*Journal of Mechanical Engineering* *Vol 15(2), 127-141, 2018*

# PLC System to Optimize Training Device of Upper Limb Spasticity

*Nurul Atiqah Othman, Noor Ayuni Che Zakaria,*
*Mohd Hanif Mohd Ramli*
*Faculty of Mechanical Engineering*
*Universiti Teknologi MARA Selangor*

*Fazah Akhtar Hanapiah*
*Faculty of Medicine*
*Universiti Teknologi MARA Selangor*

*Yee Jingye, Low Cheng Yee*
*Faculty of Mechanical Engineering and Manufacturing*
*Universiti Tun Hussein Onn Malaysia*

*Komeda Takashi*
*College of System Engineering and Science,*
*Shibaura Institute of Technology, Saitama, Japan*

## ABSTRACT

*This study provides an alternative to improvise the therapist education system by reaping the benefits of transferring industrial robotics precision into the medical and healthcare education. The proposed therapist education training simulator is driven and controlled by Programmable Logic Controller (PLC). It aims to innovate the systematic learning of physiotherapy skill sets in dealing with upper limb spasticity according to the module introduced in earlier research and publications. This research will introduce the combination of a clinical database of spasticity symptoms, the earlier work on this same simulator, and also human machine interface (HMI). This research is motivated by the purpose to avoid injury during therapy sessions due to the lack of skills and experiences of the trainee therapists. This research is to enhance the hands-on training of the trainee therapists with the proposed upper limb simulator before the real engagement with patients. This approach shows promising future improvement in therapist education strategies by combining Human Machine Interface Software Control and Data Acquisition (HMI SCADA) with PLC to further optimize the training device of upper limb spasticity simulator .*

**Keywords:** *Rehabilitation, Control Engineering, Robotic, Spasticity, PLC*

---

ISSN 1823- 5514, eISSN 2550-164X
Received for review: 2017-05-30
Accepted for publication: 2018-10-31
Published: 2018-12-15

## Introduction

In the healthcare industries, advancing the skills and upgrading trainings of staffs are necessary to increase the productivity of the trainees and trainers. Training sessions could be improved using lab facilities where trainees or students have their own workstation and more “hands-on” with minimal number of instructors. This paper focuses on introducing spasticity learning control architecture module using Programmable Logic Controller (PLC) to simulate the symptoms and embedded it into an upper limb part-task trainer, BITA1.0 introduced in [1]. The inadequate skills and experience of trainee therapists may injure the patients in therapy sessions [2]. Therefore, it is crucial to have maximum “hands-on” training with simulators or part-task trainers before their engagement with real patients [3] in a non-threatening environment [4][5]. The upper limb part-task trainer BITA1.0 uses the clinical database of upper limb spasticity collected under the ethics approval number NMRR-13-1384-18681 (IIR) published in [6].

Recently, Programmable Logic Controller (PLC) is widely applied in motion control, position control and torque control. PLC is frequently used in the big industry and commercial machineries [7]. PLC is designed for the application of multiple input and multiple outputs, signal processing, data conversion, and many other applications [8]. PLC can be defined as a main controller to automatically regulate the hardware and software based on the programmed routines. The term ‘Programmable’ is referring to a set of logical rules on setting up the connection between the inputs and the outputs. It can also be defined as an instruction set to complete the task given. The term ‘Logic’ implies that PLC interprets the outcome based on preset logical rules. ‘Controller’ is the key word of PLC. PLC holds the ability to control the outcome based on the program and logic status of the input. It will then process the information and bring out the information as being instructed to complete the task. Considering the precision and safety offered, few researchers have taken big step into implementing the technology into health sector devices [8][9].

Control architecture of BITA1.0 includes TwinCAT 3.1 – eXtended Automation Engineering (XAE) version 3.1.4018.47, Visual Studio Community 2015 and InduSoft Web Studio (IWS) version 8.0. All softwares will be explained in details in the next subchapter.

