External force/velocity control for an autonomous rehabilitation robot

Peerayuth Saekow *, Paramin Neranon and Pruittikorn Smithmaitrie

Prince of Songkla University, Faculty of Engineering, Hat Yai, Songkla 90110, Thailand

* Corresponding Author: E-mail: Peerayuth.psu@gmail.com
Telephone Number: 083-6551504

Abstract. Stroke is a primary cause of death and the leading cause of permanent disability in adults. There are many stroke survivors, who live with a variety of levels of disability and always need rehabilitation activities on daily basis. Several studies have reported that usage of rehabilitation robotic devices shows the better improvement outcomes in upper-limb stroke patients than the conventional therapy–nurses or therapists actively help patients with exercise-based rehabilitation. This research focuses on the development of an autonomous robotic trainer designed to guide a stroke patient through an upper-limb rehabilitation task. The robotic device was designed and developed to automate the reaching exercise as mentioned. The designed robotic system is made up of a four-wheel omni-directional mobile robot, an ATI Gamma multi-axis force/torque sensor used to measure contact force and a microcontroller real-time operating system. Proportional plus Integral control was adapted to control the overall performance and stability of the autonomous assistive robot. External force control was successfully implemented to establish the behavioral control strategy for the robot force and velocity control scheme. In summary, the experimental results indicated satisfactorily stable performance of the robot force and velocity control can be considered acceptable. The gain tuning for proportional integral (PI) velocity control algorithms was suitably estimated using the Ziegler-Nichols method in which the optimized proportional and integral gains are 0.45 and 0.11, respectively. Additionally, the PI external force control gains were experimentally tuned using the trial and error method based on a set of experiments which allow a human participant moves the robot along the constrained circular path whilst attempting to minimize the radial force. The performance was analyzed based on the root mean square error ($E_{RMS}$) of the radial forces, in which the lower the variation in radial forces, the better the performance of the system. The outstanding performance of the tests as specified by the $E_{RMS}$ of the radial force was observed with proportional and integral gains of $K_p = 0.7$ and $K_i = 0.75$, respectively.

1. Introduction

Stroke is a state describing the lack of blood in the brain which occurs form having ischemic stroke, arterial occlusive diseases, or Intracranial haemorrhage. This causes tissue of the brain to die, and its function to instantly stop. The body subsequently will be paralyzed, weak, and immobilized which results in the disability. Stroke and its effects are considered the major problem and cause of people’s death throughout the world. According to the World Safety Organization (WSO)’s record, it is postulated that at least one person dies in every 6 seconds, and it is the third major reason of death and disability of the world [14]. Therefore, many research papers have studied a new approach to the...
restoration of the original condition of patients with stroke based on the repeated movement to stimulate the muscles with an assistive robot. The robot is able to perform the task repeatedly and efficiently with consistent control. According to studies by Hermano and Krebs [1, 2], they have developed upper-limb robotic rehabilitation devices, namely MIT-MANUS and InMotion2, respectively. Each robot has two degrees of freedom (DOF). The user’s paretic forearm is placed in a support provided, which is attached to a robot end-effector. The user can move the handle in the transverse plane and perform goal-directed tasks which focus on human’s shoulder and elbow movement. In addition, a monitor was served to display user targets and provide visual feedback.

