ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MODELING OF DYNAMIC SYSTEMS AND NONLINEAR SYSTEM IDENTIFICATION

Ph.D. THESIS

Masoud ABEDINIFAR

Department of Mechatronics Engineering

Mechatronics Engineering Programme

MARCH 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MODELING OF DYNAMIC SYSTEMS AND NONLINEAR SYSTEM IDENTIFICATION

Ph.D. THESIS

Masoud ABEDINIFAR
(518162013)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Prof. Dr. Şeniz ERTUĞRUL

MARCH 2023

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DİNAMİK SİSTEMLERİN MODELLENMESİ VE DOĞRUSAL OLMAYAN SİSTEMLERİN TANILANMASI

DOKTORA TEZİ

Masoud ABEDİNİFAR
(518162013)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şeniz ERTUĞRUL

MART 2023

Masoud ABEDINIFAR, a Ph.D. student of ITU Graduate School student ID 518162013, successfully defended the dissertation entitled "MODELING OF DYNAMIC SYSTEMS AND NONLINEAR SYSTEM IDENTIFICATION", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Şeniz ERTUĞRUL** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Lale Canan DÜLGER** Izmir University of Economics	.....
	Assoc. Prof. Gökhan Tansel TAYYAR Istanbul Technical University	.....
	Prof. Dr. Erdinç ALTUĞ Istanbul Technical University	.....
	Assist Prof. Ebru SAYILGAN Izmir University of Economics	.....

Date of Submission : 24 February 2023
Date of Defense : 24 March 2023

FOREWORD

In this study, we suggest a framework for identifying and verifying the nonlinear model of different physical systems. The model identification and verification methodologies explained here can be employed for different physical systems with nonlinear models, such as fluidic systems, thermal systems, or mechanical systems.

Words cannot explain how thankful I am to Prof. Dr. Şeniz ERTUĞRUL, my professor and supervisor, for her tremendous guidance, constant support, and patience during my Ph.D. studies. Her vast expertise and vast knowledge have inspired me in all of my academic studies. I also could not have undertaken this journey without my defense committee members, Prof. Dr. Lale Canan DÜLGER and Assoc. Prof. Dr. Gökhan Tansel TAYYAR, who generously provided knowledge and expertise during my research. I would like to give my sincere appreciation to Assoc. Prof. Dr. Gökhan Tansel TAYYAR, who provided me an opportunity to join the project (TÜBITAK-TEYDEB) under his supervision. I would also like to extend my thanks to Assist. Prof. Dr. Şenay MIHÇIN for the opportunity to be a part of a project under her supervision at Izmir Institute of Technology. I also would like to thank Prof. Dr. Ahmet Sermet ANAGÜN for his precious recommendations throughout this research.

I would also like to thank my research fellows, Serdar Hakan ARGÜZ, Dr. Ozan KAYA, and Dr. Ömer Faruk ARGIN, for their valuable contributions at different levels of this research. I would like to thank Assist. Prof. Melek ERTOĞAN from the Department of Maritime Transportation and Management Engineering at Istanbul Technical University and Burhan SÜTCÜ from SBS Plastic for their support, hosting, insightful comments, and suggestions during this study. I would like to thank the faculty members and staff of the Izmir University of Economics for providing me the chance to work and collect data for this study in TESLA laboratories. I also want to thank graduate students and research assistants from different departments at Istanbul Technical University, Izmir University of Economics, and Izmir Institute of Technology who have always motivated me with their valuable conversations during graduate studies. It is their support and kind friendship that have made my study and life in Turkey an amazing experience.

Very special thanks to my parents, who offered me unconditional love, support, and understanding during the period of my study. I am very grateful for their sacrifices, and trust that encouraged me in achieving my goals. I would also like to express my deepest appreciation to Aysu ÇIRAK, my finacee, for her endless love, support, and care that helped me achieve my goals.

February 2023

Masoud ABEDINIFAR
(Mechatronics Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANN : Artificial Neural Networks
AR : Autoregressive
ARMA : Autoregressive with Moving Average
ARMAX : Autoregressive Moving Average with Exogenous Inputs
ARX : Autoregressive with Exogenous input
BJ : Box-Jenkins
DC : Direct Current
DH : Denavit-Hartenberg
DNN : Deep Neural Networks
FEM : Finite Element Method
FIR : Finite Impulse Response
GA : Genetic Algorithm
HS : Harmony Search
LSE : Least Squared Errors
MCMC : Monte Carlo Markov Chain
MLE : Maximum Likelihood Estimation
MSE : Mean Square Errors
NAR : Nonlinear Autoregressive
NARMA : Nonlinear Autoregressive with Moving Average
NARMAX : Nonlinear Autoregressive Moving Average with Exogenous input
NARX : Nonlinear Autoregressive with Exogenous input
NLSE : Nonlinear Least Squared Errors
NOE : Nonlinear Output Error
OE : Output Error
PRBS : Pseudo-Random-Binary-Sequence
PRMS : Pseudo-Random-Multilevel-Sequence
PSO : Particle Swarm Optimization
RMSE : Root Mean Square Errors
ROS : Robot Operating System
SA : Simulated Annealing
SNR : Signal-to-Noise Ration
SS : State-space
SSE : Sum of Squared Errors
SVD : Singular Value Decomposition
TF : Transfer Function
UR-ROS : Universal Robots-Robot Operating System
VS : Volterra Series
WOA : Whale Optimization Algorithm

SYMBOLS

$\mathbf{y(t)}$	**:** General output of the system
$\mathbf{i(t)}$	**:** General input of the system
$\mathbf{e(t)}$	**:** White Gaussian Noise
$\mathbf{G(q)}$	**:** Input nonlinear function of the Hammerstein-Wiener model
$\mathbf{S(q)}$	**:** Output nonlinear function of the Hammerstein-Wiener model
$\mathbf{f(x)}$	**:** Linear dynamic model of the Hammerstein-Wiener model
$\mathbf{V_i^{j+1}}$	**:** Particle velocity
$\mathbf{X_i^{j+1}}$	**:** Particle position
$\mathbf{pBest^j}$	**:** Particle best
$\mathbf{gBest^j}$	**:** Global best
$\boldsymbol{C_1}$	**:** Self-confidence coefficient
$\boldsymbol{C_2}$	**:** Swarm confidence coefficient
$\mathbf{CF}$	**:** Cost function of the PSO algorithm
$\mathbf{Y_j}$	**:** Real/measured output of the systems
$\mathbf{\hat{Y}_j}$	**:** Calculated output of the systems
$\mathbf{\bar{Y}}$	**:** Mean of the output
$\mathbf{L(\beta)}$	**:** Loss function of the NLSE algorithm
$\boldsymbol{\beta}$	**:** Parameter vector
$\mathbf{n}$	**:** Number of the data samples
$\boldsymbol{\lambda}$	**:** Damping coefficient of the NLSE algorithm
$\mathbf{H_0}$	**:** Null hypothesis
$\mathbf{H_1}$	**:** Alternative hypothesis
$\mathbf{SD_{\beta_k}}$	**:** Standard deviation
$\boldsymbol{\sigma^2_{\beta_k}}$	**:** Variance
$\mathbf{TS}_k$	**:** Test statistic
$\boldsymbol{\Phi(Z_k)}$	**:** Model under null hypothesis
$\boldsymbol{p_k}$	**:** Probability value of the model parameter
$\boldsymbol{\alpha}$	**:** Significance level
$\mathbf{R^2}$	**:** Coefficient of determination
$\mathbf{U(t)}$	**:** Input voltage
$\mathbf{I(t)}$	**:** Electrical current
$\boldsymbol{\omega(t)}$	**:** Angular velocity of the DC motor
$\mathbf{T_m(t)}$	**:** Torque of the DC motor
$\mathbf{J_m}$	**:** Moment of inertia of the DC motor
$\mathbf{R_m}$	**:** Resistance of the DC motor
$\mathbf{L_m}$	**:** Inductance of the DC motor
$\mathbf{K_m}$	**:** Motor constant
$\mathbf{T_f}$	**:** Generalized friction torque
$\mathbf{T_c}$	**:** Generalized Coulomb friction constant
$\mathbf{T_s}$	**:** Generalized static friction constant
$\mathbf{T_{v1}}$	**:** Generalized viscous friction constant

$\mathbf{T_{v2}, T_{v3}}$: Generalized nonlinear viscous friction constants
$\boldsymbol{\omega_s}$: Generalized Steribeck velocity constant
$\mathbf{m_s}$: Mass of the moving part of the solenoid actuator
$\mathbf{k_s}$: Stiffness of the spring of the solenoid actuator
$\mathbf{F_{mag}}$: Magnetic force
$\mathbf{F_f^s}$: Friction force of the solenoid actuator model
$\mathbf{F_c^s}$: Coulomb friction force constant of the solenoid actuator model
$\mathbf{F_s^s}$: Static friction force constant of the solenoid actuator model
$\mathbf{F_v^s}$: Viscous friction force constant of the solenoid actuator model
$\mathbf{\dot{x}_s^s}$: Steribeck velocity constant of the solenoid actuator model
$\mathbf{L_b}$: Length of the beam
$\mathbf{M_b}$: Mass of the ball
$\mathbf{J_b}$: Moment of inertia of the ball
$\mathbf{r_a}$: Radius of the ball of the ball
$\boldsymbol{\gamma}$: Angle of the motor of the ball and beam mechanism
$\boldsymbol{\beta_b}$: Angle of the beam
$\mathbf{x_b}$: Position of the ball
$\mathbf{g}$: Gravitational acceleration constant
$\mathbf{F_f^{bb}}$: Friction force of the ball and beam mechanism model
$\mathbf{F_c^{bb}}$: Coulomb friction force of the ball and beam mechanism model
$\mathbf{F_s^{bb}}$: Static friction constant of the ball and beam mechanism model
$\mathbf{F_{v1}^{bb}}$: Viscous friction force of the ball and beam mechanism model
$\mathbf{F_{v2}^{bb}, F_{v3}^{bb}}$: Nonlinear viscous friction constants of the ball and beam mechanism model
$\mathbf{\dot{x}_s^{bb}}$: Steribeck velocity constant of the ball and beam mechanism model
$\mathbf{U_l, U_r}$: Left and right breakpoints of the dead-zone nonlinearity
$\mathbf{q, \dot{q}, \ddot{q}}$: Generalized positions, velocities, and accelerations of the joints
$\mathbf{M(q)}$: Generalized Inertia matrix of the joints of the robot
$\mathbf{C(q, \dot{q})}$: Generalized Coriolis and centrifugal matrix of the joints of the robot
$\mathbf{G(q)}$: Generalized Gravity vector of the joints of the robot
$\boldsymbol{\tau^{(i)}}$: Torque of the joint i of the UR5 manipulator
$\boldsymbol{\tau_f^{(i)}}$: General friction torque of the joint i of the UR5 manipulator
$\boldsymbol{\tau_c^{(i)}}$: Coulomb friction constant of the joint i of the UR5 manipulator
$\boldsymbol{\tau_s^{(i)}}$: Static friction constant of the joint i of the UR5 manipulator
$\boldsymbol{\tau_{v1}^{(i)}}$: Viscous friction constant of the joint i of the UR5 manipulator
$\boldsymbol{\tau_{v2}^{(i)}, \tau_{v3}^{(i)}}$: Nonlinear viscous constants of the joint i of the UR5 manipulator
$\boldsymbol{\dot{q}_s^{(i)}}$: Steribeck velocity constants of the joint i of the UR5 manipulator
$\mathbf{G(s)}$: Transfer function
$\mathbf{A}$: Magnetic vector potential
$\boldsymbol{\sigma}$: Electrical conductivity
$\boldsymbol{\mu_0}$: Permeability constant
$\boldsymbol{\mu_r}$: Relative permeability
$\mathbf{j_{ext}}$: Current density
R : Radius of the plunger
L1, L2 : Length of optimization parameters of the solenoid
W : Width of the guider part of the solenoid
$\boldsymbol{\theta}$: Conic angle of the plunger of the solenoid