## Development Phase

For the development phase, suitable PLC has to be selected from a wide range of PLC offered in the market based on the requirement and computing power required of the system. Besides, related devices and software also constitute the whole hardware architecture as shown in Figure 1. Programming device can be a personal computer (PC) or laptop installed with programming software. In this research, the programming software used is TwinCAT 3.1 – eXtended Automation Engineering (XAE) version 3.1.4018.47 which is integrated with Visual Studio Community 2015. To develop a program for PLC, the programming device must be connected with the PLC through an Ethernet cable [8]. Once the programming structure is completed, the program is copied and installed into the controller and will work hand in hand with the hardware.

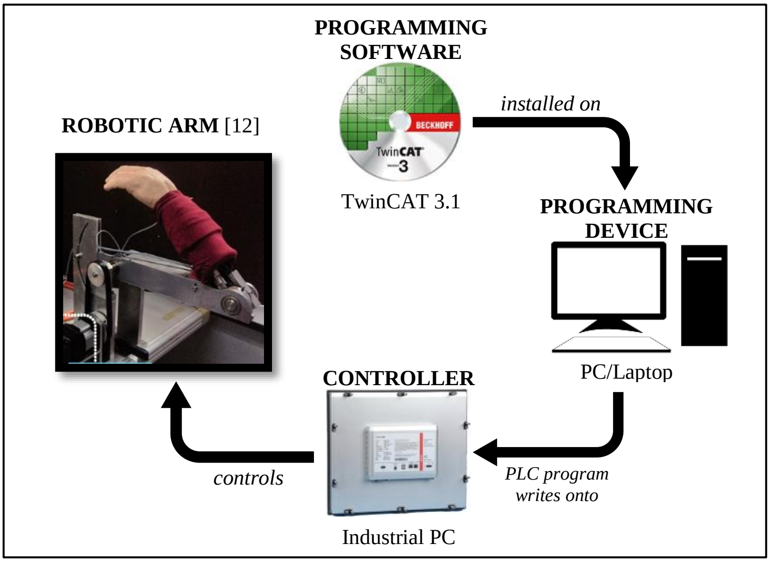


Figure 1: Hardware architecture of BITA1.0

## System Operation

An open-loop system is employed to control BITA1.0. An open-loop control system is a system that compute the input into control system directly [10][11]. PLC programming is used to determine and control the torque supplied to the robotic arm through the motor. The torque supplied is to emulate and simulate the behaviour of patient's arm with spasticity symptoms. PLC controller correlates the operational parameters to generate the output.

Figure 2 shows the block diagram representation of the system of BITA1.0. It shows the different level and step of the signal from the input and how it relates all the way to the output. The spasticity clinical profile of different Modified Ashworth Scale (MAS) levels are keyed into the PLC controller. There are two major parameters to be measured and considered by the PLC controller in order to determine the torque output at the robotic arm, which is the angle of the elbow, $\theta_e$ and the time taken for the change in elbow angle (or the rotational velocity of the elbow). In a slow passive extension, the force exhibited by BITA1.0 will be the same throughout the elbow rotation. However, if the rotational speed of the elbow exceed a preset threshold, the system will detect the rotation as a fast extension, and the torque produced will be increased to resemble the catch or jerk of a patient's spasticity symptoms.

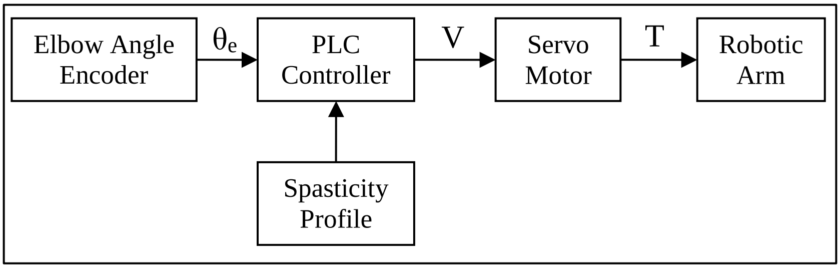


Figure 2: Block diagram representation of BITA1.0

Henceforth, the elbow angle encoder will transmit the data to the PLC controller. PLC controller will use the data and compute the rotational speed of the robotic arm. Subsequently, the PLC controller will map the input with the corresponding output by utilizing the spasticity profile from the clinical data, thus producing the corresponding voltage to the servo motor in order to produce the respective torque at the robotic arm. This system is considered as an open-loop system as there is no comparison between the actual and desired values with no self-regulating mechanism within the system. The PLC controller only maps the input to the corresponding output.