An upper-limb robotic rehabilitation device which is the Mirror-Image Motion Enabler (MIME) was developed by Peter and Van [3, 11]. The MIME robot is made up of a six-DOF paretic arm robot with the end effector, which has actuators to apply forces in goal-directed movements. The arm is attached to the end effector, which restricts wrist and hand movement. Loureiro and Harwin [4] constructed an upper-limb robotic rehabilitation device called the GENTLE/s with a virtual reality (VR) display provided to motivate users to engage in the rehabilitation therapy. The GENTLE/s includes a chair, shoulder supports, a wrist connector, an elbow orthosis, two computers, a large monitor with speakers, an exercise table, and a keypad. The arm is suspended to overcome the negative effects of gravity on the paretic arm and to help with shoulder subluxation; in the meantime, the VR can simulate three different environments: an empty room, a real room with some general shapes, and a detailed room with several objects. The Guide (ARM) robot is one of the upper-limb robotic rehabilitation devices and it was designed by Reinkensmeyer, Dewald and Rymer [5, 6]. The Guide (ARM) was utilized to assess multi-joint coordination and the workspace deficits of a paretic arm in a quantitative manner. The ARM Guide was attached to the user’s forearm with a custom splint and guides the arm on a linear path, and range of motion and constraint force components were simultaneously measured. Furthermore, there are some research papers currently focusing on high-DOF robot in rehabilitation activities. For example, the Therapy Wilmington Robotic Exoskeleton (T-WREX) robot which has five degrees of freedom and it was proposed by Housman and Reinknsmeyer [12]. The robot was initially designed to support a weak human arm in intensive movement training by reducing the effects of gravity. The T-WREX uses rubber bands to provide various levels of arm support and has position sensors at each joint, which can measure arm movement. It is able to be used in conjunction with a computer to allow the user to interact in a VR environment and play games. A six-DOF arm robot, named REHAROB, was used in upper-limb rehabilitation. This robot can perform passive physiotherapy at a constant velocity, and it was trained by therapists [7]. However, there is currently no research having studied on and developed an omni-directional mobile robot used in upper-limb stroke rehabilitation which actively help patients through exercise-based activities. Consequently, this paper highlights on the development of a behavioral control strategy using force/velocity control based on PI algorithm for the autonomous robotic trainer designed to guide a stroke patient through an upper-limb rehabilitation task. By first understanding the principle dynamics and control behavior, two sets of experiments have been conducted. PI velocity control algorithm was firstly designed and optimized to ensure the effective robot movements; secondly, the PI external force control was also experimentally tuned to provide successful and safe stroke rehabilitation tasks. The challenge is further complicated by the dynamic nature of the human-robot environment, which by its nature necessitates very careful design of the control strategy and its implementation in order to protect the human operator from the risk of harm or injury by the robot.
2. The designed robotic
This section introduces the conceptual design of the assistive robotic system as schematically illustrated in Figure 1. The system involves a four-wheel omni-directional mobile robot which coordinates with a set of physical sensors and their feedback signals have been processed by an Arduino controller in real-time. An ATI Gamma multi-axis force/torque sensor was used to measure the interactive force between the human and robot, whereas optical linear encoders were mounted to the motor actuators. The force sensor system consists of an ATI F/T Gamma sensor, an electrically shielded and twisted transducer cable and a stand-alone ATI controller in which optional analogue, parallel and serial outputs have been already provided. The ranges of force/torque measurements are up to ±130 N with 0.1 N resolution and ±10 Nm with 0.0025 Nm resolution, respectively. A faulhaber 12V-DC motor with an encoder which is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy was utilized. The motors with 64:1 planetary gearbox and 120RPM with 12CPR Encoder were used for an autonomous robotic platform. A detailed description of Arduino Mega 2560 microcontroller is based on the ATmega2560, which has 54 digital input/output pins (14 of which can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16-MHz crystal oscillator. According to the mobile robotic platform design, anodized aluminium was used for the body structures to ensure a durable and robust, yet lightweight construction, whilst plastic linkages and circuit board carriers ensure lightness yet durability. The omni-wheels adopted are wheels with small discs around the circumference which are perpendicular to the turning direction. The effect is that the wheel can be driven with full force, but it can also slide laterally with great ease.

![Figure 1. Block diagram and four-wheel omni-directional mobile robot.](image)

3. Robot dynamic model
The robotic dynamic model is essential for describing a mobile robot’s position, orientation and motions of the wheels and for analyzing and synthesizing the overall dynamic behavior of the robot. The omni-wheels robot has three degrees of freedom on the x-y plane and can perform forward-reverse movement and left-right movements. The movement in all three directions is due to the sum of and the four wheels. The kinematic equations and relations between robot velocity and various geometry specifications are expressed as the following. Oliveira and Hashemi [8, 13] have suggested that there are two coordinate frames used in the model consisting of the global frame (x, y) which represents the environment of the robot as depicted in Figure 2. The robot’s location and orientation in this global frame can be represented as (x, y, θ). The global velocity of the robot can be written as (\( \dot{x}, \dot{y}, \dot{\theta} \)) and define a local frame \([x_r, y_r]\) that is attached to the robot itself.
Transformation matrices from linear velocity on the global frame to linear on the robot frame can be expressed as the following equation:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
\cos\theta & \sin\theta & 0 \\
-\sin\theta & \cos\theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{x}_r \\
\dot{y}_r \\
\dot{\theta}_r
\end{bmatrix}
\]  

