Plug-and-play design and distributed logic control of medical devices using IEC 61499 function blocks

Majid Sorouri*

Department of Electrical and Computer Engineering,
The University of Auckland,
Building 301, 38 Princes Street,
Auckland 1142, New Zealand
E-mail: msor021@aucklanduni.ac.nz
*Corresponding author

Valeriy Vyatkin

Department of Computer Science, Electrical and Space Engineering,
Luleå University of Technology,
SE-971 87 Luleå, Sweden
E-mail: valeriy.vyatkin@ltu.se

Shengquan Xie

Department of Mechanical Engineering,
The University of Auckland,
Building 401, 20 Symond Street,
Auckland, New Zealand
E-mail: s.xie@auckland.ac.nz

Zoran Salcic

Department of Electrical and Computer Engineering,
The University of Auckland,
Building 301, 38 Princes Street,
Auckland 1142, New Zealand
E-mail: z.salcic@auckland.ac.nz

Abstract: Plug-and-play control design and interoperability of medical devices has now become a decisive mission of technological research. One of the reasons for these requirements is the ever increasing attention that the medical sector is paying these days to specific treatments each individual patient requires which mandates having medical devices that are customisable and, more importantly, are able to interact with each other to handle more complex tasks and, as well, avoid more risks. Currently, vendor-specific devices and proprietary communication systems have hindered clinical environments from thoroughly reaching these goals. Thus, this paper concentrates on the software development aspect of medical devices and proposes a novel approach for the development of distributed control logic for them using modular, reusable and interoperable software components based on the IEC 61499 function blocks (FB). This technology enables the control design of the entire application to be in one software tool and consequently, alleviates design complexity and development time. Following that, it presents a simplified case study to exhibit the viability of this methodology to be exploited for control software design of medical and rehabilitation devices.

Keywords: medical robotics; plug-and-play software components; medical devices; distributed control design; IEC 61499; function blocks; upper-limb rehabilitation device; biomechatronics; biomedical robotics.

Reference to this paper should be made as follows: Sorouri, M., Vyatkin, V., Xie, S. and Salcic, Z. (2013) 'Plug-and-play design and distributed logic control of medical devices using IEC 61499 function blocks', *Int. J. Biomechatronics and Biomedical Robotics*, Vol. 2, Nos. 2/3/4, pp.102–110.

Biographical notes: Majid Sorouri received his BSc in Mechanical Engineering from KNT University, Tehran, Iran in 2002 and MSc in Intelligent Manufacturing Systems from the University of Birmingham, UK in 2004. He has more than five years of work and teaching experience in mechatronics and industrial automation. Since 2010, he has been working toward his PhD in the Department of Electrical and Computer Engineering, The University of Auckland, New Zealand. His current research interests include multi-agent systems, semantic web technologies and distributed control design of intelligent mechatronic components for biomedical and industrial applications.

Valeriy Vyatkin is Chaired Professor (Ämnesföreträdare) of Dependable Computation and Communication Systems at Luleå University of Technology, Sweden. He also teaches part-time with The University of Auckland, New Zealand, where he has been Associate Professor and Director of the InfoMechatronics and Industrial Automation lab (MITRA) at the Department of Electrical and Computer Engineering. His research interests are in the area of dependable distributed automation and industrial informatics, including software engineering for industrial automation systems, distributed architectures and multi-agent systems applied in various industry sectors. In 2012, he has been awarded Andrew P. Sage Award for best *IEEE Transactions* paper.

Shengquan (Shane) Xie received his PhDs from Huazhong University of Science and Technology (China), and University of Canterbury (New Zealand). He is Chair Professor in Mechatronics Engineering at University of Auckland, New Zealand. His research interests include intelligent mechatronics systems, vision techniques and applications, smart sensors and actuators, bio-mechatronics and biomedical robotics. He is the Editor and Associate Editor of two international journals, and is an editorial board member and scientific advisory member for many international journals and conferences. He has published more than 200 papers in refereed international journals and conferences.

Zoran Salcic is a Professor of Computer Systems Engineering at the University of Auckland. His main research interests include digital systems design, custom-computing machines, reconfigurable computing, processor and computer systems architectures, embedded systems and their implementation, design automation tools for embedded systems, hardware-software co-design, new computing architectures and models of computation and languages for heterogeneous systems and wireless sensor and actuator networks. He has published more than 250 peer-reviewed journal and conference papers, several books and numerous technical reports. He is a Fellow of the Royal Society New Zealand (Academy of Science) and recipient of Alexander von Humboldt Research Award.

This paper is a revised and expanded version of a paper entitled ‘Distributed control design of medical devices using plug-and-play IEC 61499 function blocks’ presented at International Conference in Mechatronics and Machine Vision in Practice (M2VIP), M2VIP 2012, Auckland, 28–30 November 2012.