A more detailed structure of the PLC programming is being shown in a flowchart in Figure 3. It is shown that the process begins with switching on the power. To begin with, values for slow stretch and fast stretch and assigned to the respective range of angles. Bear in mind that the value appear in BITA1.0 is to indicate maximum and minimum torque value. 800 and 100 in BITA1.0 is maximum 8Nm and 1Nm of torque value respectively.

After the process, if the power switch is still on, the controller will detect whether the simulator arm is at the home position and whether the touch sensor is being engaged.

A sensor is attached at the elbow joint of BITA1.0 to detect whether the simulator arm is at the home position, while the touch sensor is to detect if user is engaging the simulator. If the simulator arm is not at the home position and the touch sensor is being engaged, the real time angle of the simulator arm is checked. If the angle is not within the designated angle limit, the resultant torque of the simulator arm will be zero.

If the simulator arm is within the designated range, the processor  check whether the assigned torque value is larger than 800. If true, then the resultant torque value given is 8N. This is to act as the maximum torque and a safety measure to prevent any human error in assigning an excessive torque which could cause harm to the users.

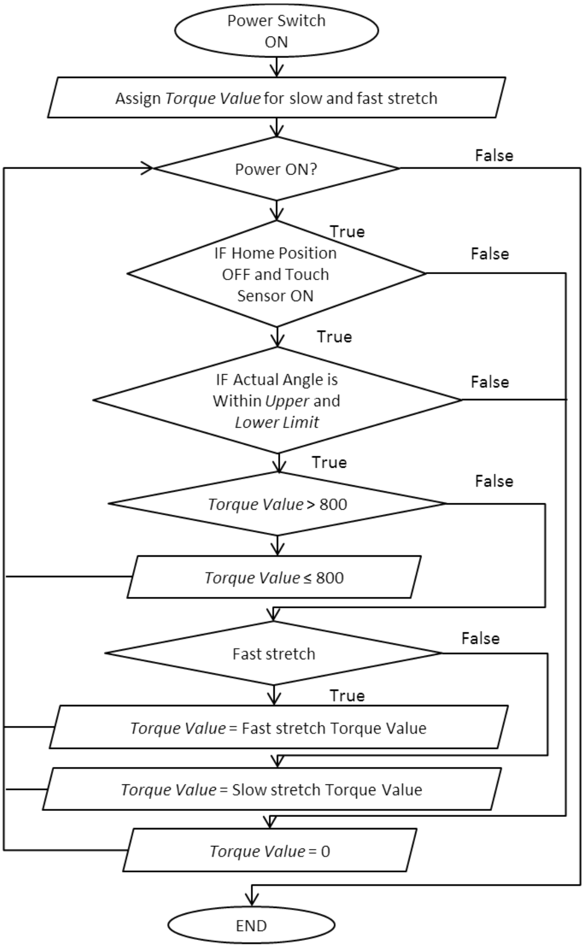


Figure 3: Flow chart of PLC programming of BITA1.0

If the assigned torque value is less than 800, the processor will determine whether the simulator arm is in a fast stretch mode. The fast stretch mode is determined by computing the instantaneous angular velocity of the BITA1.0. When the angular velocity exceeds the preset threshold of high angular velocity, the resultant torque value is taken from the data set of the fast stretch mode. Else, the resultant torque value is extracted from the data set of the slow stretch mode. The process will end following the power switch is turned off.

## BITA1.0 System Components

BITA1.0 is an integrated system of both hardwares and softwares. The overview of the setup is shown in Figure 4. All the components are arranged and kept in a PC-based box. The purpose of arranging everything in a PC-based box is to enhance the user experience, by having a nice and distinctive display system with no visible connecting wires around. It provides an uncluttered work space for the users.

