Journal of Mechanical Engineering Research and Developments
ISSN: 1024-1752
CODEN: JERDFO
Vol. 43, No. 1, pp. 369-381
Published Year 2020

Dynamic System Identification for Robot-Assisted Transcranial Magnetic Stimulation

Paramin Neranon*

Department of Mechanical Engineering, Faculty of Engineering Prince of Songkla University, Songkhla, Thailand
*Corresponding Author Email: paramin.n@psu.ac.th

ABSTRACT: Robotic-assisted Transcranial magnetic stimulation has been technically developed and widely used to enhance the long-term outcome of brain stimulation. In this paper, the overall dynamic model of the robotic Transcranial stimulation has been successfully analysed; nevertheless, its parameter estimation leads to a significant challenge due to the complicated dynamic nature of the system. To solve this limitation, system identification based on auto-regressive moving average with an exogenous signal model was consequently implemented. The predicted model was carried out and shown to be an effective data matching with the best-fit percentage of 97.10 %. The optimized gains of proportional and integral control were simulatively achieved using Ziegler–Nichols tuning method incorporated with the transfer function of the proposed dynamic model. These algorithm variables were subsequently used in the robotic implicit force control to regulate the contact forces, in which the human's head is tracked by a 3D scanner in real-time. The results demonstrate significant satisfaction with the overall qualitative performance of the robotic control, and it simplifies effective tracking of the contact force target by indicating closely the desired input with root mean square error of 0.014 N. Hence, it can be concluded that the implicit force control associated with the human's head detection system, is considered acceptable for the effective Transcranial magnetic stimulation to facilitate safe and reliable physical collaboration between the human and robot.

KEYWORDS: Implicit Force Control; Dynamic Model Identification; Robotic Transcranial magnetic stimulation

INTRODUCTION

Transcranial magnetic stimulation (TMS) is a non-invasive, painless and powerful technique of brain stimulation using magnetic field generated by the electric current at a specific area of the brain. TMS, which has been developed and used for more than 30 years, is wildly used to study human brain physiology, particularly in human motor functions and the pathophysiology of brain disorders. A TMS coil is an electrical pulse generator, which is placed to a patient's scalp to stimulate the neural elements. The repetitive TMS effects to motor function recovery of Parkinson's disease, stroke and movement disorder [1]. Traditionally, the TMS coil is held and positioned manually by a specialist doctor without precise clinical experiments. This leads to promising applications of robotics TMS systems to enhance more effective accuracy and reliable TMS.

Robot-assisted TMS systems have been developed by a number of robotic researchers. However, in order to adopt optimized force control in the non-linear robotic TMS system, it required the TMS plant transfer function [2]. The mathematical model computation is more complicated when relevant model parameters are estimated i.e. some impedance parameters of robots, force sensors, robot grippers and human environment [3-4]. To overcome the problem, system identification, which plays a crucial role to deal with the determination of overall system dynamic characteristics using applicable test results, can be strategically implemented. Therefore, this paper describes how to construct an estimated plant transfer function of a robot-assisted TMS system using MATLAB System Identification Toolbox. This technique involves in the optimal test design to effectively deliver the relative information data for an appropriate TMS model fitting. The transfer function of the TMS robot is subsequently evaluated before being utilized to further develop an optimized implicit force controller based on Proportional-Integral (PI) control. Moreover, the comparison between the simulation and real experimental results are finally conveyed.

SYSTEM ANALYSIS AND MODELLING OF ROBOTIC TMS SYSTEM

Conceptual Design and Test Procedure of Robotic TMS System

A conceptual overview of the TMS-robotic system illustrates in Figure 1, which details of in this section. A 6-DOF KUKA robot was employed to handle and manipulate the TMS coil. The human's head position and orientation around x-y-z axes are initially tracked by a 3D Kinect camera, and this can be used for guiding subsequent refinement of the position of the robot end effector. The desired robotic Cartesian position can be achieved by driving each joint position of the robot. It can be determined using the inverse kinematic technique in case of transforming between Cartesian to joint spaces, in which its kinematic model is explained in the next section. Additionally, to protect a human patient from the risk of harm or injury by the TMS robot, initially, a rubber dummy head was temporarily employed in the real tests and mounted to a fixed plate. Practically, as recommended by Barker's study, applying various contact force reported that the TMS electric pulse responses between 2 and 7 N have similar amplitude responses, while at the interactive force of 1 N slightly presents loss of the response amplitude [5]. Therefore, the suitable coil contact force of 2 N is accepted in the TMS task in this study.