1 Introduction

Over the last years, there has been an increasing demand for employing mechatronic devices for medical applications. The trace of such devices can now be found in a wide range of activities, namely, healthcare, rehabilitation, treatment and surgery and, in fact, the medical domain in general has become highly reliant on them. Along with the outstanding technological advances achieved in the development of such systems, the requirements have also dramatically increased. Nowadays medical devices are expected to address every patient’s individual needs. This mandates them to have more capabilities such as being customisable, have interaction with each other to handle more complex tasks dexterously, and avoid clinician’s faults and minimise treatment risks on patients.

Among the various efforts spent on the development of versatile medical devices, a considerable share is devoted to software development and the ratio is expected to grow rapidly, similar to the industrial automation marketplace (Vyatkin, 2013).

There exist some challenges regarding the software characteristics of medical devices manufactured by almost fifteen thousand companies (Arney et al., 2007). These are influenced by the nature of conventional software development approaches that are mainly centralised, on a case-by-case basis and demand a high amount of programming effort. The other crucial problem is that they are vendor-specific. Most medical devices do not support interoperability and merely employ proprietary protocols for system integration which lack fundamental capabilities required for collectivism among cross-manufacturer medical devices for the care of a patient (Goldmann, 2009).

In order to respond to the aforementioned challenges, researchers have set various objectives for the software part of these devices including modular and PnP control design, and distributed control, along with the implementation of open standards and interoperable technologies to provide more integral approach to the clinical environment and ultimately result in boosting treatment quality. Achieving these objectives paves the way for engineers to rapidly and more conveniently use these software components in the development of their innovative and customised medical systems. In the survey of Australian surgeons which was conducted in 2003 (Hofmann, 2007), the survey editor has evidently emphasised this essential demand when he mentioned: "Surgeons wanted 'Plug-and-Play' (PnP) components like those for computers, or a bioengineer in the OR (operating room). There was a lack of standardisation between different brands" (Patkin, 2003).

This paper aims at contributing to this research field by addressing the complexity associated with control software design and development using a distributed control architecture based on the IEC 61499 FBs and demonstrate the proposed approach on a case study of upper limb rehabilitation device. The rest of this paper is structured as follows: Section 2 provides a review of the recent literature related to control software design and implementation in medical context. Section 3 introduces the IEC 61499 distributed control architecture and illustrates its key benefits to this domain. In Section 4, a case study of an upper limb rehabilitation device is chosen as a proof of concept that is implemented with function blocks (FB). The conclusions are made in Section 5 and finally, a road map for future development is drawn in Section 6.

2 Related work

There have been numerous studies related to control software design and implementation of medical devices to foster PnP design, and interoperability of devices that use mechatronic components along with simplifying their integration with broader medical environments. This section provides insight into some of these practices.

The recent work in robotic rehabilitation has mainly focused on the development of complex robotic mechanisms characterised by many degrees-of-freedom to facilitate sophisticated movements, such as walking and hand movement (Marchal-Crespo, 2013). Similarly, Gunasekara et al. (2012) review the control methods used in the exoskeleton robots and describe that the structure of a control system for a robot is affected by its degree of freedom (DOF). According to them, exoskeleton robots with higher DOF exhibit enhanced manoeuvrability; however, centralised control systems do not support their implementation. Therefore, distributed control configuration is highlighted as a promising solution to reach high manoeuvrability as well as handling control of high DOF robots.

Kim et al. (2010) elaborate on the interoperability of distributed medical devices interconnected in an open space by introducing a network-aware supervisory system (NASS). They describe three crucial features as *real*-time, *inter-device dependencies* and *unreliable communication* that intricate the design of safety-critical supervisory systems in medical environments. Taking into account these considerations, they concentrate on safety and error avoidance in surgical environments and developed a NASS to incorporate medical devices and diminish risk of harm or death in surgical processes. Also, to demonstrate interoperability, a case study of an airway-laser surgery is presented. For instance, before laser activation request is responded to, the oxygen flow has to stop prematurely to avoid accidental burn on humans. Likewise, patient safety by interoperability of PnP medical devices is addressed in Arney et al. (2010). In this research, a clinical case study of a ventilator and a simulated X-ray interacting during surgery is described.

Hofmann et al. (2007) propose a new standard for an integrated clinical environment manager (IECEMAN), which describes medical devices by a meta-model and, as well, provides a communication protocol allowing PnP connectivity among compliant medical devices.

One of the leaders in promoting interoperability among medical equipment is the medical device plug-and-play (MD PnP) programme established in 2004. They adopt medical device interoperability by developing databases and tools for a wide range of applications, such as diagnosis, treatment, research, safety and so on (Medical Device 'Plug-and-Play' Interoperability Programme, http://www.mdpnp.org/). In this approach, they develop and support open standards, including integrated clinical environment (ICE), ASTM F2761-2009. Through the functions elaborated on in the 'patient-centric ICE' standard, some capabilities, such as real-time decision support, safety interlocks and closed-loop control can be achieved.

