Identification and Characteristics of Parallel Actuation Robot's Leg Configuration Using Hammerstein-Wiener Approach

Geogia Guni and Addie Irawan
Robotics & Unmanned System (RUS) research group, Faculty Electrical & Electronics Engineering, Universiti Malaysia Pahang
E-mail: addieirawan@ump.edu.my

***Abstract*-- This paper presents the modeling of a leg of Quadruped with Parallel Actuation Leg (QPAL) robot. QPAL leg designed with 3 Degree of Freedom (DOF) configuration with indirect or parallel actuation for each joint mimicking a muscle of life form creature such as insect and bugs classified into shoulder, thigh and shank parts. Indirect actuation configuration on its leg makes this robot has different perspective on joint rotational drive and control. Therefore, this project has taken initiative to identify and modeling this indirect actuation joint by using system identification (SI) in order to obtain a mathematical model of each joint of QPAL robot's leg. A system identification approach was implemented by employing a Hammerstein-Wiener (HW) model as model structure. The state-space model and the transfer function are designed and generated using Hammerstein-wiener modeling procedures start with experiment setup and data collection from experiment. Continue with data processing, selecting model structure, estimation and validation of the model using system SI toolbox in MATLAB®. The best percentage fits for Joint 1, Joint 2 and Joint 3 are 71.06%, 79.14% and 71.35% respectively, meaning that the estimated model is almost tracking the real output data from the experiments. The model for Joint 1 is ideally acceptable and highly applicable since the correlation curves lie between the confidence interval. While the model for Joint 2 and Joint 3 are considered well and acceptable as the correlation curves are almost lies between the confidence interval. The balances 28.94%, 20.86%, and 28.65% are losses due to nonlinear factor such as friction, backlash, torque, and external disturbance.**

***Index Terms*—Parallel actuation leg, nonlinear Hammerstein-wiener, Joint identification model.**

I. INTRODUCTION

Robots are mainly used to replace human workers in dangerous tasks, high precision or in routine and repetitive works. Research and development in robotics had explored tremendous foundation towards mimicking life form creature especially human. In robotics control point of view, the imprecision will occur along the robot movement or operation and it may be caused by the structural or control algorithms. Due to the imprecision, the robot dynamics parameters also will not able to be brought together into the robot model. On the other hand, knowledge on the parameter values robot must comply with the robot system in detail to get a good robot model[1].

However, the uncertainties in modeling a robot will cause difficulties in forming a good model. Therefore, SI is often required to take uncertainty into the robot model. Thus, SI is widely used in engineering and non-engineering areas as it offers the possibility to build a model from experimental data[1].

Today, robot motion control is a major concern among robot developers, and current development is focusing on improving the performance of the robot, minimizes development costs, improves security, and introduces new functions. Thus, the modeling and identification of a robot system is required in order to control and simulate the system accurately. Generally, the objective of modeling and identification for a robot system is to obtain a suitable mathematical model of the robot system. Hence, the main problem studied in this thesis is the identification of unknown parameters that will be used in the mathematical modeling of QPAL's leg system.

Therefore, this paper presents the modeling of an indirect actuation joint by using SI in order to obtain a mathematical model of each joint of QPAL robot's leg. The SI technique was applied by using a HW as the model structure. The technique concerned on the modeling of the joints of QPAL robot leg is based on the experimental data. The experimental data are obtained from the robot leg real system, which then be used in HW model to build a black-box identification model. This joint-by-joint modeling gives flexibility to the parameterization of nonlinear models of QPAL's leg. The modeling is generated by using SI toolbox of MATLAB® software.

II. RELATED RESEARCH REVIEW

A. Parallel Actuated Robot

Parallel robots have been used over the years, including in the field of astronomy until the flight simulator and is becoming increasingly popular, especially in the machine tool industry. J.-P. Merlet mentioned that, a parallel robot can be defined as a closed loop kinematic chain mechanism with

excellent performance in terms of accuracy, rigidity, and the capability to handle large loads. A parallel robot is made up of an end-effector with (n) DOF, and a permanent base connected together by at least two independent kinematics chains and the actuation takes place through (n) simple actuators[2].

The first parallel robot industrial was designed and patented by Willard L. V. Pollard as shown in Figure 1[3]. A parallel actuated robot development began in the early 1960s, when the first six-linear jack system that functions as a universal tire testing machine is invented by Gough and Whitehall. Years later, in 1965, a platform manipulator called Stewart platform was developed by D. Stewart, which serves as a flight simulator. Since then, there is growing interest in the development of parallel actuated robots[4].

In addition, the closed kinematic chain mechanism has an inverse kinematics easier than the conventional open kinematic chain mechanism. A part from that, the closed kinematic chain manipulator has a better application where the needs of work space and movability is low but require extreme dynamic loads, high speeds and great precision motion[5].

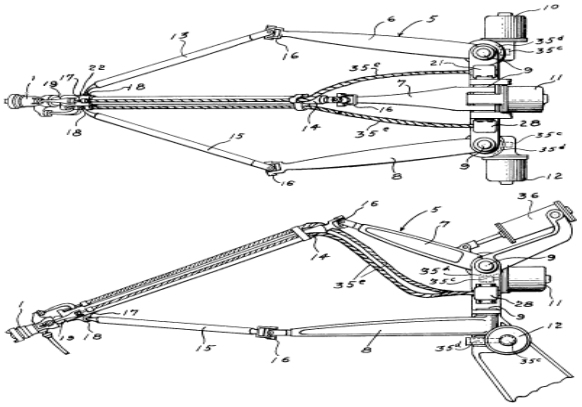


Fig. 1. An illustration of the first spatial industrial parallel robot in 1942[6]

B. Underactuated Robot

Underactuated robot can be described as a robot that composed of underactuated manipulators which made up of active and passive joints in serial chain mechanism[7]. The term called underactuated in robotics means, is to have the number of actuators less than the number of DOF or joints[8]. Moreover, for a robot which a has large redundancy is available for dexterity, large number of DOF and particular task completion such as snake-like robots and multi-legged mobile robots, the underactuated construction enables a more convenient design, simple control mechanism and communication system, weightless and consume less energy compared to a fully-actuated robot. Bergman et al mentioned that when working in dangerous areas or handling hazardous materials, the underactuated robot is very advantageous in the terms of reliability or fault-tolerant design of fully-actuated manipulators. Thus if one or more of the joint actuators fails, means that one or more DOF of the manipulator is also fails. In this case, the failed (passive) joint can still be controlled using the dynamic coupling with the functioning (active) joints, hence the manipulator can still use all of its DOF as initially planned. In addition, there are several advantages of using underactuated system in robotics. As stated earlier, the underactuated robot has less number of actuators than the number of DOF, thus reduced the quantity of actuators for a robot manipulator will reduce energy utilization, and useful to the field in which the energy efficiency is a main factor, like space robots. Next, reducing several actuators enable a more compact design leads to total size and weight minimization. Therefore, this will eventually decrease the development cost and running power[9].

C. System Identification and Its Types

SI is the field of mathematical modeling of dynamic systems from experimental input and output data. In order to make the data extremely informative about the system properties, the input and output data are typically collected from a test or experiment of a real-world system are designed and executed to generate this data[1]. The process of system identification can be outlined in a few steps as follows with reference to Figure 2:

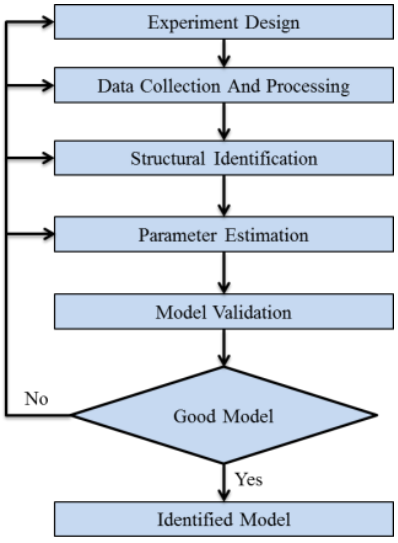


Fig. 2. Process flow of SI modeling

- Experiment design - prepare the experiment setup, decide what signals to measure, choice of sampling time and of excitation signals.
- Data collection and processing – collecting input and output data from the experiment setup and process the data. Eliminating biases, trends, outliers, etc.
- Structural identification - selection of model structure that is required for observation.
- Parameter estimation - executing an identification algorithm and defining the best model criteria to represent the real system.
- Model validation - validating the performance capability of the model in defining the real systems.

Generally, the selection of identification model is based on the available information. Therefore, a better model and more similarities between the system and the model can be constructed if there is more information available from the

system. There are three types of identification models are common in system identification: White Box Modeling, Black Box Modeling and Grey Box Modeling. The white box modeling assumes that the structure and the parameters of the system are completely known and all the complete physical knowledge is available. White box models can be constructed from that information alone without any observations[10].