$\mathbf{K_{bb}}$: Kinetic energy term for the ball and beam mechanism
$\mathbf{U_{bb}}$: Potential energy term for the ball and beam mechanism
$\mathbf{J_{Beam}}$: Moment inertia of the beam of the ball and beam mechanism
$\boldsymbol{\theta_i}$: Joint angles of the UR5 manipulator (DH parameter)
$\mathbf{d_i}$: Offset of the link i (DH parameter)
$\mathbf{a_i}$: Length of the link i ((DH parameter))
$\boldsymbol{\alpha_i}$: Twist angle of the link i (DH parameter)
$\mathbf{T_i^{i-1}}$: Transformation matrix between frame i-1 and i
$\mathbf{R_i^{i-1}}$: Rotation matrices between frame i-1 and i
$\mathbf{t_i^{i-1}}$: Translation vectors between frame i-1 and i
$\mathbf{v_{tp}}$: Linear velocity of the tip-point of the UR5 manipulator
$\boldsymbol{\omega_{tp}}$: Angular velocity of the tip-point of the UR5 manipulator
$\mathbf{J_{V_{tp}}, J_{\omega_{tp}}}$: Geometric Jacobians of the tip-point of the UR5 manipulator
$\mathbf{J_{v_{mci}}}$: Jacobian matrices of the center of the mass of the link i
$\mathbf{J_{v_{mci}}, J_{\omega_{mci}}}$: Geometric Jacobians of the center of the mass of the link i
$\mathbf{r_{mc_i}^0}$: Center of mass of the link i regarding frame 0
$\mathbf{m_i}$: Mass of the link i of the UR5 manipulator
$\mathbf{mc_i}$: Center of mass of the link i regarding frame i

LIST OF TABLES

Page

LIST OF FIGURES

Page

MODELING OF DYNAMIC SYSTEMS AND NONLINEAR SYSTEM IDENTIFICATION

SUMMARY

One of the primary goals of science is to identify and describe the structures and physical laws of nature. When the data corresponding to the input and output of a physical system is available, but the underlying rules and the structure of the system are unknown, it is essential to employ various approaches to determine these rules and structures. Determination of the underlying rules and structure of a system, particularly in some operation regions, is a difficult task because of the existence of some nonlinearities in the structure of the model. Therefore, choosing a reliable approach to identify the structure of the model in the different working regions of the system is crucial. For this purpose, system identification has been established as a critical technique for assisting in the modeling of complex engineering systems. System identification includes all processes of establishing a mathematical model of the systems by measured input-output datasets. The developed mathematical models using system identification methods are commonly used for monitoring, controller design, fault detection, system response prediction, optimization, and other purposes. The procedure of system identification could be classified into three steps: First, the structure of the mathematical model has to be determined. The structure of the mathematical model could be represented with linear or nonlinear models. Second, the unknown coefficients of the mathematical model should be determined by simulation or experimental input-output datasets. Finally, the model with the identified parameters has to be validated with the new input-output datasets.

The major aims of this research could be listed as: In the first step, it is planned to develop transparent nonlinear mathematical models of the mechanical systems in a way that each term of the model could be physically interpreted. These models are called "white-box" models, which are developed using physical rules like Kirchhoff's and Newton's rules. Second, the thesis aims to properly determine the nonlinear models of the physical systems utilizing an appropriate system identification methodology. Third, it aims to investigate the existence of the identified physical phenomena, like nonlinear frictional terms, and dead-zone using different statistical methods. To fulfill these purposes, the following steps are performed:

First, the general mathematical models of some physical systems are developed. The mathematical models of the physical systems include linear and various nonlinear equations. The linear equations of the model are developed utilizing some physical rules like Kirchhoff's and Newton’s rules, etc. For the nonlinear part of the models, the nonlinear equations of some physical phenomena, like nonlinear friction equations and dead-zone, along with time-delay, are compiled and added to the general mathematical model of the physical systems. Then, the appropriate input signals are generated to stimulate all the dynamics of the physical systems in their different working regions. This is performed to capture the effect of all the possible existing nonlinearities in the system’s output. In the next step, the output of the mathematical models is collected,

and input-output data sets are established. Then, the Particle Swarm Algorithm (PSO) algorithm is coded to determine the unknown parameters of the general mathematical model of the system using input-output datasets. The PSO algorithm's results are evaluated by utilizing the conventional Nonlinear Least Squared Errors (NLSE) estimation method. Afterward, various statistical tests, including the confidence interval test and the null hypothesis test, are executed to investigate the identification results' validity. Finally, using some model evaluation criteria such as Mean Squared Errors (MSE) and coefficient of determination (R^2), the capability of the determined models in computing the output of the real systems is evaluated. The framework suggested in this thesis is implemented for four case studies as benchmark problems, ranging from simple to complex in two steps.

Initially, two case studies, namely a Direct Current (DC) motor, and a solenoid actuator are chosen, and their mathematical models with various combinations of nonlinearities are constructed in the first stage. The simulation data for both the DC motor and solenoid actuator models are established by utilizing the nonlinear models. First, all kinds of friction nonlinearities are incorporated into the real mathematical models of these components, followed by adding some likely friction nonlinearities to check the effectiveness of the identification algorithms. After that, the identification and validation frameworks are utilized to ascertain the model parameters and verify the credibility of the outcomes. Furthermore, a PSO algorithm with multiple cost functions is used to optimize the design parameters of a solenoid actuator to improve its performance.

The second stage involves obtaining actual experimental data from real mechanical systems, which is then utilized to examine the framework developed in the simulation studies. The initial benchmark problem involves collecting real data from the experimental apparatus of the ball and beam mechanism by providing appropriate input signals. Moreover, the identification algorithm's effectiveness is tested for various experimental conditions for the mechanism of the ball and beam. In the second benchmark problem, real data is acquired from a 6-degree-of-freedom (DOF) UR5 robotic manipulator by providing appropriate trajectories. Then, the model parameters are determined, and the reliability of the outcomes is examined using the identification and validation frameworks.

DİNAMİK SİSTEMLERİN MODELLENMESİ VE DOĞRUSAL OLMAYAN SİSTEMLERİN TANILANMASI

ÖZET

Bilimin temel amaçlarından biri, doğanın yapısını ve fiziksel yasaları tanılamak ve açıklamaktır. Fiziksel bir sistemin girdi ve çıktılarına karşılık gelen veriler mevcut olduğunda, ancak sistemin temsil eden matematik modelin yapısı ve katsayıları bilinmediğinde, matematik modeli ve katsayıları belirlemek için çeşitli yaklaşımlar kullanılır. Birçek sistemde doğrusal olmayan terimlerin bulunması nedeniyle model tanılama zor bir çalışmadır ve modelin yapısını ve katsayılarını tanılamak için güvenilir bir yaklaşım seçmek çok önemlidir. Bu amaçla, sistem tanılama, karmaşık mühendislik sistemlerinin modellenmesine yardımcı olmak için gerekli bir teknik olarak belirlenmiştir.

Sistem tanılama; ölçülen girdi-çıktı veri setleri ile sistemlerin matematiksel bir modelini geliştirme süreçlerinin tümünü içerir. Sistem tanılama yöntemleri kullanılarak geliştirilen matematiksel modeller genellikle izleme, denetleyici tasarımı, hata tespiti, sistem yanıt tahmini, optimizasyon ve diğer amaçlar için kullanılabilir. Sistem tanılama prosedürü üç adımda sınıflandırılabilir: İlk olarak, model yapısı tanımlanmalıdır. Matematiksel modelin yapısı doğrusal veya doğrusal olmayan terimlerle temsil edilebilir. İkinci olarak, matematiksel modelin bilinmeyen katsayıları benzetim veya deneysel girdi-çıktı veri setleri ile belirlenmelidir. Son olarak, belirlenen parametrelere sahip modelin yeni girdi-çıktı veri setleri ile doğrulanması gerekmektedir.

Sistem tanılama ve ilgili kavramlara kısa bir genel bakıştan sonra, bu çalışmanın temel amaçları aşağıdaki gibi sıralanabilir: İlk adımda, mekanik sistemlerin şeffaf doğrusal olmayan matematiksel modellerinin, modelin her bir teriminin fiziksel olarak yorumlanabileceği şekilde geliştirilmesi planlanmalıdır. Bu modeller, Kirchhoff ve Newton kuralları gibi fiziksel yasalar kullanılarak geliştirilen "beyaz kutu" modelleri olarak adlandırılmaktadır. İkinci olarak, tez, uygun bir sistem tanılama yaklaşımı kullanarak fiziksel sistemlerin doğrusal olmayan modellerini doğru bir şekilde tanılamayı amaçlamaktadır. Üçüncü olarak, doğrusal olmayan sürtünme terimleri ve ölü bölge gibi tanılanan fiziksel olayların varlığını farklı istatistiksel yöntemler kullanarak araştırmayı amaçlamaktadır. Bu amaçları yerine getirmek için aşağıdaki adımlar gerçekleştirilmiştir:

İlk olarak, fiziksel sistemlerin genel matematiksel modelleri geliştirilmiştir. Sistemlerin matematiksel modelleri doğrusal ve doğrusal olmayan denklemleri içermektedir. Modelin doğrusal denklemleri Kirchhoff ve Newton yasaları gibi bazı fiziksel kurallar kullanılarak geliştirilmiştir. Modellerin doğrusal olmayan kısmı için, doğrusal olmayan sürtünme denklemleri ve ölü bölge gibi fiziksel olayların doğrusal olmayan denklemleri, zaman gecikmesi ile birlikte derlenir ve fiziksel sistemlerin genel matematiksel modeline eklenir. Ardından, fiziksel sistemlerin farklı çalışma bölgelerindeki tüm dinamiklerini uyarmak için uygun giriş sinyalleri üretilir. Bu,

sistemin çıktısındaki tüm olası mevcut doğrusal olmayan terimlerin etkisini yakalamak için gerçekleştirilir. Bir sonraki adımda, modellerin çıktıları toplanır ve girdi-çıktı veri setleri oluşturulur. Daha sonra giriş-çıkış veri setlerini kullanarak sistemin genel modelinin parametrelerini belirlemek için Parçacık Sürü Algoritması (PSO) kullanılır. PSO algoritmasının sonuçlarını karşılaştırmak için geleneksel Doğrusal Olmayan En Küçük Karesel Hatalar (NLSE) tahmin yöntemi kullanılır. Ardından, tanılama sonuçlarının güvenilirliğini incelemek için güven aralığı testi ve boş hipotez testi gibi farklı istatistiksel testler yapılmıştır. Son olarak, Ortalama Karesel Hata (MSE) ve belirleme katsayısı (R^2) gibi bazı model değerlendirme kriterleri kullanılarak, tanılanan modellerin gerçek sistemlerin tepkisini hesaplama kabiliyeti değerlendirilir. Bu tezde önerilen çerçeve, iki adımda basitten karmaşığa değişen dört vaka çalışması için kıyaslama problemleri olarak uygulanmıştır.