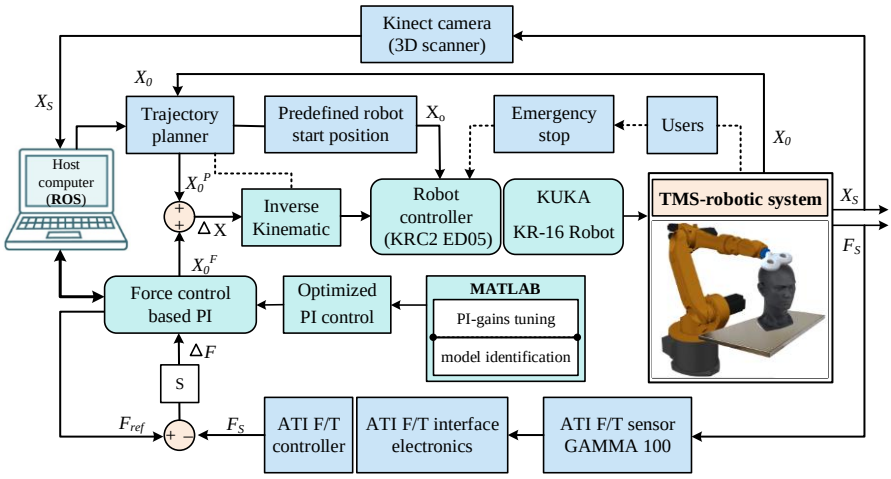

Figure 1. Control block diagram of the robotic TMS system

Once the interactive contacting force between the head and robot occurs, the robot force control approach is continuously executed until the desired contact force is completely adopted. Practically, if a small movement of unrestrained head occurs due to the postural sway or the subject altering their sitting position, it should be tracked and corrected in real-time. Consider the force control schematic diagram in Figure 1. It composes of a multi-axis ATI force/torque sensor mounted between the robot end effector and the stimulating coil in order to measure the force feedback from the patient's head. A stand-alone ATI controller converts all strain gauge signals into the magnitudes of the Cartesian force/ torque components. The head position and orientation and interactive force data are directly transferred to the robotic control system running under the real-time Robot Operating System (ROS) through serial and TCP/IP communication. The information is sent and packed into a single package of additional information including a checksum used to verify the effective data. Additionally, safety issues are crucial aspects of human-robot interaction and were addressed in the robot safety section.

Kinematic Model of a KUKA KR-16 Robot

This section briefly explains the kinematic model of the KUKA KR-16 manipulator arm which relates to forward and inverse kinematics [6-7]. In case of forward kinematics, the location and pose of the robot end effector are controlled by the given links and joints of the robot (depicted in Figure 2), in which the Denavit–Hartenberg parameters evaluation used to achieve the kinematic chain of the robot rigid bodies is shown in the following table [8, 18].

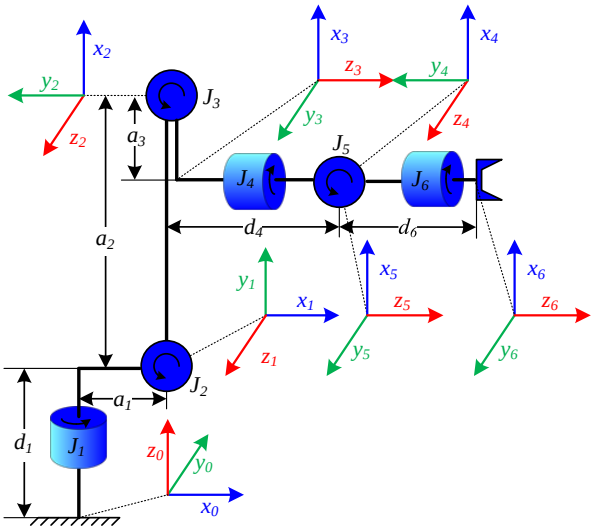

Figure 2. Manipulator kinematic chain and D-H parameters

Table 1. DH parameter estimations for the KUKA robot

Joint	θ°	$d(mm)$	$a(mm)$	α°	θ°_{min}	θ°_{max}
J_1	θ_1	d_1=675	a_1=260	-90	-185	185
J_2	θ_2	d_2=0	a_2=680	0	-155	35
J_3	θ_3-90°	d_3=0	-a_3=-35	-90	-130	154
J_4	θ_4	d_4=670	a_4=0	90	-350	350
J_5	θ_5	d_5=0	a_5=0	-90	-130	130
J_6	θ_6	d_6=115	a_6=0	0	-350	350

The KUKA robot joints are named as J_1 to J_6, and the lengths of the robot linkages are denoted as a_i and d_i. The transformation matrices (T_i^{i+1}) of the robot can be calculated as explained in the following equations, in which the following notation is used: $Rot_{z,\theta i}$ and $Rot_{x,\alpha i}$ denote the rotation with θ_i and α_i around z and x axes. The terms $Trans_{z,di}$ and $Trans_{x,ai}$ stand for translation with d_i and a_i along x and z axes respectively. The parameters: c_i and s_i represent $\cos(\theta_i)$ and $\sin(\theta_i)$, and then the transformation matrices can be presented as:

$$T_i^{i+1} = Rot_{z,\theta i}.Trans_{z,di}.Trans_{x,ai}.Rot_{x,\alpha i} \quad (1)$$

$$T_0^1 = Rot_{z,\theta 1}.Trans_{a1,0,d1}.Rot_{x,-90^{\circ}},\ T_1^2 = Rot_{z,\theta 2}.Trans_{a2,0,0},\ T_2^3 = Rot_{z,\theta 3-90^{\circ}}.Trans_{-a3,0,0}.Rot_{x,-90^{\circ}} \quad (2)$$

$T_3^4 = Rot_{z,\theta 4}.Trans_{0,0,d4}.Rot_{x,90^{\circ}}$, $T_4^5 = Rot_{z,\theta 5}.Rot_{x,-90^{\circ}}$ and $T_5^6 = Rot_{z,\theta 6}.Trans_{0,0,d6}$

$$T_0^1=\begin{bmatrix} C_1 & 0 & -S_1 & a_1C_1 \\ S_1 & 0 & C_1 & a_1S_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\quad T_1^2=\begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\quad T_2^3=\begin{bmatrix} S_3 & 0 & C_3 & -a_3S_3 \\ -C_3 & 0 & S_3 & a_3C_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$T_3^4=\begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},\quad T_4^5=\begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\quad T_5^6=\begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