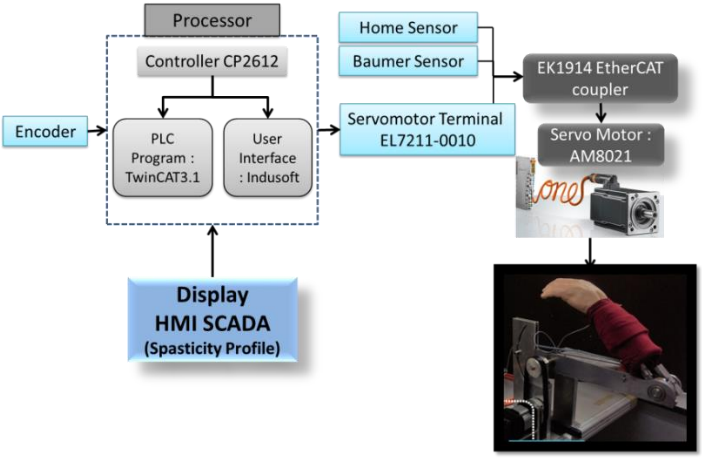


Figure 4: Components of the system.

Refering to Figure 5, BITA1.0 is using an industrial PC (IPC) instead of a normal PC in order to capitalize on its ease-of-use and its convenience. IPC is an industrial PC with multi-touch screen. It is easy to be installed and highly portable. The model of the IPC used is CP2616 from Beckhoff Automation. It is a compact multi-touch build in panel PC with 12” front panel display. The touch screen panel is built in PC with 1GHz ARM Cortex$^{TM}$ processor, on-board Ethernet adapter with RJ45 connection, and 128 kbyte NOVRAM for fail-safe storage of TwinCAT process data. The controller (CP 2612) acts as the brain of the system. PLC program of the system is implemented and installed into the controller (CP2612) by using TwinCAT3.1 software.

The display system employed by BITA1.0 is Human Machine Interface Software Control and Data Acquisition (HMI SCADA) with Indusoft as the software. The graphical user interface (GUI) that bridges the PLC program of the machine and user is shown in Figure 6.

Figure 5: PC-based controller CP2612 (Back view).

Figure 6: HMI SCADA display.

A Baumer sensor as shown in Figure 7 is installed in the system to provide feedback to the controller. A Baumer sensor is a small, slim, and sensitive capacitive sensor to detect the presence of any user engaging the wrist of the part task trainer. The feedback from Baumer sensor will directly send the signal to EK1914 EtherCAT coupler.

Figure 7: Baumer sensor to detect the engagement of user.

Figure 8 shows Beckhoff EtherCAT Coupler EK1914. The function of EK1914 coupler is to couple EtherCAT terminal EL7211 to 100BASE-TX EtherCAT networks by using CAT 5 Ethernet OR EtherCAT cable.

Digital input terminal (EL7211-0010) as shown in Figure 9 is a servomotor EtherCAT terminal integrated with One Cable Technology (OCT). EL7211-0010 connects the servomotor AM8021 to the coupler EK1914. The OCT combines a motor cable and absolute feedback system into a single cable.

The purpose is to minimize the connection wiring between the input and the processor.

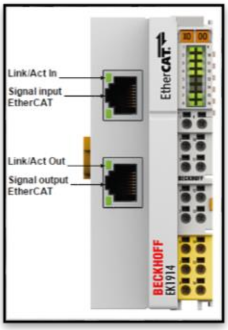


Figure 8: Beckhoff EtherCAT coupler EK1914.

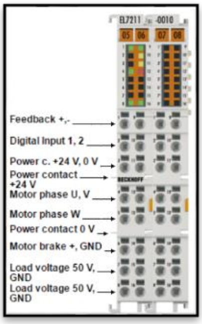


Figure 9: Digital input terminal EL7211-0010.

Figure 10 shows the Beckhoff Servomotor AM8021. Beckhoff Servomotor AM8021 is connected to terminal EL7211 to have the signals sent to EK1914. PLC program in the controller CP2612 will capture the real time position of servomotor AM8021 and the position will be translated to the corresponding torque in accordance to the MAS level. The position and applied torque are displayed through the HMI SCADA.

Figure 10: AM8021 Servomotor.

The combination of all the components and PLC program listed above forms the final prototype of BITA1.0. The full developed system of BITA1.0 is shown in Figure 11.