İlk adımda, bir Doğru Akım (DC) motoru ve bir solenoid eyleyici olmak üzere iki vaka çalışması seçilmiş ve bunların doğrusal olmayan özelliklerinin çeşitli kombinasyonlarına sahip matematiksel modelleri oluşturulmuştur. Ardından, modellerden elde edilen benzetim verileri, hem DC motorun hem de solenoid eyleyicinin doğrusal olmayan modelleri kullanılarak oluşturulmuştur. İlk olarak, bu bileşenlerin gerçek matematiksel modeline tüm doğrusal olmayan sürtünme terimleri eklenmiştir. İkinci olarak, tanılama algoritmalarının işlevselliğini kontrol etmek için bileşenlerin gerçek matematiksel modeline bazı olası doğrusal olmayan sürtünme terimleri eklenmiştir. Ardından, model parametrelerini belirlemek ve sonuçların güvenilirliğini kontrol etmek için tanılama ve doğrulama çerçeveleri kullanılmıştır. Buna ek olarak, bir solenoid eyleyicinin tasarım parametreleri, solenoid eyleyicinin performansını artırmak için çoklu maliyet fonksiyonlarına sahip bir PSO algoritması kullanılarak optimize edilmiştir.

Solenoid eyleyici, hidrolik valfler gibi çeşitli endüstriyel uygulamalarda kontrol elemanı olarak kullanılır. Farklı bileşenlerde bir aktüatör olarak kullanıldığında bu bileşenle ilgili iki ana zorluk vardır: Birincisi, aktüatörün verimliliğini temsil eden yüksek bir manyetik kuvvete ihtiyaç vardır. İkincisi, manyetik kuvvetin aktüatörün çalışma bölgesinde tutarlı olması gerekir. Bu zorlukların üstesinden gelmek için bobin özelliklerinin ve bileşenin geometrisinin optimize edilmesi gerekir. Bu araştırmada, bobinin çapı, sarım sayısı ve akım gibi özellikleri sabit tutulmuş ve değiştirilmemiştir. Bununla birlikte, bileşenin geometrik tasarım parametrelerinin (çeşitli parçaların genişliği, yüksekliği ve yarıçapı) manyetik kuvvet üzerindeki etkisi incelenmiş ve en önemli olanlar optimizasyon parametreleri olarak kabul edilmiştir. Bir sonraki adımda, solenoid aktüatörün optimum tasarım parametrelerini elde etmek için çoklu maliyet fonksiyonlu PSO algoritması kullanılmıştır.

İkinci adımda, gerçek mekanik sistemlerden gerçek deneysel veriler elde edilerek, benzetim çalışmaları için geliştirilen çerçeve gerçek veri setleri kullanılarak test edilmiştir. İlk kıyaslama probleminde, top ve kiriş sisteminin deney düzeneğinden uygun giriş sinyalleri sağlanarak gerçek veriler elde edilmiştir. Ayrıca, tanılama algoritmasının uygulanabilirliği farklı deneysel koşullar için test edilmiştir. Kıyaslama probleminde, gerçek veriler 6 serbestlik dereceli (DOF) UR5 robotik manipülatörden uygun yörüngeler sağlanarak elde edilmiştir. Ardından, model parametrelerini belirlemek ve sonuçların güvenilirliğini incelemek için tanılama ve doğrulama çerçeveleri kullanılmıştır.

UR5 robot kolunun dinamik modelini oluşturmak için aşağıdaki adımlar izlenmiştir. İlk olarak, robotun kinematiğini hem konum hem de hız seviyelerinde geliştirmek için Denavit-Hartenberg (DH) yaklaşımı kullanılmıştır. Bu yaklaşım kullanılarak, uç nokta

konumu ve referans koordinat çerçevesine göre yönelimi, eklem açıları cinsinden parametrik olarak hesaplanmıştır. Ayrıca robotun dönüşüm matrisleri de hesaplanmıştır. Daha sonra, UR5 robot kolunun kinematiği konum seviyesinde hesaplandıktan sonra manipülatörün kinematiği hız seviyesinde elde edilmıştır. Ardından, uç nokta Jacobian'ı kullanılarak, uç noktanın astar ve açısal hızları, eklem hızları cinsinden parametrik olarak hesaplanır. Bir sonraki adımda, UR5 manipülatörünün bağlantılarının kütle merkezi için Jacobian matrisleri hesaplanmıştır. UR5 manipülatörünün kinematiği hem konum hem de hız seviyelerinde hesaplandıktan sonra robotun ters dinamik modeli kurulmuştur. Daha sonra, UR5 manipülatörünün dinamik modelini geliştirmek için deney düzeneğinden gerçek veriler elde edilmiştir.

Özetle, tezin ana hedefleri, farklı mekanik sistemlerde gerçek dünyadaki doğrusal olmayan olayların model tanılamasına katkıda bulunmak ve model tanılama metodolojisinin güvenilirliğini incelemek için istatistiksel yöntemler kullanmaktır. Tezin katkıları aşağıdaki gibi sıralanabilir:

- Basit sistemlerden daha karmaşık sistemlere kadar fiziksel sistemlerin, fiziksel olarak yorumlanabilir doğrusal olmayan denklemlere sahip matematiksel modelleri geliştirilmiştir. Bu amaçla, fiziksel olarak yorumlanabilen ve mekanik sistemlerin performansını bozan farklı sürtünme terimleri PSO algoritması kullanılarak belirlenmiştir.

- Belirleme algoritmasının sonuçlarının güvenilirliği istatistiksel analiz testleri kullanılarak test edilmiştir. Bu amaçla, fiziksel bileşenlerin belirlenen model parametreleri sıfır hipotezleri ve güven aralığı testleri ile incelenmiştir. Literatürde bu yöntemler daha önce bu amaçlar için kullanılmamıştır.

- Farklı vaka çalışmalarının doğrusal olmayan modelleme problemlerini çözmek için daha önce bu amaçla kullanılmamış geleneksel bir tahmin yöntemi olan Doğrusal Olmayan En Küçük Kareler Hataları (NLSE) tahmin yöntemi kullanılmıştır.

- Doğrusal olmayan model tanılama için bir çerçeve araştırılmış ve basitten karmaşık problemlere kadar dört deneysel çalışma için uygulanmıştır. Bu amaçla, model tanılama algoritması ilk olarak bir doğru akım motoru ve bir solenoid eyleyici içeren iki örnek olay kullanılarak oluşturulan benzetim verileri ile geliştirilmiştir. Daha sonra, çerçevenin işlevselliği, bir bilye ve kiriş mekanizması ve bir UR5 robotik manipülatör dahil olmak üzere iki karmaşık dinamik sistemden elde edilen deneysel verilerle incelenecektir.

- Model tanılama ve istatistiksel doğrulama algoritmalarının işlevselliği, gerçek dünya sistemindeki doğrusal olmayan terimlerin özellikleri değiştirilerek farklı deneyler yapılarak test edilecektir. Örneğin, top ve kiriş mekanizması arasındaki sürtünme, topun temas yüzeyi ile mekanizmanın kirişi arasına yapışkan bir bant eklenerek değiştirilecektir. Daha sonra, sistemin doğrusal olmayan özellikleri değiştirildiğinde algoritmanın uygulanabilirliğini incelemek için çerçeve tekrarlanacaktır.

- Çoklu maliyet fonksiyonlarına sahip bir PSO algoritması kodlanmış ve bir solenoid eyleyicinin optimum tasarım parametrelerini elde etmek için kullanılmıştır. Optimal parametreler elde edildikten sonra, solenoid eyleyici bu parametreler kullanılarak üretilmiş ve sistemden deneysel veriler toplanmıştır. Benzetim ve deneysel çalışmaların sonuçları karşılaştırılmış ve tartışılmıştır.

1. INTRODUCTION

In this chapter, the significance of the study and its implications for the industry are discussed. First, the background of the problem is provided. Second, the purposes of the thesis are explained. Third, the contributions of the thesis are discussed to highlight its differences from the earlier research and to show the superiority of the thesis over other methods. Eventually, the thesis outline is given at the end of this chapter.

1.1 Background of the Problem

One of the primary goals of science is to identify and describe the structures and physical laws of nature. When the input and output data of a physical system are available, but the underlying rules and structure are unknown, various approaches must be employed to identify them. Identifying the underlying rules and structure of a system can be a difficult task, especially in certain operational regions, because the structure of the model may contain nonlinearities. Therefore, choosing a reliable approach to identify the structure of the model in the different working regions of the system is crucial. To address this challenge, system identification has been recognized as an essential technique to assist in modeling complex engineering systems. System identification encompasses all the processes involved in establishing a mathematical model of a physical system using measured input-output datasets [1]. The developed mathematical models using system identification methods are commonly used for monitoring, controller design, fault detection, system response prediction, optimization, and other purposes [2-5]. The process of system identification can be categorized into three steps [6]: First, identifying the model structure, which can be represented using linear or nonlinear models. Second, determining the unknown coefficients of the mathematical model using simulation or experimental input-output datasets. Finally, validating the model with new input-output datasets using the identified parameters.

In the literature, various strategies are suggested to build mathematical models of real-world systems. These methods could be categorized into linear and nonlinear modeling

approaches. In dealing with modeling problems, the first choice is to use linear models, as they are simple and easily interpretable. The literature on solving linear modeling problems is extremely rich and well-defined. Meanwhile, to solve real-world modeling problems, the linear models become insufficient and fail to deal with these problems since all the physical systems represent nonlinear behavior in their different working regions. In these cases, one of the solutions is to linearize the nonlinear model of the system around one or some of its operating points using various linearization methods such as the Taylor series [7,8]. However, in the case of a system with strong nonlinear behavior or wide working regions, using these techniques may produce inadequate results, as a large number of linear models are needed to represent the dynamic behavior of the system [9]. Therefore, when a system's dynamic structure includes nonlinearities, it becomes necessary to explore nonlinear system identification methods. Nonlinearity is a broad term that can have different meanings in various engineering fields. Nonlinearity, according to system dynamics, is anything that causes a system to violate the principle of homogeneity. Systems with nonlinearities can be mathematically described by a set of differential equations containing various nonlinearities.