The homogenous transformation from the robot base coordination to the robot end effector is formulated by:

$$T_1^6=T_0^1T_1^2T_2^3T_3^4T_4^5T_5^6=\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where, r_{i1}, r_{i2} and r_{i3} are orthonormal orientation vectors of x-y-z axes respectively, and p is a position vector. The inverse kinematics of the robot used to transform motion specifications into corresponding joint space can be defined as the determination of each robot joint regarding the position and orientation of the robot end effector. Two techniques of the inverse kinematic solution are made up of mathematical or algebraic (always implemented with higher DOF robots) and geometrical approaches. Thus, the geometrical method was appropriately used to solve the KUKA KR-16 inverse kinematic problem and it gives:

$$\theta_1=Atan2(p_x,p_y)-Atan2\left(d_3,\pm\sqrt{p_x^2+p_y^2-d_3^2}\right) \tag{5}$$

$$\theta_2=Atan2\{\left[(-a_3-a_3c_3)p_z-\left(c_1p_x+s_1p_y\right)(d_4-a_2s_3)\right],\\ [(a_2s_3-d_4)\,p_z-(a_3+a_2c_3).(c_1p_x+s_1p_y)]\}-\theta_3 \tag{6}$$

$$\theta_3=Atan2(a_3,d_4)-Atan2\left(a_3c_3-d_4s_3,\pm\sqrt{a_3^2+d_4^2+(a_3c_3-d_4s_3)^2}\right) \tag{7}$$

$$\theta_4=Atan2[(-r_{13}s_1+r_{23}c_1),(-r_{13}c_1c_{23}-r_{23}s_1c_{23}+r_{33}s_{23})]$$
$$s_{23}=\frac{(-a_3-a_2c_3)p_z+(c_1p_x+s_1p_y)(a_2s_3-d_4)}{p_z^2+(c_1p_x+s_1p_y)^2} \tag{8}$$
$$c_{23}=\frac{(a_2s_3-d_4)p_z-(a_3+a_2c_3)(c_1p_x+s_1p_y)}{p_z^2+(c_1p_x+s_1p_y)^2}$$

$$\theta_5=Atan2(s_5,c_5) \tag{9}$$

$$\theta_6=Atan2(s_6,c_6) \tag{10}$$

To analyse the relation between the robot Cartesian velocity and joint space velocity, the Jacobian matrix or $J(q)$, representing the matrix of all first-order partial derivatives of a vector-valued function, was implemented. Therefore, the robot manipulator Jacobian $\xi(t)$ can be expressed as in Equation (11). When the Jacobian is given as a square matrix, then $\dot{q}$ can be determined as in Equation (12)

$$\xi = \begin{bmatrix} v_n^0 \\ \omega_{0,n}^0 \end{bmatrix} = J(q)\frac{dq}{dt} = [J_1(q), J_2(q), \dots, J_n(q)] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (11)$$

$$\dot{q} = J^{-1}(q)\xi \quad (12)$$

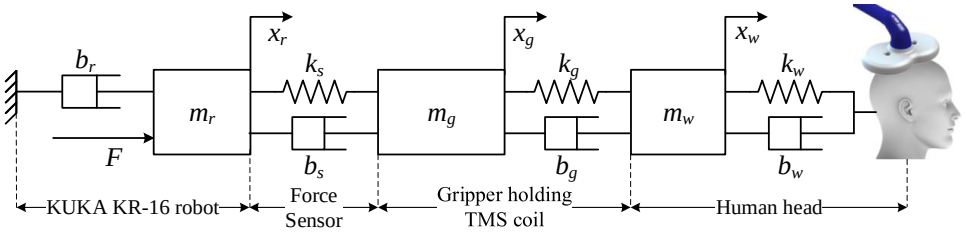

Figure 3. Second-order-lumped-mass model of the Robotic-assisted TMS system

The following notation is used: q_1 to q_n are the robot joint variables, $\omega_{0,n}^0$ is the angular velocity of the end effector and v_n^0 is a linear velocity of the end effector. This finding is subsequently used to compute the velocity of the robot end effector of any point in the robot real-time path control, which is addressed in the robotic implicit force control implementation for TMS section.

Dynamic Mechanical Model of Robotic TMS

This section explains a mathematical dynamic model of the robotic TMS system**.** Initially, the system is assumed to be sub-equivalent-lumped**-**mass systems, illustrated in Figure 3. The KUKA KR**-**16 robot can be modelled by the rigid body model **(**m_r**)** associated with effective viscous damping **(**b_r**)** to ground. The robot itself can be driven by force F generated by robot force control**.** The multi-axis force sensor is represented as a spring and dashpot system, in which the stiffness and viscous damping are k_s and b_s respectively**.** The total mass, stiffness and damping factor of the robot gripper while holding the rigid-body TMS coil are m_g, k_g and b_g. The human model is confined to the environment component, which is contacted by the stimulating coil**.** The human mass represents as m_w, and the viscous damping and stiffness are b_w and k_w**.** In addition, the displacements of the masses **(**m_r, m_g and m_w**)** are respectively defined as x_r, x_g and x_w**.**

There are many ways to establish state variables to describe a system space. By applying Newton's 2nd law of motion, the system dynamic equations of the rigid body masses of m_r, m_g and m_w are computed. The design of the state space for the dynamic model of the TMS specifically requires the characteristic Equations (17-22). Then, the plant transfer functions of the system can be examined using modelling presented in the state space. This section describes how the transfer functions of a series of lumped-parameter models of the robotic TMS system are examined. The state-space equations based modelling were formulated and given in the following equations.

