Praise Worthy Prize

International Review of Electrical Engineering (I.R.E.E.), Vol. 6, N. 2
March-April 2011

Hammerstein Based Modeling of Traveling Wave Ultrasonic Motors

Hamed Mojallali, Mohamadreza Ahmadi

Abstract – *Traveling Wave Ultrasonic Motors (TWUSMs) possess extreme nonlinear properties such as dead-zone and saturation reverse effect, which are reliant on the driving conditions. These characteristics make modeling and control of TWUSMs quite problematic. In this paper, a new scheme for the identification of TWUSM's Hammerstein model consisting of a nonlinear static function followed by a linear dynamical model is introduced. The nonlinear static function is identified using the Bezier–Bernstein polynomial functions. The identification method is based on a hybrid scheme including the inverse de Casteljau algorithm, the least squares method, and the Levenberg-Marquart (LM) algorithm. Simulation results and their validation with the data derived from experiments demonstrate the efficiency of the proposed scheme.*

Keywords: *Travelling Wave Ultrasonic Motor, Modeling, Bezier-Bernstein Polynomial Functions, Hammerstein Model, Levenberg-Marquart Algorithm*

Nomenclature

$\varphi(t)$	Phase difference of applied voltage
$\omega(t)$	Angular velocity of rotor
$y(t)$	Hammerstein model output
$\eta(t)$	Gaussian random noise with zero mean and variance σ^2
$u(t)$	Hammerstein model input
$V(t)$	Nonlinear subsystem output in Hammerstein model
n_a	Input lag for linear subsystem of Hammerstein model
n_b	Output lag for linear subsystem of Hammerstein model
k_m	Steady state gain of TWUSM dynamical behavior
τ	Time constant of TWUSM dynamical behavior
$B_j^d(\cdot)$	d th order Bezier-Bernstein basis function
U_j	Horizontal coordinate
V_j	Vertical coordinate
P_j	Predetermined control points (knots)
$\hat{\omega}(t)$	Estimated angular velocity of rotor
J	Jacobian Matrix
μ	Damping parameter of the LM algorithm
h_{lm}	Step of the LM algorithm
h_{gn}	Step of the Guass-Newton Algorithm
λ	Gain ratio of the LM algorithm

I. Introduction

Ultrasonic motors (USMs) are a type of actuators and electromechanical devices which exhibit several particular excellent performance features such as high holding torque, high torque at low speed, silent operation, simple structure, compact size and no electromagnetic interference [1]. Due to such characteristics, USMs have been widely used in numerous practical applications, e.g., in robots, medical instruments, cameras, aeronautics, MEMS and many others [2]. Several types of piezoelectric ultrasonic motors (PEUM) have been suggested and designed. The large family of USMs is mostly categorized by two distinct methods. USMs are distinguished based on their functionality into two major groups: linear and rotary motors. Another classification approach is established upon the type of the propagating wave in the stator of USMs. With this classification, standing wave and traveling wave USMs can be identified. The advantages and various applications of traveling wave ultrasonic motors (TWUSMs), compared to other types of USMs, have attracted more research attention to them [2].

The modeling of Motors has been subject to several research activities and investigations [3]-[8]. Existing modeling methods for TWUSMs are mostly accomplished on the basis of equivalent circuit methods [1], [9]–[10], analytical approaches [11]–[13] and finite element modeling (FEM) [14]. Recently, black box models for TWUSMs based on Hammerstein structure were also suggested [15], [16]. However, there are a number of serious drawbacks to these models. The method put forward in [16] is established upon "Try and Error". The proposed algorithm in [15] depends on amplitude as the input for modeling, which is not a

Manuscript received and revised March 2011, accepted April 2011

conventional method for controlling TWUSMs. Moreover, both methods lack computational simplicity, they are significantly time consuming, and deficient in having a uniform formula for backward and forward rotation directions of TWUSM.

The Hammerstein model is composed of a nonlinear static memoryless subsystem which is in series with a linear dynamic block. Thus far, the Hammerstein model has received major attention in modeling a myriad of nonlinear systems including chemical processes, DC/DC convertors, electrically stimulated muscles, actuators, RF transmitters, stretch reflexes and etc. [17]-[24]. As a consequence of this wide variety of applications, different identification algorithms for Hammerstein models are vastly addressed in literature [25]. One avenue that the researchers have followed in their quest to identify Hammerstein systems is by the means of Bezier-Bernstein approximation. Hong and Mitchel [26] were the first to propose such identification algorithm for SISO Hammerstein systems. Albeit their approach was successful in identifying numerical examples, it suffered from several weak points which made it almost impossible to be applied for identifying real world nonlinear processes. In the first place, the algorithm only considered the delayed versioned of the inputs to the nonlinear subsystem, whereas in most practical cases the inputs to the nonlinear blocks are not delayed. A more subtle disadvantage is that the Gauss-Newton algorithm used to estimate the nonlinear coefficients is, up to a point, slow, inaccurate, and quite sensitive to the choice of initiating knots.

In the proposed scheme in this paper, the Hammerstein model is considered in a rather comprehensive mode, that is, both the delayed and un-delayed versions of the input appear in the system model. Additionally, the Levenberg-Marquart algorithm is utilized for estimating a mixture of the nonlinear and linear parameters. The LM algorithm has been previously applied in many nonlinear least squares problems ranging from applications in nuclear to biomedical engineering [22]-[27].

The aim of this paper is to introduce a new Hammerstein based model for TWUSM. The chief virtue of the proposed scheme is that given an input/output data set from a TWUSM, the model can be readily identified. Whereas, other identification methods for TWUSM require partial or complete modification when the system is changed or the conditions are altered. The nonlinear subsystem in Hammerstein model is parameterized by a Bezier curve, which is a linear combination of a set of Bernstein basis functions. These Bernstein basis functions are fabricated over the input data by exploiting the inverse de Casteljau algorithm [28]. The remaining parameters in the model are approximated using the least squares algorithm and the Levenberg-Marquart (LM) algorithm subject to constraints.

The balance of this paper proceeds as follows. In the subsequent section, a brief discussion on the operating principles of the TWUSM is provided and then the Hammerstein model for TWUSM is outlined. The proposed identification algorithm is discussed in section III. Section IV considers the descriptions of the implemented experimental set-up. Simulation results and their verification with data from experiments are presented in section V. The paper ends with conclusions in section VI.

II. Traveling Wave Ultrasonic Motor

II.1. Operating Principle

Fig. 1 shows the cutaway view of a typical USR60 traveling wave ultrasonic motor, as discussed in this paper. The motor consists of two basic parts: the stator vibrates with a frequency in the ultrasonic range, and the rotor is driven by the stator via frictional forces. Stator is composed of an elastic body and a thin piezo-ceramic ring which is bonded under the elastic body. The piezo-ceramic ring has the function of exciting traveling bending waves and is shown in Fig. 2.

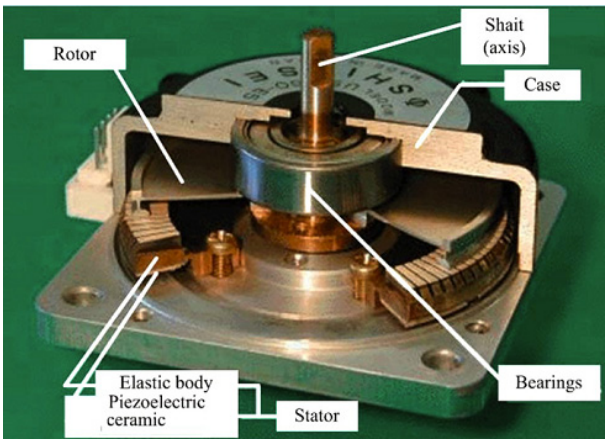


Fig. 1. Cutaway view of Shinsei USR60 TWUSM

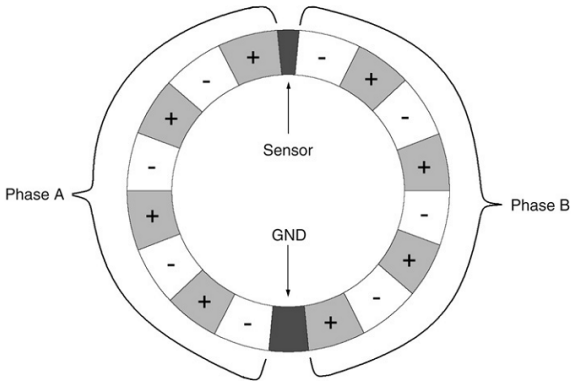


Fig. 2. The piezoceramic ring of the experimental ultrasonic motor

The ring is divided into two halves: phase A and phase B. These two phases are separated by a sensor and ground parts which are a quarter and 3 quarters of a wavelength, respectively. Each phase (A or B) includes n

segments. Each segment is a half wavelength and polarized adversely with respect to the adjacent one. Phase A and phase B are a quarter of the wavelength out of phase, spatially. The phases are excited by two sinusoidal voltages which are temporally 90◦ out of phase. Therefore, a traveling wave is generated and the particles of the stator surface move elliptically.