In the literature, a general and complete approach that could cover all the nonlinear modeling problems is not available due to the complexities and diversities of these problems. In general, all real-world mechanical systems show nonlinear behavior in their specific operating zones. The fundamental problem in dynamic modeling processes is identifying and interpreting nonlinear terms in these systems. Nonlinear terms such as Steribeck, Coulomb, static, viscous frictions, or their different combinations, may exist in a mechanical component. These nonlinearities, which are severely limiting the performance of a system, are some of the most important nonlinearities that may occur in electro-mechanical actuators such as direct current motors, electromagnetic components such as solenoid actuators, etc. In general, inaccuracies in dynamic models are caused by disregarding the existing nonlinearities in mathematical models of mechanical systems. The nonlinearities may lead to the malfunction of the system since they cause inaccuracies in the controller of the systems, limit cycles, instabilities, steady-state error, etc. [10–13]. As a result, building parsimonious models (the simplest models among all possible models which could perfectly demonstrate the dynamic behavior of the physical system) of mechanical

systems that incorporate nonlinearities is required to examine the systems' performance.

The two most commonly used methods for modeling systems with nonlinearities are the black-box and white-box approaches. In black-box modeling, the model's structure is chosen and the parameters are determined without considering the physical understanding of the mechanical system [14]. These models simply map the input data to the output data using a mathematical function, based on fitting the mathematical model to the system's input and output data. These modeling methods are generally based on fuzzy rules, Artificial Neural Networks (ANN), etc. In the literature, different black-box modeling methods are suggested to develop mathematical models of systems containing structural nonlinearities such as friction terms. In some studies, different Nonlinear Autoregressive with Exogeneous (NARX) inputs black-box models are suggested to represent the dry friction exiting in the dynamic model of the physical systems [15,16]. In another study, a Nonlinear Autoregressive Moving Average with Exogenous input (NARMAX) black-box model is employed to represent the nonlinear model of the bearing mechanism [17]. In another study, a kernel-based Volterra Series black-box modeling method was employed to establish a nonlinear model of the system with friction nonlinearities [18]. In another study, Naerum et al. utilized a wavelet network to establish a mathematical model of the friction nonlinearities in the Phantom Omni robot [19]. These black-box modeling methods do not require any prior presumptions regarding the model structure and aim to establish a general model which maps the input of the system into output. Black-box approaches are useful when the primary objective is to fit nonlinear models to input-output data without considering the physically meaningful mathematical model of the system. However, these modeling methods come with certain drawbacks, such as increased model complexity, and do not represent any physical aspect of the system.

On the other hand, there are other methods named "white-box" models which are based on physical rules existing in nature. These models transparently explain the dynamic behavior of the systems with various physical rules [20]. To accurately demonstrate the dynamical behavior of real-world systems, their mathematical models, which contain some nonlinearities, have to be identified correctly using these "white box" models. The white-box models are one of the most preferred modeling approaches in engineering applications since they show the physical relationship between the model's

variables. The nonlinearities, such as friction terms, dead-zone, etc., which are physically interpretable, could be explained with white-box models. Therefore, employing some nonlinear system identification approaches for the determination of these nonlinearities is essential. In general, the friction terms are divided into two groups, including static and dynamic friction models [21]. The common friction type that occurs when two contacting surfaces stick together is static friction (stiction friction) [22]. The breakaway force needed to overcome this friction and commence a relative motion determines the value of this friction. However, static friction models are incapable of capturing the friction effect when the system operates at extremely low speeds or when it operates in the pre-sliding regime. To make up for these deficiencies, some references present dynamic friction models with Coulomb friction constant [23,24]. The Coulomb friction is constant and is independent of the magnitude of the velocity. This type of friction is always in the reverse direction of the system's velocity and is only affected by the direction of movement. The magnitude of this friction depends on the normal force acting on the surfaces and the characteristics of the contacting surfaces. The Steribeck velocity effect is another type of friction that is proposed in [25] and is incorporated in various nonlinear friction models [26,27]. This kind of friction appears when there is lubricative material between contacting interfaces and commonly happens when the system operates around zero velocity. The other type of friction is viscous friction, which is zero at zero velocity and whose component rises as velocity increases [28]. In [29], Wolf and Iskandar investigated the nonlinear effect of viscous friction on contacting surfaces. They added nonlinear terms to the general nonlinear friction model to capture the nonlinear effect of viscous friction at high velocities. The nonlinear friction terms mentioned above, including static, Coulomb, viscous, and Steribeck effects, or their different combinations, may exist in a mechanical component. Therefore, it is vital to identify the exact nonlinearity that exists in the model of the mechanical system.

In the literature, various nonlinear model identification approaches are suggested to find the unknown coefficients of the mathematical models. The Least Squared Errors (LSE) estimation approach, its derivatives, maximum likelihood approaches, etc. are the most commonly utilized classical methods for this purpose. For instance, the unknown friction coefficients of a 2 DOF robotic arm are determined using classical LSE, weighted LSE, and extended Kalman filtering approaches [30-32]. The general

nonlinear friction model included Coulomb and linear viscous friction terms. In these studies, the output data without any noise were utilized for parameter determination purposes. In other work, a maximum likelihood estimation-based method is employed to find the unknown coefficients of the mathematical model of the frictions in the IRB 6620/ABB robotic arm [33]. Although LSE and maximum likelihood estimation approaches are the most prominent approaches for the parameter determination of nonlinear models, they have some shortcomings. The LSE estimation approach is susceptible to the existing noise in the identification data, which may lead to inaccurate estimations [34,35]. In addition, it may converge to the local optimum values [36] or even lead to physically unrealistic solutions [37]. The Maximum Likelihood Estimation (MLE) method is another alternative to parameter determination methods. In [38], the MLE method is employed to determine the friction coefficients of the KUKA 361 manipulator. The friction model in this study includes the parameters of the Coulomb and linear viscous friction terms and other types of friction are ignored in the general model of the joints. In other similar works, the MLE approach is employed to determine the unknown coefficients of the friction terms [39-41]. The maximum likelihood estimation approaches could perfectly address the issue of noisy data; however, these approaches are computationally expensive and may be unable to converge to consistent solutions [42-44].

The presence of dead-zone nonlinearity is another type of nonlinear behavior that can be found in the mathematical models of mechanical systems. Dead-zone nonlinearity can occur between the input and output of a physical system, and it can negatively impact the performance of the system's controller. It is a significant source of instability in mechanical systems. Additionally, time-delay is another phenomenon that can exist in the dynamic behavior of real-world operating systems. The presence of a time-delay can degrade the performance of the system and potentially lead to instability if the system lacks an appropriate controller.

In general, if the mathematical model of a system contains nonlinearities such as nonlinear frictions or dead-zone along with a time-delay structure, any parameter estimation approach may be employed to determine the unknown coefficients of the mathematical models. Finding a non-zero parameter, on the other hand, does not imply the presence of the nonlinear term in the model of the physical system or vice versa.

As a result, a thorough statistical investigation of the verification methods is needed to assure the identification process's trustworthiness.

To sum up, the thesis motivation comes from the need for developing white-box models of physical systems with some nonlinearities that are physically interpretable. These models include some real-world phenomena such as nonlinear friction, dead-zone, and time-delay. Consequently, the existence of these phenomena has to be investigated and accurately identified. In addition, once the models of the systems are identified, the statistical methods have to be investigated and employed to examine the credibility of the identified model parameters.

1.2 Purpose of the Thesis

The main aims of this study could be listed below:

The thesis has outlined a three-step approach to achieving its objectives. Firstly, it aims to develop "white-box" models of mechanical systems that have physically interpretable terms, based on physical rules like Kirchhoff's and Newton's laws. These models are transparent and provide insight into the underlying physics of the system. Secondly, an appropriate system identification method will be utilized to accurately determine the nonlinear models of the physical systems. Thirdly, the thesis will investigate the presence of identified physical phenomena, such as nonlinear frictional terms, dead-zone, and time-delay, using various statistical methods. To achieve these goals, the thesis will undertake the following steps:

The first step of the process involves developing general mathematical models of physical systems, including linear and various nonlinear equations. The linear equations are based on physical rules like Kirchhoff's and Newton's laws, while the nonlinear equations include physical phenomena such as nonlinear friction equations, dead-zone, and time-delay. Input signals are then generated to stimulate the dynamics of the physical systems, and the resulting output is collected to establish input-output datasets. Next, the Particle Swarm Algorithm (PSO) is used to determine the parameters of the general model, and the results are compared with those obtained using the conventional Nonlinear Least Squared Errors (NLSE) estimation approach. Statistical tests such as the confidence interval test and the null hypothesis test are conducted to verify the identification results. The models are then evaluated using

Mean Squared Errors (MSE) and the coefficient of determination (R^2). This framework is applied to four case studies of varying complexity.

In the first step, two case studies, including a Direct Current (DC) motor and a solenoid actuator, are selected, and their mathematical models with various combinations of the nonlinearities are constructed. Then, the simulation data from the models were established by employing the nonlinear mathematical models of both the DC motor and solenoid actuator. Firstly, all types of friction nonlinearities are added to the real mathematical model of these components. Secondly, some of the probable friction nonlinearities are added to the actual mathematical model of the components to examine the functionality of the identification algorithms. Then, the identification and validation frameworks are employed to find the model parameters and check the reliability of the results. In addition, the design parameters of a solenoid are optimized using a PSO algorithm with two cost functions to improve the performance of the solenoid actuator.

In the second step, the real experimental data are acquired from the real mechanical systems. Then, the framework developed for simulation studies is tested using real data sets. In the first benchmark problem, the real data are acquired from the experimental apparatus of the ball and beam mechanism by providing appropriate input signals. In addition, the applicability of the identification algorithm is examined for different experimental conditions. In the benchmark problem, the real data are acquired from a 6-degree-of-freedom (DOF) UR5 robotic manipulator by providing appropriate trajectories. Then, the identification and validation frameworks are employed to find the model parameters and examine the reliability of the results.

1.3 Contributions of the Thesis

In this thesis, the main goals are to contribute to the model identification of real-world nonlinear phenomena in different mechanical systems and to employ some statistical methods to examine the credibility of the model identification methodology. The contributions of the thesis could be listed as follows:

- A PSO algorithm with multiple cost functions is coded and employed to obtain the optimal design parameters of a solenoid actuator. After obtaining the optimal

parameters, the solenoid actuator is fabricated using these parameters, and the experimental data is collected from the system.

Mathematical models of some physical systems from simple to more complicated systems, with physically interpretable nonlinear equations, are developed. For this purpose, different frictional terms which are physically interpretable and deteriorate the performance of the mechanical systems are determined utilizing the PSO algorithm.

- The reliability of the results of the identification algorithm will be tested using statistical analysis tests. For this purpose, the determined model parameters of the physical components are examined by null hypotheses and confidence interval tests. In the literature, these methods have not been utilized for these purposes previously.
- A conventional estimation method Nonlinear Least Squares Errors (NLSE) estimation method will be employed to solve the nonlinear modeling problems of the different case studies which have not been utilized for this purpose previously.
- A framework for the nonlinear model identification is applied to four case studies as benchmark case studies from simple to complex problems. For this aim, the model identification algorithm is first developed by simulation data generated using two case studies including a direct current motor and a solenoid actuator. Then, the functionality of the framework will be examined by experimental data acquired from two complicated real-world mechanical systems including a ball and beam mechanism and a UR5 robotic manipulator.
- The functionality of the model identification and statistical validation algorithms will be tested by conducting different experiments by changing the characteristics of the nonlinear terms in a real-world system. For instance, the friction between the ball and beam mechanism will be changed by adding an adhesive tape band between the contact surface of the ball and the beam of the mechanism. Then, the framework will be repeated to examine the applicability of the algorithm when the nonlinearities of the system were changed.