(1)

The relationship between the translational velocity of wheel \( v_i \) and the global velocity of the robot in the environment \( (\dot{x}, \dot{y}, \dot{\theta}) \) is presented. The translation of velocity of wheel can be divided into two parts which are translation and rotation.

\[
V_i = V_{\text{trans},i} + V_{\text{rot}}
\]  

(2)

Translation as shown in Figure 3 can map the vector \( V_{\text{trans},0} \) onto the vectors \( \dot{x} \) and \( \dot{y} \) as follows:

\[
V_{\text{trans},0} = -\sin(\theta) \dot{x} + \cos(\theta) \dot{y}.
\]  

(3)

The vector for all wheels takes into consideration that the vector \( v_i \) are position and offset by \( \theta + \alpha_i \) as follows:

\[
V_{\text{trans},i} = -\sin(\theta + \alpha_i) \dot{x} + \cos(\theta + \alpha_i) \dot{y},
\]  

(4)

where \( \alpha_i = \) starting point and count degrees in the clockwise direction as positive, \( \alpha_0 = 0^\circ, \alpha_1 = 90^\circ, \alpha_2 = 180^\circ \) and \( \alpha_3 = 270^\circ \).

Once the mobile robot has a pure rotation, it gives:
\[ V_{rot} = R \dot{\theta}, \]  

where \( R \) is distance between wheel and center robot (\( R = 0.155 \) m).

Substituting the equations (4) and (5) into the equation (2) gives:

\[ V_i = - \sin(\theta + \alpha_i) \dot{x} + \cos(\theta + \alpha_i) \dot{y} + R \dot{\theta}. \]  

The translation velocity of the hub is related to the angular velocity \( \dot{\phi}_i \) of the wheel as:

\[ v_i = r \dot{\phi}_i, \]  

where \( r \) is wheel radius (\( r = 0.029 \) m) and \( i = 0,1,2,3 \) represents indices of the wheels.

Substituting the equation (7) into the equation (6) gives:

\[ \dot{\phi}_i = \frac{1}{r} \left( - \sin(\theta + \alpha_i) \dot{x} + \cos(\theta + \alpha_i) \dot{y} + R \dot{\theta} \right). \]  

Then equation (8) can be transformed to matrix representation (9) as follows:

\[
\begin{bmatrix}
\dot{\phi}_1 \\
\dot{\phi}_2 \\
\dot{\phi}_3 \\
\dot{\phi}_4
\end{bmatrix} = \frac{1}{r} \begin{bmatrix}
-\sin(\theta) & \cos(\theta) & R & 0 \\
-\sin(\theta + \alpha_1) & \cos(\theta + \alpha_1) & R & \dot{x}
\end{bmatrix} \begin{bmatrix}
\cos(\theta) & \sin(\theta) & 0 & 1 \\
-\sin(\theta) & \cos(\theta) & 0 & 1
\end{bmatrix} \begin{bmatrix}
\dot{x}_r \\
\dot{y}_r \\
\dot{\theta}
\end{bmatrix}.
\]  

Simply converting the global to robot coordinates and substituting equation (1) into the equation (9) lead to:

\[
\begin{bmatrix}
\dot{\phi}_1 \\
\dot{\phi}_2 \\
\dot{\phi}_3 \\
\dot{\phi}_4
\end{bmatrix} = \frac{1}{r} \begin{bmatrix}
-\sin(\theta) & \cos(\theta) & R & 0 \\
-\sin(\theta + \alpha_1) & \cos(\theta + \alpha_1) & R & \dot{x}
\end{bmatrix} \begin{bmatrix}
\cos(\theta) & \sin(\theta) & 0 & 1 \\
-\sin(\theta) & \cos(\theta) & 0 & 1
\end{bmatrix} \begin{bmatrix}
\dot{x}_r \\
\dot{y}_r \\
\dot{\theta}
\end{bmatrix}.
\]  

Therefore, the local frame matrix relation can be expressed the following equation:

\[
\begin{bmatrix}
\dot{\phi}_1 \\
\dot{\phi}_2 \\
\dot{\phi}_3 \\
\dot{\phi}_4
\end{bmatrix} = \begin{bmatrix}
0.00 & 34.48 & 5.35 \\
-34.48 & 0.00 & 5.35 \\
0.00 & -34.48 & 5.35 \\
34.48 & 0.00 & 5.35
\end{bmatrix} \begin{bmatrix}
\dot{x}_r \\
\dot{y}_r \\
\dot{\theta}
\end{bmatrix}.
\]  

where \( \dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3, \dot{\phi}_4 \) are angular velocities of the wheels and \( \dot{x}_r, \dot{y}_r, \dot{\theta} \) are velocities of the robot in local frame.