The sensor section is used for measuring the amplitude and the phase of the traveling wave which is used to control the excitation of the piezo-ceramic ring. The rotor is pressed against the stator by a disk spring, and a thin contact layer is bonded to the rotor in the contact region. Therefore, the vibration of the stator with high frequency and small amplitude is rectified into the lower frequency macroscopic rotary motion of the rotor by friction. The speed of the TWUSM can be controlled by the frequency of the two-phase voltages, the amplitude of the two-phase voltages and the phase difference between the two-phase voltages. More information about the working principles of traveling wave ultrasonic motors can be found in [1].

II.2. Hammerstein Model Structure

Generally, the Single-Input Single-Output (SISO) Hammerstein model can be defined by:

$$\begin{aligned} y(t) &= -a_1 y(t-1) - a_2 y(t-2) + \\ &- \cdots - a_{n_a} y(t-n_a) + b_0 V(u(t)) + \\ &+ b_1 V(u(t-1)) + \cdots + b_{n_b} V(u(t-n_b)) + \eta(t) \end{aligned} \quad (1)$$

The gain of the linear subsystem is thus given by:

$$G = \frac{\sum_{i=0}^{n_b} b_i}{1 + \sum_{j=1}^{n_a} a_j} \quad (2)$$

In case of TWUSMs, the proposed Hammerstein model is modified according to the motor driving characteristics. Phase difference is considered as the input to the Hammerstein model, inasmuch as it is a commonplace control input for TWUSM. Not only can phase difference be applied for rotation in both directions, but also it can be used to modify motor speed. Besides, the rotor speed is suggested as the output of the Hammerstein model. The Hammerstein system for TWUSM can be modeled by:

$$\omega(t) = -\sum_{i=1}^{n_a} a_i \omega(t-i) + \sum_{k=0}^{n_b} b_k v(\varphi(t-k)) + \eta(t) \quad (3)$$

The intermediate signal $v(t)$ is the output of the nonlinear subsystem which cannot be measured. Given an observational (phase difference /motor speed) data set $\{\varphi(t), \omega(t)\}_{t=1}^{N}$, where $t \in (1, N)$ denotes the sampled times, the goal is to identify the nonlinear gain function and subsequently the Hammerstein model.

TWUSM exposes a dynamical behavior that can be explained by the following transfer function [29], [30]:

$$\frac{\omega(s)}{V(s)} = \frac{k_m}{\tau s + 1} \quad (4)$$

The typical values of k_m and τ for USR60 are 10.25 and 0.0035 sec, respectively. It is assumed that k_m is included in the nonlinear gain function; thus, the gain of the linear subsystem could be maintained as one. The Hammerstein model for TWUSM is illustrated in Fig. 3.

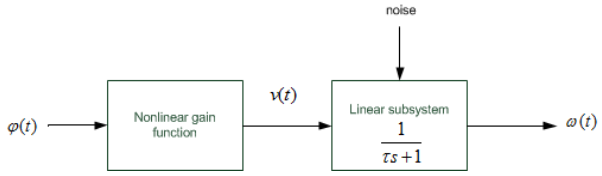


Fig. 3. The Hammerstein model structure for TWUSM

III. The Identification Algorithm

III.1. Modeling of the Nonlinear Gain Function Using Bezier-Bernestein Polynomials

Bezier curve is a parametric curve characterized by Bernstein basis functions [31]-[33]. With a set of preset two dimensional control points, the Bezier curve can be readily constructed through the de Casteljau algorithm. The univariate Bernstein polynomial basis functions $B_j^d(x)$ are the expansion of $[x + (1-x)]^d$ [31]-[37], described by:

$$B_j^d(x) = \binom{d}{j} x^j (1-x)^{d-j} \quad (5)$$

where j and d are nonnegative integers satisfying $j \le d$ over the region $x \in [0,1]$. The total number of the univariate d^{th} order Bernstein polynomials is $d+1$. It has been shown that Bernstein polynomials can be computed using the recursion given below [32]:

$$B_j^d(x) = (1-x) B_j^{d-1}(x) + x B_{j-1}^{d-1}(x) \quad (6)$$

$\psi(*)$, which is an arbitrary nonlinear function, can be approximated using the Bezier-Bernstein polynomials as:

$$\psi(u(t)) = \sum_{j=0}^{d} B_j^d(x(u(t))) \delta_j \quad (7)$$

where δ_j 's are the weights to be determined from the input-output data, $u(t) \in \Re^1$ is the input sample at the time t, and $B_j^d\left(x\left(u(t)\right)\right)$, $j = 0,1,\ldots,d$ denote the corresponding dth order Bezier-Bernstein basis functions. To determine these basis functions, one has to perform the one-to-one mapping:

$$\Omega(*) : u(t) \in \left[min\left(u(t)\right), max\left(u(t)\right)\right] \to x\left(u(t)\right) \in (0,1)$$

The univariate de Casteljau algorithm, which is widely implemented in Bezier curve construction, can realize the mapping:

$$\Omega^{-1}(*) : x \in (0,1) \to u(x) \in \left[min\left(u(t)\right), max\left(u(t)\right)\right]$$

The de Caseljau algorithm is described next.

With a set of predetermined control points (knots) in 2-Dimensional space $P_j = \left[U_j, V_j\right]^T \in \Re^2$, $j = 0,1,\ldots,d$, the de Casteljau algorithm is a recursion defined over these control points represented as [32]:

$$P_j^{(r)}(x) = (1-x)P_j^{(r-1)}(x) + xP_{j+1}^{(r-1)}(x) \qquad (8)$$

where:

$$P_j^{(0)}(x) = P_j,\ r = 1,2,\ldots,d,\ j = 0,\ldots,d-r \text{ and}$$
$$P_j^{(r)}(x) = \left[U_j^{(r)}, V_j^{(r)}\right]^T$$

It can be concluded that the main idea of the de Casteljau algorithm is upon recursively subdividing a curve and adding points to refine the knots. Note that if only the information about one of the coordinates is available, still recursion (8) is valid and defined as:

$$U_j^{(r)}(x) = (1-x)U_j^{(r-1)}(x) + xU_{j+1}^{(r-1)}(x) \qquad (9)$$

However, as stated earlier, the inverse procedure of the above algorithm is required to accomplish the mapping from $u(t) \in \left[min\left(u(t)\right), max\left(u(t)\right)\right]$ to $x\left(u(t)\right) \in (0,1)$, which is known as the inverse de Casteljau algorithm. Previously, the inverse of de Casteljau algorithm has been proposed through iterative error feedback [27].

Considering the phase difference $\varphi(t)$ as the input, the inverse de Casteljau's algorithm is applied to map each input data to $x \in [0,1]$, so that $x\left(\varphi(t)\right)_{t=1}^N$ can be utilized to fabricate the Bernstein polynomial basis functions $B_j^d\left(x\left(\varphi(t)\right)\right)_{t=1}^N$.

The algorithm is described as follows [27]:

given a desired mapping point $\varphi(t)$, a set of knots $\Phi_j \in \left[\Phi_0, \Phi_d\right] \in \left[min(\varphi), max(\varphi)\right]$, $j = 0,\ldots,d$, are preset. Denote the iteration step in the following procedure as n

A. Initially, set $n = 1$, and x as a random number in the interval $(0,1)$.

B. Calculate the corresponding first component of Bezier curve points, $\Phi_j^{(r)}$, using the de Casteljau recursive formula:

$$\Phi_j^{(r)}(x) = (1-x)\Phi_j^{(r-1)}(x) + x\Phi_{j+1}^{(r-1)}(x) \qquad (10)$$

until $r = d$, producing an intermediate approximated point $\hat{\varphi}\left(x^{(n)}\right)$:

$$\hat{\varphi}\left(x^{(n)}\right) = \left(1-x^{(n)}\right)\Phi_0^{(d-1)}\left(x^{(n)}\right) + x^{(n)}\Phi_1^{(d-1)}\left(x^{(n)}\right) \qquad (11)$$

C. The difference between the desired point φ and the estimated point $\hat{\varphi}\left(x^{(n)}\right)$ is used to adjust the search direction of x. A new point is created as

$$\tilde{\varphi}\left(x^{(n)}\right) = \hat{\varphi}\left(x^{(n)}\right) + \gamma\left[\varphi - \hat{\varphi}\left(x^{(n)}\right)\right] \qquad (12)$$

where γ, the learning rate, is a very small positive integer ($0 \le \gamma \le 1$).