Meanwhile, if the model construction is based on the experimental data, then it is an input-output or behavioral model. The input-output model can also be described as the black box model because the model is characterized only with its input-output behavior without any detailed information about the system structure. In the black box modeling, the model structure does not define the structure of the physical system, therefore structural elements of the model have no physical meaning. On the other hand, the structure of the model has to be selected that is flexible enough to represent a large class of systems[10].

Actual system usually lies anywhere in between the white box and the black box model. Some physical information is available, but it is not completed, this type of modeling is called the grey-box modeling. The structure of the model of is selected based on the available physical insight, thus the structure of the model will correspond to the physical system. At the same time, the parameters of the model are unknown or only partly known, so they must be obtained from the observed data of the system. The model will be fitted empirically using observations. The common example of grey-box modeling is a physical modeling. The more complete the physical insight the "lighter" grey box model can be constructed and vice versa. The "darkness" of model is based on the unknown and known information of the system to be modeled, as shown in Figure 3[10].

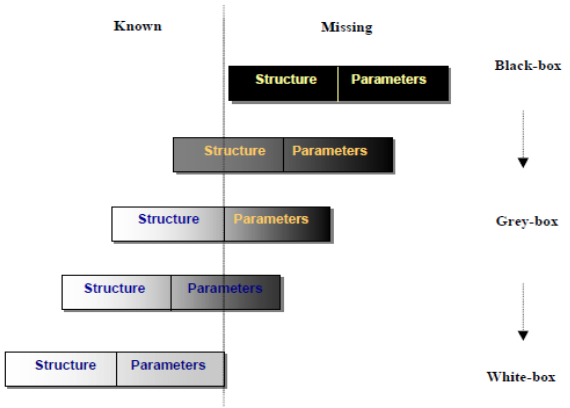


Fig. 3. Identification models based on the prior information

D.Nonlinear Model Identification

SI is an essential tool in technical field where most physical systems are nonlinear processes. Almost all of the systems are nonlinear where the output is a nonlinear function of the input elements. On the other hand, the linear model is always adequate to precisely define the system dynamics, usually via fitting the system with the experimental suitable linear models. *Hussain et al* in their paper indicated that, the SI technique of the dynamics models are represented by mathematical relations between the system's inputs (u) and outputs (y) at time (t). These mathematical relations will be applied to estimate the current output from previous inputs and outputs. The equation of nonlinear model for discrete time can be expressed below, where the function (f) is a nonlinear model that contain nonlinear elements represent the arbitrary nonlinearities of the systems[11].

$$y(t) = f(\mathrm{u}(t-1),..., y(t-1), \mathrm{u}(\mathrm{t}-2), \mathrm{y}(\mathrm{t}-2),...) \quad (1)$$

In addition, there are several types of nonlinear models available to describe a system dynamic, such as nonlinear autoregressive exogenous model (NLARX), Hammerstein model, Wiener model and HW model. However, HW model had proven is the best to describe the nonlinear dynamic systems. In order to estimate HW models a uniformly sampled time-domain data are needed where the data are contains of single-input and single-output (SISO) channels[11].

Therefore, the goals of SI are to acquire the best suitable mathematical model for the real system by using the actual data. The best fitting model will be useful for getting a good understanding on the real dynamic system and also to predict or simulate the behavior of the system, especially to act as control mechanism for the design and analysis of the controller that is depend on the actual system model[11].

E. Nonlinear Arx Model

A nonlinear ARX (NLARX) model is the extended of linear ARX models to the nonlinear situation as shown in Figure 4 and expressed in Eq. (2). According to *Lennart Ljung*, the function (f) depending on the finite number of previous inputs (u) and outputs (y), where (n_a) is the number of past output terms, (n_b) is the number of past input terms used to predict the current output and (n_k) is the delay from the input to the output (the number of samples)[12].

$$y(t) = f(\mathrm{y}(\mathrm{t}-1),..., \mathrm{y}(\mathrm{t}-\mathrm{n}_a), \mathrm{u}(\mathrm{t}-\mathrm{n}_k),..., \mathrm{u}(\mathrm{t}-\mathrm{n}_k-\mathrm{n}_b+1)) \quad (2)$$

Therefore, this model is used to describe the nonlinear extensions of linear models. The structure enables complex nonlinear behavior to be modeled using flexible nonlinear functions, such as wavelet and sigmoid networks. The NLARX model is usually used as a black-box identification model because the nonlinear function of the NLARX model is a flexible nonlinearity estimator with parameters that need not have physical significance[12].

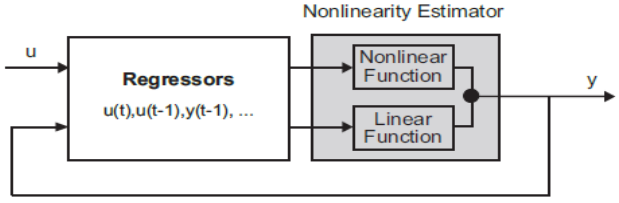


Fig. 4. The structure of a NLARX model

F. Nonlinear Hammerstein-Weiner Model

Lennart Ljung stated that, the HW models can be used to defined the dynamic systems that using one or two static nonlinear blocks in series with a linear block, where the linear block is a discrete transfer function and epitomizes the dynamic component of the model[12]. Figure 5 shows the structure of HW Model that represents the dynamic system using input and output static nonlinear blocks in between dynamic linear blocks which is distorted by static nonlinearities. Furthermore, the HW structure can also be used to capture the physical nonlinear effects in the system that will affect the input and output of the linear system[11].

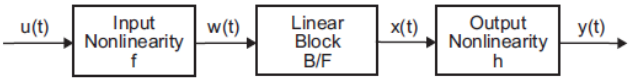


Fig. 5. The fundamental structure of a HW model

Since the applications of HW Model depending on the inputs, thus if the output of the system depends nonlinearly on the input, it can break down the input and output relationship of two or more elements that are interconnected. This structure is recommended because they have a simpler block representation, transparent relationship to linear systems, and is easier to be carryout than heavy-duty nonlinear models. With reference to Figure 2.5, the HW model can be outlines as a combination of three series blocks that can be expressed as Eq. (2.3) where $u(t)$ is an input data which has the same dimension as $\mathrm{w}(t)$ [13].

$$w(t) = f(\mathrm{u(t)}) \tag{3}$$

For the second block:

$$x(t) = \frac{B}{F} w(t) \tag{4}$$

Meanwhile, Eq. (4) is a linear transfer function and $x(t)$ has the same dimension as $y(t)$, where (B) and (F) are similar to polynomials in the linear Output-Error model. For n_y outputs and n_u inputs, the linear block is a transfer function matrix containing entries[13, 14]:

$$\frac{B_{j,i}(q)}{F_{j,i}(q)}$$

where:

$$j = 1, 2, ..., n_y$$

$$i = 1, 2, ..., n_u$$

For the third block:

$$y(t) = h(\mathrm{x(t)}) \tag{5}$$

On the other hand, Eq. (5) is defined as a nonlinear function that maps the output of the linear block to the system output, which $w(t)$ and $x(t)$ are internal variables that define the input and output of the linear block, respectively. As (f) acting as an input port of the linear block, this function is called the input nonlinearity. Since (h) also acting as an output port of the linear block, this function is called the output nonlinearity. Thus if a system consists of more than one inputs and outputs, the functions (f) and (h) must be define for each input and output signal[13, 14].

However, it is not compulsory to contain both the input and the output nonlinearity in the model structure. If a model consists only the input nonlinearity (f), then it is a Hammerstein model. In the same way, when the model consists only the output nonlinearity (h), thus it is a Wiener model. Therefore, the nonlinearities (f) and (h) can be defined as a scalar function, meaning that one nonlinear function for each input and output channel. Hence the process of HW model estimates the output $y(t)$ can be summarize into three steps as follows[13, 14]:

- Evaluates Eq. (3) from the input data, where, $w(t)$ is the input to the linear transfer function $\frac{B}{F}$. The input nonlinearity is a static (memoryless) function, where the value of the output at given time (t) depending to the input value at time (t). The input nonlinearity can be defined as a sigmoid network, wavelet network, saturation, dead zone, piecewise linear function, one dimensional polynomial, or a custom network. It is possible to remove the input nonlinearity.
- Determines the output of the linear block using $w(t)$ with Eq. (4) as the initial conditions, where the configuration of the linear block will be done by defining the numerator (B) and denominator (F) orders.
- Estimates the model output by transforming the output of the linear block using the nonlinear function (h), as it mentioned in Eq. (5).