$$\frac{dx_r(n)}{dt} = \dot{x}_r(n) \quad (13)$$

$$\ddot{x}_r(n) = \frac{F(n)}{m_r} - \frac{(b_r+b_s)\dot{x}_r(n)}{m_r} + \frac{b_s\dot{x}_g(n)}{m_r} - \frac{k_s x_r(n)}{m_r} + \frac{k_s x_g(n)}{m_r} \quad (14)$$

$$\frac{dx_g(n)}{dt} = \dot{x}_g(n) \quad (15)$$

$$\ddot{x}_g(t) = \frac{b_s\dot{x}_r}{m_g} - \frac{(b_s+b_g)\dot{x}_g}{m_g} + \frac{b_g\dot{x}_w}{m_g} + \frac{k_s x_r}{m_g} - \frac{(k_s+k_g)x_g}{m_g} + \frac{k_g x_w}{m_g} \quad (16)$$

$$\frac{dx_w(n)}{dt} = \dot{x}_w(n) \tag{17}$$

$$\ddot{x}_w(t) = \frac{b_g \dot{x}_g}{m_w} - \frac{(b_g + b_w)\dot{x}_w}{m_w} + \frac{k_g x_g}{m_w} - \frac{(k_g + k_w)x_w}{m_w} \tag{18}$$

$$\dot{X}(t) = AX(t) + BF(t) \tag{19}$$

$$Y(t) = CX(t) + DF(t) \tag{20}$$

where X is called the state vector, Y is called the output vector, F is called the input vector, A is called the state matrix, B is called the input matrix, C is called the output matrix, and finally, D is called the feed-through matrix [9-10]. According to the TMS system, vector-matrix and system output forms can be written in the state space representation as:

$$\frac{d}{dt}\begin{bmatrix} x_r \\ \dot{x}_r \\ x_g \\ \dot{x}_g \\ x_w \\ \dot{x}_w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{k_s}{m_r} & -\frac{(b_r + b_s)}{m_r} & \frac{k_s}{m_r} & \frac{b_s}{m_r} & & \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_s}{m_g} & \frac{b_s}{m_g} & -\frac{(k_g + k_s)}{m_g} & -\frac{(b_g + b_s)}{m_g} & \frac{k_g}{m_g} & \frac{b_g}{m_g} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_g}{m_w} & \frac{b_g}{m_w} & -\frac{(k_g + k_w)}{m_w} & -\frac{(b_g + b_w)}{m_w} \end{bmatrix} \begin{bmatrix} x_r \\ \dot{x}_r \\ x_g \\ \dot{x}_g \\ x_w \\ \dot{x}_w \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m_r} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{21}$$

$$\text{and } \prime = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_r \\ \dot{x}_r \\ x_g \\ \dot{x}_g \\ x_w \\ \dot{x}_w \end{bmatrix} + [0]F$$

The following equations explain how the system transfer functions are represented based on the proposed state-space form. Consider the state space system is shown in Equations (26) and (27). Ogata (2002) mentioned that the transfer matrix, G(s), which relates to the output vector Y(s) and the input vector F(s) can be derived as [11]:

$$Y(s) = G(s)F(s) \tag{22}$$

After taking the Laplace transforms of the state space equations, the transfer matrix ($G(s)$) can be derived as:

$$G(s) = C[(sI - A)^{-1}B\,] + D \tag{23}$$

Hence, the plat transfer functions are then given by:

$$G_1(s) = \frac{x_r(S)}{F(S)} = \frac{A}{D}, G_2(s) = \frac{x_g(S)}{F(S)} = \frac{B}{D}\,, G_3(s) = \frac{x_w(S)}{F(S)} = \frac{C}{D} \tag{24}$$

where,

$$A = (k_g k_s + k_g k_w + k_s k_w + b_g k_s S + b_s k_g S + b_g k_w S + b_w k_g S + b_s k_w S + b_w k_s S + b_g b_s S^2 + b_g b_w S^2 + b_s b_w S^2 + b_g m_g S^3 + b_g m_w S^3 + b_w m_g S^3 + b_s m_w S^3 + k_g m_g S^2 + k_g m_w S^2 + k_w m_g S^2 + k_s m_w S^2 + m_g m_w S^4)$$