D. The desired solution of x at iteration step $(n+1)$ is computed such that $\tilde{\varphi}(x^{(n)})$ is the first order Bezier point with respect to the two end knots $\Phi_0^{(d-1)}(x)$ and $\Phi_1^{(d-1)}(x)$. The solution is therefore given by:

$$x^{(n+1)} = \frac{\tilde{\varphi}\left(x^{(n)}\right) - \Phi_0^{(d-1)}\left(x^{(n)}\right)}{\Phi_1^{(d-1)}\left(x^{(n)}\right) - \Phi_0^{(d-1)}\left(x^{(n)}\right)} \qquad (13)$$

The procedure continues until $\left\|\varphi - \tilde{\varphi}\left(x^{(n)}\right)\right\| \le \varepsilon$, where ε is an arbitrarily small positive number close to zero. If $\left\|\varphi - \tilde{\varphi}\left(x^{(n)}\right)\right\| \le \varepsilon$ set $x = x^{(n)}$. Otherwise, set $n = n+1$ and go to step 2.

Once the input data is mapped to $(0,1)$, one can fabricate the Bernstein basis functions $B_j^d\left(x\left(\varphi(t)\right)\right)_{t=1}^{N}$ according to the initially chosen knots Φ_j, $j=0,\ldots,d$. Subsequently, the estimated nonlinear function $\hat{v}_d(t)$ can be explained by:

$$\hat{v}_d(t)=\sum_{j=0}^{d}B_j^d\left(x\left(\varphi(t)\right)\right)\delta_j \tag{14}$$

However, δ_j, $j=0,\ldots,d$ still has to be calculated. The approximated output $\omega(t)$ can be expressed by substituting equation (14) in (3):

$$\begin{aligned}\hat{\omega}(t)=&-\sum_{j=1}^{n_a}a_i\omega(t-i)+b_0\sum_{j=0}^{d}\delta_j B_j^d\left(x\left(\varphi(t)\right)\right)+\\&+\ldots+b_{n_b}\sum_{j=0}^{d}\delta_j B_j^d\left(x\left(\varphi(t-n_b)\right)\right)\end{aligned} \tag{15}$$

Denote:

$$\hat{\omega}\left(-\omega(t-1),\ldots,-\omega(t-n_a),\varphi(t),\ldots,\varphi(t-n_b),\hat{a},\hat{b},\hat{\delta}\right)$$

where $\hat{\omega}$ is the estimated Hammerstein model output, $\hat{a}=\left[a_1,\ldots,a_{n_a}\right]$, $\hat{b}=\left[b_1,\ldots,b_{n_b}\right]$, and $\hat{\delta}=\left[\delta_0,\ldots,\delta_d\right]$.

The approximation accuracy can be increased as the sum of square error is minimized:

$$SSE=\sum_{t=1}^{N}\left[\omega(t)-\hat{\omega}(t)\right]^2 \tag{16}$$

With this choice of cost function at hand, the parameter identification challenge is retranslated to a nonlinear least squares problem. It is obvious that $\hat{\omega}$ is not linear to $\hat{b}$ and $\hat{\delta}$ yet it can be translated as a linear regression from $\omega(t)$ in vector $\hat{a}$. The proposed parameter estimation procedure is performed in two stages: (a) based on the estimated resultant model structure using the Bezier–Bernstein polynomial functions, the least squares algorithm is applied to estimate the parameters in the autoregressive (AR) part of the linear subsystem; (b) the remaining parameters are approximated using the LM algorithm subject to the constraint of unit gain [38].

III.2. Determining the Parameters $\hat{a}_j$ Using the Least Squares Method

Eq. (3) can be rewritten as matrixes in regression form:

$$\omega=W\Theta+H \tag{17}$$

with:

$$W=\begin{bmatrix}w\left(F(1)\right)\\w\left(F(2)\right)\\\ldots\\w\left(F(N)\right)\end{bmatrix} \tag{18a}$$

where:

$$\begin{aligned}&w\left(F(t)\right)=\\&=\left[-\omega(t-1),\ldots,-\omega(t-n_a),B_0^d\left(x\left(\varphi(t)\right)\right),\right.\\&\left.\ldots,B_0^d\left(x\left(\varphi(t-n_b)\right)\right),\ldots,B_d^d\left(x\left(\varphi(t-n_b)\right)\right)\right]\\&\in\Re^{n_a+(d+1)(n_b+1)}\end{aligned} \tag{18b}$$

and:

$$\begin{aligned}\Theta=&\left[\hat{a}^T,(b_0\delta_0),\ldots,(b_0\delta_d),\ldots,\left(b_{n_b}\delta_0\right),\right.\\&\left.\ldots,\left(b_{n_b}\delta_d\right)\right]^T\in\Re^{n_a+(d+1)(n_b+1)}\end{aligned} \tag{18c}$$

$H=\left[\eta(1),\ldots,\eta(N)\right]^T$ is the presumed Gaussian noise associated with the system. The least squares solution of Θ is:

$$\hat{\Theta}=\left[W^TW\right]^{-1}W^T\omega \tag{19}$$

from (19) one can simply derive $\hat{a}$, which is a sub vector of $\hat{\Theta}$. At this point, a sequence $r(t)$ as a collateral model output, based on the least squares solutions of $\hat{a}$, and the output data is generated:

$$r(t)=\omega(t)+\hat{a}_1\omega(t-1)+\cdots+\hat{a}_n\omega(t-n_a) \tag{20}$$

However, the intermediate model output $r(t)$, still needs to be interpreted by the unknown parameters b and δ. The approximated version of $r(t)$ based on Bezier-Bernstein Basis functions is:

$$\begin{aligned}\hat{r}(t)=&b_0\sum_{j=0}^{d}\delta_j B_j^d\left(x\left(\varphi(t)\right)\right)+\\&+\cdots+b_{n_b}\sum_{j=0}^{d}\delta_j B_j^d\left(x\left(\varphi(t-n_a)\right)\right)\end{aligned} \tag{21}$$

Proposition 1: It is postulated that W is nonsingular. Consequently, the minimization of $\sum_{t=1}^{N}\left[\omega(t)-\hat{\omega}(t)\right]^2$ is

identical to that of $\sum_{t=1}^{N}\left[r(t)-\hat{r}(t)\right]^2$. The proof for proposition 1 is available in [26].

III.3. Method Applying the Levenberg-Marquart Algorithm to approximate b and δ

LM algorithm is a modified version of the Gauss-Newton algorithm which converges to an optimal solution through an iterative procedure. A chief disadvantage of the Gauss-Newton algorithm which arose in many practical cases is that the Jacobian matrix turns out to be singular; therefore, a valid inverse matrix could not be obtained. Since the cost function is in the quadratic form, it is convenient to employ the LM algorithm.

The LM algorithm introduces a new multi-functional parameter μ called the damping parameter. The LM step is defined as follows [33]-[35]:

$$h_{lm}=\left\{J^TJ+\mu I\right\}^{-1}J^TF \tag{22}$$

where I is a unit matrix with the same size as J^TJ, and F is the function to be minimized.
Including the damping parameter leads to several encouraging outcomes [41]:

a. For all positive values of the damping parameter the coefficient matrix is positive definite, and consequently h_{lm} is decent direction.
b. For large values of μ we have

$$h_{lm}=-\frac{1}{\mu}J^TF=-\frac{1}{\mu}F' \tag{23}$$

which is a small step in the steep decent direction. A desirable feature, if the current intermediate solution is far from the answer.

c. For small values of the LM step, $h_{lm}=h_{gn}$. This happens in the final steps when the algorithm is converging.