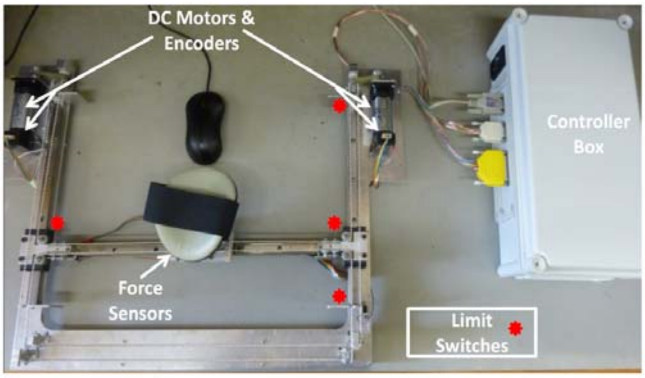
The past, present and future of medical devices, encompassing both their software and hardware characteristics, capabilities and challenges are comprehensively described in the US Government report (High Confidence Software and Systems Coordinating Group, 2009). Most of the recent literature on PnP medical devices has been dedicated to dynamic connectivity of devices and their synchronisation (Kim et al., 2010). However, in spite of the multitude of research that has taken place over the years on this subject, biomedical engineers are still lacking proper platforms that enable them to build clinical systems and integrate cross-vendor medical devices in a PnP and interoperable manner. Even the current standards available for this sector have rarely been adopted by medical manufacturers, and this is due to their complexity along with providing weak support for legacy devices (Hofmann, 2007).

The research presented in this paper proposes a reliable solution based on the IEC 61499 standard to enhance the process of distributed control design and interconnectivity of devices in medical environments, into a robust and manageable method.

3 A rehabilitation use case

To realise the potential advantages of the proposed method, a case study which presents how to design modular, reconfigurable and distributed control (Sorouri et al., 2012) for rehabilitation, and in general, medical devices using FBs, was undertaken. The case study selected device is a two-degrees-of-freedom upper limb rehabilitation device (Figure 1) that has been designed, manufactured and optimised by a number of students at the mechatronics laboratory in the Department of Mechanical Engineering of the University of Auckland (Wei, 2011; Noellat, 2011; Dass, 2008).

Figure 1 The upper limb rehabilitation device (see online version for colours)



The aim of designing such a device is to help a vast number of people who have survived a stroke but still suffer from its negative effects on the body in the form of lost ability to move limbs. Researchers have proven that task-oriented recursive movements can considerably help those patients recover the mobility of their damaged limbs (Wei, 2011). The device has been designed to provide such required therapy to the patients, either at clinical centres or patients' homes, without having the complexity of conventional devices, as well as having a reasonable price. In fact, simplicity of use, being safe, light-weight and portable has made it an attractive solution for patient's personal therapy, and enables them to utilise the device at home while receiving feedback from the device about their therapy activities through an intuitive graphical user interface (Rupp et al., 2009).

3.1 Device specifications

3.1.1 Mechanical components

The mechanical components of the device include a light U-shaped aluminium frame along with a sliding bar, which holds the armrest (for strapping the patient's hand) and is driven by metal chains transmitting power from the motors to the armrest. A number of plastic idler pulleys are mounted around the device to conduct the chain movement around the device and the chain route has been designed in a way that each motor rotation will cause the armrest to move along one of its diagonals (45º and 135º). As a result, in order to move in various directions, a combination of both motor rotations is required. For instance, for the armrest to move forward (away from where patient is sitting), the left motor must rotate anticlockwise and the right motor must rotate clockwise with the same speed, so that the resultant forces move the device forward. The device also assists a patient to use a computer mouse to interact with the computer.

3.1.2 Electrical components

The electrical parts of the device include: two brushed DC motors as actuators, two rotary encoders for identifying motor positions, four limit switches for safety consideration (identifying the boundary limits), two accelerometers as head tilt sensors (for two axes) embedded in a helmet to measure the head's angles and, finally, two strain gauges for armrest force measurement (to receive feedback force). Two analogue servo drives are also employed to magnify the 0–10 voltage sent out from the PLC analogue outputs to be fed into the two 24 DC motors. Also, PID control was employed for both motors to be able to follow the given set points of the trajectories.

3.1.3 Device operation modes

To satisfy the treatment requirements, the device's control programme is considered to run under the following six operation modes (Wei, 2011):

- Manual mode: In this mode, the patient can move the mouse cursor over the virtual joystick pad and simply click to intuitively specify the desired coordinates.
- Head tracking mode: In this method, a helmet with strapped accelerometer sensors is used to determine where the patient is looking at the monitor, and moving their hand on the workspace based on the measured tilt angle.
- Assist mode: If the patients are not totally disabled, but the force they apply to the armrest is not adequate to move it, force sensors will measure the forces in two directions and, based on that, guide the armrest toward that direction.
- Resist mode: This mode is utilised when patient's limb strengths are almost completely recovered and they merely require some more exercises to complete their treatment. Therefore, a threshold will be defined for each force sensor and if the exerted force to the armrest is greater than that threshold, the device will produce the same amount of force, but in the opposite direction to avoid armrest movement.
- Trajectory mode: In this mode, therapist's or patient's attempt to guide the arm through predefined trajectories defined in the device programme or record the manual movement of the armrest at the first trial and repeat it several times.
- Game mode: Special purpose games can significantly assist patients by increasing their motivation and making the rehabilitation process more appealing (Andrade et al., 2013). The interactive game developed for this device is to encourage patients during the repetitive and tiring therapy process so that they withstand the rehabilitation exercises better and longer.