$$B = ((k_s + S b_s)(k_g + k_w + b_g S + b_w S + m_w S^2))$$

$$C = ((k_g + S b_g)(k_s + S b_s))$$

$$D = (k_g k_s k_w + b_g b_r b_s S^3 + b_g b_r b_w S^3 + b_g b_s b_w S^3 + b_r b_s b_w S^3 + b_g b_r k_s S^2 + b_r b_s k_g S^2 + b_g b_r k_w S^2 + b_r b_w k_g S^2 + b_g b_s k_w S^2 + b_g b_w k_s S^2 + b_s b_w k_g S^2 + b_r b_s k_w S^2 + b_r b_w k_s S^2 + b_g b_r m_g S^4 + b_g b_s m_g S^4 + b_g b_s m_r S^4 + b_g b_r m_w S^4 + b_g b_w m_r S^4 + b_r b_w m_g S^4 + b_g b_s m_w S^4 + b_s b_w m_g S^4 + b_r b_s m_w S^4 + b_s b_w m_r S^4 + b_r k_g m_g S^3 + b_g k_s m_g S^3 + b_s k_g m_g S^3 + b_g k_s m_r S^3 + b_s k_g m_r S^3 + b_g k_w m_r S^3 + b_r k_g m_w S^3 + b_r k_w m_g S^3 + b_w k_g m_r S^3 + b_g k_s m_w S^3 + b_s k_g m_w S^3 + b_s k_w m_g S^3 + b_w k_s m_g S^3 + b_r k_s m_w S^3 + b_s k_w m_r S^3 + b_w k_s m_r S^3 + b_g m_g m_r S^5 + b_g m_r m_w S^5 + b_r m_g m_w S^5 + b_w m_g m_r S^5 + b_s m_g m_w S^5 + b_s m_r m_w S^5 + k_g k_s m_g S^2 + k_g k_s m_r S^2 + k_g k_w m_r S^2 + k_g k_s m_w S^2 + k_s k_w m_g S^2 + k_s k_w m_r S^2 + k_g m_g m_r S^4 + k_g m_r m_w S^4 + k_w m_g m_r S^4 + k_s m_g m_w S^4 + k_s m_r m_w S^4 + m_g m_r m_w S^6 + b_r k_g k_s S + b_r k_g k_w S + b_g k_s k_w S + b_s k_g k_w S + b_w k_g k_s S + b_r k_s k_w S$$

As mentioned previously, it is complicated to estimate some unknown model parameters i.e. mass, stiffness and damping parameters in the TMS dynamic equations. For an instant, impedance characteristic of human is normally affected by environment activities, and, and this leads to rough parameter estimation [12]. The MATLAB system identification was consequently adopted, which explains in the following section.

Robotic TMS system identification and its implementation

System identification is a method to mathematically construct dynamic characteristic models using test input and output information. The estimated model is always required to be implemented in controlling the plant, designing and synthesizing a controller for the plant, simulating the controlled plant [13-14]. It can be first assumed that the robotic-assisted TMS is modelled as a black box with unknown model parameters. Several parametric model structures describing system transfer functions and its differential equations are available to assist in developing an unknown parameter model. The auto-regressive moving average model with an exogenous signal (ARMAX) is one of the most robust techniques in the parametric modelling of time series used for studying complex dynamical systems. ARMAX was selected since it has more flexibility in the handling of dynamic disturbance modelling, and is suitable for parameter estimation of nonlinear or non-convex problems [15].

The Z–transform is reasonable for analysis of practical model identification. It can be said that the relationship between the eigenvalues and the poles of the ARMAX-based estimated model can be developed beyond the basics that the eigenvalues can be determined through the poles of the model. The ARMAX model structure can be presented in the following equation [19]. The current observation output: $y(k)$ is the sum of its own past, and the system input is defined as $u(k)$. The noise signal moderated into the plant is $e(k)$. The structural model equation representing the relationship between the system inputs can be expressed as:

$$y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \cdots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t-1) + \cdots + c_{n_c} e(t-n_c) \quad (25)$$

$$A(z)y(t) = B(z)u(t) + C(z)e(t) \quad (26)$$

where, $A(z)$, $B(z)$and $C(z)$ are polynomial equations and can be derived from:

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a}$$

$$B(z) = b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b} \quad (27)$$

$$C(z) = 1 + c_1 z^{-1} + \cdots + a_{n_a} z^{-n_c}$$

The Matlab Identification Toolbox was executed to appropriately estimate the model unknown parameters by selecting a set of a polynomial of n_a, $n_b$and n_c in order to minimize the forecasting errors and to obtain the effective model. In addition, to present the system-parameter estimation technique under consideration adequately, the model validation finally verifies a proposed identified model. According to the ARMAX implementation, a set of preliminary experiments involving a robot-assisted TMS system has been developed and carried out.

The main objective of the experiments is to evaluate the dynamic responses of the contact force between the stimulating coil and the human head in real-time. The key parameters are captured in real-time based sampling time of 0.004 s, which are made up of the interactive force setpoint of 2 N (ARMAX input) and the measured contact force (ARMAX output). The test information collected were initially normalized before being appropriately identified their relationship using Matlab Identification Toolbox based on ARMAX command. The discrete-time ARMAX model based on Equation (30) was identified, and its results are given as:

$$A(z) = 1 - 0.3855 z^{-1} - 0.6006 z^{-2}$$

$$B(z) = 0.008147 z^{-1} + 0.005766 z^{-2} \quad (28)$$

$$C(z) = 1 + 0.1275 z^{-1} + 0.03972 z^{-2}$$

Model quality validation has been executed in the final stage of the model estimation process in order to verify the simulated model outputs and the actual outcomes. Three model validation techniques were then utilized consisting of the percentage of best fit (R^2), the mean sum of square error (MSE) and the final prediction error (FPE). Best fit validation is computed by the determination coefficient R^2 and can be analysed in Equation (33), and MSE is formulated in Equation (34) Additionally, FPE equation can be expressed in Equation (35), to be noted that most estimated models with high accuracy has smallest FPE. After implementing the identification-based ARMAX model, it can be concluded that fit to estimation data is 97.10 %, MSE of 0.00024 was achieved and FPE value is 0.00025. This indicates the actual and estimated data have high similarity in terms of the system responses. Thus, it can be postulated that the acceptable percentage of the model fitting estimation is more than 80% [16].

$$R^2 = 100 \times \left(1 - \frac{\sum_{i=1}^{N} \varepsilon^2}{\sum_{i=1}^{N} (y - \hat{y})^2}\right)\% \quad (29)$$

$$MSE = \sum_{i=1}^{N} \frac{(y - \hat{y})^2}{N} \quad (30)$$

$$FPE = \frac{1}{N} \sum_{i=1}^{N} (y - \hat{y})^2 \left[\frac{(1 + \frac{P}{N})}{(1 - \frac{P}{N})}\right] \quad (31)$$

where, y and $\hat{y}$ are actual and calculated outputs,

N is a number of experimental data, and

P is a number of parameters (n_a+ n_b+ n_c).