It can be comprehended from the above corollaries that the damping parameter is able to influence both the direction and the size of a step. So, a nonlinear optimization algorithm without a specific line search can be achieved.

Another important concept in the LM algorithm is the gain ratio λ. Gain ratio is also commonly used in trust region algorithms such as Powell's dog leg method [38], [41]. λ is defined as the ratio between the actual and predicted decrease in function value, and it can be computed using the following equation:

$$\lambda=\frac{F(x)-F(x+h)}{L(0)-L(h)} \tag{24}$$

where the denominator is the gain predicted by the linear model. In the course of the LM algorithm, the damping parameter is increased as the gain ratio becomes small, since $L(h)$ is a poor estimation of $F(x+h)$. On the other hand, a large value of the gain ratio indicates a good approximation; therefore, the damping parameter is decreased. The gain predicted by the linear model is calculated as:

$$\begin{aligned}&L(0)-L(h_{lm})=-h_{lm}^TJ^TF(x)-0.5h_{lm}^TJ^Th_{lm}=\\&=-0.5h_{lm}^T\left(2J^TF(x)-\left(J^TJ+\mu I-\mu I\right)h_{lm}\right)=\\&=0.5h_{lm}^T\left(\mu h_{lm}-J^TF(x)\right)\end{aligned} \tag{25}$$

One should note that both $-h_{lm}^TJF(x)$ and $h_{lm}^Th_{lm}$ are positive; thus, $L(0)-L(h_{lm})$ is also positive and nonzero. A comprehensive discussion of the LM algorithm can be found in [41]. Note that the constraint of unit gain resulting from the special structure of the Hammerstein model should be taken into account so an enhanced algorithm could be achieved. In other words, the algorithm would avoid being trapped in local minima [26], [38]. Using (18) and (19) the model residual is represented as $e\left(\hat{b},\hat{\delta},t\right)=r(t)-\hat{r}(t)$.

The Jacobian matrix J with respect to $\left[b^T,\delta^T\right]^T$ is given by:

$$J=\begin{bmatrix}\frac{\partial}{\partial b_0}e\left(\hat{b},\hat{\delta},1\right) & \dots & \frac{\partial}{\partial b_{n_b}}e\left(\hat{b},\hat{\delta},1\right)\\ \frac{\partial}{\partial b_0}e\left(\hat{b},\hat{\delta},2\right) & \dots & \frac{\partial}{\partial b_{n_b}}e\left(\hat{b},\hat{\delta},2\right)\\ \dots & \dots & \dots\\ \frac{\partial}{\partial b_0}e\left(\hat{b},\hat{\delta},N\right) & \dots & \frac{\partial}{\partial b_{n_b}}e\left(\hat{b},\hat{\delta},N\right)\end{bmatrix}\\ \begin{bmatrix}\frac{\partial}{\partial \delta_0}e\left(\hat{b},\hat{\delta},1\right) & \dots & \frac{\partial}{\partial \delta_d}e\left(\hat{b},\hat{\delta},1\right)\\ \frac{\partial}{\partial \delta_0}e\left(\hat{b},\hat{\delta},2\right) & \dots & \frac{\partial}{\partial \delta_d}e\left(\hat{b},\hat{\delta},2\right)\\ \dots & \dots & \dots\\ \frac{\partial}{\partial \delta_0}e\left(\hat{b},\hat{\delta},N\right) & \dots & \frac{\partial}{\partial \delta_d}e\left(\hat{b},\hat{\delta},N\right)\end{bmatrix} \tag{26}$$

where:

$$\begin{aligned}&\frac{\partial}{\partial b_i}e\left(\hat{b},\hat{\delta},t\right)=\sum_{j=0}^{d}B_j^d\left(x\left(\varphi(t-i)\right)\right)\hat{\delta}_j\\&i=0,1,\dots,n_b\end{aligned} \tag{27a}$$

$$\frac{\partial}{\partial \delta_j} e\left(\hat{b},\hat{\delta},t\right) = \sum_{j=0}^{d} B_j^d\left(x\left(\varphi\left(t-i\right)\right)\right)\hat{b}_h \qquad (27b)$$
$$j = 0,\ldots,d$$

The algorithm is described as follows:

I) Set $n=0$, $v=\alpha$, $\mu=\beta$. $\hat{b}$ and $\hat{\delta}$ are generated as random vectors with appropriate dimensions. It is worth noting that the corresponding length of the vectors $\hat{b}$, and $\hat{\delta}$ is n_b+1, and $d+1$, respectively.

II) Apply the Levenberg-Marquart algorithm subject to the normalization constraint in order to maintain the gain of linear subsystem as one.

II.a) LM step

$$A^{(n)} = \left(J^{(n)}\right)^T J^{(n)} \qquad (28a)$$

$$h_{lm} = \left\{A^{(n)} + \mu I\right\}^{-1}\left(J^{(n)}\right)^T e \qquad (28b)$$

$$\begin{bmatrix}\hat{b}^{(n+1)}\\ \hat{\delta}^{(n+1)}\end{bmatrix} = \begin{bmatrix}\hat{b}^{(n)}\\ \hat{\delta}^{(n))}\end{bmatrix} + h_{lm} \qquad (28c)$$

where $e = \left[e\left(\hat{b},\hat{\delta},1\right),\ldots,e\left(\hat{b},\hat{\delta},N\right)\right]^T$.

II.b) Apply the constraint of the unit gain (parameter normalization). Calculate $\hat{G}$ the estimated gain of the linear subsystem:

$$\hat{G}^{(n+1)} = \frac{\sum_{i=0}^{n_b}\hat{b}_i^{(n+1)}}{1+\sum_{j=1}^{n_a} a_j} \qquad (29)$$

Then:

$$\hat{b}^{(n+1)} \leftarrow \frac{\hat{b}^{(n+1)}}{\hat{G}^{(n+1)}}$$
$$\hat{\delta}^{(n+1)} \leftarrow \hat{G}^{(n+1)}\hat{\delta}^{(n+1)} \qquad (30)$$

Subsequently, calculate:

$$\lambda = \frac{\left\|e^{(n)}\right\|^2 - \left\|e^{(n+1)}\right\|^2}{0.5h_{lm}^T\left(\mu^{(n)}h_{lm} - J^T e^{(n)}\right)} \qquad (31)$$

II.c) If $\lambda > 0$ meaning the solution matrix has a better estimation, step is accepted. Then:

$$X = \begin{bmatrix}\hat{b}^{(n+1)}\\ \hat{\delta}^{(n+1)}\end{bmatrix} \qquad (32a)$$

$$A^{(n+1)} = \left(J^{(n+1)}\right)^T J^{(n+1)} \qquad (32b)$$

where X denotes the solution matrix. Afterwards, the parameters and the estimation error matrix are updated

$$\mu^{(n+1)} = \mu^{(n)}\, max\left(1/3, 1-\left(2\lambda-1\right)^3\right) \qquad (33a)$$

$$v = \alpha \qquad (33b)$$

$$e^{(n+1)} = e^{(n)} \qquad (33c)$$

On the other hand, if λ is not positive, the step is not accepted and μ is renewed:

$$\mu^{(n+1)} = \mu^{(n)}\upsilon \qquad (34a)$$

$$\upsilon^{(n+1)} = \alpha^{(n)}\upsilon \qquad (34b)$$

The algorithm is continued at step 2 till the approximation criterion, i.e. $\left\|\sum_{t=1}^{N} e^2(t)\right\| < \xi$, where $0 < \xi << 1$, is satisfied.

A simple flowchart of the overall identification algorithm is depicted in Fig. 4.

IV. Experimental Setup

A testbed was designed to assemble the piezo-electric motor (PEM) and the necessary instruments. A picture of the experimental testbed is given in Fig. 5. On the testbed, the motor is mounted on a shaft which is connected to the sensors. A stop block is boarded on the shaft to prevent the shaft from sliding through the bearing when the normal force is applied to the motor. The stop block is designed such that it only touches the rotating part of the bearing, so no extra friction is added. A support for the shaft between the force gauge and the motor is placed on a carriage which is kept on ball bearings to reduce resistance.

The velocity of the motor is measured using an HEDS-9040 encoder with an HEDS-5140 A11 encoder wheel. The encoder is connected to the 6071EDAQ-card. MATLAB is used to process the data and calculate the exact position of the motor. The encoder and the encoder wheel are located at the right most aluminum block of the experimental testbed.