4. Fundamental of Force/Velocity Control

This section explains robot force/velocity control algorithms, which are a fundamental requirement in the achievement of the robot’s real-time control. External force control involves any contact force between the mobile robot and its environment. In the meantime the robot velocity control algorithm has perform satisfactorily stable performance for every robot movement. The key performance issue in design of a robotic force/velocity control system is that of stability and reliability that means the robot has to perform in a stable and reliable manner while operating and contacting with various unknown stiffness environments and moving along the constrained path. Using external force/velocity control (see Figure 4.) developed by De Schutter and Van Brussel [9] is considered to offer a better solution regarding the safety constraints, simplicity and implementation efficiency. The most crucial aspect in
this control method is to achieve a suitable compromise between the system response and stability, where the response time was required to be as short as possible.

Proportional plus integral (PI) control is most widely utilized in industrial process control applications because it is relatively easy to implement and no other type of control can match its simplicity and clear functionality, such as motor speed or position control as suggested by De Schutter and Van Brussel [9]. The PI (proportional plus integral) force control algorithms were successfully implemented with discrete sampling periods. The discrete form of the proportional integral equation is needed to approximate the integral of the error and because this technique facilitates an increase in the accuracy and stability of control system. Therefore, it was decided to apply simple proportional plus integral robot force control in this project.

An incremental discrete-time PI control algorithm [10] with sampling time period \( \tau \) and the discrete time interval \( k \) can be calculated by applying the equations (12) and (13) and it gives:

\[
U(k) = k_p e(k) + k_i \sum_{j=0}^{k} e(j).
\]  

(12)

The incremental PI control value represented by \( \Delta u(k) \) can be calculated as:

\[
\Delta u(k) = k_p e(k) + k_i \sum_{j=0}^{k} e(j) - k_p e(k-1) + k_i \sum_{j=0}^{k-1} e(j)
\]  

(13)

\[
\Delta u(k) = k_p [e(k) - e(k-1)] + k_i e(k)
\]  

(14)

where \( u(k) \) is the PI control output; \( k_p \) is proportional gain; \( k_i \) is integral gain; \( f_d \) is desired force, which is initially defined as 0, and \( f_s \) is the actual force (measured by the ATI force control).

A block diagram of external force/velocity control based on the PI control is subsequently shown in Figure 5 where \( e \) is the error defined as the difference in magnitude between the desired \( (f_d) \) and actual \( (f_s) \) forces, while \( d_e \) is the change in error \( (e) \). The PI control output can be determined as the incremental displacement \( (\Delta U_k) \) modified by the previous computed value of \( (\Delta U_{k-1}) \), which is scaled before being transferred to the robot.
In the same way, the output of the PI velocity control implemented was assumed to be the incremental position $\Delta u(k)$, where increment PI velocity control algorithm can be calculated by applying the equation as shown below.

The incremental robot PI velocity control is defined by

$$u(k) = u(k-1) + kp[e(k) - e(k-1)] + ki\Delta e(k)$$  \hspace{1cm} (16)

The input $e$ of PI control was generated as the difference in magnitude between the desired ($V_d$) and actual ($V_s$) velocities, and the input $de$ is the change in error ($e$) where $V_d$ is desired velocity, and $V_s$ is the actual velocity (measured by the encoder sensor), respectively.