Regarding the computing results of the ARMAX model, the system transfer functions without noise $C(z)$ ($ARMAX_1$) and with noise $C(z)$ ($ARMAX_2$) are as follows:

$$ARMAX_1 = \frac{0.008147z^{-1} + 0.005766z^{-2}}{1 - 0.3855z^{-1} - 0.6006z^{-2}} \quad (32)$$

$$ARMAX_2 = \frac{0.01561 + 0.00199z^{-1} + 0.0006201z^{-2}}{1 - 0.3855z^{-1} - 0.6006z^{-2}} \quad (33)$$

MATLAB discrete-time to continuous-time model conversion was applied to convert the transfer function $ARMAX_2$ into a continuous-time transfer function ($ARMAX_{TF}$) as in Equation (38), and this is used in the robotic PI force control.

$$ARMAX_{TF} = \frac{0.01561s^3 + 6.541s^2 + 9321s + 1.802e06}{s^3 + 252.7s^2 + 6.331e05s + 1.371e06} \quad (34)$$

Robotic control implementation for TMS

Implementation and results of robotic implicit force control based on PI algorithm

To deliver more reliability and stability in the robotic TMS, implicit position-based force control, which is a principle method in the achievement of the robot's real-time path control, was implemented. Practically, it is suited to control commercial robots as they have been designed as positioning devices. The type of control provides the force control rule to respond simultaneously to the interactive force (F_s) between the human head and robot end effector. It can compensate for positioning variations in the robot manipulator. As suggested that the position-based force control is appropriately associated with simple control schemes, in this study, it was hence decided to apply PI control to provide the smallest possible error, since the derivative term in PID is too sensitive to noise, and this could introduce a robot destabilizing effect.

For better understanding, consider the control block diagram as depicted in Figure 1. The X and F are 6×1 vectors representing the Cartesian position and orientation and Cartesian force and moment respectively. The S is a 6×6 diagonal selection matrix, where each element becomes a one for position control or zero for no position control. The input of PI force control (ΔF) is the difference between the desired force (F_{ref}) and F_s. Each output from the force control (X_0^F) yields an equivalent position (X_s), which is directly modified by an initial reference position (X_0^P). An incremental position (ΔX) can be computed by subtracting X_s from a current robot position (X_0). Remarkably, the optimized PI gains selected in the force control was appropriately defined using PI control design based on Ziegler–Nichols tuning method in MATLAB based on the proposed transfer function in Equation (38). This strives to keep the force overshoot response to a very minimum and to provide rapid response at the same time. However, PI control has a nonlinear effect, namely integral windup, which has to be considered. This occurs when there is a large change in a setpoint, and the Integral term responds to accumulated errors. It results in excessive overshoot from the setpoint value, and to overcome this limitation, the Integral anti-windup was implemented using a discrete-time equation with sampling time τ, the time interval k, PI control output $U(k)$ and its error $e(k)$ as:

$$U(k) = U(k-1) + k_P[e(k) - e(k-1)] + k_I e(k) \quad (35)$$

MATLAB Simulink block diagram of the robotic TMS system along with PI position-based force control was developed as shown in Figure 4 [17]. Correspondingly, Ziegler–Nichols tuning method, which has been widely used because of their simplicity, deliveries optimized proportional (k_p) and integral (k_i) gains of 1.176 and 2.850 respectively. The test results, where a system setpoint was fixed at the contact force of 2 N, display in Figure 5. This yields the rise time, settling time and overshoot performance of 0.596 s, 0.957 s and 0.386 %. As the frequency response was examined, then a closed-loop stability analysis was applied, and the evaluation revealed satisfactorily stable performance with a phase margin of 88.00 degrees at 3.47 rad/s. It can be summarized that

the set of conveyed PI gains demonstrates significant satisfaction with the overall control performance in the robotic TMS system, and this is subsequently used in the robot implicit force control approach.