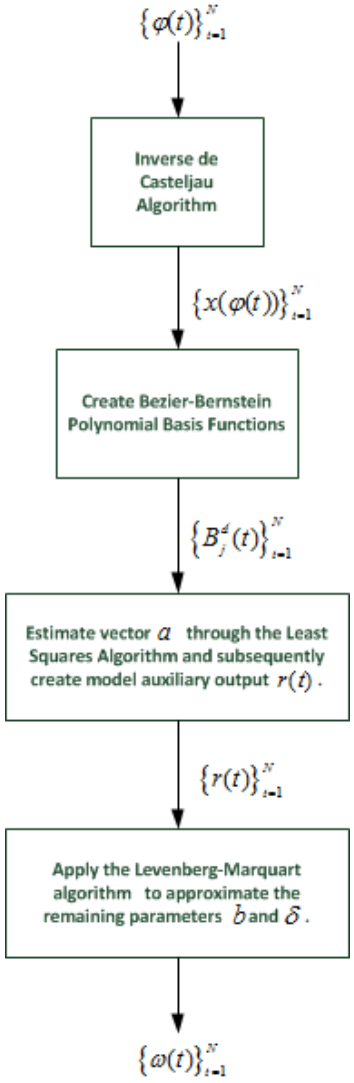


Fig. 4. A simple flowchart of the overall algorithm

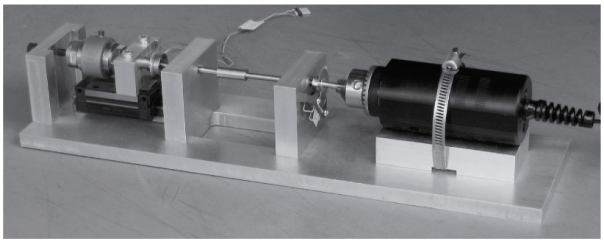


Fig. 5. A picture of the experimental testbed equipped with the sensors

The driver for the encoder wheel is implemented in two functions: an ordinary function which sets up the encoder and starts the measuring, and a call back function which is called every 1000 samples calculating the average speed over these 1000 samples. The speed can then be read through a global variable, which is updated at each run of the call back function. Two sinusoidal excitation signals are necessary to drive the motor. For this purpose, a signal generator board is designed. The board includes a PIC microprocessor, two signal generators, and an analogue circuit. The PIC is controlled from PC software via a RS232 connection. The PIC manages the functionality of the signal generators by sending certain commands to them. The signal generators produce square waves with a frequency determined by the micro-controller. The analog circuit converts the square waves to sinusoidal waves and amplifies it.

By sending proper commands to the micro-controller, the frequency and the phase shift between the signals are adjustable. The amplitude of the output signals is manually adjustable by changing the input voltage of the board. The overall driving and testbed preparations are illustrated in Fig. 6.

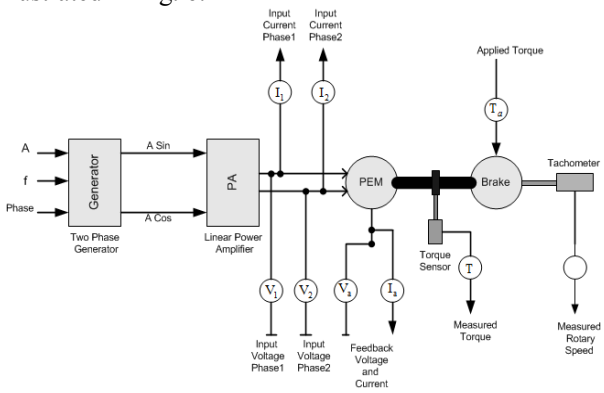


Fig. 6. Schematic drawing of the designed experimental setup

V. Results and Discussions

By discretizing Eq. (4), the input and output lag values (n_a and n_b) for the dynamical subsystem of TWUSM can be simply computed as 1 and 0, respectively [23],[24]. About 40 samples of the input/output data derived from experiments with USR60 are used for comparison with the simulation results of the proposed identification scheme. The polynomial degree of Bezier-Bernstein Basis functions was set to $d=18$, and a set 19 knots were selected as:

$$\{\Phi_k\}=\left\{\begin{bmatrix}-90,-70,-50,-30,-20,-10,-5,\\ -2,-1,0,1,2,5,10,20,30,50,70,90\end{bmatrix}\right\}$$

Note that the proposed scheme is not sensitive to the choice of knots, as long as the knots are selected closer to each other in the most nonlinear region.The inverse de Casteljau algorithm with learning rate $\gamma=0.01$ was applied to map the input data to $(0,1)$. Next, a sequence of regressors $B_j^d\left(x\left(\varphi(t)\right)\right)$, $j=0,1,\ldots,18$ were generated. Consequently, 19 Bezier-Bernstein basis functions were created as illustrated in Fig. 7. It is important to note that these functions are quite dense where most of the nonlinearity exists (here, the most nonlinear part is within $\varphi\in[-20,20]$ around the dead-zone).

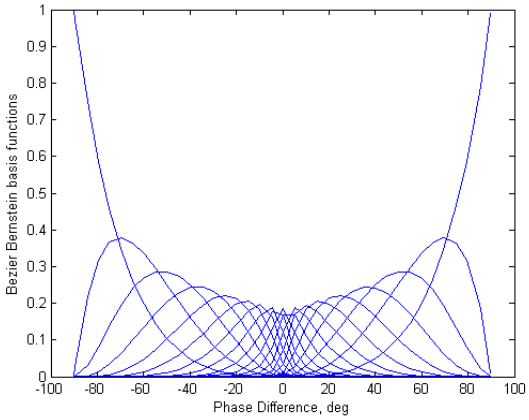
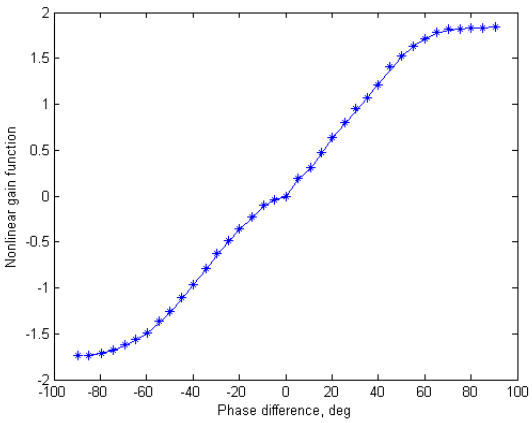


Fig. 7. The resultant Bezier-Bernstein Basis functions fabricated from the 19 preset knots in section 5

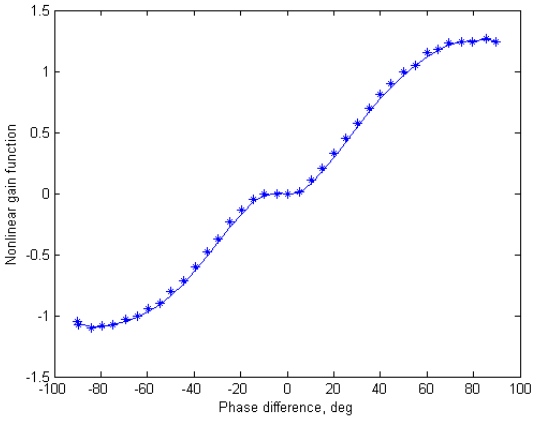
Using the input data, the least squares algorithm as discussed in section III.B was applied to estimate the vector a. The model auxiliary output $r(t)$ is formed on the basis of the estimated values of a.

Henceforth, the task of the identification algorithm is to minimize $\sum_{t=1}^{N} e^2(t)$ through the LM algorithm, that is, $\chi^* = arg\left[min\left\{ \sum_{t=1}^{N} e^2(t), \forall \hat{\chi} \right\} \right]$, subject to $G = 1$ where $\chi = \left[a^T, b^T, \delta^T \right]^T$.