III. QPAL ROBOT SYSTEMS & CONFIGURATION

A. QPAL Robot System Overview

QPAL Robot was designed and developed in February 2014. The robot was designed for medium capacity multi-purpose applications such as advanced firefighting systems, mine detection, simple tunnel system studies etc. In addition to the statically stable and active suspension (multi- joints) robot system configuration, this robot can be used for various difficult tasks and uneven terrain. As its name implies, the robot was designed with four legs, each with a 3-DOF and it is

driven by entirely by electrical energy. On the control panel, a special computer-based control with single board design namely QBoard uses a relatively microcontroller unit (MCU) that is placed in the center of the robot body[15], as shown in Figure 6.

The body frame was minimized in order to maintain the overall size and weight of the robot down. Framework for the design of the body is made of aluminum rod with dimensions of 0.44m length, 0.21m width and height of 0.12 m. In order to reduce the quantity of hardware required to attach the whole assembly, the body frame share its hardware attachment with the shoulder. The electronics devices are installed as close as possible to the center of the body frame to be easily covered by a shell and reduce the quantity of wires wiring[15].

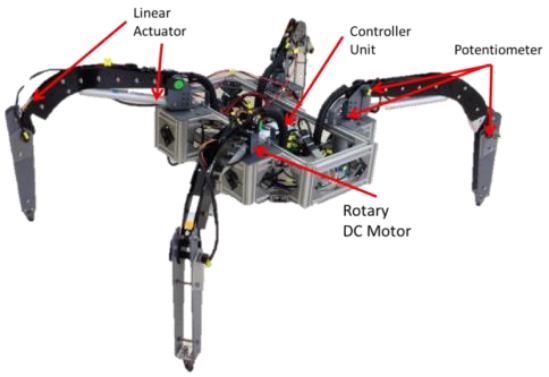


Fig. 6. QPAL robot system configuration

B. QPAL Leg Design and Configuration

The shoulder of the robot was intended to hold the motor for each leg and to ensure that all attachments for the legs and motor can be connected in the same assembly. The movement of the shoulder are directly controlled to enable the horizontal rotation clockwise and anti-clockwise direction meanwhile an indirectly control system with a parallel actuation were used to control the other joints. On top the motors; there is another surface to hold the potentiometer for measuring the rotation of the shaft. Shaft potentiometers help support the end of the motor shaft to prevent the shaft from fully cantilevered[15].

In order to control the movement of the leg, a DC linear actuator M1 and M2 are mounted as shown in Figure7. The same DC linear actuator is used to control the height of the thigh from the axle on the thigh joint to the axle on the shank; the structure should not interfere with the wires. Both the DC linear actuator is completed by using an indirect control system to cause a different direction either clockwise or counter-clockwise. This allows the leg to perform a motion. The shank of the quadruped robot was designed by the need of what the robot required to move properly. The legs should be kept as close as possible to the robot body to minimize the amount of torque on each joint. The design of the leg is intended to be one piece to make the assembly of the robot easier. Another DC linear actuator is used to allow the leg to be pulled in both clockwise and counter clockwise directions. To reduce the friction when the leg contacted with the ground, a semi-sphere rubber ball are used at the end effector as shown in the green circle in Figure 7[15].

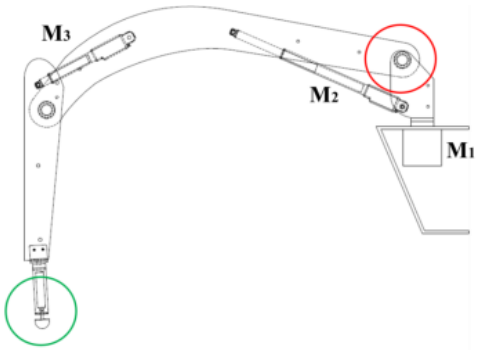


Fig. 7. QPAL robot leg structure

C. QPAL Leg Coordination and Kinematics

QPAL Robot was designed with three degree of freedom (3-DOF) since it has three different links for each leg in order to mimic the leg structure of a quadruped creature through the shoulder, thigh and shank links, as shown in Figure 8. The calculation of kinematics for this robot is based on the shoulder point known as shoulder coordinate system (SCS) as shown in Figure 9. Meanwhile, the calculations of kinematics based on the body coordinate system (BCS) are determined from SCS for each leg, as shown in Figure 3.4. For the inverse kinematics calculation the configuration of angle for each DOF is used[15], as shown in Figure 10.

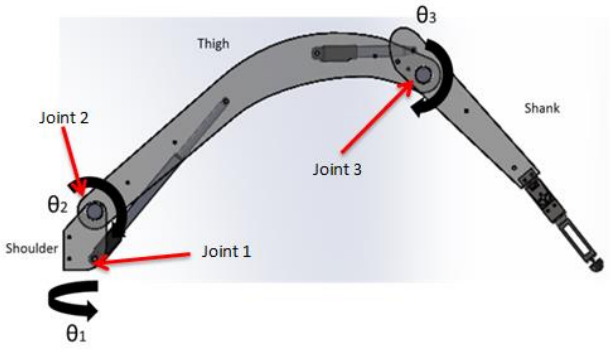


Fig. 8. A 3-DOF QPAL robot leg with shoulder, thigh and shank links

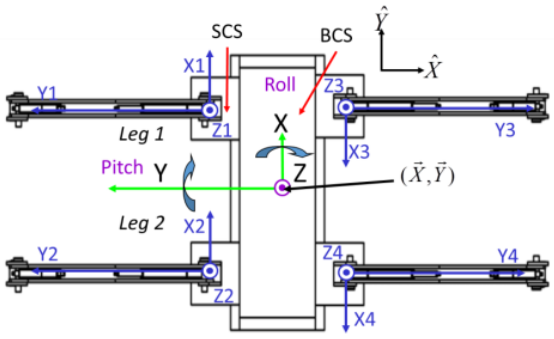


Fig. 9. Coordinate system used for QPAL robot

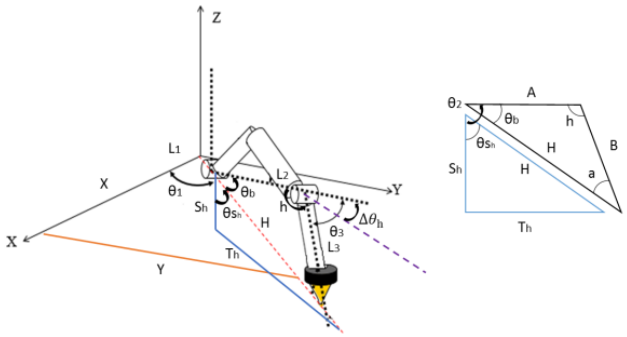


Fig. 10. QPAL geometrical-based coordination system for one leg

D. QPAL Robot Control System Overview

As mentioned in before, QPAL robot is a computer-based control with single board design namely QBoard. The control system mechanism of the QPAL robot is designed with QBoard, digital multiplexer and full bridge drivers to manipulating a single leg, as shown in Figure 11. Through the study of QPAL robot control system, it can be separated into two different categories which are the low-level I/O port read/write operations and high-level locomotion to allow the robot walking on the flat ground surface. Due to QPAL's complexity, this method has some weaknesses. As a solution, Arduino Mega is selected to overcome the weaknesses and MATLAB® software is used to build the system architecture for operating the robot. QPAL robot made up of 12 controllable joints, where each leg consists of three sources of feedback from the potentiometer to make a closed-loop control system for the robot. Thus, the QBoard system control system is designed to allow dual driver controllers for each leg for testing purpose, where the main microcontroller runs the complete operating system, while the dual driver software is fully application specific and the leg position of QPAL's is controlled by using an individual dual driver[15].

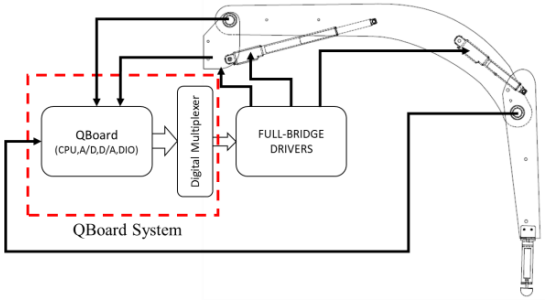


Fig. 11. QPAL robot control system design

III. QPAL ROBOT LEG'S IDENTIFICATION AND MODELING

A. Introduction

The experiment setup for this project consists of one leg of QPAL robot with computer-based controller QBoard[15] connected with the personal computer (PC) with embedded SIMULINK MATLAB® software. The SIMULINK program as shown in Figure 12 is selected as the software platform for control and monitoring QPAL robot's leg movement. Figure 13 shows the real experiment setup for QPAL leg system. The project use voltage input signal to each joints and the output is the angle of the joints, with sampling time of 0.1s for model estimation and validation. The identification and modeling technique is performed by using Figure 2 as a reference.