5. **External Implementation**

In order to improve the stability and performance of the autonomous robotic trainer designed to guide a stroke patient through an upper-limb rehabilitation task, the proportional integral (PI) force/velocity control technique has been clearly addressed in the last section. However, one of the most principal challenges in the control development is to design a set of appropriate PI gains to achieve an effective system response. Various techniques for PI gain tuning have been developed, such as the Ziegler-Nichols, Cohen-Coon, Chien-Hrones-Reswick or manual (trial and error) techniques. In this paper, Ziegler-Nichols tuning method was adopted to establish appropriate PI gains for the robot’s velocity control algorithm. However, as complicated robot force control while the robot interacts with the human environment, a trial-and-error tuning method based on a circular tracking experiment was used in the robot’s external force control.

5.1. **Experiment 1: Gain tuning of the PI velocity control**

Ziegler-Nichols method was applied to simplify the determination of PI velocity control’s gain parameters to meet the desired performance specifications, in which the system overshoot response is kept to minimum but at the same time the response should be as fast as possible. However, in this paper this tuning technique still needs operator’s experience for a fine tuning until the system specification being completely achieved. As the systematic approach based on the method of Ziegler and Nichols rules was applied to simplify PI gains, its performance was analyzed based on the root mean square error ($E_{\text{RMS}}$) of the velocity. It can be said that the performance of the system response can be evaluated in terms of variation in the radial forces, in which the lower the variation in radial forces, the better the performance of the system. The equation used to calculate the magnitude of error deviations of $E_{\text{RMS}}$ is expressed as:
\[ E_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(X_d - X_s)^2}{n}}, \] 

where \( n \) is the number of evaluated values, \( X_d \) is desired value, and \( X_s \) is the actual value (measured by the encoder sensor).

Properly tuning the gain parameters for the PI velocity control algorithm can be a challenge because, if the time constants in the process are large, the time to do the optimization can be excessively long. This following process presents how the appropriate gain tuning was achieved. Firstly, the autonomous rehabilitation robot was pulled by constant forces using a mass of 5N (simulating a human pulling force). The integral term \( (K_i) \) was turned off and only the proportional gain \( (K_p) \) was initially set to zero. The \( K_p \) component subsequently increased slowly while the robot velocity was simultaneously measured and monitored until it can be noted that the velocity profile exhibited sustained oscillations. At this point, the experiment finalizes the critical gain, and the \( K_p, T_u \) unknown parameters were calculated. Thus, using Ziegler and Nichols technique gave the proportional gain \( (K_p) \) of 0.45 and integral gain \( (K_i) \) of 0.10. Nevertheless, as mentioned earlier, to adopt the set of proper PI gains this method still needs operator’s experience for a fine tuning. A set of the experiments were undertaken to examine the relationship between the adjusted proportional gain and root mean square error \( (E_{RMS}) \) of the velocity where \( V_d \) is desired velocity, and \( V_s \) is the actual velocity. The test results are illustrated in Table 1.

5.1.1. Experimental results: \( K_p \) gain fine tuning (robot velocity control algorithm)

| \( K_p \) Gain | \( E_{RMS} \) of velocity [RPM] |
|----------------|-------------------------------|
| 0.25           | 14.07                         |
| 0.35           | 8.17                          |
| 0.45           | 4.86                          |
| 0.55           | 5.44                          |
| 0.65           | 5.93                          |
| 0.75           | 8.13                          |
| 0.85           | 9.61                          |

After using Ziegler-Nichols technique to roughly estimate the proportional and integral gains, the operator’s experience was needed for a fine tuning to delivery an optimized control system. A set of the experiments was proposed and undertaken to examine the relationship between the adjusted proportional gain and root mean square error \( (E_{RMS}) \) of the velocity where \( V_d \) is desired velocity, and \( V_s \) is the actual velocity. The gain tuning for the proportional velocity control was successfully implemented. Based on the results of the experiments, it can be concluded that the best performance as specified by the lowest \( E_{RMS} \) of the velocity is observed with the proportional \( K_p = 0.45 \), in which this value was subsequently employed in the robot velocity control in order to finalize the appropriate integral gain \( K_i \).
5.1.2. Experimental results: $K_i$ gain fine tuning (robot velocity control algorithm)

Table 2. The results of preliminary tests to evaluate the gain $K_i$

| $K_i$ Gain | $E_{RMS}$ of velocity [RPM] |
|------------|-----------------------------|
| 0.09       | 6.49                        |
| 0.10       | 6.20                        |
| 0.11       | 4.86                        |
| 0.12       | 5.38                        |
| 0.13       | 5.63                        |

In the same way, Table 2 shows the results of preliminary tests to evaluate the gain $K_i$ presented in the performance results of the tests in which the range of the integral gain ($K_i$) were between 0.09 and 0.13. The system performance can be also identified based on the $E_{RMS}$ of velocity as in the previous case. The best performance of this test is represented by minimum $E_{RMS}$ of velocity, and was achieved at a gain $K_i$ of 0.11.