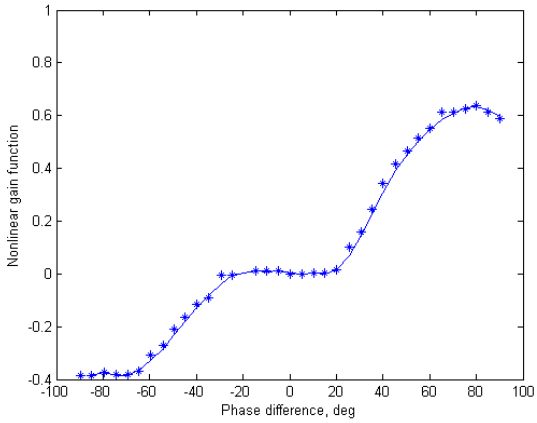
One should note that α should be chosen quite carefully. In conventional nonlinear least squares problems solved using the LM method, the value of α is chosen as 2 [41]. However, this choice of α, fails to appropriately approximate the nonlinear parameters in the proposed identification algorithm in this paper due to the number of parameters to be tuned and the abundance of local minimums. A large value of α results in fast convergence, the algorithm would get trapped in local minima though. For very small values of α, the LM algorithm loses its convergence speed, and , more or less, its accuracy. The most suitable value of α was obtained empirically as 1.01. This would guarantee less iteration cycles, and thus a relatively fast convergence. Furthermore, this would lead to a more precise estimation of the parameters. Taking into account the above considerations, the LM algorithm as discussed in section III.C was applied. The resultant sum of square errors was obtained as $SSE = 0.0000535$. The approximated rotor speeds are shown in Figs. 8 and estimation errors are provided in Figs. 9.



(a)

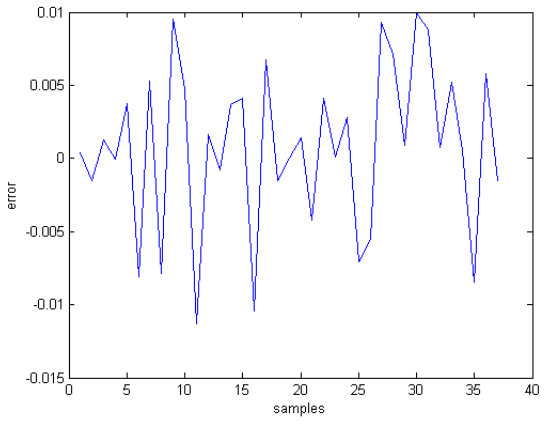


(b)

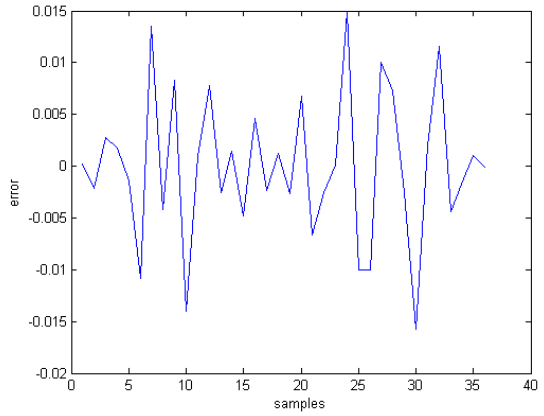


(c)

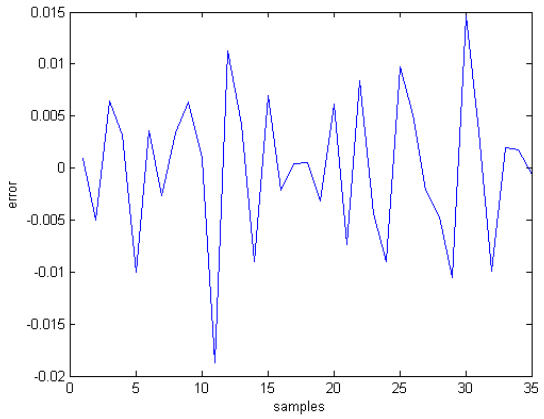
Figs. 8. The estimated nonlinear function (solid line), and the experimental data from USR60 (crosses) for different torque values of 0.0 (a), 0.1 (b), 0.2 (c) at driving frequency of 41kHz



(a)



(b)



(c)

Figs. 9. The estimation error per samples associated with different torque values of 0.0 (a), 0.1(b), and 0.2 (c) at driving frequency of 41kHz

Similarly, the simulation was performed for different torque values and different driving frequencies.

The available input/output data were in the range of $\varphi \in \left[0,90^{\circ}\right]$; therefore, a set of 11 knots were preset $\{\Phi_k\} = \{[0,10,20,60,65,68,70,75,80,85,90]\}$.

The constructed Bernstein basis functions are shown in Fig. 10.

The simulation results are also available in Figs. 11-13.

According to the results, the proposed algorithm shows significant preciseness compared with other published identification approaches for TWUSMs.

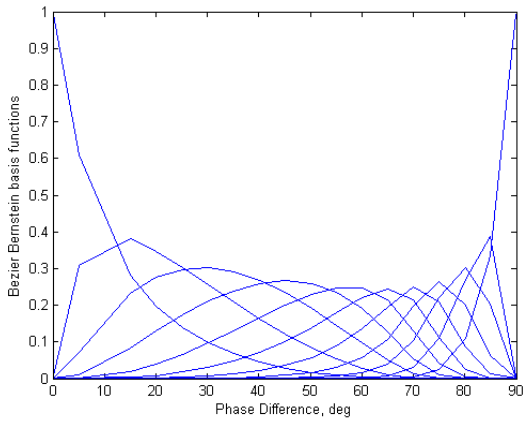
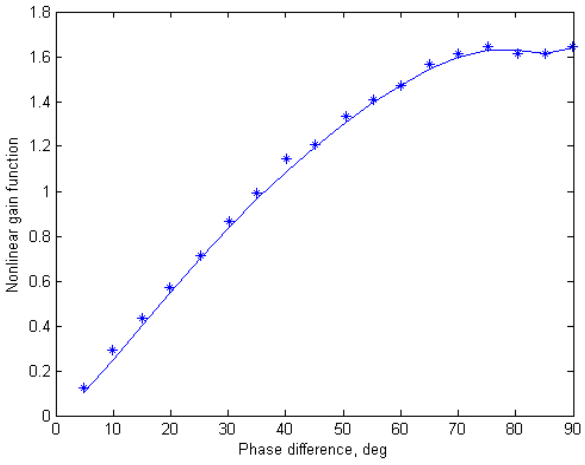


Fig. 10. Bezier-Bernstein Basis functions used to approximate the nonlinear gain functions in driving frequencies other than 41KHz illustrated in Figs. 7-9



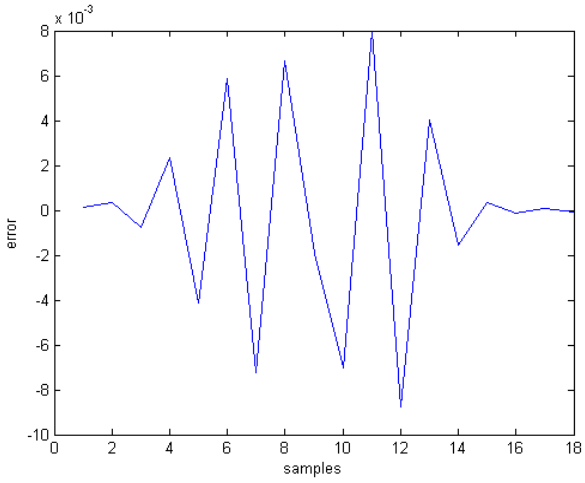


Fig. 11(a)

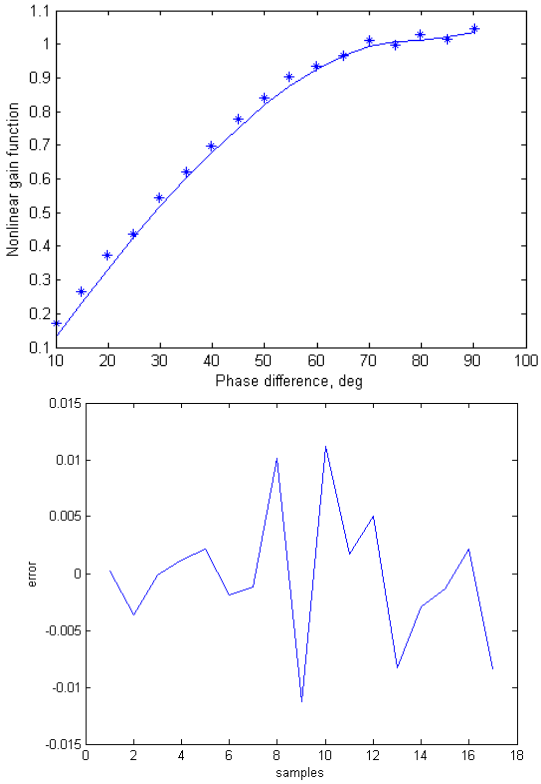


Fig. 11(b)