1.4 Outline of the Thesis

Following the introduction, the remaining sections of the thesis continue with the description of the system identification methodologies in Chapter 2. The

methodologies and framework for nonlinear model identification and statistical verification methods are provided in Chapter 3. In Chapter 4, benchmark nonlinear modeling problems such as a DC motor, a solenoid actuator, a ball and beam mechanism, and a UR5 robotic manipulator are presented. In Chapter 5, the results of the utilized framework in this study are presented and discussed. Finally, the conclusion of the research and future works are provided in Chapter 6.

2. SYSTEM IDENTIFICATION

The aim of system identification methods is to establish a model that accurately reflects the dynamic characteristics of the actual system. Within system identification, linear and nonlinear models are the two most widely used types of model structures. In this chapter, the most well-known linear and nonlinear modeling methods are reviewed. In addition, the parametrization methods, excitation signals, and model validation criteria as part of the system identification concept are provided.

2.1 Linear Models

Linear models assume that the relationship between various inputs and outputs of the system follows a linear pattern, while non-linearities are regarded as sources of uncertainty. The basic linear model of a system, which is also called the Box-Jenkins (BJ) model, could be demonstrated by a generalized polynomial model as equation (2.1) [48].

$$y(t) = \frac{B(q)}{A(q)F(q)}u(t) + \frac{C(q)}{A(q)D(q)}e(t) \tag{2.1}$$

where $y(t)$, $u(t)$, and $e(t)$ represent system input, system output, and white Gaussian noise, respectively. $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are polynomials that are expressed as equation (2.2) - equation (2.6).

$$A(q) = 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na} \tag{2.2}$$

$$B(q) = b_1 q^{-1} + b_2 q^{-1} + \cdots + b_{nb} q^{-nb} \tag{2.3}$$

$$C(q) = 1 + c_1 q^{-1} + \cdots + c_{nc} q^{-nc} \tag{2.4}$$

$$D(q) = 1 + d_1 q^{-1} + \cdots + d_{nd} q^{-nd} \tag{2.5}$$

$$F(q) = 1 + f_1 q^{-1} + \cdots + f_{nf} q^{-nf} \tag{2.6}$$

The unknown parameters of the polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ have to be determined with an appropriate parametrization method. However, to decrease the number of unknown parameters, it is possible to build simpler linear models that are subsets of the basic model represented in equation (2.1). The following models are built using the basic linear model by setting one or more polynomials to 1.

2.1.1 Autoregressive (AR) models

The Autoregressive (AR) models are the simplest structures that are used to generate models where the outputs depend only on prior outputs [48]. It is often used in applications such as linear prediction error in time-series analysis [49]. The AR model that includes only the terms of the polynomial $A(q)$ is demonstrated by equation (2.7).

$$y(t) = \frac{1}{A(q)} e(t) \tag{2.7}$$

2.1.2 Autoregressive with exogenous inputs (ARX) models

The Autoregressive with exogenous Inputs (ARX) models are the most basic models that incorporate the stimulus signals [48]. They are generally utilized to represent the models of systems with high orders. The major drawback of these models is that the disturbances are considered part of the dynamic model of the physical system [50]. Therefore, the variance of the model grows with increasing the order of the model. The ARX models have identical sets of poles in their deterministic and stochastic parts, and they could be represented as the equation (2.8).

$$y(t) = \frac{B(q)}{A(q)} u(t) + \frac{1}{A(q)} e(t) \tag{2.8}$$

2.1.3 Autoregressive with moving average (ARMA) models

An Autoregressive with Moving Average (ARMA) model was first suggested in [51], and has been widely used in different applications, such as forecasting in economic and engineering systems. The ARMA model could be divided into two parts containing auto-regressive (AR), which includes parameters of $A(q)$, and moving average (MA), which includes the parameters of $C(q)$. The ARMA model could be represented as the equation (2.9).

$$y(t) = \frac{C(q)}{A(q)} e(t) \tag{2.9}$$

2.1.4 Autoregressive moving average with exogenous input models

The Autoregressive Moving Average with Exogenous Inputs (ARMAX) models are similar to the ARX model; however, they include one additional term incorporating the moving average error ($C(q)$) in their stochastic part [52]. These models are represented by the equation (2.10).

$$y(t) = \frac{B(q)}{A(q)} u(t) + \frac{C(q)}{A(q)} e(t) \tag{2.10}$$

2.1.5 Output error (OE) models

The Output Error (OE) models represent the dynamics of the systems separately [53]. They do not use a polynomial to simulate the disturbance's characteristics. These models are represented by the equation (2.11).

$$y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \tag{2.11}$$

2.1.6 Finite impulse response (FIR) models

The Finite Impulse Response (FIR) model is a linear model that is commonly employed for filtering purposes [54]. In these models, the current output of the system is calculated using the previous and current input values. The FIR model could be demonstrated by the equation (2.12).

$$y(t) = B(q)u(t) + e(t) \tag{2.12}$$

2.1.7 Transfer function (TF) models

The aforementioned general linear models are often utilized for stochastic control applications because they separate the deterministic and stochastic aspects of a system. However, in traditional control engineering, the deterministic component of the system takes precedence over the stochastic component. However, the Transfer Function (TF) model is typically used to explain solely the system's deterministic part. TF models are

one of the most basic approaches for representing the linear relationships between the input and the output of the systems [55].

2.1.8 State-Space (SS) models

The State-space (SS) models are extensively employed to model both linear and nonlinear behaviors of dynamic systems [55]. Three types of variables, including, state variables, input, and output, are involved in representing the dynamic systems with these models. The state-space model of the continuous-time systems could be represented as equation (2.13) and equation (2.14).

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.13}$$

$$y(t) = Cx(t) + Du(t) \tag{2.14}$$

where $x(t)$, $u(t)$, and $y(t)$ demonstrate the state, the input, and the output variables of the system, respectively. A, B, C, and D represent the system's state, system's input, system's output, and direct transmission matrices, respectively.

Each of the listed linear models has its own set of benefits and drawbacks, and they are widely employed in the modeling of different systems. Meanwhile, there are other various ways to represent linear models which are extensively explained in the literature [56-58]. However, real-world systems are likely to be nonlinear to some extent, with the nonlinearity generated by one or a combination of different factors. In many engineering applications, linearizing a nonlinear model around some operation points is a frequent and successful strategy [59,60]. However, if the nonlinearity of the physical system is strong or the system has wide working regions, this technique may yield poor results. Therefore, a large number of linearized models are required to demonstrate the dynamic behavior of the system. Consequently, when the system exhibits nonlinear behavior, the performance of the linear model is insufficient, and nonlinear system identification models are preferred. In the next subsection, different nonlinear modeling methods will be presented.

2.2 Nonlinear Models

In various engineering applications, nonlinear system identification approaches are more favorable than linear ones, since all physical systems are intrinsically nonlinear.

The selection of a nonlinear model entails determining an appropriate model to demonstrate the input-output relationship. In the literature, various nonlinear models are extensively utilized to model real-world systems. Since the scope of the nonlinear system identification methods is broad, some of these models, such as the Volterra Series (VS), block-oriented models, etc., will be investigated in the next few sub-sections.

2.2.1 Volterra series (VS) models

The Volterra Series (VS) is a non-parametric system identification method that was first introduced in [61], and its application in solving engineering problems was extensively reviewed in [62]. In VS models, the output of the system at a given time is equal to the infinite sum of the multi-dimensional convolutional integrals of the system's previous inputs. A set of multidimensional weighting functions known as Volterra kernels completely characterizes the dynamical features of the nonlinear system in the Volterra series form. These Volterra kernels form the foundation of the Volterra series method for nonlinear analysis and system identification. The Volterra series in continuous-time systems could be represented as the equation (2.15).

$$y(t) = h_0 + \sum_{n=1}^{M} \int_a^b \cdots \int_a^b h_n(T_1, \cdots, T_n) \prod_{j=1}^{n} u(t - T_j) dT_j \tag{2.15}$$

where the output and the input of the system are demonstrated with $y(t)$ and $u(t)$, respectively. h_n represents the n^{th}-order Volterra kernel.

The Volterra series is extensively utilized for nonlinear modeling problems such as hysteresis nonlinearity [63], nonlinear stiffness, damping modeling [64], etc. However, the main disadvantage of these models is their computational complexity, which necessitates the estimation of a large number of unknown parameters [65]. Because the number of model parameters rises as the order of the nonlinearities increases, they represent poor performances for strong nonlinearities.

2.2.2 Block-oriented models

The block-oriented models, such as the Hammerstein and the Wiener nonlinear models, might be considered an alternative to existing nonlinear modeling techniques [66]. The block-oriented models include some static nonlinear components and a

dynamic Linear Time-Invariant (LTI) component that are represented with blocks. The linear components of the model could be constructed using a parametric model (transfer function model, ARX, etc.) or a nonparametric model (frequency or impulse response, etc.). On the other hand, the nonlinear component could be incorporated with some static nonlinearities such as wavelet and sigmoid networks, dead-zone and saturation nonlinearities, etc. [67].

Block-oriented models allow for various ways to connect the components of the system, which include both linear and nonlinear blocks. The connections could be made in a variety of configurations such as series, parallel, or feedback. One popular block-oriented model is the Hammerstein model, which comprises a linear model of the physical system in series with a static nonlinearity component located in the input of the system. Another frequently used block-oriented model is the Wiener model, in which the static nonlinear component is located in the output of the system and connected to the linear component in series.

The combination of the Hammerstein and the Wiener nonlinear models is known as the Hammerstein-Wiener models, which are widely used to represent a dynamic model of various physical systems [67]. They are utilized for the nonlinear modeling of an electromagnetic actuator [68], an electro-hydraulic control system [69], driver systems [70], etc. The Hammerstein-Wiener models include the system's linear dynamics along with their input static nonlinearities and output static nonlinearities [71].

The Hammerstein-Wiener model contains input nonlinear model ($G(q)$), output nonlinear model ($S(q)$), and linear model ($f(x)$) blocks that could be represented in Figure 2.1. The system's input and system's output are denoted by *u(t)* and *y(t)*, respectively.

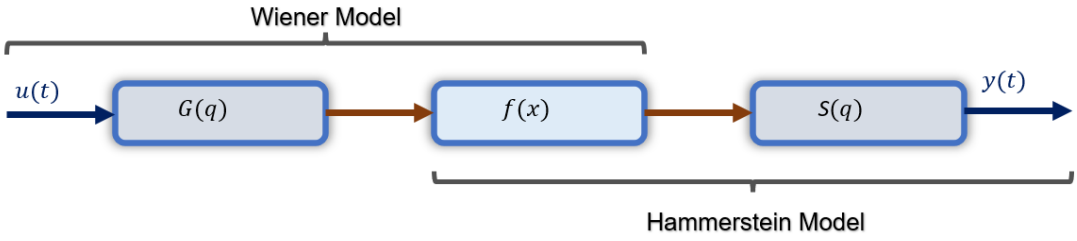


Figure 2.1 : The Hammerstein-Wiener model with input nonlinear model (*G(q)*), output nonlinear model (*S(q)*), and linear model (*f(x)*) blocks.