5.2. Experiment 2: Gain tuning of the PI force control

As suggested by De Schutter and Van Brussel [9], proportional integral (PI) control is appropriate for robot force/position control in order to provide the smallest possible force control error. The journal of Nerano and Bicker [10] also stated that according to the Ziegler-Nichols method, $K_u, T_u$ and the critical gain can be calculated once the $K_p$ component is slowly increased until the robot motion profile exhibited sustained oscillations; however, at this point if the robot system has very high unstable oscillation, this could damage the robot. Therefore, the proper gains of PI external force control were experimentally tuned using the trial-and-error method based on a set of circular tracking experiments which allow a human participant to move the robot along the constrained circular path whilst attempting to minimize the radial force. The designed circle was drawn on a green template mounted on the fixed platform where the circular tracking test in an X-Y plane required the operator to track a circle radius selected of approximately 25 cm. The robot was positioned such that its pointer was at the home position as shown in Figure 6. The performance was analyzed based on the root mean square error ($E_{RMS}$) of the radial forces, in which the lower the variation in radial forces, the better the performance of the PI control system.

Ten participants were adopted and required to perform a set of tests in order to become familiar the test rig before completing five repetitions of the substantive experiments. Prior to running the real-time circular tracking test, the human was requested to sit down in comfortable position in front of the test rig and naturally grasp the supporter provided using only one hand. When the timer trigger is activated, the human participant is asked to start moving the robot from the home position towards the constrained circular path. In the meantime, the robot position and velocity components are simultaneously detected and monitored in real time. The human operator is then required to continually track the constrained circular path in a clockwise direction until the test is totally completed when a full revolution has been achieved and the timer is stopped.
Figure 6. experimental set-up.

5.2.1. Experimental results: $K_p$ gain tuning (robot force control algorithm)

As a trial-and-error tuning based on a circular tracking experiments was adopted. Firstly, the integral term ($K_i$) was initially turned off and only the proportional gain ($K_p$) ranging between 0.2 – 0.8 with 0.1 resolution was set. Choosing the optimum empirical control gains is a cautious exercise if a high proportional gain is preferable as its implementation will increase the transparency of the operation; nevertheless, the gain can only be increased to a certain upper limit; if the gain is set too large the system becomes unstable which could easily damage the mobile robot. The following table and figures demonstrate a representative of results of the relationship between the proportional gain and the root mean square error ($E_{\text{RMS}}$) of the radial force applied to the robot.

| $K_p$ Gain | $E_{\text{RMS}}$ of force [N] |
|------------|--------------------------------|
| 0.2        | 4.33                           |
| 0.3        | 1.71                           |
| 0.4        | 1.92                           |
| 0.5        | 1.45                           |
| 0.6        | 1.39                           |
| 0.7        | 0.96                           |
| 0.8        | 0.98                           |
| 0.9        | 1.29                           |
Figure 7. Circular tracking tests to evaluate the gain $K_p$.

Figure 7 presents the examples of the performance results with the different values of proportional gain $K_p$ ranging between 0.2 and 0.9 in which the system performance can be identified based on the $E_{RMS}$ of force. The highest performance of this test is represented by the minimum $E_{RMS}$ of the radial force and was achieved at a gain $K_p$ of 0.7, where the minimum $E_{RMS}$ value is 0.96 N. Additionally, as expected the tangential force applied to the robot decreases when the gain $K_p$ increases, and is approximately inversely proportional to the $K_p$ gain value. Hence, the optimized proportional gain proposed was then utilized in $K_i$ gain tuning of the robot force control algorithm expressed in the following section.

5.2.2. Experimental results: $K_i$ gain tuning (robot force control algorithm)

To optimize the integral gain ($K_i$), the gain $K_p$ was initially set at 0.7, and then tuning of the integral gain is marked by increasing $K_i$ until the best performance indicating by the minimum $E_{RMS}$ of the radial force is achieved. The same procedure for the first circular tracking test developed for evaluating
the performance of the gain $K_p$ was used. The same group of the participants was employed to perform the assigned tests in which a range of integral gains varying from 0.25 to 1.25 with 0.25 resolution.