Figs. 11. The estimation error (right), the estimated nonlinear function (solid line), and the experimental data from USR60 (crosses) (left) in no load condition associated with different driving frequency values of 41.5kHz (a), and 42kHz(b)

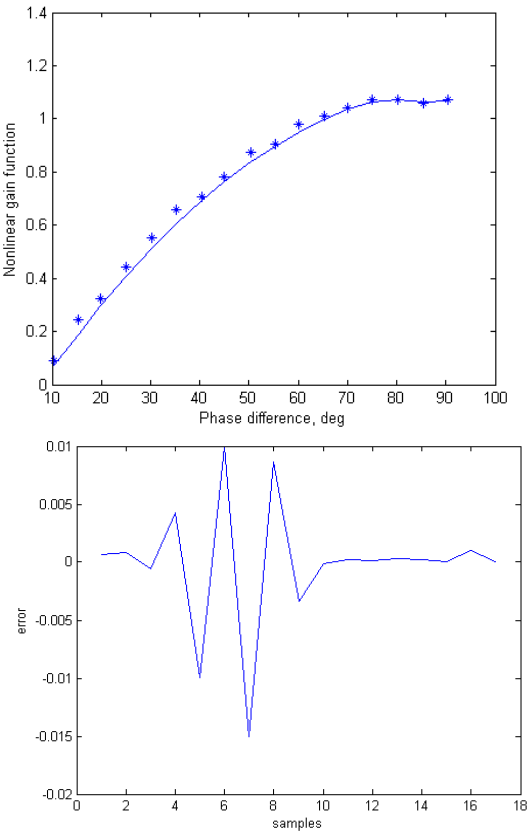


Fig. 12(a)

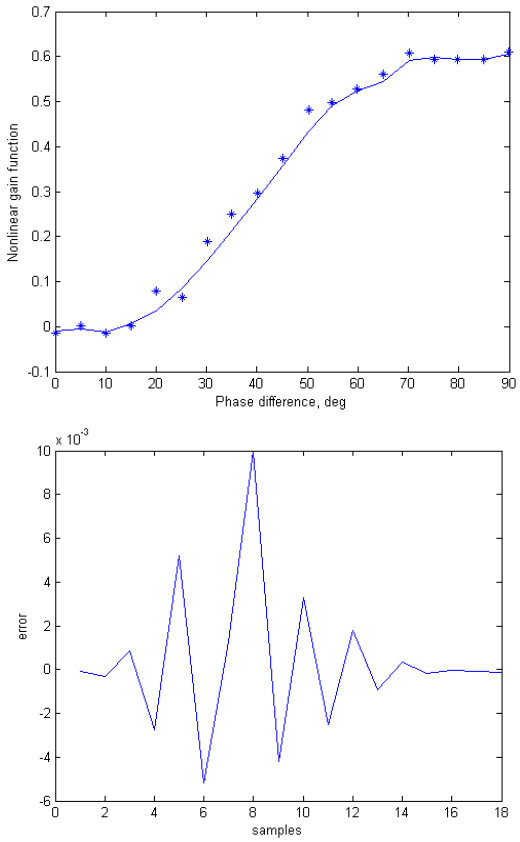


Fig. 12(b)

Figs. 12. The estimation error(right), the estimated nonlinear function (solid line), and the experimental data from USR60 (crosses) (left) at torque value of 0.1 associated with different driving frequency values of 41.5kHz (a), and 42kHz(b)

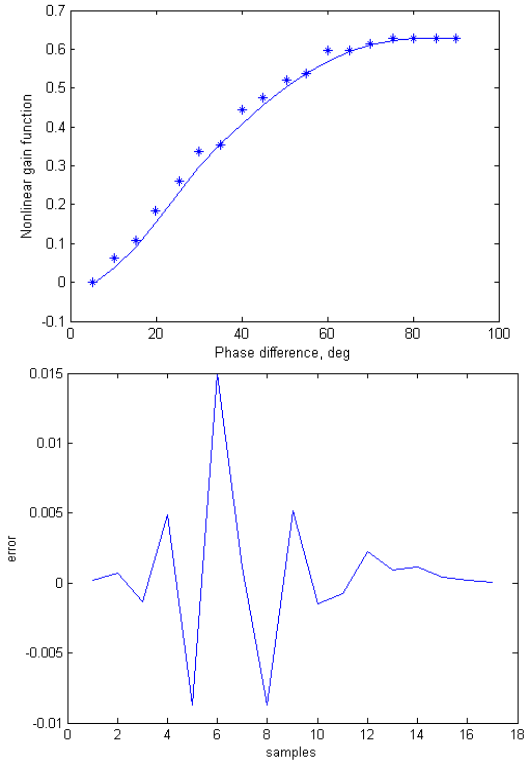


Fig. 13(a)

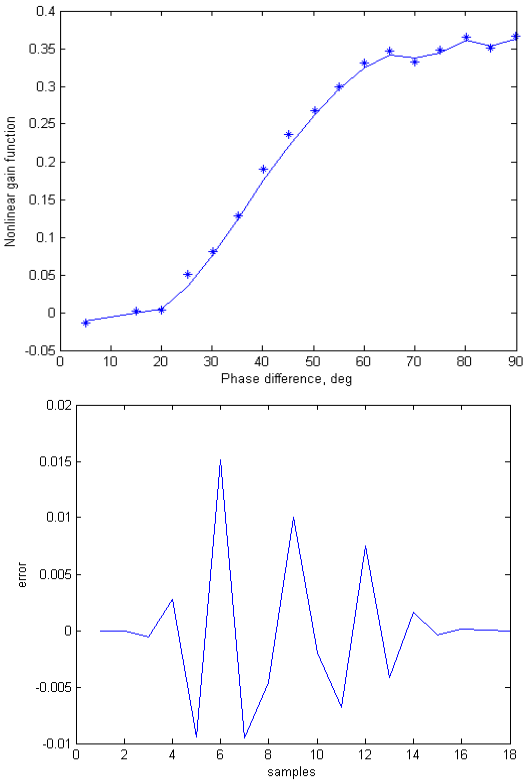


Fig. 13(b)

Figs. 13. The estimation error (right), the estimated nonlinear function (solid line), and the experimental data from USR60 (crosses) (left) at torque value of 0.2 associated with different driving frequency values of 41.5kHz (a), and 42kHz(b)

VI. Conclusion

This paper presents a new, fast and straightforward scheme for the identification of TWUSMs using Bezier-Bernstein polynomial functions. The inverse de Casteljau algorithm is implemented to map the sampled input data. Subsequently, the Bezier-Bernstein basis functions are formed. Then, using the least squares method followed by the LM algorithm, the unknown parameters are estimated. For the purpose of comparison, simulation results based on the proposed scheme and experimental data are also provided, which verified the algorithm's accuracy. Concluding from the results of this study, the identification and modeling scheme has proved to be able to precisely approximate the nonlinear characteristics of TWUSMs.

References

[1] T. Sashida,T. Kenjo, An introduction to ultrasonic motors (Oxford Science Publications, 1993).

[2] K. Uchino, Piezoelectric ultrasonic motors: Overview, Smart Materials and Structures, vol. 7, 1998, pp. 273-285.

[3] M. R. Ghazanchaei, A. Vahedi, M. Mirimani, A. Baktash, Sensitivity Analysis of Hysteresis Motor Through a Circuit Coupled 3D FEM Model, International Review of Electrical Engineering (IREE), vol. 5, no. 4, pp. 1543-1549, 2010.

[4] M. Shahnazari, A. Vahedi, Improved Dynamic Average Modeling of Synchronous Machine with Diode-Rectified Output, International Review of Electrical Engineering (IREE), vol. 4, no. 6, pp. 1248-1258, 2009.

[5] H. Akbari, H. Meshgin-Kelk, J. Milimonfared, Induction Machine Modeling Including the Interbar Currents Using Winding Function Approach, International Review of Electrical Engineering (IREE), vol. 4. n. 1, pp. 269-277, 2009.

[6] M. A. Nasr Khoidja, B. Ben Salah , Design, Modelling and Control of a Linear Induction Motor, International Review of Electrical Engineering (IREE), vol. 2. no. 1, pp. 414-424, 2007.