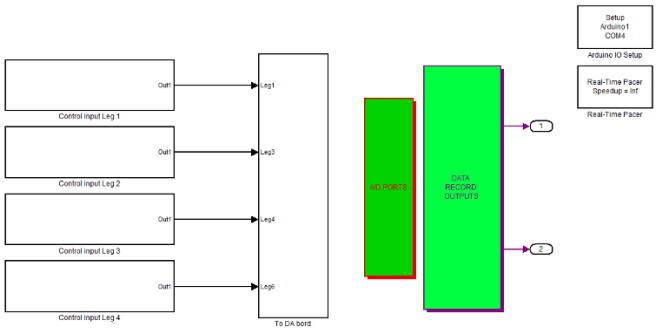


Fig. 12. Block diagram of control system for QPAL robot leg

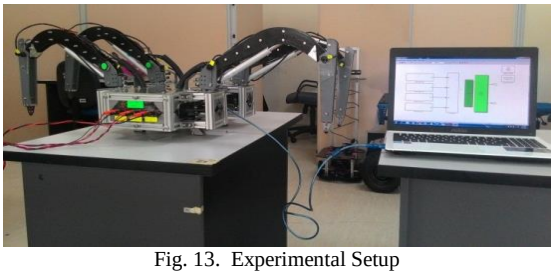


Fig. 13. Experimental Setup

B. Data Preprocessing

The raw data collected during the experiment need to be preprocessed before starting the identification process. Since measured data frequently have offsets, slow drifts, outliers, missing values, and other irregularities, thus SI toolbox will eliminate such irregularities by executing process such as detrending, filtering, resampling, and reconstruction of missing data. Moreover, the toolbox can analyze the appropriateness of data for identification and provide diagnostics on the persistence of excitation, existence of feedback loops, and presence of nonlinearities[16]. The data preprocessing can be done by selecting the 'Quick Start' option in SI App, which is the pre-processing shortcut process for the experimental data. The 'Quick Start' option instantaneously performs the following four actions: removes the means from the experimental data (input and output data), it splits the data (detrended data) into two parts, specify the first part as estimation data for models (or working data) and specify the second part as validation data[12].

C. Selecting Model Structure

Selecting the right model structure is prerequisite before its estimation and the selection of model structure is based upon understanding of the physical systems. Since the leg system is a nonlinear system, nonlinear SI is used as the model structure. There are two types of nonlinear model structure in SI: NLARX model and HW model. Therefore, the selection between these two models is decided based on the comparison of the highest best fits results for Joint 1, Joint 2 and Joint 3. The estimation for selecting between the NLARX and HW model with respect to the raw data was generated with by using the default setting of both model.

D. Selecting Model Structure

SI Toolbox offers a few scalar nonlinearity estimators $f(x)$ for HW models, where nonlinearity estimators are available for both the input and output nonlinearities (f) and (h), respectively[17]. Every nonlinearity estimator resembles to an object class in SI toolbox. As the HW model is estimated in the app, SI toolbox will generate and constructs objects based on these classes. In addition, the nonlinearity estimators can also be created and configured at the command line[12].

In addition, there are six types of numerical search method used for iterative parameter estimation of HW model provided in SI toolbox: auto (default), Gauss-Newton (gn), Adaptive Gauss-Newton (gna), Levenberg-Marquardt (lm), Trust-Region Reflective Newton (lsqnonlin) and Gradient Search (grad)[12]. Therefore, for this project, the estimation of nonlinearity for the input and output channel together with varying of linear order and the search method is done by using the trial and error methods. Then, SI toolbox GUI will generate the best fits to the experimental data for each joints of QPAL leg system.

E. Model Validation

SI toolbox provides feature to validate the accurateness of identified models using independent sets of experimental data from a real system. For a given set of input data, the toolbox will generates the output of the identified model and compare the output with the experimental output data from a real system. Additionally, the toolbox can also computes the prediction error and produce time-response and frequency-response plots with confidence bounds to visualize the effect of parameter uncertainties on model responses. Moreover, the toolbox also able to analyze the identified model using time-response and frequency-response plots, such as step, impulse, Bode plots, and pole-zero maps by dragging the identified model into the LTI viewer[16].

F. Linearization

Since the control design and linear analysis methods using Control System Toolbox software require linear models, thus the estimated nonlinear model in SI toolbox must be linearize so that the model can be used for control design and linear analysis purpose. According to *Lennart Ljung*, there are two techniques to determine a linear approximation of nonlinear models: linear approximation for a given input signal and tangent linearization. In MATLAB® software, *linapp* command is used to generate a linear approximation of a nonlinear model for a given input signal, which the resulting model is only valid for the same input that is used to generate the linear approximation. Meanwhile, the *linearize* command is used to computes tangent approximation of the nonlinear dynamics that is accurate near the system operating point, where the resulting model is a first-order Taylor series approximation for the system about the operating point, which is defined by a constant input and model state values[12]. As for this project, the linear approximation for a given input signals technique is used to linearize the HW model of the joints of QPAL leg system. The *linapp* command will generate the best linear approximation of a NLARX or HW model for a given input or a randomly generated input in a mean-square-error sense, where the resulting linear model might only be valid for the same input signal as the one that is used to computes the linear approximation. The *linapp* command also determines the best linear model that is structurally similar to the original nonlinear model and delivers the best fit between a given input and the corresponding simulated response of the nonlinear model. In order to generate a linear approximation of a nonlinear model for a given input, the necessary variables are as follows[12, 18]:

- Nonlinear ARX model (*idnlarx* object) or Hammerstein-Wiener model (*idnlhw* object).
- Input signal for which needed to obtain a linear approximation, specified as a real matrix or an iddata object.

The specified input signal used by *linapp* command to compute a linear approximation can be outlined as follows[12]:

- For nonlinear ARX models, linapp estimates a linear ARX model using the same model orders n_a, n_b and n_k as the original model.
- For Hammerstein-Wiener models, linapp estimates a linear Output-Error (OE) model using the same model orders n_b, n_f and n_k.

Generally, the *idnlhw* object can be generated by importing the identified HW model from the SI App into the MATLAB® workspace. The HW model is then linearized by using the following syntax in the MATLAB® command window:

```
[X,U] = findop(sys,'steady',InputLevel,OutputLevel)
SYS = linearize(NLSYS,U0,X0)
t_fsys = tf(sys)
```

where,
[X,U] = findop(sys,'steady',InputLevel,OutputLevel) returns the operating-point state values, (X), and input values, (U), for the *idnlarx* model, (sys), by using steady-state input and output specifications[19].

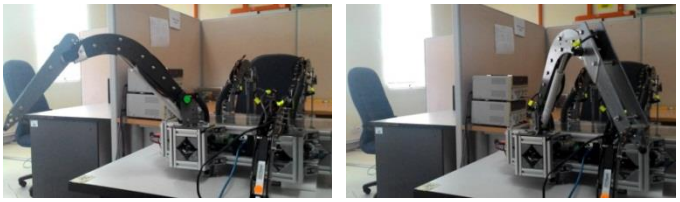
SYS = linearize (NLSYS,U0,X0) linearizes the *idnlhw* model (NLSYS) around the operating point specified by the input (U0) and state values (X0), where, (X0) must not contain equilibrium state values[20].

tf_sys = tf(SYS) transforms the *idnlhw* model (SYS) into transfer function form. The output (tf_sys) is a (tf) model object representing (SYS) as a discrete time transfer function[21].

IV. RESULTS AND ANALYSIS

A. Experiment Setup and Running Procedure

Figure 14 to 16 shows the running procedure for each joints of the QPAL's leg system. The experimental data is obtained via moving the joints from its minimum to its maximum angle to make the data very informative about the leg system by using Figure 12 as the software platform to control QPAL's leg movement. The data is obtained by using sampling time of 0.1s.