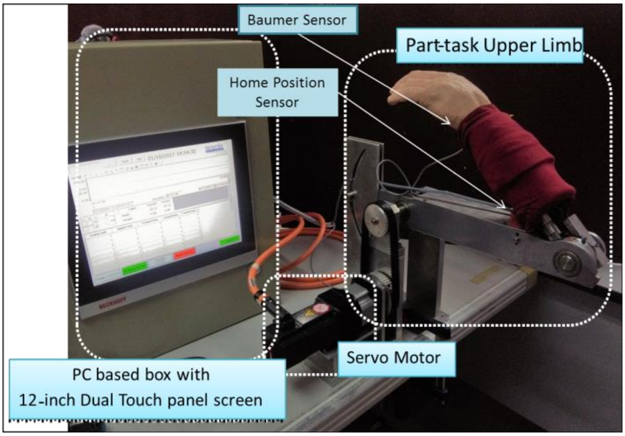


Figure 11: Prototype BITA1.0.

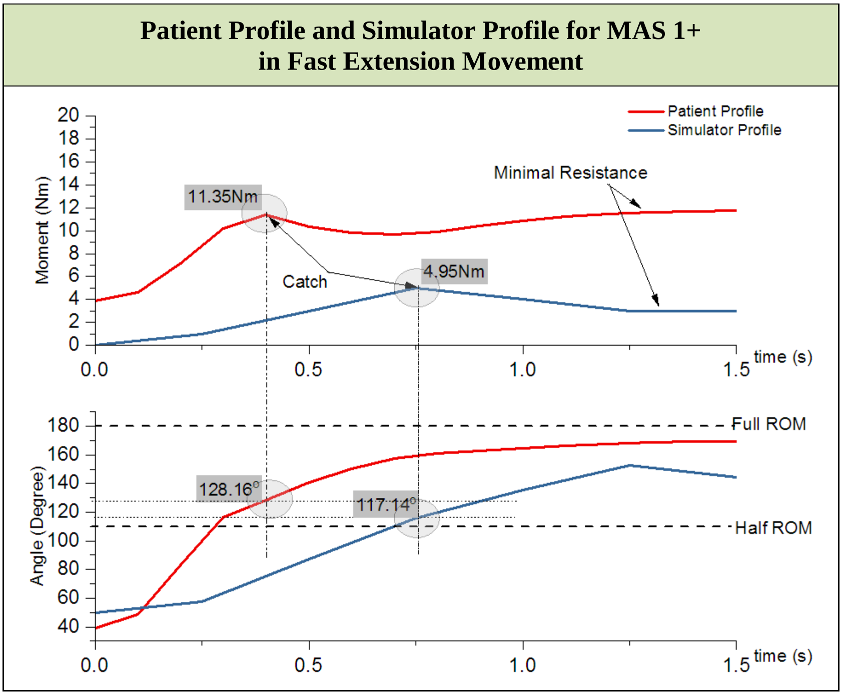
## Results and Discussions

After the implementation of the controller on BITA1.0, the study is followed by the evaluation phase. The evaluation phase serves to ensure that BITA1.0 is capable of simulating the human characteristics and reactions according to the clinical profile of spasticity. The evaluation process was conducted by having an expert rehabilitation physician to evaluate the performance of the simulator. The results are elaborated in the subsequent part below. The diagram hypothesis of each MAS level description is taken from the study in [1].

Figure 13 shows the comparison of the results between the real patient profile and BITA1.0. The outcome from BITA1.0 is compared to the real patient clinical data to evaluate the similarity of the movement pattern, range of motion, and catch position. In a fast extension, the catch torque and the angle of occurrence position are 4.95[Nm] at 117.14° and 11.35[Nm] at 128.16° for BITA1.0 and real patient respectively. The graph shows that both real patient profile and BITA1.0 satisfy the clinical Range of Motion (ROM).

As the catches occur at angular positions greater than 110° (more than half of the ROM) and close to the full end of ROM, the results can be considered as satisfied the hypothesis of condition of MAS 1+ in [1]. Nonetheless, the accuracy of both fast and slow extensions are not precise enough due to the low sampling rate.

Figure 12: Evaluation of BITA1.0 by an expert rehabilitation physician.