In this simple game, the patient must move the armrest to reach the goal circle at random positions.

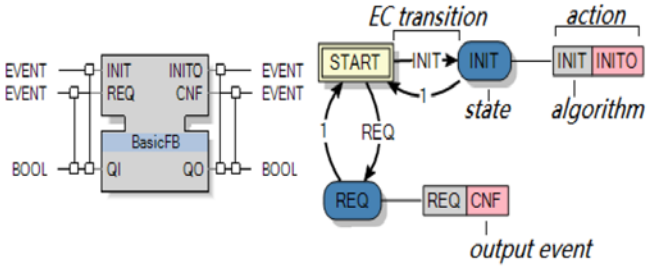
4 Component-based distributed programming architecture of the IEC 61499

The IEC 61499 standard emerged to facilitate the encapsulation of functionality and intelligence into software components defined as FB and to distribute them across the entire network of devices. So far, this technology has successfully been implemented both in industrial and research environment aiming at tackling the aforementioned control design and implementation concerns. Some of the areas where FBs have been successfully employed include smart grid systems (Vyatkin et al., 2010), baggage handling systems (Black and Vyatkin, 2010), building automation systems (Deng et al., 2013) and shoe manufacturing (Brusaferri et al., 2009). Other use cases of the IEC 61499 technology along with its benefits have been reported in Vyatkin (2011). However, the range of application and effectiveness of this technology is not confined to the specified industry sector and, likewise, can tap into medical and healthcare applications. This is due to the fact that the general concept of control logic design is similar in different domains and, as well, the accuracy and precision of the current industrial machineries and robots implemented with FBs are somehow comparable with the advanced robots employed in the medical sector. For instance, there is a Delta robot prototype that was developed in 2010 by NxtControl (http://www.nxtcontrol.com; http://vimeo.com/33566705) that is fully programmed and controlled with the IEC 61499 technology. This is why a similar Delta robot type (when customised and armed with the appropriate tools) can be used in surgical operations, as it has already been used for maxillofacial surgery (Lueth et al., 1998; Cleary and Nguyen, 2001). Furthermore, this implementation demonstrates how high speed and real-time control of robots can be achieved through this technology.

This open standard enables biomedical engineers to initially simulate and debug their whole application's control code, independent of the hardware brand/type which then can be conveniently chosen at later design stages. In addition, it allows them to dexterously manage the complexity of such medical systems and distribute control of their medical application across a number of medical devices.

The primary constituent of FB architecture is a basic function block (BFB). It consists of a group of data and event inputs/outputs along with a state machine known as executive control chart (ECC) which is in charge of executing the invoked algorithms to achieve the intended functionality of the FB (Figure 2). Also, each FB can be linked with a visualisation entity (HMI FB) to be used for testing, simulation and monitoring purposes.

Figure 2 A BFB, (a) interface and (b) ECC (see online version for colours)



(a) (b)

Source: Cheng et al. (2011)

By interconnecting two or more BFBs, more capable blocks, named composite function blocks (CFBs) can be created inheriting their functionalities from their constituting BFBs. Once all the required functionalities are either selected from the FB libraries (drag and drop) or manually created (in cases when a particular functionality does not exist), they can be instantiated as many times as required and connected to one another to build the whole application. Currently, NxtStudio (http://www.nxtcontrol.com) and ISaGRAF (Albertos and Mareels, 2010) are two commercial software tools which are compliant with the IEC 61499 and their code is deployable into various available hardware such as Siemens, Wago, Beckhoff and Advantech controllers (Vyatkin, 2011). However, the number of these conformable controllers is growing.

5 Implementation of the use case based on the IEC 61499 FB

5.1 Control software

To implement the control design for the upper-limb rehabilitation device, at first, the physical components that take part in the application must be identified informally and their interactions with the rest of the system to create the desired behaviour be specified as illustrated in Figure 3. Once the physical components are selected, the necessary FBs implementing. sensing, control or actuation, are selected from the range of available library elements and instantiated in the application environment. These FBs are modular and application-independent such as the two sensor FBs each representing two limit switches (Figure 3). Then, each FB is configured based on the specific values and physical characteristics of its associated parts. These essential software components can then take part in the application as stand-alone FBs or may be embedded into other FBs.

The next and most crucial milestone is to design the control logic for each component based on their behaviour. A controller FB needs to analyse the obtained information (from the plant and other blocks) and based on that information, make the appropriate decisions. The 'If-Statement' conditions represented by arrows in the ECC are considered as *decisions*, due to the fact that they identify when and how an actuation must take place.