2.2.3 Nonlinear ARMAX (NARMAX) models and their derivatives

Nonlinear Autoregressive Moving Average with exogenous (NARMAX) Inputs Models are non-parametric, nonlinear models that are extensions of linear ARMAX models [72].

In NARMAX models, the output of the model is computed by previous outputs, inputs, and residuals. They are generally used to develop nonlinear models of physical systems, such as models of a gas turbine, an aircraft system, a diesel engine, etc. [73-75].

In the literature, different derivatives of the NARMAX model are also extensively used to model nonlinear systems. Some of these models and their applications could be addressed as follows.

The Nonlinear Autoregressive with exogenous Inputs (NARX) model is an extent nonlinear form of the ARX model that is widely utilized to model nonlinear dynamical systems [76]. This well-known method is employed to determine the nonlinear model of a typical rotor-bearing system in [77], the nonlinear modeling of a piezo-acoustic transmission system [78], etc.

In the literature, other extensions of the linear models, such as Nonlinear ARMA (NARMA), Nonlinear OE (NOE), Nonlinear AR (NAR), etc., are applied to solve different nonlinear modeling problems. In [79], the collisions of the vehicle are modeled using NARMA and the model structure, and in [80], the structure selection and the determination of the coefficients of the NAR model are investigated.

NARMAX models and their variations are commonly used to model nonlinear systems in the real world, but they have some drawbacks that must be considered. One potential limitation is that these models are not well-suited for modeling systems with high levels of nonlinearity [81]. Another challenge with these methods is selecting an appropriate model structure that accurately represents the underlying dynamics of the physical system [82]. Furthermore, these models have poorer performance compared to other methods like Hammerstein-Wiener models when applied to real systems, such as multi-input, and multi-output magnetic bearings [83]. In addition, computing the unknown coefficients for high-order models requires powerful computers and is time-consuming.

2.3 Parametrization Methods

After deciding on the structure of the system model, it is necessary to choose a suitable method for identifying the unknown coefficients of the mathematical model. Many different techniques have been explored in the literature to find the unknown parameters of both linear and nonlinear models. Extensive research has been conducted on this topic.

The approaches like Least Squares Error (LSE) estimation and Maximum Likelihood Estimation (MLE) methods are commonly utilized approaches to determine unknown parameters of linear equations or models.

2.3.1 Least squared error (LSE) estimation algorithm and its derivatives

The methodology which is widely used to determine the unknown coefficients of linear models is the Least Squared Error (LSE) estimation algorithm, which is well-known and extensively used [84]. This technique works by minimizing the sum of the squares of the model residuals to estimate the coefficients of a system. When it comes to polynomial linear models, the LSE approach is frequently used as a curve-fitting method to ensure that the model's output matches the output data obtained from the original system. A lot of research has been conducted on the LSE and its applications, and this has been presented in [85, 86]. To solve LSE problems, methods like Singular Value Decomposition (SVD), QR factorization, and normal equations are commonly used [87-89]. Moreover, there are other variations of the LSE algorithm, including linearly constrained least squares, minimum-norm solutions to linear equations, regularized least squares, and total least squares, which have been discussed and reviewed in [90].

The LSE method for linear models is computationally simple to code and easily implementable for real-world problems, however, it has some potential drawbacks. Some of these drawbacks could be listed as follows. The LSE algorithm could be affected by noise, and consequently, it would produce inaccurate results [91]. Additionally, the algorithm may fail to find the unknown coefficients of the nonlinear models [92].

For nonlinear models, since the model has nonlinear characteristics, the linear LSE approach is not capable of providing correct results in estimating the unknown

coefficients of the dynamic model of the physical system. However, the Nonlinear Least Squared Errors (NLSE) which was first introduced in [93] is generally utilized for nonlinear system identification problems.

In the literature, various numerical algorithms, such as Steepest Descent, Newton-Raphson, Quasi-Newton, Gauss-Newton, Levenberg-Marquardt, Nelder-Mead, etc., were proposed to solve the NLSE problems [94-98]. Each of these methods has distinct characteristics with some benefits and drawbacks, which are briefly discussed as follows.

The Gradient Descent method, also known as the Steepest Descent technique, is commonly used for solving NLSE problems [99]. Another well-known iterative algorithm for NLSE problem-solving is the classical Newton-Raphson method, which is widely applied to estimate time-varying coefficients of a system's dynamic model [100]. The Hessian matrix is computed first in the classical Newton method, and then its inverse is obtained. However, calculating Hessian matrices is often challenging, and they may not always be available. In such cases, the Quasi-Newton approach is used to iteratively approximate the inverse of the Hessian matrix [101]. Successive gradients obtained from the objective function are used to estimate the inverse of the Hessian matrix in the Quasi-Newton approach, making it a fast and straightforward method that does not require the actual calculation of the Hessian matrix.

The Gauss-Newton method is another popular approach for solving NLSE problems, particularly for estimating gray system model parameters [102,103]. The Levenberg-Marquardt technique is a combination of Gauss-Newton and gradient descent approaches and is useful when both approaches' characteristics are necessary [104,105]. While the gradient descent method works well in early iterations, it becomes slower as it approaches the global optima, and Gauss-Newton produces better results near the global optima.

Additionally, derivative-free methods like the Nelder-Mead approach are employed to solve NLSE problems. The Nelder-Mead approach, also known as the simplex search algorithm, is a derivative-free method widely used in non-differentiable nonlinear problems [106]. This method does not require calculating or approximating the derivatives of the modeling problem's loss function.

2.3.2 Maximum likelihood estimation (MLE) method

The Maximum Likelihood Estimation (MLE) is another approach widely used to determine the unknown coefficients of both linear and nonlinear models. Unlike the LSE method, MLE involves constructing a likelihood function that takes into account both the observed data and unknown parameters [107]. The likelihood function is then maximized to determine the optimal solutions. The MLE method provides efficient and consistent results, especially when formulated appropriately, in contrast to the LSE method [108]. However, the MLE approach also has its drawbacks. For instance, when dealing with a small number of data samples, the MLE method may produce biased estimations that are inaccurate [109]. Additionally, implementing the MLE method may become computationally slow and challenging for large datasets and complex nonlinear models [110].

2.3.3 Parameterization using evolutionary algorithms.

The evolutionary algorithms are another type of parameter estimating approach that is widely used to find the unknown parameters of both linear and nonlinear models. These methods are population-based, and they transform the mathematical modeling problem into an optimization problem and solve it iteratively. In these methods, the objective functions are sums of squared errors, mean squared errors, etc. that have to be minimized to be smaller than the pre-set value. In other words, the errors between the actual output and the computed output utilizing the model identified with evolutionary algorithms are minimized. Approaches like Particle Swarm Optimization (PSO), Simulated Annealing (SA), Whale Optimization Algorithm (WOA), Harmony Search (HS), Genetic Algorithm (GA), etc. are considered the most well-known population-based methods [111-115]. Each of these methods has some special characteristics as well as pros and cons. In the literature, various comparative investigations are performed to compare these algorithms. Some of the applications of these methods in estimating the parameters of the mathematical models are as follows. The unknown coefficients of the nonlinear model of the multi-input, multi-output twin rotor are estimated utilizing the PSO method [116]. The GA algorithm is utilized to determine the coefficients of the nonlinear damping and stiffness model [117]. The unknown coefficients of the nonlinear models of different solar cells, such as photovoltaic modules and single and double diodes, are identified using the SA method [118]. The applications of the WOA algorithm for different purposes, such as

parameter estimation of the models of fuel cells, are investigated and reviewed in [119]. The HS method and its various applications, such as estimating parameters of nonlinear models, are discussed in [120].

In the literature, other parameter estimation methods such as moment estimation, Bayesian estimation, and so on are extensively investigated and utilized for both linear and nonlinear models [121-123].

2.4 Excitation Signals For System Identification

Once the model structure and parameterization approaches are selected, then adequate pairs of input-output data have to be provided for system identification. Data acquisition is a challenging step in system identification since the quality of the determined model strongly relies on the data obtained from the system [124]. The output data of a dynamic system is collected by exciting the system with an appropriate input signal. In the literature, different input signals are suggested for exciting real-world systems. The most well-known input for linear system identification is Pseudo-Random-Binary-Sequence (PRBS) [125]. The signal shifts periodically between two constant values [126]. Using the PBRS as an input signal helps to capture the dynamics of the system more efficiently since it includes both negative and positive changes in the input signal. In some papers, the PRBS input signals are utilized for nonlinear system identification, such as in plito plant system modeling [127] and ship maneuvering hydrodynamic modeling [128]. However, they are not appropriate for nonlinear system identification purposes as they are not able to excite all the dynamics of the systems [129]. For nonlinear system identification purposes, another variant of the PRBS signal called Pseudo-Random-Multilevel-Sequence (PRMS) signals is utilized [130-132]. The other types of excitation signals that are widely utilized for identification of the nonlinear systems are sinusoidal and multi-sinusoidal signals [133]. Some examples of the utilization of sine wave signals for system identification purposes are as follows: Xu employed a summation of two sinusoidal signals with distinct frequencies for the model identification of a second-order dynamic system [134]. In other work, multi-sine wave input signals are employed to identify nonlinear Hammerstein and Wiener models that are connected in parallel [135]. In addition, there are still some other types of signals, such as chirp signals, etc., that are proposed to be utilized as input excitation signals [136].

2.5 Model Evaluation Criteria for System Identification

The final stage of the system identification process involves using suitable statistical techniques to assess the accuracy of the identified model. Many different standards are found in the literature to assess the quality of identified models. One of the earliest and most commonly used statistical methods to verify the reliability of the identified models is by conducting cross-correlation and auto-correlation tests between the system's input and residuals [137,138]. These tests help to determine the relationships and connections between two sequences. These tests have various applications in the system identification process, some of which are:

In one study, Ling et al. utilized cross-correlation and auto-correlation tests to examine the accuracy of the ARX and ARMAX models they identified for an electro-hydraulic actuator system [139]. In another study, Shariff et al. used a statistical correlation test to verify the correctness of the linear and nonlinear ARX models they identified for a steam distillation system [140]. Other commonly used measures to determine the similarity between identified and actual model outputs, as found in the literature, include the coefficient of determination (R^2), Mean Square Errors (MSE), Root Mean Square Errors (RMSE), and Sum of Squared Errors (SSE), among others [141-143]. Furthermore, some papers use histograms to illustrate the differences between the identified model and actual model outputs. For instance, histograms were used to show the error between the identified polynomial models and the actual outputs of the wind speed model in [144] and the prediction error of the pipe flow model in [145].

To sum up, the procedure of the system identification could be summarized as follows:

The first step in the system identification process is to design an appropriate input signal to stimulate the system and obtain the corresponding output signal. This can be performed by conducting experiments or creating simulation models of the physical systems. After the data sets are prepared, the structure of the system's model needs to be identified. Different types of models, such as black-box or white-box models, can be employed for this purpose. Then, an appropriate parametrization method is utilized to determine the unknown parameters of the mathematical model. The identified model with the determined parameters is then tested with new input signals. If the model satisfies the predetermined criteria, it is chosen as the final model for the system. However, if the model fails to meet the evaluation criteria, the process is repeated from

the beginning. The system identification procedure can be illustrated using a flowchart, as shown in Figure 2.2.