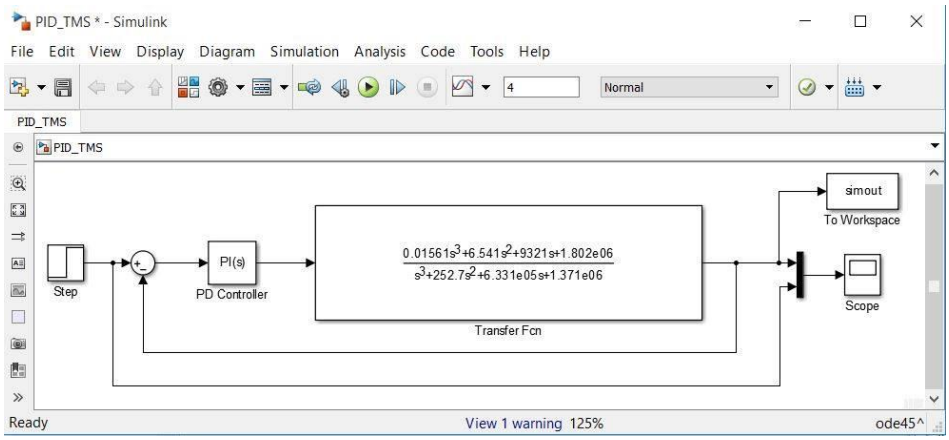

Figure 4. MATLAB simulation diagram

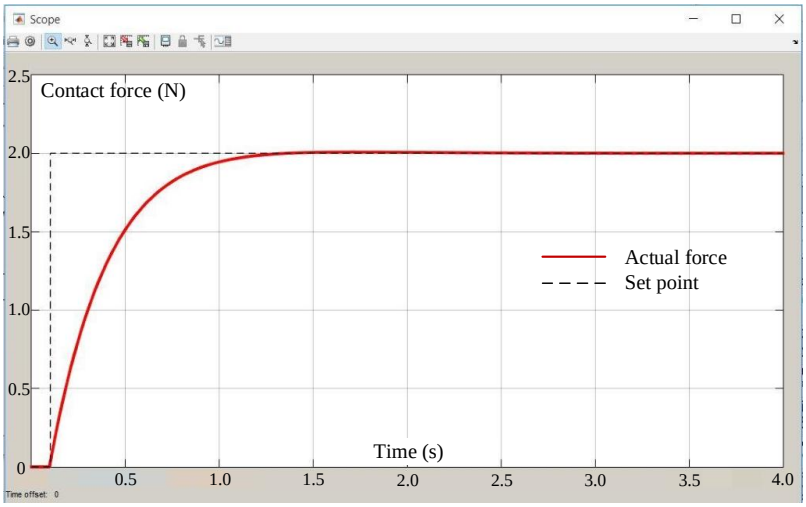

Figure 5. MATLAB simulation results

Safety Issues in Robot-assisted TMS

The most crucial issue in the robotic TMS system is safety because any failures occurring might become very serious problems**.** Hardware and software safety issues have been given due consideration**.** Stand-alone emergency stop and manual control panel (MCP) are available in order to manually deactivate the system by both administrator and patient if accidents are detected**.** The speed of the end effector was limited to 10 mm/s**.** A timeout was assigned for the serial and TCP/IP communication. If the transmission has not been completed yet after the timeout has elapsed, then all communication will be immediately terminated**.** Safety threshold forces **(**$\widehat{F_x}$, $\widehat{F_y}$ and $\widehat{F_z}$**)** were individually defined. If one or more interactive forces are greater than the proposed threshold values, then the robot will be suddenly stopped and waiting for the reset command in order to ensure the safety of the human participant.

Results of Proposed Robotic Force Control based on the PI Control Approach

The implementation of the optimized PI gains for the robot force control was successfully executed. The qualitative evaluation of the control approach is then carried out. The experimental procedure begins with initially setting the fixed contact force of 2 N as suggested. The robot end effector attached by the TMS coil is moving to the human's head position with appropriate orientation using the 3D tracking camera. After completely approaching the target, the robotic implicit force control based on the proposed PI control algorithm is simultaneously activated. In the meantime, interactive force is continuously measured respective to time. Figure

6 provides the results obtained from the real test. Root Mean Square Error (RMSE) presents statistically the comparison between two data sets. RMSE can be formulated as the following equation, where y and $\hat{y}$ are the actual and desired contact force data, and N is a number of captured experimental data.

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(y - \hat{y})^2}{N}} \tag{36}$$

The RMSE technique was implemented to evaluate the qualitative performance, and it gives 0.014 N. It means that the real interactive force exhibits to track closely the desired input with approximately zero error at the steady state. Notably, the overshoot response is not presented, that means the controlled robot system has the capability to modify the interactive force without any higher force amplitude applied to the patient's head. It can also protect the human subject from the risk of harm or injury by the KUKA robot. Therefore, this PI control algorithm is considered acceptable for the effective robot-assisted TMS system, and it offers the safe and reliable haptic interaction between the patient and the KUKA robot manipulator.

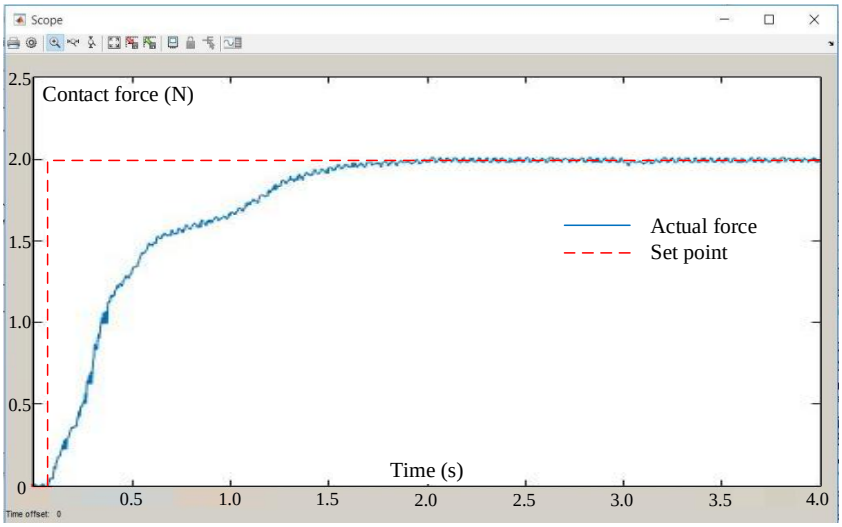

Figure 6. Results obtained from the Robotic TMS system

CONCLUSION

This paper initially extends the dynamic model of the robot-assisted TMS system and its transfer function; however, estimating the relevant impedance parameters has encountered a problematic aspect by reason of dealing with the complicated dynamic characteristic of the human behaviour and the TMS system. In order to overcome this challenging issue, MATLAB model identification using ARMAX model was utilized to facilitate the appropriately approximated discrete-time and continuous-time transfer functions of the robotic TMS system. The model validation was summarized that the percentage of best fit of 97.10 % was achieved, and the FPE and MSE outcomes based on the smallest values are 0.0002, and of 0.0002 was achieved indicating that calculated mathematical models are a highly effective and reliable model.