The procedure chosen for the controller FB design is known as *self*, *task*, and *environment* perception (STEP) which is defined in Sorouri et al. (2012). Following these steps the control development process is completed leading to the precise definition of states, conditions and algorithms of a controller FB. Taking into account this approach, six separate controller FBs (one for each operation mode) were designed for this device. They operate in parallel and constantly provide the desired X and Y positions for the armrest. The hand position is also illustrated by an armrest position FB (FB No. 7 in Figure 4) that instantly receives the current position of the armrest and updates its location on the display. Once the XY coordinates are generated by the different mode FBs, they will be passed to the actuators based on the user's selected mode and converted into encoder counts. These values are fed to the two motor FBs with PID controllers as set points along with other data from motor encoders and limit switches.

Figure 3 Layout of the interactions between the constituent components in the upper-limb device (see online version for colours)

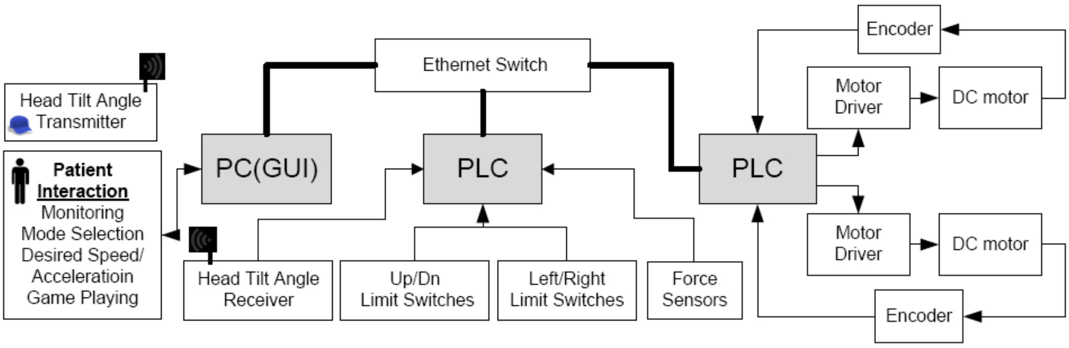
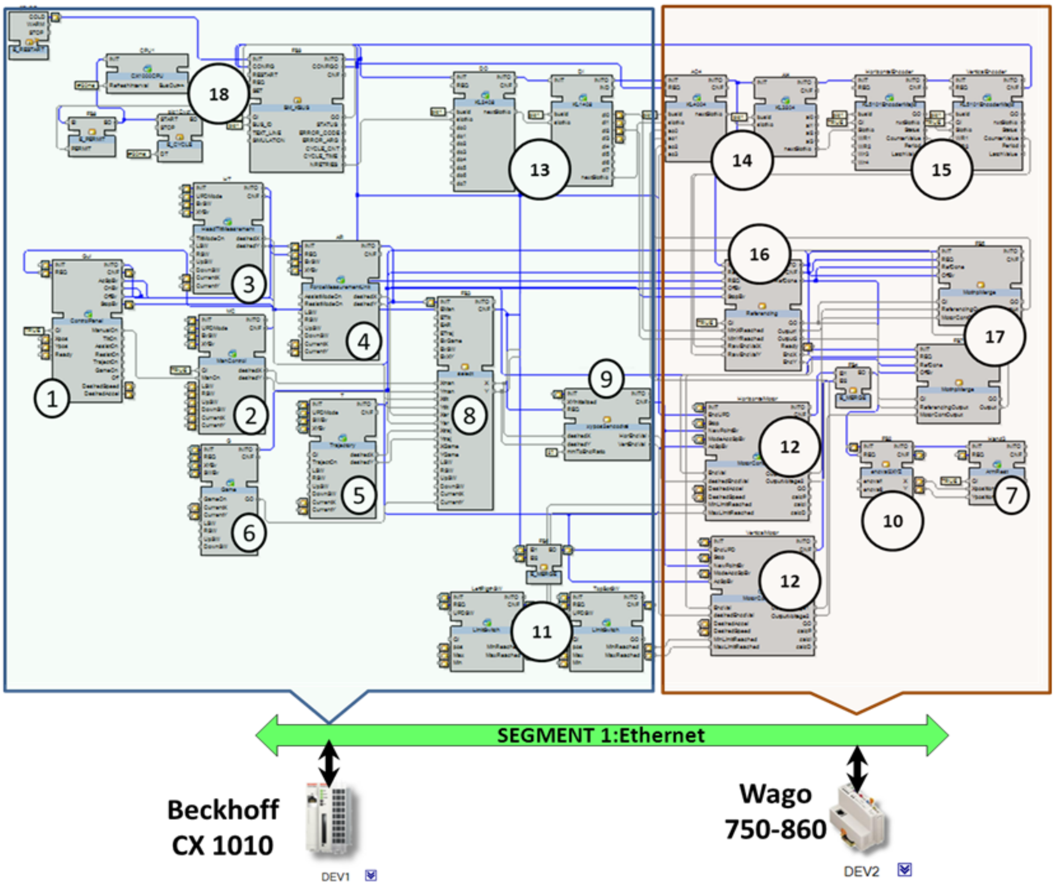