**Table 4.** The results of preliminary tests to evaluate the gain $K_i$

| $K_i$ Gain | $E_{RMS}$ of force [N] |
|------------|-----------------------|
| 0.25       | 2.44                  |
| 0.50       | 1.26                  |
| 0.75       | 0.91                  |
| 1.00       | 1.03                  |
| 1.25       | 1.27                  |

Table 4 presents the performance results of test different values of integral gain $K_i$ ranging between 0.25 and 1.25. The system performance can be identified based on the $E_{RMS}$ of force. The best performance of this test is represented by minimum $E_{RMS}$ of force, and was achieved at a gain $K_i$ of 0.75.

The performance of the circular tracking test for different values of integral gain ($K_i$) is illustrated above. It can be noted that the best performance of the $K_i$ tuning test is 0.75, in which the lowest $E_{RMS}$ value is 0.9. To sum up, based on a set of experiments which allow a human participant to move the robot along the constrained circular path whilst attempting to minimize the radial force, the performance was analyzed based on the root mean square error ($E_{RMS}$) of the radial forces, in which the lower the variation in radial forces, the better the performance of the system. The outstanding performance results to optimized proportional ($K_p$) and integral ($K_i$) gains are observed with 0.7 and 0.75, respectively.

6. **Conclusion**

This paper has presented a new development of the force/velocity control for an autonomous rehabilitation robot system. This new robotic system has its emphasis on real-time control, programmability and repeatability in which the robot provides as advantages for guiding a stroke patient through an upper-limb rehabilitation. A well tuned proportional plus integral (PI) control was successful developed and it allows the robot to meet the desired response and satisfactorily stable performance. Based on the experimental results, it was confirmed that the best performance of the robotic force/velocity control as specified by the $E_{RMS}$ of radial force has been observed with proportional and integral gains of the velocity control equal to 0.45 and 0.11, respectively. And proportional and integral gains of force control are 0.7 and 0.75, respectively. Therefore the development of an autonomous robotic trainer designed to guide a stroke patient through an upper-limb rehabilitation task has been carried out successfully.
Reference

[1] Krebs H I, et al., J. Neuroengineering Rehabil. 1(1) p 5
[2] Krebs H I, Hogan N, Aisen M L and Volpe B T 1998 IEEE Trans. Rehabil. Eng 6(1) pp 75–87
[3] Lum P S, Burgar C G, Shor P C, Majmundar M and Van der Loos M 2002 Arch. Phys. Med. Rehabil. 83 (7) pp 952–959
[4] Loureiro R, Amirabdollahian F, Topping M, Driessen B and Harwin W 2003 Auton. Robots 15(1) pp 35–51
[5] Reinkensmeyer D J, Kahn L E, Averbuch M, McKenna-Cole A, Schmit B D and Rymer W Z 2014 J. Rehabil. Res. Dev. 37(6)
[6] Reinkensmeyer D J, Dewald J P A and Rymer W Z 1990 IEEE Trans. Rehabil. Eng. 7(1) pp 1–11
[7] Fazekas G, Horvath M and Toth A 2006 Int. J. Rehabil. Res. 29(3) pp 251–254
[8] Oliveira H P, Sousa A J, Moreira A P and Costa P J 2009 Modeling and Assessing of Omni-Directional Robots with Three and Four Wheels
[9] De Schutter J and Van Brussel H 1988 Int. J. Robot. Res. 7(4) pp 18–33
[10] Neranon P and Bicker R 2016 Therm. Sci. 20(2) pp 537–548
[11] Lum P S, Burgar C G, der Loos M V, Shor P C, Majmundar M and Yap R 2005 9th Int. Conf. on Rehabilitation Robotics pp 511–514
[12] Housman S J, Le V, Rahman T, Sanchez R J and Reinkensmeyer D J 2007 2007 IEEE 10th Int. Conf. on Rehabilitation Robotics pp 562–568
[13] Hashemi E, Jadidi M G and Babarsad O B 2009 Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE Int. Symp. pp 272–277
[14] http://world-stroke.org/ [4 January 2016].