[7] M. Hasni, O. Touhami, R. Ibtiouen, M. Fadel , Modeling and Parameter Identification of a Synchronous Machine by Using Singular Perturbation Approach, , International Review of Electrical Engineering (IREE), vol. 1. no. 3, pp. 418-425, 2006.

[8] P. Macko, P. Fedor, D. Perduková , Simplified Fuzzy Model of an Induction Motor, International Review of Electrical Engineering (IREE), vol. 1. no. 2, pp. 270-276, 2006.

[9] N. El Ghouthi, Hybrid modeling of traveling wave piezoelectric motor, Ph.D. thesis, Dept. Control Eng., Aalborg Univ., Aalborg, Denmark, 2000.

[10] H. Mojallali, R. Amini, R. Izadi-Zamanabadi, Ali A. Jalali, Systematic modeling of for free stators of rotary piezoelectric ultrasonic motors, IEEE/ASME Trans. Mechatronics, vol. 12, 2007, pp. 219–223.

[11] N. Hagood and J. Andrew, "Modeling of a rotary piezoelectric ultrasonic motor", IEEE Trans. Ultrason., Ferroelect.Freq. Control, vol. 42, pp. 210–24, 1995.

[12] F. Giraud, B. Lemaire-Semail, Causal modeling and identification of a traveling wave ultrasonic motor, Eur. Phys. J. Appl. Phys. , vol. 21,2003, pp. 151–159.

[13] M. Zhou, Contact analysis and mathematical modeling of traveling wave ultrasonic motor, IEEE Trans. Ultrason., Ferroelect. , Freq. Control, vol. 51, 2004, pp. 668–679.

[14] A. Frangi, A.Corigliano, M.Binci, P. Faure, Finite element modelling of a rotating piezoelectric ultrasonic motor, Ultrasonics, vol. 43, no. 9, 2005, pp. 747–755.

[15] X. Zhang, Y. Tan, Modeling of ultrasonic motors with dead-zone based on Hammerstein model structure, J. Zhejiang Univ. Sci. A, vol. 9, no. 1, 2008, pp. 58-64.

[16] N. Bigdeli, M. Haeri, Modeling of an ultrasonic motor based on Hammerstein model structure, Proc. ICARCV04, no. 2,2004, pp. 1374-78.

[17] R. K. Pearson, M. Pottman, Gray-box identification of block-oriented nonlinear models, J. Process Control, vol. 10, no. 4, 2000, pp. 301-315.

[18] H.T. Su, T.J. McAvoy, Integration of multilayer perception neural networks and linear dynamic models: A Hammerstein modeling approach, Ind. Eng. Chem. Res., vol. 26,1993, pp. 1927-1936.

[19] A. Balestrino, A. Landi, M. Ould-Zmirli, L. Sani, Automatic nonlinear auto-tuning method for Hammerstein modeling of electrical drives, IEEE Trans. Ind. Electron., vol. 48, no.3,2001, pp. 645-655.

[20] D.K. Rollins, N. Bhandari, A. M. Bassily, G.M. Colver, S.T. Chin, A continuous-time nonlinear dynamic predictive modeling method for Hammerstein processes, Ind. Eng. Chem. Res., vol. 42, no. 4, 2003, pp. 860-872.

[21] J. G. Smith, S. Kamat, K.P. Madhavan, Modelling of PH process using wavenet based Hammerstein model, J. Process Control, vol. 17, 2007, pp. 551-561.

[22] D.T. Westwick, R.E. Kearney, Identification of a Hammerstein model of the Stretch reflex EMG using separable Least Squares, Proc. 22nd EMBS international Conf., 2000, pp. 23-28.

[23] K. J. Hunt, M. Munih, N. N. Donaldson, F. M. D. Barr, Investigation of Hammerstein hypothesis in the modelling of Electrically Stimulated Muscle, IEEE Trans. Biomedical Eng., vol. 45, no. 8,1998, pp. 998-1009.

[24] F. Alonge, F. D'Ippolito, F. M. Raimondi, S. Tumminaro, Nonlinear modelling of DC/DC converters using the Hammerstein's approach, IEEE Trans. Power Electronics, vol. 22, no. 4,2007, pp. 1210-1221.

[25] U. Mehta, S. Majhi, Identification of a class of Wiener and Hammerstein-type nonlinear processes with monotonic static gains, ISA Transactions, no. 49, no. 4,2010, pp. 501-509.

[26] X. Hong , R.J. Mitchell, Hammerstein model identification algorithm using Bezier-Bernstein approximation, IET Control Theory Appl. , vol. 1, no. 4, 2007, pp. 1149-1159.
[27] S. Kwon, M. Lynch, M. Prokop, Decoupling PI controller design for a normal conducting RF cavity using a recursive Levenberg-Marquart algorithm, IEEE Trans. Nuclear Sci., vol. 52, no. 1,2005, pp. 440-449.
[28] X. Hong, C.J. Harris, Generalized neuro fuzzy network modeling algorithms using Bezier-Bernstein polynomial functions and additive decomposition, IEEE Trans. Neural Networks, vol. 11, 2000, pp. 889-902.
[29] T. Senjyu, S. Yokoda, Y. Gushiken, K. Uezato, Position control of ultrasonic motors with adaptive dead-zone compensation, Proc. IAS98, 1998, pp. 506-512.
[30] T. Senjyu, T. Kashiwagi, K. Uezato, Position control of ultrasonic motors using MRAC with dead-zone compensation, IEEE Trans. Ind. Electron., vol. 48, 2001, pp. 1278-1285.
[31] Y. Shin, Modified Bernstein Polynomial and their connectionist interpretation, Proc. IEEE World Congress on Computational Intelligence, vol. 3,1994, pp. 1433-1438.
[32] G. Farin, Curves and Surfaces for Computer-Aided Design: A Practical Guide (Boston, MA: Academic, 1994).
[33] T. Pavildis, Algorithms for Graphics and Image Processing (Rockville, MD: Computer Science Press, 1982).
[34] M. B. Egerstedt, C. F. Martin, A note on the connection between Bezier curves and linear optimal control, IEEE Trans. on Automatic Control, vol. 49, no. 10, 2004, pp. 1728-1731.
[35] K. Narooi, A. K. Taheri, A new model for prediction the strain field and extrusion pressure in ECAE process of circular cross section, Applied Mathematical Modelling, vol. 34,2010, pp. 1901–1917.
[36] K. K. Gorowara, On Bezier curves and surfaces, Proc. IEEE Aerospace and Electronics Conf. (NAECON 1988), vol. 2, 1988, pp. 754 -756.
[37] M. Zettler, J. Garloff, Robustness analysis of polynomial parameter dependency using Bernstein expansion, IEEE Trans. Automatic Control, vol. 43, no. 3, 1998, pp. 425-431.
[38] M.J.D. Powell, Problems related to unconstrained optimization, in Murray, W. (Ed.), Numerical methods for unconstrained optimisation (Academic Press, London & New York, 1972, pp. 29-55).
[39] K. Levenberg, A Method for the Solution of Certain Non-linear Problems in Least Squares, Quarterly of Applied Mathematics, vol. 2, no. 2,1944, pp. 164–168.
[40] D. Marquardt, An Algorithm for the Least-Squares Estimation of Nonlinear Parameters, SIAM Journal of Applied Mathematics, vol. 11, no. 2, 1963, pp. 431–441.
[41] K. Madsen, H. Nielsen, O. Tingleff, Methods for Non-Linear Least Squares Problems, Technical University of Denmark, 2004. Lecture notes, available at http://www.imm.dtu.dk/courses/02611/nllsq.pdf

Authors' information

Electrical Engineering Department,
Faculty of Engineering,
University of Guilan,
Rasht, Iran.

Hamed Mojallali was born in Fouman, Iran, in 1974. He Received the Ph.D. degree in Control Engineering from Iran University of Science and Technology (IUST), Iran, in 2006. He is currently assistant professor at Electrical Engineering Department, University of Guilan, Rasht, Iran. His current research activities include modeling and system identification, evolutionary algorithms and hybrid systems.

Mohamadreza Ahmadi was born in Rasht, Iran, in 1988. He has received his BSc. in electrical engineering with first class honors from University of Guilan, Iran, in 2010. Currently, he is pursuing his masters' studies in electrical engineering at the same university. He is a gold member of IEEE Iran section. His current research interests include system identification, modeling, estimation, nonlinear filtering, and hybrid systems.