(a) (b)

Fig. 14. Joint 1 running procedure, (a) minimum angle position (b) maximum angle position

(a) (b)

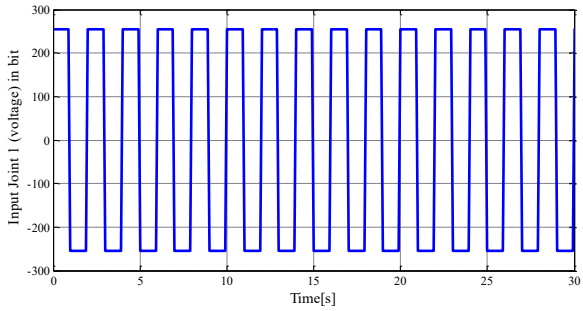
Fig. 15. Joint 2 running procedure, (a) minimum angle position (b) maximum angle position

(a) (b)

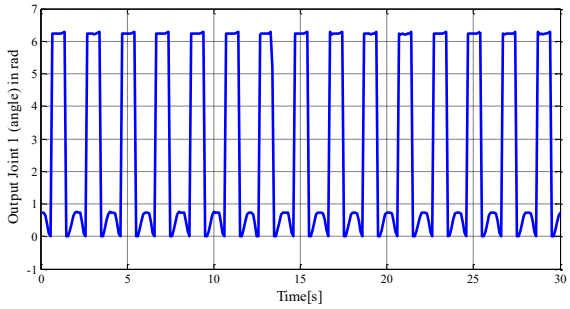
Fig. 16. Joint 3 running procedure, (a) minimum angle position (b) maximum angle position

B. The Preprocessed Data

The raw data are preprocessed to remove the means before the estimation process. The data also splits into estimation and validation data. The input and output data obtained from the experimental data for each joints of QPAL leg as shown from Figure 17 to 19. On the other hand, Figure 20 to 22 shows the experimental data after the data preprocessing process.

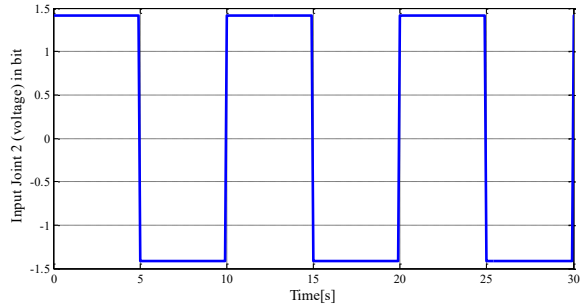


(a)

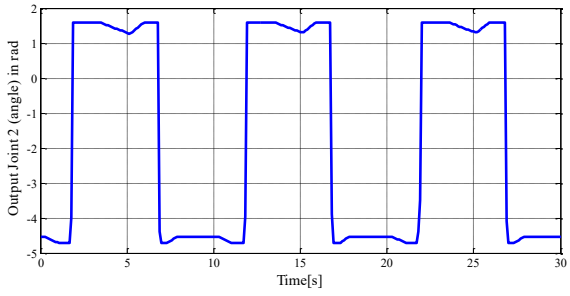


(b)

Fig.17. The experimental data for Joint 1(sample for first 30 seconds); (a) input data, (b) output data

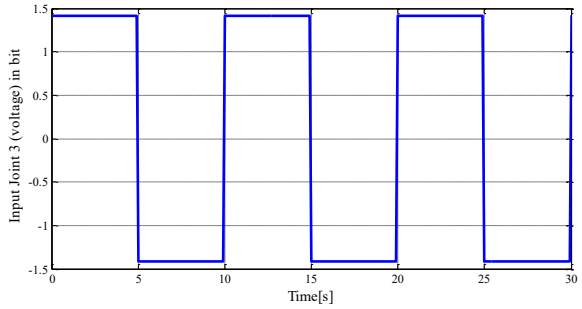


(a)

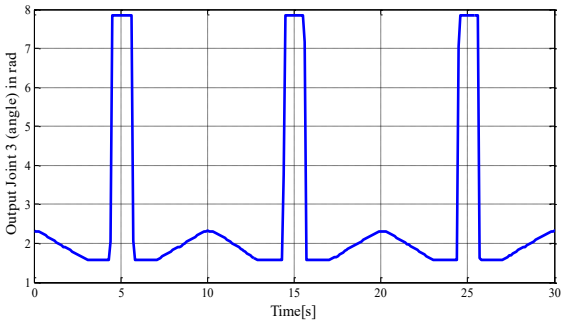


(b)

Fig. 18. The experimental data for Joint 2 (sample for first 30 seconds); (a) input data, (b) output data



(a)



(b)

Fig. 19. The experimental data for Joint 3 (sample for first 30 seconds); (a) input data, (b) output data

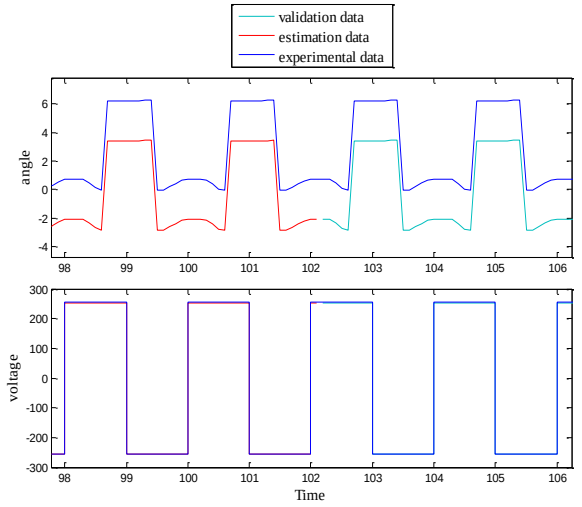


Fig. 20. The preprocessed experimental data for Joint 1

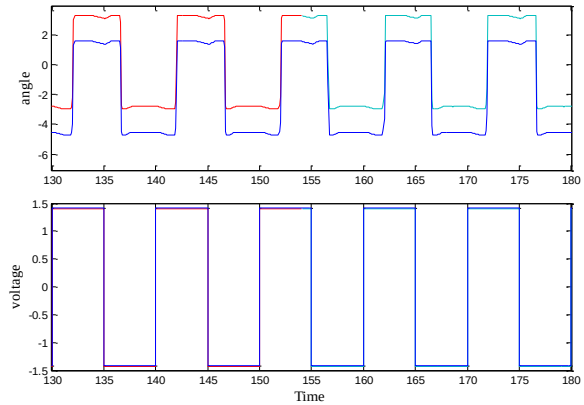


Fig. 21. The preprocessed experimental data for Joint 2

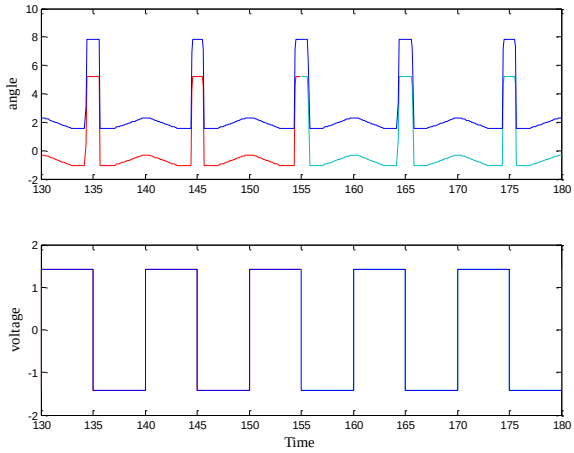
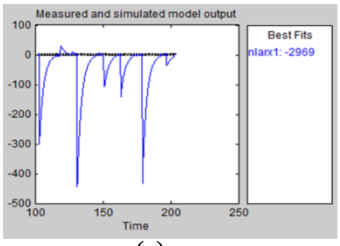


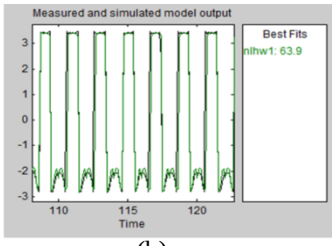
Fig. 22. The preprocessed experimental data for Joint 3

C. Selected Model Structure

As stated in Section 4.3, the selection between NLARX model and HW model is decided based on the comparison of the highest best fits results for Joint 1, Joint 2 and Joint 3. Thus, the estimation of NLARX model with respect to the raw data is generated with $n_a = 2$, $n_b = 2$, $n_k = 1$ while (f) is applied as a wavelet network. These values are the default setting for NLARX model. Meanwhile, the estimation of HW model is computed with $b_n = 2$, $f_n = 3$, $k_n = 1$, where input and output nonlinearity estimators is a piecewise linear. These values are also the default setting for HW model. From Figure 23 to 25 shows the best fits of both model structures obtained with the default setting. From the results, it can be concluded that for the joints of QPAL leg system, the HW model provide a better result compared to the NLARX model. Thus, HW model is selected as the model structure for this project to obtain the mathematical model for these joints of QPAL leg system.

