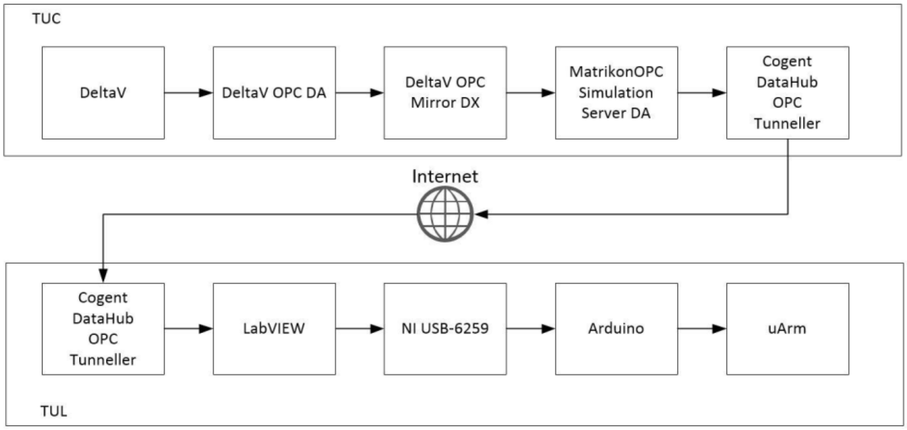

Figure 16. Dataflow from DeltaV in USN to the uArm located at Technical University of Lodz

A good source of information on many aspects related to programing discussed here can be found in [15].

4. Conclusions

The Polish team of students created a LabVIEW program for controlling a robotic arm locally in Lodz, and made an application for the robotic arm to be used in solving environmental issues.

The Norwegian team further developed the LabVIEW program to be able to communicate with DeltaV using OPC. An operational connection between Telemark University College and Lodz University of Technology is established by using Cogent OPC DataHub Tunneller. DeltaV is used for manual update of the variables used to control the robotic arm. DeltaV OPC Mirror is used to map the OPC tags from the DeltaV OPC Server to a MatrikonOPC Simulation Server. The transferred OPC data is read in LabVIEW and transferred to the Arduino connected to the uArm.

The communication between LabVIEW and DeltaV is working, and the LabVIEW user interface has the option to activate DeltaV control. In the DeltaV interface, it is possible to manually change the control variables.

It is important to note that in robotics, much faster PLC systems are used. As described above, DeltaV is conceived as a DCS normally used in batch and continuous processes, normally not rquiring swift response. Normal response times in Delta V is in the range of 100ms to some seconds.

However, as an acdemic projcet testing cross-platform operations, the remote control by the team in Porsgrunn, Norway of a robot arm in Lodz in Poland was successfully accomplisehd. .

6. Acknowledgement

Advice and support by experts at National Instruments, particularly Tom Arne Danielsen and at Emersions Process Management are greatly appreciated. Emerson Process Mangement supported USN with issues related to license in the usage of DeltaV modules. The mobility of student and staff between TUL and TUC/HSN was facilitated by the EEA project.

7. References

[1] Skourup C , John Pretlove J: 2011, “Remote inspection and intervention - Remote robotics at work in harsh oil and gas environments”, ABB Review 2/11, pp 50-55.

[2] Anisi D A , Skourup C: “A step-wise approach to oil and gas robotics”, 2012, Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Norwegian University of Science and Technology, Trondheim, Norway, May 31 - June 1, 2012.

[3] http://www.bbc.com/future/story/20140516-i-operate-on-people-400km-away

[4] Schreckenghost D , Milam T, Fong T ; “Measuring Performance in Real Time during Remote Human-Robot Operations with Adjustable Autonomy“, IEEE Intelligent Systems, September/October 2010, pp1541-1672.

[5] http://www.nasa.gov/centers/ames/K10/

[6] Isaksen K , Andersen D, Thorsen S A, Hagen R, Hagen S: 2012 , «DeltaV Prosessautomasjonssystem”, IA5510 Prosjekt IA5, Student Project Report, Telemark University College.

[7] Kellett P, , Robotics Industries Association, “Robotic Warehousing: A New Market Opportunity for Robot Manufacturers and Integrators?”, 2011, http://www.robotics.org/content-detail.cfm/Industrial-Robotics-News/Robotic-Warehousing:-A-New-Market-Opportunity-for-Robot-Manufacturers-and-Integrators/content_id/2574

[8] Heyer C. ,2010 . Human-robot interaction in future industrial robotics applications. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4749{4754. Taipei, Taiwan.

[9] Fathizadeh M, Cody J E: “Design of a mobile robot and use of RFID for fast Warehousing”, AC 2012-4826, American Society for Engineering Education,

[10] http://www.bostondynamics.com/

[11] EVOL uARM, http://store.ufactory.cc/uarm-metal/

[12] Cogent DataHub. “Advanced OPC Tunnelling”, 2015. [Online]. Available at: http://www.opcdatahub.com/Features/OPC_Tunnelling.html
[13] Cogent DataHub. “DataHub OPC Tunneller”, 2015. [Online]. Available at: http://www.cogentdatahub.com/Products/DataHub_OPC_Tunneller.html
[14] Cogent DataHub. “OPC Support”, 2015. [Online]. Available at: http://www.cogentdatahub.com/Features/OPC_Support.html
[15] http://home.hit.no/~hansha