**Patient Profile and Simulator Profile for MAS 1+ in Fast Extension Movement**

Figure 13: Comparison between spasticity profile of real patient and BITA1.0 for MAS 1+ in fast extension.

A comparison is being made between this upgraded system of BITA1.0 and the previous system developed by the previous researcher in [6].

Table 1: Comparison of Upper Limb Spasticity Part-task Trainer (2014) and BITA1.0 (2017)

| **Aspect** | **Upper Limb Spasticity Part-task Trainer (2014)** | **BITA1.0 (2017)** |
|---|---|---|
| Hardware | • Magneto-Rheological brake<br>• DC servo motor<br>• DC Motor Driver 2x15A Lite | • AM8021 Servo Motor<br>• EK1914 coupler<br>• EL7211-0010 servomotor EtherCAT terminal |
| Software | • Arduino program ver. 1.0.1<br>• C language | • TwinCAT3.1 – eXtended Automation Engineering (XAE) version 3.1.4018.47<br>• C++ language |
| Controller | Arduino Mega ADK | CP 2612 (TwinCAT data processor) |
| Graphical User Interface (GUI) | Parallax Data Acquisition tool (PLX-DAQ) | • InduSoft Web Studio version 8.0<br>• HMI SCADA |

The previous system employed MR brake and direct current (DC) servomotor to produce resistance while Arduino Mega ADK is used as the controller. Figure 14 shows the profile outcomes of both systems.

By observing Figure 14, it is indicated that Arduino controller is unable to closely emulate the behavior of spasticity. No clear and distinctive catch is detected in the system. Based on the findings of evaluation process of Upper Limb Part-task Trainer (2014) [6], the DC Motor Driver 2×15A Lite is unstable in producing the resistance. Overshoot-voltage and current occurs when the motor power source is switched on and jerk is produced during the arm extension movements. Strain Gauge Module was used in the prototype to measure fast and slow stretch and it is found to be unsuitable as the measurement device due to its instability and robustness in measuring strain.

The system used in BITA1.0 is different with the previous prototype. By employing new system and components, the outcome shows clear improvements. Beckhoff Servomotor AM8021 is able to produce standstill torque, thus the resistance produced is more stable than DC Motor. BITA1.0 exhibits an improved result from the previous prototype as discussed.

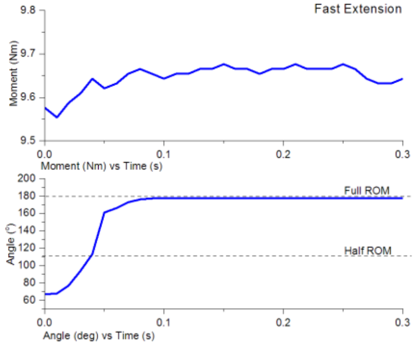
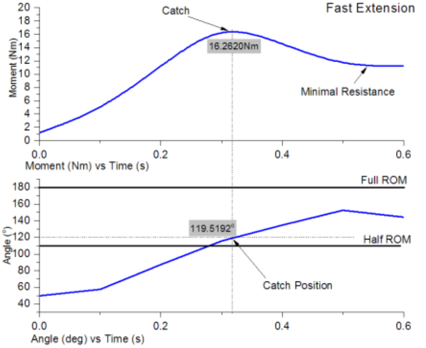
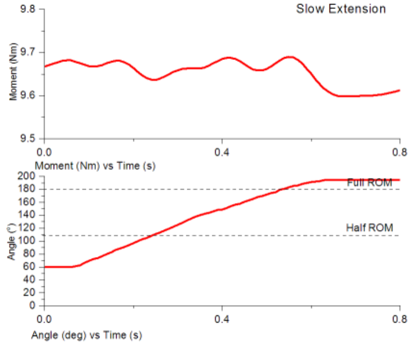
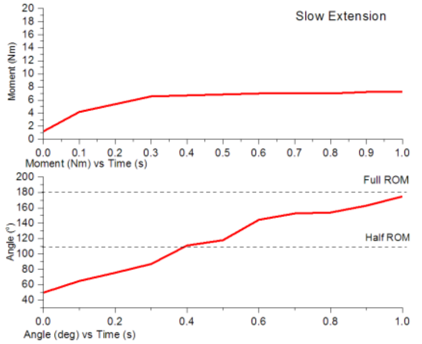
| **Upper Limb Spasticity Part-task Trainer (2014)** | **BITA1.0 (2017)** |
|---|---|
|  |  |
|  |  |

Figure 14: Result outcome by using Arduino (left) and Programmable Logic Controller (right).