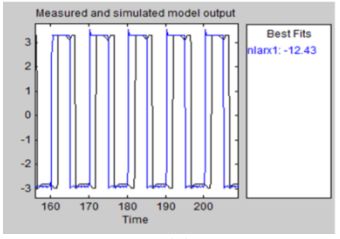


(a)

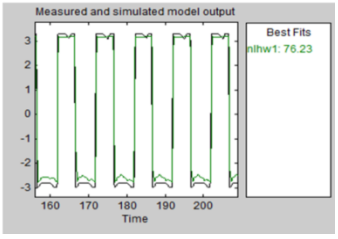


(b)

Fig. 23. Model output for Joint 1 using (a) NLARX Model, (b) HW Model

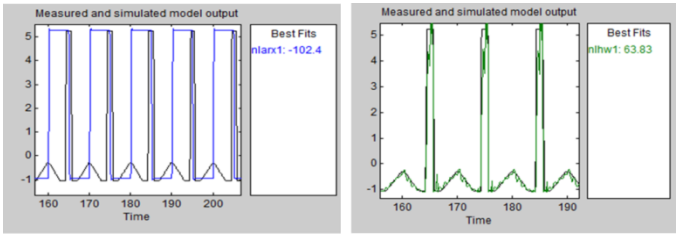


(a)



(b)

Fig. 24. Model output for Joint 2 using (a) NLARX Model, (b) HW Model



(a) (b)

Fig. 25. Model output for Joint 3 using (a) NLARX Model, (b) HW Model

D.Estimated Model Output Analysis

After the selection of model structure is finalized, the estimation of the model continued with the estimation of the nonlinearity estimator for the input and output together with varying the linear order by using the trial and error technique since there is no information available regarding the joints system.

The estimation process continues until it achieves the highest possible best fits for the model. After an uncountable of estimation trials, the model that provide the best fits yield the linear order of $b_n = 3$, $f_n = 3$, $k_n = 1$, estimated with sigmoid network as the input nonlinearity and piecewise linear as the output nonlinearity. The validation data shows 71.06% best fits with low Final prediction error (FPE) = 0.054 and loss function = 0.04841 by using Levenberg-Marquardt (lm) algorithm, meaning that the estimated model nearly tracking the real output data from the experiments, as shown in Figure 26, while Figure 27 shows the close-up for the first upper peak and the lower peak waveform of the data.

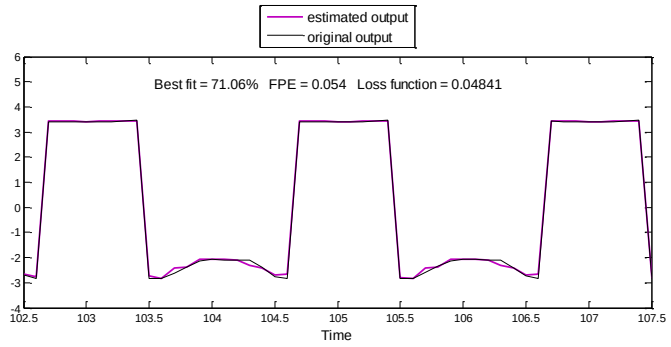
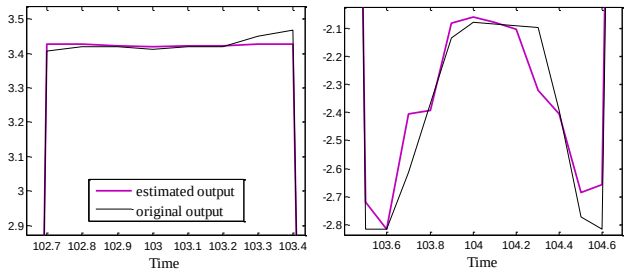


Fig. 26. Measured versus simulated model outputs for Joint 1(sample for first 5 seconds)



(a) (b)

Fig. 27. Joint 1 model output close-up; (a) upper peak (b) lower peak

The estimated model must undergo the linearization process to obtain the mathematical model for the system. As results, the state-space model and the transfer function representing the estimated model for joint 1 in the form of discrete-time function can be shown in Eq. (6) and Eq. (7) below. In addition, the step response result from the SI toolbox shows a good oscillatory reaction to the final steady value for the estimated model, as shown in Figure 28. Meanwhile, Figure 29 proved that the estimated model system is stable since all the poles of the transfer function lies within the unit circle of the z-plane.

Joint 1 discrete-time state-space model:

$$a = \begin{bmatrix} -0.4951 & 0.8738 & 0.5013 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 0.462 \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

$$c = \begin{bmatrix} -0.009381 & 0.03261 & 0.05884 \end{bmatrix}$$

$$d = \begin{bmatrix} 0 \end{bmatrix}$$

Joint 1 discrete-time transfer function:

$$tf = \frac{-0.004334z^2 + 0.01507z + 0.02719}{z^3 + 0.4951z^2 - 0.8738z - 0.5013} \quad (7)$$

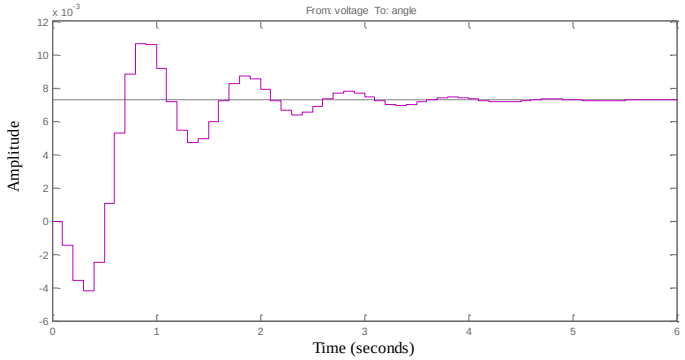


Fig. 28. Estimated model step response for Joint 1

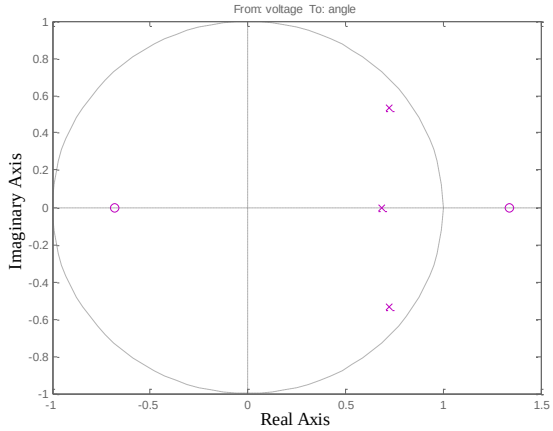


Fig. 29. Estimated model zero and pole output for Joint 1

Figure 30 shows the estimated model output computed using HW model after varying the nonlinearity estimator of input and output channel together with varying of linear order for trial and error methods. The best fit of 79.14% is obtained for the experimental data of Joint 2 with low Final prediction error (FPE) = 0.101 and loss function = 0.09762. The model is estimated with linear order of $b_n = 2$, $f_n = 2$, $k_n = 1$, with saturation as the input nonlinearity and piecewise linear as the output nonlinearity by using default search method. Since the validation data shows 79.14% best fits, it can be concluded that the estimated model is very nearly tracking the real output data from the experiments. The lower and upper peak waveform close-up can be shown as in Figure 31.

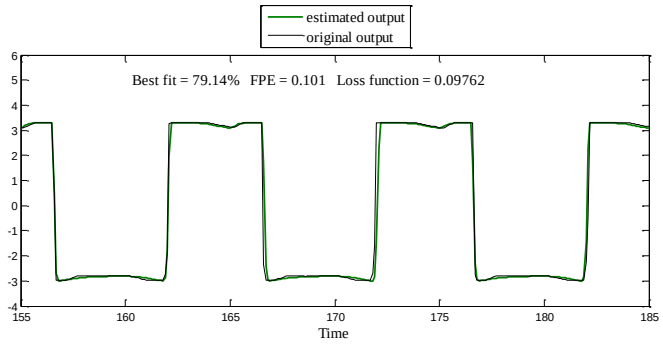
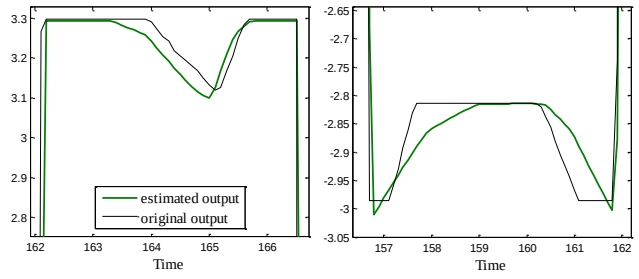


Fig. 30. Measured versus simulated model outputs for Joint 2 (Sample for first 30 seconds)



(a) (b)

Fig. 31. Joint 2 model output close-up; (a) upper peak (b) lower peak

The discrete-time state-space model and the discrete-time transfer functions that represents the estimated model for Joint 2 after the linearization was performed is shown in Eq. (8) and Eq. (9). Figure 32 shows the step response result from the SI toolbox shows a good oscillatory reaction to the final steady value for the estimated model, while Figure 33 proved that the estimated model system is stable since all the poles of the transfer function lies within the unit circle of the z-plane.