Figure 4 Upper limb rehabilitation device implemented in NxtStudio IDE with a network of FBs (see online version for colours)



Notes: Network of FBs includes: 1 – control panel, 2 – manual mode, 3 – head tilt measurement mode, 4 – assist/resist mode, 5 – trajectory mode, 6 – game mode, 7 – armrest positioning, 8 – mode selector, 9 – XY to encoder value converter, 10 – encoder to XY converter, 11 – up/down and left/right limit switches, 12 – motor controllers, 13 – digital IO modules, 14 – analogue IO modules, 15 – encoder reader modules, 16 – initialisation mode (reference positioning), 17 – merge initialisation and motor controller output, 18 – hardware management FBs.

The motor controller FB receives the necessary motion settings, such as speed and acceleration, and feeds them into the velocity profile FB which utilises a set of algorithms to interpolate the positions that the armrest has to travel to. As a safety interlock, the operation of each motor is interconnected to the other one. For instance, motors' controllers are designed in a way that failure in one motor will promptly stop the other motor to avoid undesirable movement in the armrest.

At the later stage, the number of IOs required for this application to be interfaced with the device must be identified. Thus, the intended control hardware and IO modules (e.g., analogue/digital/encoder reader) are dragged and dropped from the FB libraries to the application. The network of FBs with different encapsulated functionalities that are involved in this application are illustrated in Figure 4.

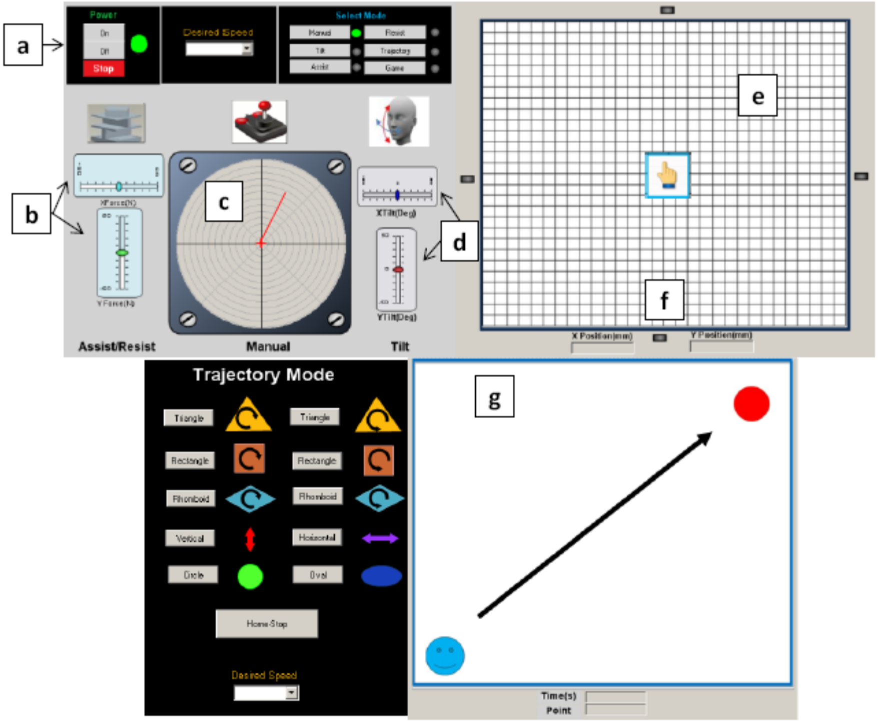
5.2 Control hardware

The ultimate vision in the intelligent mechatronics approach is that every mechatronic unit can be equipped with a microcontroller, executing a part of the entire systems' control application. In this study, to demonstrate some of the capabilities of the proposed method, such as distributed control and hardware independence, the developed FB network was divided into just two parts allocated for execution on two different hardware controllers (also shown as PLCs in Figure 3), namely a Beckhoff CX 1010 and a Wago 750–860 controller (Figure 4). However, in general, according to IEC 61499, every FB can be assigned for execution to any available compliant hardware. The communication between the two hardware controllers is achieved through the Ethernet network as depicted in Figure 4. Prior to this scenario, the device had been tested under centralised control in a way that the whole FB network was implemented in a single Beckhoff hardware controller. The device performance observed under the two 'centralised' and 'distributed' conditions were identical and no significant difference in the device behaviour was noticed. The NxtStudio editor and its visualisation both function on Windows XP and Win7 platforms and are able to communicate with the controllers through TCP/IP.