## Qualitative Study: Evaluator Response

Besides being evaluated experimentally and quantitatively, the simulator has to be evaluated qualitatively to determine its effectiveness by assessing the response of evaluator. The questionnaire investigates a predefined set of procedures involved in analyzing spasticity symptoms and collecting evidence within the immediate boundaries of current developed simulator.

The single respondent involved in this qualitative survey has been part of this research development, thus semi-structured interviews with close-ended questions were designed in optimizing data collection. The data collection is an 'internal validation'. The semi-structured interview is structured and analyzed using 5-point Likert Scale.

Table 2: List of questions for semi-structured interview

| No. | Questions |
|---|---|
| **1** | The design appearance of Part-task Trainer of Upper Limb Spasticity Simulator is realistic. |
| **2** | Physical contact impression when handling the Part-task Trainer is similar to examination of a human upper limb. |
| **3** | Range Of Motion (ROM) of the Part-task Trainer is simulating human ROM movement. |
| **4** | 12-inch Dual Touch Display is convenient for clinical teaching and learning process. |
| **5** | Graphical User Interface - All information on the display system is important for clinical teaching and learning process. |
| **6** | Slow extension movement - The resistance feedback during slow extension movement is emulating patients' spasticity profile. |
| **7** | Fast extension movement - The resistance feedback during fast extension movement is emulating patients' spasticity profile. |
| **8** | As an overall impression, the Part-task Trainer of Upper Limb Spasticity Simulator is satisfactory and can be used as a teaching tool. |

Figure 15 shows the score of BITA1.0 as rated by the evaluator. The score rating ranges from minimum of 0 to maximum of 5. For Question (1) and Question (2) regarding the appearance of the simulator, score rating of 3 were given. The score of 3 can also be understood as a neutral stand of neither agree nor disagree with the statement. The evaluator thinks that aesthetic designers are needed in the development process to give a better impression and human-like appearance for the simulator. Comment given is that "the simulator need to get involvement of designers where the muscle mass of the forearm also needs to be looked into."

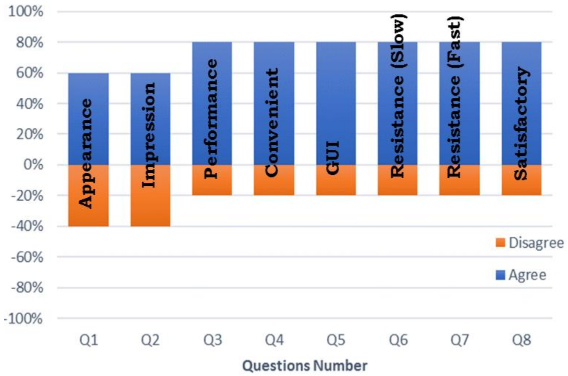


Figure 15: Performance rating of BITA1.0

As for Question (3), with respect to the simulation of human upper limb ROM, the evaluator responded with a score of 4. Score of 4 in a score range of 1 to 5 can be interpreted as an agreement to the statement. The comment given by the evaluator is that "it is important to have a 'drop' phenomenon where the arm should follow gravity rules in relaxed condition according to physician's order". Currently, the simulator is fixed with last position where the servomotor will have the limb retained at its last angle when no input is given.

Question (6) and Question (7) are to assess the functionability of the PLC program coding with regards to the human non-linear muscle movement. Respondent has to refer to the previous evaluation conducted before answering these questions. The main finding for these two questions was interpreted through the keyword used. Respondent used the term 'jerky movement' to express the spasticity symptom that does not fully match the human upper limb spasticity behaviour despite the fairly high score of 3 are given to both question. This means a linear input data mapping to the database is not enough to deliver human non-linear muscle movement and non-linear modelling has to be considered for the improvement of the system.