Joint 2 discrete-time state-space model:

$$a = \begin{bmatrix} 1.976 & -0.9761 \\ 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad (8)$$

$$c = \begin{bmatrix} -0.02534 & 0.02529 \end{bmatrix}$$

$$d = \begin{bmatrix} 0 \end{bmatrix}$$

Joint 2 discrete-time transfer function:

$$tf = \frac{-0.05067z + 0.05058}{z^2 - 1.976z + 0.9761} \quad (9)$$

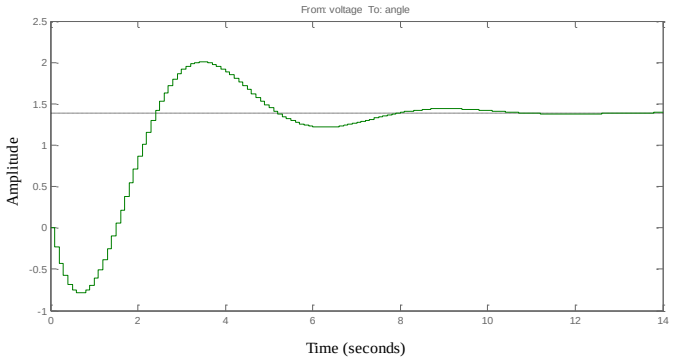


Fig. 32. Estimated model step response for Joint 2

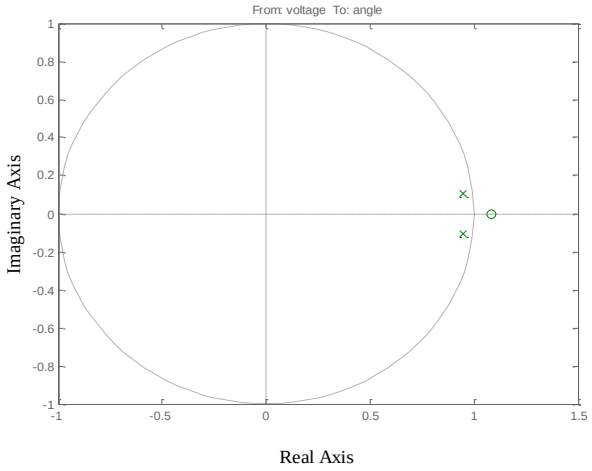


Fig. 33. Estimated model zero and pole output for Joint 2

Figure 34 shows the model output for Joint 3 estimation using HW model. The upper and lower peak close-up for the estimated output can be shown as in Figure 35. Through numerous of estimation trials, the highest best fits for Joint 3 point to 71.35% by using Adaptive Gauss-Newton (gna) algorithm with low Final prediction error (FPE) = 0.1271 and loss function = 0.1228. This model is obtained with linear order of $b_n = 3, f_n = 1, k_n = 3$, estimated with saturation as the input nonlinearity and piecewise linear as the output nonlinearity. As the validation data shows 71.35% best fits, it can be concluded that the estimated model is almost tracking the real output data from the experiments. As a result from the linearization, the state-space model and the transfer function representing the estimated model for joint 3 in the form of discrete-time function can be specified as in Eq. (10) and Eq. (11). Figure 36 shows the step response result from the SI toolbox shows a good oscillatory reaction to the final steady value for the estimated model. Whereas, Figure 37 proved that the estimated model system is stable since all the poles of the transfer function lies within the unit circle of the z-plane.

Joint 3 discrete-time state-space model:

$$a = \begin{bmatrix} 0.9848 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

$$c = \begin{bmatrix} 0 & 0 & -7.138e^{-10} & 1.642e^{-9} & -7.958e^{-10} \end{bmatrix}$$

$$d = \begin{bmatrix} 0 \end{bmatrix}$$

Joint 3 discrete-time transfer function:

$$tf = \frac{-1.428e^{-9}z^2 + 3.285e^{-9}z - 1.592e^{-9}}{z^5 - 0.9848z^4} \quad (11)$$

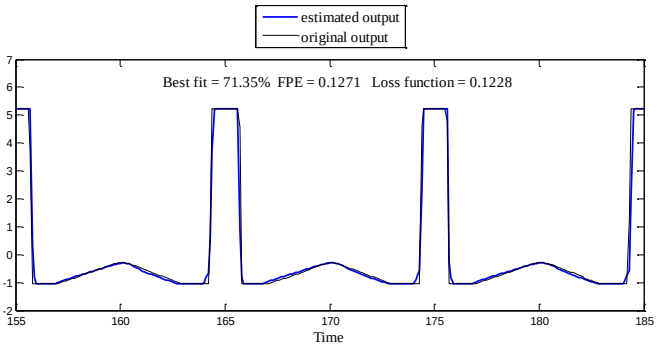
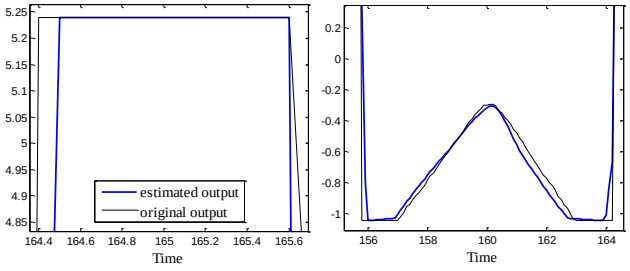


Fig. 34. Joint 3 model output, original output (black) and estimated output (blue) for the same input signal



(a) (b)

Fig. 35. Joint 3 model output close-up; (a) upper peak (b) lower peak

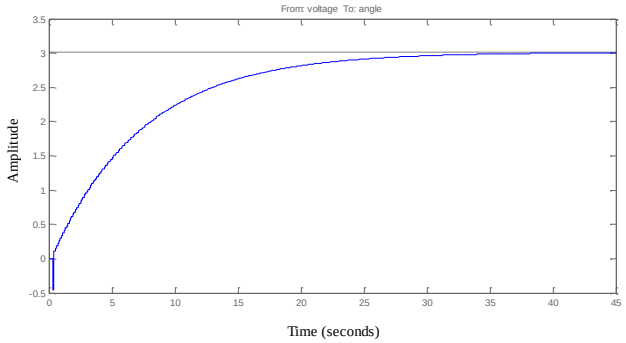


Fig. 36. Estimated model step response for Joint 3

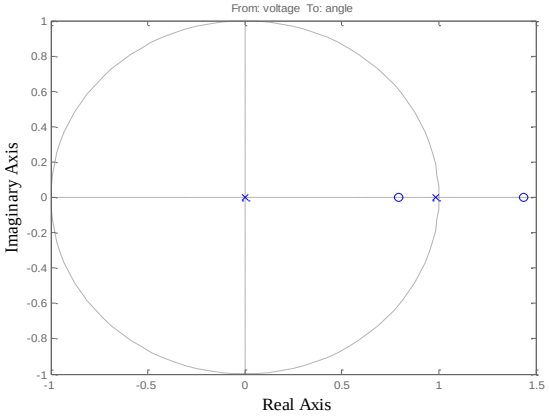


Fig. 37. Estimated model zero and pole output for Joint 3

E. Residual Analysis Results

Figure 38 to 40 shows the autocorrelation of output (angle) residuals along with the cross-correlation between input (voltage) and output (angle) residuals. In order to obtain the residual analysis for these three models the confidence interval (dashed lines) for this estimation is set to 100%. The top axis shows the autocorrelation of residuals for the output (whiteness test) and the bottom axis shows the cross-correlation between input and output residual (independence test). The horizontal scale (the number of lags) is the time difference (in samples) between the signals at which the correlation is estimated[17, 22]. The whiteness test for all three models indicates that the output residuals are uncorrelated since they fall within the confidence interval. Moreover, the independence test also shows that there is no correlation between the input and the output residual as they also fall within the confidence interval. Therefore, both tests proved that the all three models are good and acceptable.