As the ARMAX model has the capability to effectively identify the TMS plant transfer function, subsequently the PI force control algorithm was simulated based on Ziegler–Nichols gains-tuning method. It strategically offers the optimized k_p and k_i gains of 1.176 and 2.850 respectively. The gain parameters produced was used in the robotic implicit PI force control. The test results of the real robotic TMS reported that this provides an effective matching performance between the setpoint and actual outcomes, which is proved by the RMSE value of 0.014 N at the steady state. It can be concluded that as the quantitative measurement of the robotic control approach was validated and demonstrates effective performance and stability. Therefore the proposed robotic Transcranial magnetic stimulation using PI implicit force control is significantly considered acceptable for the TMS system.

REFERENCES

[1] L. Zorn. et al. "Design and Evaluation of a Robotic System for Transcranial Magnetic Stimulation". *IEEE Transactions on Biomedical Engineering, March 2012*, vol. 59, no. 3, 805-815. 2012.

[2] A.T. Ansari, H. Kala, S. Abirami, K. Thivakaran and R.A.R. Zepherin, "Model identification and comparison of different controller for the air-temperature process. *International Conference on Circuits, Power and Computing Technologies (ICCPCT-2015), Nagercoil*, pp. 1-6, 2015.

[3] S. Sathishkumar, M. Kannan. "Topology Optimization of Integrated Combustion Engine Piston Using FEA Method (CAE Tools)". *Acta Mechanica Malaysia*, vol. 2, no. 1, pp. 01-05, 2019.

[4] Jirgl, M., L. Obsilova, J. Boril and R. Jalovecky. "Parameter identification for pilot behaviour model using the MATLAB system identification toolbox". *International Conference on Military Technologies (ICMT), Brno*, pp. 582-587. 2017.

[5] T. Barker. "The history and basic principles of magnetic nerve stimulation". *Electro-encephalogr Clin Neurophysiol Suppl*. vol. 51, pp. 3-21, 1999.

[6] R. S. Hartenberg, and J. Denavit. "Kinematic Synthesis of Linkages. New York: McGraw-Hill. 1964.

[7] I. ul Haq. "Analytical Approximate Solution Of Non-Linear Problem By Homotopy Perturbation Method (HPM)". *Matrix Science Mathematic*, vol. 2, no. 1, pp. 3(1): 20-24. 2019.

[8] OH Yeon Taek. "Study of Driving Torque through Analysis of Dynamic Characteristics on Industrial Robot". *Journal of Mechanical Engineering Research and Developments*, vol. 40, no. 4, pp. 547-554.

[9] N. S. Nise. "Control Systems Engineering (6th ed.)". *John Wiley & Sons, Inc*. ISBN 978-0-470-54756-4. 2010.

[10] A. Habib, M. A. Nassar. "Modelling of Deposition and Erosion Processes Along A 180° Open Canal Bend by Nays2dh in iRIC". *Engineering Heritage Journal*, vol. 3, no. 2, pp. 1-5. 2019

[11] K. Ogata, "Modern control engineering (4th ed.)". Upper Saddle River, NJ: New Delhi: Prentice Hall. MLA Citation. Ogata, Katsuhiko. 2002.

[12] Rahman, Md. Mr, R. Ikeura and K. Mizutani. "Investigating the impedance characteristic of human arm for development of robots to co-operate with human operators". *IEEE SMC 1999 Conference Proceedings*, vol.2. 676 - 681. 1999.

[13] W., Tang, Z. Liu and Q. Wang. 2017. "DC motor speed control based on system identification and PID auto tuning". *36th Chinese Control Conference (CCC), Dalian*, pp. 6420-6423, 2017.

[14] S. Ishtiaq, A. Sajid, R.A. Wagan. "Review Paper on Wearable Computing its Applications and Research Challenges". *Acta Electronica Malaysia*, vol. 2, no. 1, pp. 3(2): 37-40. 2019.

[15] Ruslan, F. A., K. Haron, A. M. Samad and R. Adnan. "Multiple Input Single Output (MISO) ARX and ARMAX model of flood prediction system: Case study Pahang". *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA), Batu Ferringhi*, pp. 179-184, 2017.

[16] P. Neranon, and R. Bicker. "Human-human interaction using a behavioural control strategy". IEEE Conference on Technologies for Practical Robot Applications (TePRA), Woburn, MA, 1-6. 2013.

[17] K. R. Ashok and K. Balaji. "PI and Sliding Mode Speed Control of Permanent Magnet Synchronous Motor Fed from Three Phase Four Switch VSI". Journal of Mechanical Engineering Research and Developments ISSN: 1024-1752, vol. 40, no. 4, pp. 716-725, 2017.

[18] J. Denavit, and R.S. Hartenberg. "A kinematic notation for lower-pair mechanisms based on matrices". *Trans.ASME E, Journal of Applied Mechanics*, vol. 22, pp. 215-221. 1955.

[19] F. Rabbani. "Model Identification and Validation for a Heating System using MATLAB System Identification Toolbox". *IOP Conference Series Materials Science and Engineering*, DOI:101088/1757-899X/51/1/012022. Vol 5, no. 1, pp. 10, 2013.