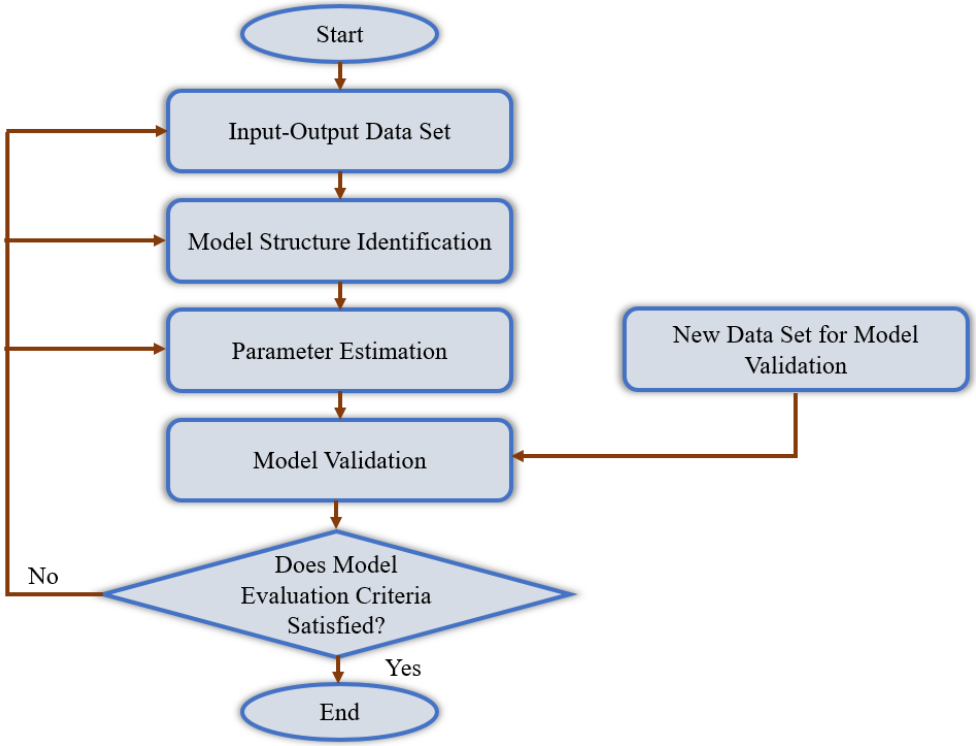


Figure 2.2 : The flowchart of the system identification.

After reviewing system identification approaches, the methodology employed in this study is explained in Chapter 3.

3. METHODOLOGY

This chapter explains the framework for creating mathematical models of physical systems with nonlinearities and verifying the accuracy of the identification process through statistical analysis. The topics covered in this chapter include: in Section 3.1, the PSO algorithm is introduced as a means of identifying mathematical models for nonlinear systems. Additionally, the NLSE estimation method is proposed in Section 3.2 as a way of determining parameters for nonlinear models. Section 3.3 covers statistical analysis methods, such as hypotheses and confidence interval tests, which are used to verify the validity of the determined parameters in the mathematical models. Finally, Section 3.4 discusses various model evaluation criteria, including the coefficient of determination (R^2) and the Mean Square Error (MSE), which can be used to further assess the quality of the models and their ability to predict the output of the system.

3.1 Particle Swarm Optimization (PSO) Algorithm

One of the most well-known nature-inspired and swarm-based optimization methodologies is the Particle Swarm Optimization (PSO) algorithm. This optimization method mimics the social behavior of birds inside a flock with the goal of reaching food [146]. A bird population approaches its goal by combining personal and population experience. They continually update their current position based on their best position in combination with the population's best position and reunite to form an optimal configuration.

Elberhart and Kennedy were motivated by the socio-psychological behavior of birds to apply this notion of social interaction to solve real-world problems [146]. This iterative swarm intelligence method is inspired by nature. It begins with a population of possible solutions. Each particle in this case indicates a possible solution to the given problem. The population is updated with each iteration by updating the velocity and position of each person. The personal best ($pBest^j$) and global best ($gBest^j$) values are used to calculate these updates. The particles in the $pBest^j$ matrix is impacted by their positions, but the particles in the $gBest^j$ matrix is influenced by the best position obtained by any member of the whole population. Finally, each particle of the population will then converge to this new position which is called the global

optima. For this purpose, the position and the velocity of the particles are repeatedly updated to converge to the global optimal value. The adjustment of velocity is a vital aspect of this algorithm since it is the fundamental mechanism utilized to alter the position of a particle to seek an ideal solution in the solution space.

Several comparative research in the literature emphasizes the advantages of the PSO algorithm over other evolutionary techniques. While searching for the global optimal value, the PSO technique outperforms the GA algorithm in terms of iteration number, accuracy, and programming simplicity [147]. As compared to the GA algorithm, the PSO algorithm produces more accurate results for tackling complex optimization problems [148]. The PSO algorithms can produce high-quality solutions in less time and with more reliable convergence to the global optimal solution than other optimization approaches [149]. According to [150], the global optima values found by the PSO algorithm are not highly affected by the initial values. The PSO algorithm has fewer tuning parameters, and the influence of parameters on solutions is thought to be less sensitive when compared to other population-based optimization algorithms [151]. Furthermore, the PSO algorithm's other characteristics, such as being derivative-free, simple to code, and easily applicable to real-world problems, make it an ideal approach for solving complex nonlinear problems.

In this research, due to the referred superiorities of the PSO algorithm, it is utilized to determine the parameters of the mathematical models of physical systems. The main steps of the PSO algorithm could be listed as follows: First, initial positions are assigned to the particles randomly. In the next step, the particle velocity (V_i^{j+1}) and position (X_i^{j+1}) are updated by the current value of particle velocity (V_i^j) and the current position of the particle (X_i^j) through equation (3.1) and equation (3.2) [152].

$$V_i^{j+1} = \chi[V_i^j + C_1(pBest^j - X_i^j) \times rnd_1 + C_2(gBest^j - X_i^j) \times rnd_2] \tag{3.1}$$

$$X_i^{j+1} = X_i^j + V_i^{j+1} \tag{3.2}$$

where, $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ are the self-confidence coefficient and swarm confidence coefficient, respectively. The $\boldsymbol{rnd_1}$ and $\boldsymbol{rnd_2}$ are the random numbers distributed between [0 1]. The particles and globally optimal solutions are represented with

$\boldsymbol{pBest^j}$ and $\boldsymbol{gBest^j}$, respectively. The inertia weight χ is defined as equation (3.3) [153].

$$\chi = \chi_{max} - \frac{\chi_{max} - \chi_{min}}{Iter_{max}} \times Iter \tag{3.3}$$

where the number of the iteration and the number of the maximum iteration are represented with *Iter* and $\boldsymbol{Iter_{max}}$. The maximum and minimum values of the inertia weight are represented as $\boldsymbol{\chi_{max}}$ and $\boldsymbol{\chi_{min}}$, respectively. The values of the $\boldsymbol{\chi_{max}}$ and $\boldsymbol{\chi_{min}}$ are originally proposed to be 0.9 and 0.4, respectively, to make a balance between local and global exploration and guarantee convergence to an adequately optimal solution with a fewer number of iterations [153].

In this thesis, the nonlinear mathematical modeling problem is solved with the PSO algorithm. For this purpose, the physical systems model parameters are considered to be global optimal solutions that have to be determined by the PSO algorithm. To employ the PSO algorithm for solving nonlinear mathematical problems, an appropriate cost function has to be formulated for the problem. In this case, the cost function of the PSO is formulated using the Root Mean Squared Error (RMSE). The RMSE represents the error between the *n* number of real model outputs ($\boldsymbol{Y_j}$) and the computed outputs ($\boldsymbol{\hat{Y}_j}$) utilizing the mathematical model with determined parameters through the PSO algorithm. In each stage of the optimization procedure, the value of the RMSE has to be minimized up to some specified threshold value. The optimal parameters of the model should be chosen such that the cost function meets the stopping criterion. The cost function or cost function of the PSO algorithm could be formulated as equation (3.4).

$$Cost\ Function = \frac{1}{n}\sum_{j=1}^{n}(Y_j - \hat{Y}_j)^2 \tag{3.4}$$

The algorithm updates the coefficients of the mathematical model of the physical systems in each iteration. Once one of the stopping criteria, including the maximum iteration or threshold value, is satisfied, then the algorithm stops. The global best ($gBest^j$) values of the parameters that were obtained in the last iteration are considered to be the determined parameters (β) of the mathematical models.

3.2 The Nonlinear Least Squared Errors (NLSE) Estimation Algorithm

When we are dealing with nonlinear models, the linear LSE algorithm fails to provide correct results in determining the coefficients of the mathematical models. Consequently, the Nonlinear Least Squared Errors (NLSE) estimation method which is a more comprehensive version of the LSE algorithm, is suggested to deal with nonlinear models. The NLSE technique iteratively refines the solutions of the problem after estimation by the linear LSE estimation approach. Generally, to solve nonlinear modeling problems, the technique employs the Levenberg-Marquardt approach. The Levenberg-Marquardt method is an iterative process that begins with the starting values of model coefficients [154,155]. The Levenberg-Marquardt is a hybrid approach that uses two distinct minimization techniques including Gauss-Newton and Steepest Descent approaches. The Gauss-Newton approach reduces the Sum of Squared Errors (SSE) by presuming the least squares equation is locally quadratic and then calculating the minimum value of the quadratic equation. On the other hand, the Steepest Descent approach reduces the SSE of the problem by changing the parameters in the gradient descent direction. The Steepest Decent approach performs perfectly when the model parameters are distant from their optimum value. Therefore, the Levenberg-Marquardt approach behaves more like the Steepest Decent approach when the model parameters are distant from their optimum value. On the other hand, the Gauss-Newton approach represents better performance when the model parameters are near their optimum values.

Hence, the Levenberg-Marquardt method behaves more like the Gauss-Newton approach when the estimation values of the parameters are near their optimum values.