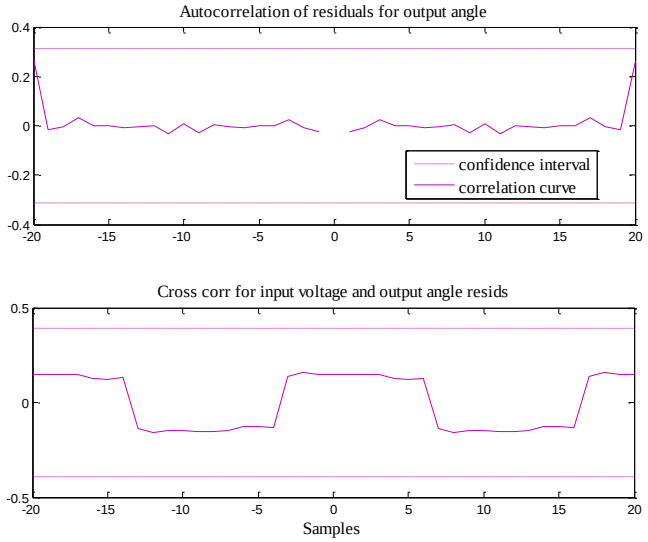


Fig. 38. Autocorrelation of residual and cross correlation analysis of estimated model for Joint 1 model

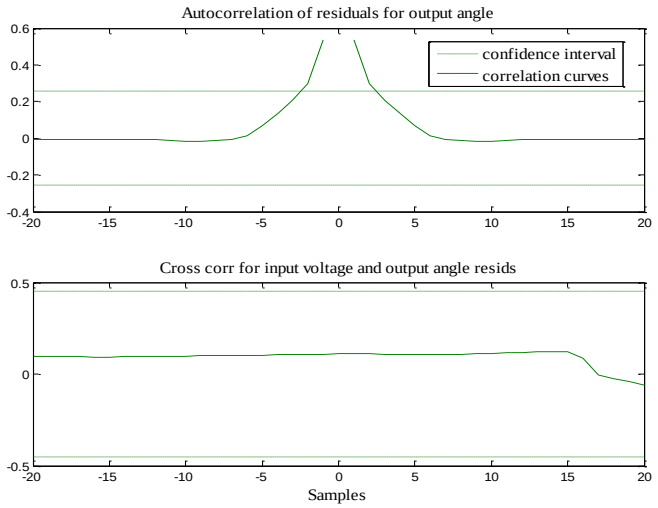


Fig. 39. Autocorrelation of residual and cross correlation analysis of estimated model for Joint 2 model

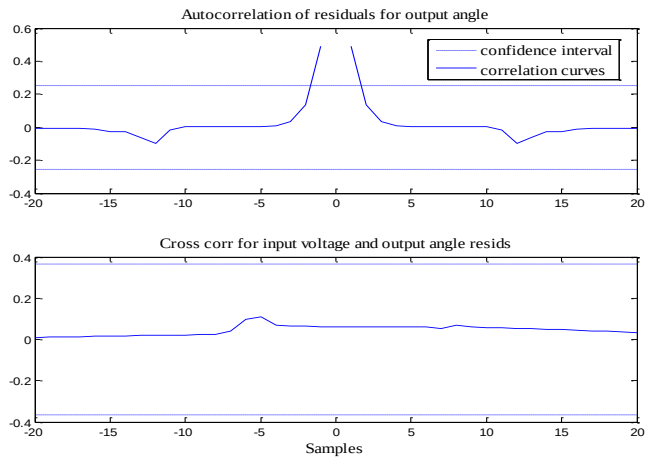


Fig. 40. Autocorrelation of residual and cross correlation analysis of estimated model for Joint 3 model

V. Conclusion

As for conclusion the system identification technique successfully applied to each joints of QPAL leg system to obtain the best representation of mathematical model for the system. With reference to all the candidate structure models studied, HW models represent the behavior for each joints of QPAL leg system quite well. HW model offers a greater reproduction of the actual data on the entire analyzed period. Another interesting feature of the HW model lies in its simplicity to consider the nonlinearities and the possibility to add new nonlinearities, and also its easy execution. These good results show that the black-box model can easily estimate the joints system of QPAL leg.

There are seven nonlinearity estimators for both input and output have been studied throughout this project, which are the piecewise linear, sigmoid network, wavelet network, saturation, dead zone, one-dimensional polynomial, and unit gain. As well as six search methods that is used for this project, which are the auto, Gauss-Newton, Adaptive Gauss-Newton, Levenberg-Marquardt, Trust-Region Reflective Newton and Gradient Search. Through the trial and error methods by varying the nonlinearity estimators, search methods and the linear order, the HW model will result in the highest best fits percentage with low prediction errors and loss function.

All three models are considered well and highly applicable since they pass the whiteness test and the independence test. The stability of the estimated model is proved as the poles for all the models lies within the unit circle of the z-plane. Also, the step response results show that the models have a good oscillatory reaction to the final steady state value. It can be concluded that, the balances 28.94%, 20.86%, and 28.65% are losses due to nonlinear factor such as friction, backlash, torque, and other external disturbance. The best fits percentage can give better percentage if the nonlinear factors are also considered. Meanwhile, the search method, nonlinearity estimators and the linear order involved in this project can be further studied to improve the results.

The mathematical models obtained from this project can contribute to the method for development and implementation of other controller for QPAL leg system. Thus, this project will provide greater opportunities for future work such as development of robust controllers, validation process, and comparing result with real system. Future research includes the identification for the other leg of QPAL robot and the development of robust controller for QPAL leg system also can be done by using the mathematical model obtained from this project to provide sophisticated control system for QPAL robot.

VI. Acknowledgment

The research work is fully supported by Universiti Malaysia Pahang (UMP) Research Grant under the Research and Innovation Department, RDU RDU100378.

VII. References

[1] A. B. Rehiara, "System Identification Solution for Developing an AdeptThree Robot Arm Model."

[2] J.-P. Merlet, *Parallel robots* vol. 74: Springer Science & Business Media, 2012.
[3] W. L. V. Pollard, "The first spatial industrial parallel robot ", 1942.
[4] Y. Patel and P. George, "Parallel manipulators applications—a survey," *Modern Mechanical Engineering*, vol. 2, p. 57, 2012.
[5] K.-M. Lee and D. K. Shah, "Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator," *Robotics and Automation, IEEE Journal of*, vol. 4, pp. 361-367, 1988.
[6] I. Bonev, "The True Origins of Parallel Robots," 2015.
[7] M. Bergerman and Y. Xu, "Robust joint and Cartesian control of underactuated manipulators," *Journal of dynamic systems, measurement, and control*, vol. 118, pp. 557-565, 1996.
[8] P. La Hera, "Underactuated mechanical systems: Contributions to trajectory planning, analysis, and control," 2011.
[9] Y.-L. Gu and Y. Xu, "Under-actuated robot systems: dynamic interaction and adaptive control," in *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference on*, 1994, pp. 958-963.
[10] G. Horváth, "Neural networks in system identification," *Nato Science Series Sub Series III Computer And Systems Sciences*, vol. 185, pp. 43-78, 2003.
[11] M. N. M. Hussain, A. M. Omar, P. Saidin, A. A. A. Samat, and Z. Hussain, "Identification of Hammerstein-Weiner System for Normal and Shading Operation of Photovoltaic System," *International Journal of Machine Learning and Computing*, vol. 2, p. 239, 2012.
[12] L. Ljung, "System Identification ToolboxT M User's Guide R2014b. The MathWorks," *Inc. Natick, MA*, 2014.
[13] R. Abbasi-Asl, R. Khorsandi, and B. Vosooghi-Vahdat, "Hammerstein-Wiener Model: A New Approach to the Estimation of Formal Neural Information," *Basic and Clinical Neuroscience*, vol. 3, pp. 45-51, 2012.
[14] (2015). *Identifying Hammerstein-Wiener Models - MATLAB® & Simulink*. Available: http://www.mathworks.com/help/ident/ug/identifying-hammerstein-wiener-models.html#bq4noaq
[15] H. L. Jiun, "Development of quadruped robot with parallel actuation leg configuration (framework and control)," Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, 2014.
[16] (2015). *Features - System Identification Toolbox*. Available: http://www.mathworks.com/products/sysid/features.html#model-identification-from-data
[17] L. Ljung, "System Identification ToolboxT M User's Guide R2015b. The MathWorks," *Inc. Natick, MA*, 2015.
[18] (2015). *Linear Approximation of Nonlinear Black-Box Models - MATLAB® & Simulink*. Available: http://www.mathworks.com/help/ident/ug/linear-approximation-of-nonlinear-black-box-models.html#brjumso
[19] (2015). *Compute operating point for Nonlinear ARX model - MATLAB® idnlarx/findop*. Available: http://www.mathworks.com/help/ident/ref/idnlarxfindop.html
[20] (2015). *Linearize Hammerstein-Wiener model - MATLAB® idnlhw/linearize*. Available: http://www.mathworks.com/help/ident/ref/idnlhwlinearize.html
[21] (2015). *Create transfer function model, convert to transfer function model - MATLAB® tf*. Available: http://www.mathworks.com/help/control/ref/tf.html#zmw57dd0e73123
[22] D. Araromi and B. Adegbola, "System Identification of Nylon-6 Caprolactam Polymerization Process," 2011.