5.3 Graphic user interfaces

For this application, three graphic user interfaces (GUIs) are considered to be displayed for the patient on the PC to enable user interaction with the rehabilitation device, including *general*, *trajectory* and *game* canvases. When the FB network is completed by inserting the necessary FBs in the application, the graphical interface of each FB that has been developed at earlier design stages will be dragged and dropped into the associated project canvases to provide the desired visualisation and control panel for the users in an easy and user-friendly manner. As illustrated in Figure 5, these canvases comprise a number of displays, such as work space, allowing the user to monitor the armrest position, safety limit switches status, reading tilt angles and applied forces. In addition, the user is able to configure the device by entering the desired position, speed and acceleration, toggle between different modes and so on.

Figure 5 Different GUIs for the upper limb rehabilitation device implemented in NxtStudio IDE (see online version for colours)



Notes: a – control panel, b – force measurement of armrest, c – virtual joystick pad (manual mode), d – head tilt angle, e – workspace, f – XY coordinates of armrest, g – game mode.

5.4 The benefits gained through usage of FB in the case study

The control code of this case study had initially been developed by a number of students using conventional programming methods (e.g., programming in C language and executed on microcontrollers) as reported in Wei (2011), Noellat (2011), Dass (2008) and Sorouri et al. (2012). Exploiting FBs in the control of this rehabilitation device revealed a number of benefits compared with previous approaches including:

- The readability and maintainability of the control code was improved;
- Thanks to the modularity of the software in FBs and ease of reconfigurability, the software evolvement took place along with modification of physical components and made the design process easier and faster. For instance, it is possible to rapidly add more complex trajectories based on exploitation of patients' feedback and implementing them as control algorithms (Malosio et al., 2010).
- The design of the entire control system along with the visualisation was implemented in one software tool (NxtStudio IDE);
- Using prefabricated, modular and reusable software components from numerous FB libraries (e.g., limit switches, motors, hardware modules) in the application, saved a significant amount of development time and, more importantly, allowed rapid and flexible adaptation of the system based on the specific needs of stroke patients (Mullins et al., 2005).
- Hardware independence of the design allowed changes in the hardware at any of the design stages. This brings more flexibility to the design and benefits from current proprietary devices with hard-coded functionalities.

6 Conclusions and future work

While the IEC 61499 architecture is not new in the design and control of mechatronic components in industrial automation systems, we demonstrated its applicability as a mature technology in medical devices domain and delineated the remarkable benefits it can provide to this sector such as PnP software design, reusability of software components and the distribution ability of those components across various hardware controllers, openness, system level design and interoperability. These advantages are of particular importance for handling complexity and customisation of large medical applications where software components can be conveniently assembled together in a PnP manner similar to physical components to build diverse medical systems.

Then, a case study was used to illustrate this method in the control of an upper-limb rehabilitation device and the performance and safety of the device was tested. This use case demonstrated interoperability among two different hardware units executing FBs representing mechatronic components of the target device; however, the concept can be conveniently applied to different medical devices interacting to fulfil their common goals.

In continuation of the current research, three research directions are envisaged as our future works:

1 using semantic web technologies for developing an engine to automatically generate such required FB networks based on requirements

2 examining this technique on more complex, redundant and network-connected medical mechatronic devices

3 using formal verification methods, such as model-checking approach (Patil et al., 2012), to automatically verify the control model of this case study designed with FBs for safety, liveliness and functional properties of the medical equipment.

This is due to the fact that at the moment there is no widely accepted method for verification of software in medical devices (Arney et al., 2007).

References

Albertos, P. and Mareels, I. (2010) *Feedback and Control for Everyone*, Springer, Verlag Berlin Heidelberg.

Andrade, K.d.O., Fernandes, G., Martins, J., Roma, V.C., Joaquim, R.C. and Caurin, G.A.P. (2013) 'Rehabilitation robotics and serious games: An initial architecture for simultaneous players', in *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP*, pp.1–6.

Arney, D., Goldman, J., Whitehead, S. and Lee, I. (2010) 'Improving patient safety with x-ray and anesthesia machine ventilator synchronization: a medical device interoperability case study', in A. Fred, J. Filipe and H. Gamboa (Eds.): *Biomedical Engineering Systems and Technologies*, Vol. 52, pp.96–109, Springer, Berlin Heidelberg.

Arney, D., Jetley, R., Jones, P., Insup, L. and Sokolsky, O. (2007) 'Formal methods based development of a PCA infusion pump reference model: generic infusion pump (GIP) project', in *High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, HCMDSS-MDPnP, Joint Workshop on*, pp.23–33.

Black, G. and Vyatkin, V. (2010) 'Intelligent component-based automation of baggage handling systems with IEC 61499', *Automation Science and Engineering, IEEE Transactions on*, Vol. 7, No. 2, pp.337–351.

Brusaferri, A., Ballarino, A. and Carpanzano, E. (2009) 'Enabling agile manufacturing through reconfigurable control solutions', in *Emerging Technologies & Factory Automation, ETFA 2009, IEEE Conference on*, pp.1–8.

Cheng, P., Yan, J., Vyatkin, V. and Jennings, S. (2011) 'Distributed IEC 61499 material handling control based on time synchronization with IEEE 1588', in *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2011 International IEEE Symposium on*, pp.126–131.