The touch screen and display system, Graphical User Interface (GUI), and overall impression are being mentioned in Question (4), (5), and (8). Evaluator gives out score of 4 out of 5 for all three questions. This shows the satisfaction of the evaluator upon BITA1.0 in these three areas. However, in order to be qualified as one of the effective teaching and training tools, the improvement and perfection of this system should be targeted.

## Conclusion

Programmable logic controller (PLC) is widely used in the automation industries. Hence, it can be considered as an innovation to transfer the technology into application in the area of bio-mechanics. In this study, PLC is being implemented into the system of BITA1.0. The results from the analysis shows that BITA1.0 is able to emulate the behaviour of spasticity of a patient during the flexion and extension movement by reproducing the resistance based on MAS level. Besides, the physical appearance of BITA1.0 closely mimics the upper limb of a human being.

Through this research, the PLC technology is proven to be able to replace the conventional controller in improving the overall stability of the system with higher performance as compared to the controller employed in the previous prototype. Despite the fact that there is still room for improvement and perfection of the BITA1.0 system, this innovation has been proven to enhance the overall performance of a part task trainer in emulating the spasticity behavior of a real patient, which could lead to a higher quality training routine for a novice physiotherapist or a physiotherapist-to-be.

## References

[1] F. A. H. and C. Y. L. N. A. Othman, F. Idris, N. A. C. Zakaria, “Supporting Conventional Qualitative Clinical Evaluation of Upper Limb Spasticity with Quantitative Data Measurement In Accordance to the Modified Ashworth Scale,” *IEEE EMBS*, no. Conference on Biomedical Engineering and Sciences (IECBES), pp. 731–736, 2016.

[2] K. Words, “Physiotherapy for the Arm and Hand after Stroke,” vol. 85, no. 8, pp. 417–425.

[3] B. R. Brewer, S. K. McDowell, and L. C. Worthen-Chaudhari, “Poststroke upper extremity rehabilitation: a review of robotic systems and clinical results.,” *Top. Stroke Rehabil.*, vol. 14, no. 6, pp. 22–44, 2007.

[4] N. A. C. Zakaria, T. Komeda, C. Y. Low, K. Inoue, and S. Raczak, “Emulation of Spasticity Symptoms in Upper Limb Part-Task Trainer for Physiotherapist Education,” *Appl. Mech. Mater.*, vol. 393, pp. 999–1004, 2013.

[5] N. Ayuni, C. H. E. Zakaria, and U. Mara, “High-fidelity Part-Task Trainer of Upper Limb Disorderfor Physiotherapist Education 2 Development of Part-task Trainer,” pp. 165–170.

[6] N. A. C. Zakaria *et al.*, “Evaluation of upper limb spasticity towards the development of a high fidelity part-task trainer,” *2Nd Int. Conf. Syst. Intell. Challenges Prod. Prod. Eng.*, vol. 15, pp. 817–826, 2014.

[7] H. Wright, “Introduction to programmable logic controllers ( PLCs ) and the operational function of main system modules,” pp. 1–6, 2010.

[8] A. R. Kiran, B. V. Sundeep, S. C. Vardhan, and N. Mathews, “The principle of programmable logic controller and its role in automation,” *Int. J. Eng. Trends Technol.*, vol. 4, no. 3, pp. 500–502, 2013.

[9] F. Idris, N. Ayuni, C. Zakaria, C. Yee, F. Akhtar, and N. Atiqah, “System Integration of an Upper Limb Disorder Part-Task Trainer with PC- based Control,” *Procedia - Procedia Comput. Sci.*, vol. 105, no. December 2016, pp. 328–332, 2017.

[10] J. C. Willems, “Feedforward Control, Pid Control Laws, and Almost Invariant Subspaces,” *Syst. Control Lett.*, vol. 1, no. 4, pp. 277–282, 1982.

[11] Y. Birbir and H. S. Nogay, “Design and implementation of PLC-based monitoring control system for three-phase induction motors fed by PWM inverter,” *Int. J. Syst. Appl. ...*, vol. 2, no. 3, pp. 128–135, 2008.

[12] C. Z. Noor Ayuni, T. Komeda, and C. Y. Low, “Design of upper limb patient simulator,” *Procedia Eng.*, vol. 41, no. Iris, pp. 1374–1378, 2012.