Cleary, K. and Nguyen, C. (2001) 'State of the art in surgical robotics: clinical applications and technology challenges', *Computer Aided Surgery*, Vol. 6, No. 6, pp.312–328.

Dass, R.K. (2008) *Upper Limb Rehabilitation Device for Home Use*, Project Report, Department of Mechanical Engineering, University of Auckland.

Deng, Y., Sorouri, M., Pang, C. and Vyatkin, V. (2013) 'A distributed automation architecture enabling simulation-in-the-loop of energy-efficient buildings', in *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, pp.542–547.

Goldmann, J.M. (2009) *Medical Devices Interoperability to Enable System Solutions at the Sharp Edge of Healthcare Delivery*, White House Homeland Security Council Biodefense Directorate Conference presentation, April.

Gunasekara, J.M.P., Gopura, R.A.R.C., Jayawardane, T.S.S. and Lalitharathne, S.W.H.M.T. (2012) 'Control methodologies for upper limb exoskeleton robots', in *System Integration (SII), 2012 IEEE/SICE International Symposium on*, pp.19–24.

High Confidence Software and Systems Coordinating Group (2009) *High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care*, A Research and Development Needs Report, NITRD, February.

Hofmann, R.M. (2007) *Modeling Medical Devices for Plug-and-Play Interoperability*, Master of Engineering Thesis, Massachusetts Institute of Technology.

Implementation of Delta Robot with the IEC 61499 at NxtControl Co. [online] http://vimeo.com/33566705.

Kim, C., Sun, M., Mohan, S., Yun, H., Sha, L. and Abdelzaher, T.F. (2010) 'A framework for the safe interoperability of medical devices in the presence of network failures', presented at the *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, Stockholm, Sweden.

Lueth, T.C., Hein, A., Albrecht, J., Demirtas, M., Zachow, S., Heissler, E., Klein, M., Menneking, H., Hommel, G. and Bier, J. (1998) 'A surgical robot system for maxillofacial surgery', in *Industrial Electronics Society, IECON '98, Proceedings of the 24th Annual Conference of the IEEE*, Vol. 4, pp.2470–2475.

Malosio, M., Pedrocchi, N. and Tosatti, L.M. (2010) 'Robot-assisted upper-limb rehabilitation platform', presented at the *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, Osaka, Japan.

Marchal-Crespo, L. (2013) 'Control strategies for robot-assisted training – literature review and experimental impressions', in J.L. Pons, D. Torricelli and M. Pajaro (Eds.): *Converging Clinical and Engineering Research on Neurorehabilitation*, Vol. 1, pp.115–120, Springer, Berlin Heidelberg.

Medical Device 'Plug-and-Play' Interoperability Programme [online] http://www.mdpnp.org/.

Mullins, J., Mawson, C. and Nahavandi, S. (2005) 'Haptic handwriting aid for training and rehabilitation', in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, Vol. 3, pp.2690–2694.

Noellat, C. (2011) *Development of an Upper Limb Rehabilitation Device for Home Use*, Department of Mechanical Engineering, University of Auckland.

Nxtcontrol Company [online] http://www.nxtcontrol.com.

Patil, S., Vyatkin, V. and Sorouri, M. (2012) 'Formal verification of intelligent mechatronic systems with decentralized control logic', presented at the *17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Krakow, Poland.

Patkin, M. (2003) 'What surgeons want in operating rooms', *Minimally Invasive Therapy & Allied Technologies*, November, Vol. 12, No. 6, pp.256–262.

Rupp, R., Plewa, H., Hofer, E.P. and Knestel, M. (2009) 'MotionTherapy@Home – a robotic device for automated locomotion therapy at home', in *Rehabilitation Robotics, ICORR 2009. IEEE International Conference on*, pp.395–400.

Sorouri, M., Patil, S. and Vyatkin, V. (2012) 'Distributed control patterns for intelligent mechatronic systems', *10th International Conference on Industrial Informatics*, Beijing, China.

Sorouri, M., Vyatkin, V. and Xie, S. (2012) 'Distributed control design of medical devices using plug-and-play IEC 6199 function blocks', presented at the *M2VIP, 19th International Conference on Mechatronics and Machine Vision in Practice*, Auckland New Zealand.

Vyatkin, V. (2011) 'IEC 61499 as enabler of distributed and intelligent automation: state of the art review', *IEEE Transactions on Industrial Informatics*, Vol. 4, No. 4, pp.768–781.

Vyatkin, V. (2013) 'Software engineering in industrial automation: state of the art review', *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 3, pp.1234–1249.

Vyatkin, V., Zhabelova, G., Higgins, N., Ulieru, M., Schwarz, K. and Nair, N.K.C. (2010) 'Standards-enabled smart grid for the future energy web', in *Innovative Smart Grid Technologies (ISGT)*, pp.1–9.

Wei, C-C. (2011) *Development of an Upper Limb Rehabilitation Device for Home Use*, MSc thesis, Department of Mechanical Engineering, University of Auckland.