The Levenberg-Marquardt approach determines the parameter vector ($\boldsymbol{\beta}$) of the nonlinear function so that the loss function ($\boldsymbol{L}(\boldsymbol{\beta})$) is minimized [156]. The loss function $\boldsymbol{L}(\boldsymbol{\beta})$ for n data samples is calculated using equation (3.5).

$$L(\beta) = \sum_{i=1}^{n} (Y_i - F(X_i, \beta))^2 \tag{3.5}$$

where Y_i and $F(X_i, \beta)$ represent the real output and calculated output of the model. In each step, the β is updated, and the new parameters are computed to be $\beta + \delta$. Then,

Taylor series expansion is employed to linearize $F(X_i, \beta + \Delta\beta)$ to obtain the variation of the parameter values ($\Delta\beta$) as shown in equation (3.6) [156].

$$F(X_i, \beta + \Delta\beta) \approx F(X_i, \beta) + J_i \Delta\beta \tag{3.6}$$

where the Jacobian matrix of the algorithm is represented with J_i and is computed by equation (2.6) [156].

$$J_i = \frac{\partial F(X_i, \beta)}{\partial \beta} \tag{3.7}$$

Then, equation (3.6) is replaced in equation (3.5), and the $L(\boldsymbol{\beta} + \Delta\boldsymbol{\beta})$ is then calculated to be as equation (3.8) [156].

$$L(\beta + \Delta\beta) = \sum_{i=1}^{n} (Y_i - F(X_i, \beta) - J_i \Delta\beta)^2 \tag{3.8}$$

The equation (3.8) could be shown as a vector notation as equation (3.9) [156].

$$\begin{aligned} L(\beta + \Delta\beta) = {} & [Y_i - F(\beta)]^T |Y_i - F(\beta)| - 2[Y_i - F(\beta)]^T J_i \Delta\beta \\ & + \Delta\beta {J_i}^T J_i \Delta\beta \end{aligned} \tag{3.9}$$

where $\boldsymbol{J_i^T J_i}$ is the approximate Hessian Matrix. By taking the derivation of the equation (3.9) concerning $\Delta\boldsymbol{\beta}$ and equalizing the result to zero, the equation (3.10) is yield [156].

$$({J_i}^T J_i)\Delta\beta = Ji^T [Y - F(\beta)] \tag{3.10}$$

J_i is rectangular and generally, it is not a square matrix [156]. However, the term ${J_i}^T J_i$ is a $m \times m$ square matrix (m is the number of the parameters). Hence, the term $(Y_i - F(\beta))$ on the right side of the equation (3.10) is a vector with the size of m. The results of the equation (3.10) yield n number of linear equations which could be solved for the δ. In the next step, the equation (3.10) is modified by Marquardt and the diagonal elements of the Hessian Matrix along with the damping coefficient (λ) are included in the equation (3.10) [157]. The revised equation which is the basis of the Levenberg-Marquardt algorithm is rewritten as equation (3.11).

$$\left({J_i}^T J_i + \lambda . diag({J_i}^T J_i)\right)\Delta\beta = {J_i}^T [Y_i - F(\beta)] \tag{3.11}$$

Using the equation (3.11), the diagonal elements of the Hessian matrix are updated in each iteration and the procedure of tuning the Hessian matrix is called damping. The parameter λ is a damping coefficient that is updated at each iteration. If the decreasing rate of the $L(\beta)$ becomes swift, then the smaller values of the λ are being utilized which makes the algorithm act like the Gauss-Newton approach. However, if the decrease in the value of the $L(\beta)$ becomes smaller which means that the residuals do not change significantly in each iteration, the algorithm uses larger λ values. It makes the algorithm behave like the Steepest Decent approach.

By rewriting equation (3.11), the value of the $\Delta\beta$ is updated in each iteration which is calculated using equation (3.12) [156].

$$\Delta\beta = {J_i}^T [Y_i - F(\beta)]\left({J_i}^T J_i + \lambda . diag({J_i}^T J_i)\right)^{-1} \tag{3.12}$$

Finally, using equation (3.12), the parameter vector is updated, and the updated parameter vector is represented with equation (3.13) [156].

$$\beta_{i+1} = \beta_i + \Delta\beta \tag{3.13}$$

When the decreasing rate of the $L(\beta)$ falls below the predefined value (termination tolerance), the iterations of the algorithm terminate, and the last obtained parameter vector becomes the solution. In this research, the NLSE method is performed for nonlinear model parameter determination which is provided in the Optimization Toolbox of MATLAB and the results are compared with those determined by the PSO algorithm.

Once the model parameters are determined, the accuracy of these parameters is examined using statistical analysis methods.

3.3 Analysis of the Results Using Statistical Tests

In general, if the model structure containing nonlinearities is available, any conventional technique or evolutionary algorithm may be employed to find the unknown coefficients of these mathematical models. Finding non-zero parameters, on

the other hand, does not imply the presence of the nonlinear term in the mathematical model of the physical system or vice versa. As a result, a thorough study of statistical verification tests is necessary to assure the identification process's trustworthiness. In the literature, there is no general approach to examining the determined parameters of the models. Meanwhile, the confidence interval and the null hypothesis tests are commonly utilized techniques in statistics science that are commonly employed to examine the credibility of the parameter of the regression [158]. In the meanwhile, they have been given less attention in the nonlinear mathematical modeling of physical systems to test the trustworthiness of the calculated parameters. In this study, the hypothesis test and confidence interval test are conducted to test the credibility of the results of the PSO algorithm. Both hypothesis and confidence interval tests are inferential procedures that rely on an estimated sample distribution. A hypothesis test employs the information from sample data to test a specified hypothesis. On the other hand, the confidence interval test approximates the models' parameters using the information extracted from sample data.

3.3.1 Statistical hypothesis test

The statistical hypothesis test states that the mathematical model parameters are equal to zero or not [159]. The stages of the hypothesis test could be described below:

First, the hypotheses equation, containing the null hypothesis ($\boldsymbol{H_0}$) and the alternative hypothesis ($\boldsymbol{H_1}$), are defined by equation (3.14) [158,159]:

$$\begin{cases} H_0: \beta_k = 0 \\ H_1: \beta_k \neq 0 \end{cases} \tag{3.14}$$

where β_k represents the model parameters of the physical system. The null hypothesis (H_0) states that the parameter β_k does not exist in the mathematical model of the physical system and it is equal to zero. It infers that the parameter has to be omitted from the model. However, the alternative hypothesis (H_1) declares that the parameter β_k exist in the mathematical model of the physical system. When H_1 is correct, it means that the H_0 is rejected and vice versa. The significance level (α) is the likelihood of rejection of the null hypothesis when it is accepted [160,161]. The significance level value which is utilized during the hypotheses test helps to determine which of the hypotheses are supported by underlying data.

Second, the standard deviations of the parameters are computed using equation (3.15) [158, 159].

$$SD_{\beta_k} = \sqrt{\sigma_{\beta_k}^2} \tag{3.15}$$

where the model parameters' variances are represented as $\sigma_{\beta_k}^2$ and their values are obtained by the diagonal components of the variance-covariance matrices.

Third, the model parameters' standard errors (SE_{β_k}) is computed using equation (3.16) for n data samples [158,159].

$$SE_{\beta_k} = \frac{SD_{\beta_k}}{\sqrt{n}} \tag{3.16}$$

Fourth, the values of the test statistic (TS_k) are computed using equation (3.17) [158, 159].

$$TS_k = \frac{\beta_k - \beta_0}{SE_{\beta_k}} \tag{3.17}$$

where the parameters assigned to the $\boldsymbol{H_0}$ are represented with $\boldsymbol{\beta_0}$ which are equal to zero.

Fifth, the probability value corresponding to each of the model parameters ($\boldsymbol{p_k}$) are computed from the models ($\boldsymbol{\Phi(Z_k)}$) under the $\boldsymbol{H_0}$ using equation (3.18).

$$p_k = 2(1 - \Phi(Z_k)) \tag{3.18}$$

Finally, using equation (3.18), the credibility of the H_0 which states that the parameter β_k of the mathematical model is equal to zero, is examined. The value of the p_k is used to assess the relevance of the parameter β_k to the model. The H_0 is rejected if the p_k of the parameter β_k is equal to or less than the value of α ($p_k \leq \alpha$), and the H_1 is accepted. In this situation, it is possible to infer that the H_1 is correct, which indicates that the nonlinear model's parameter β_k is not equal to zero. If the p_k of the parameter β_k is being calculated larger than the pre-set α value ($p_k > \alpha$), it is possible to deduce that the β_k is equal to zero and it does not exist in the mathematical model of the physical system. In other words, it infers that the corresponding nonlinear term of the parameter β_k does not exist in the system's model. In this study, the significance level

(α) is considered to be 0.01 which means that the calculated values of p_k for each of the parameters (β_k) of the model are assessed according to this value. To sum up, the hypotheses test procedure for the model parameters is demonstrated in Figure 3.1.

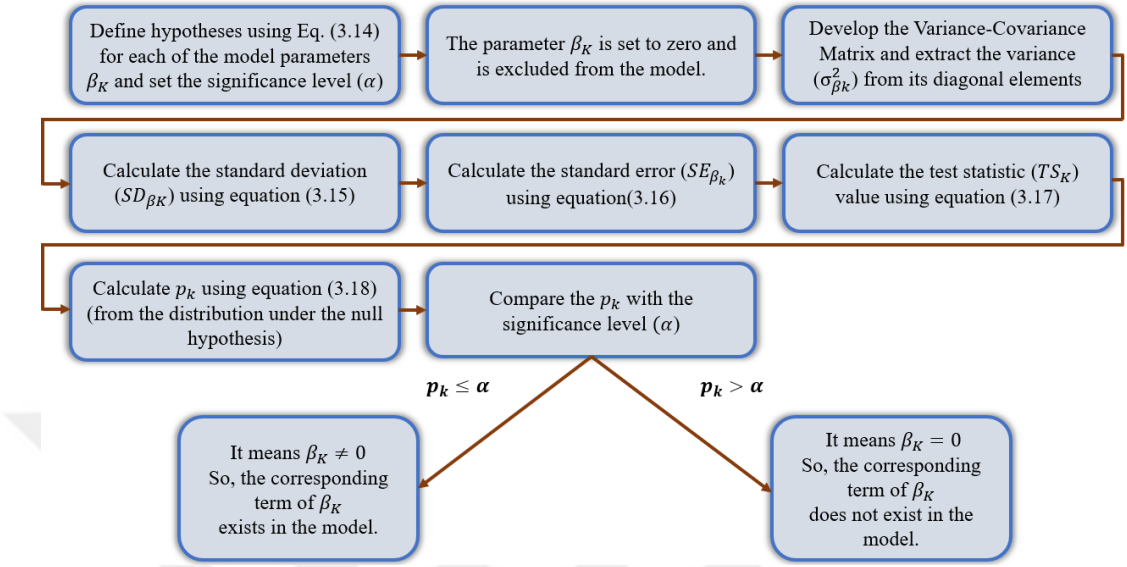


Figure 3.1 : The flowchart of the statistical hypotheses test.

3.3.2 Statistical confidence interval test

The second statistical analysis approach which is employed to test the determined parameters of the mathematical model of the systems is the confidence interval test. The confidence interval $\boldsymbol{CI_k}$ for each parameter $\boldsymbol{\beta_k}$ of the model could be calculated utilizing equation (3.19) [159].

$$CI_k = \beta_k \pm 2.58\ (SE_{\beta_k}) \tag{3.19}$$

where SE_{β_k} is the model parameters' standard errors which are calculated using equation (3.16). The parameters β_k which contain 0 values in their confidence intervals CI_k the corresponding term of the parameter β_k do not exist in the mathematical model of the system. On the contrary, if the confidence intervals CI_k of a model parameter β_k do not contain a 0 value, it could be concluded that the corresponding term of the parameter β_k exists in the dynamic model of the physical system. It could be stated that the tighter the confidence intervals of the model parameters, the more accurate the identification process. The flowchart for obtaining the confidence intervals for model parameters could be shown in Figure 3.2.

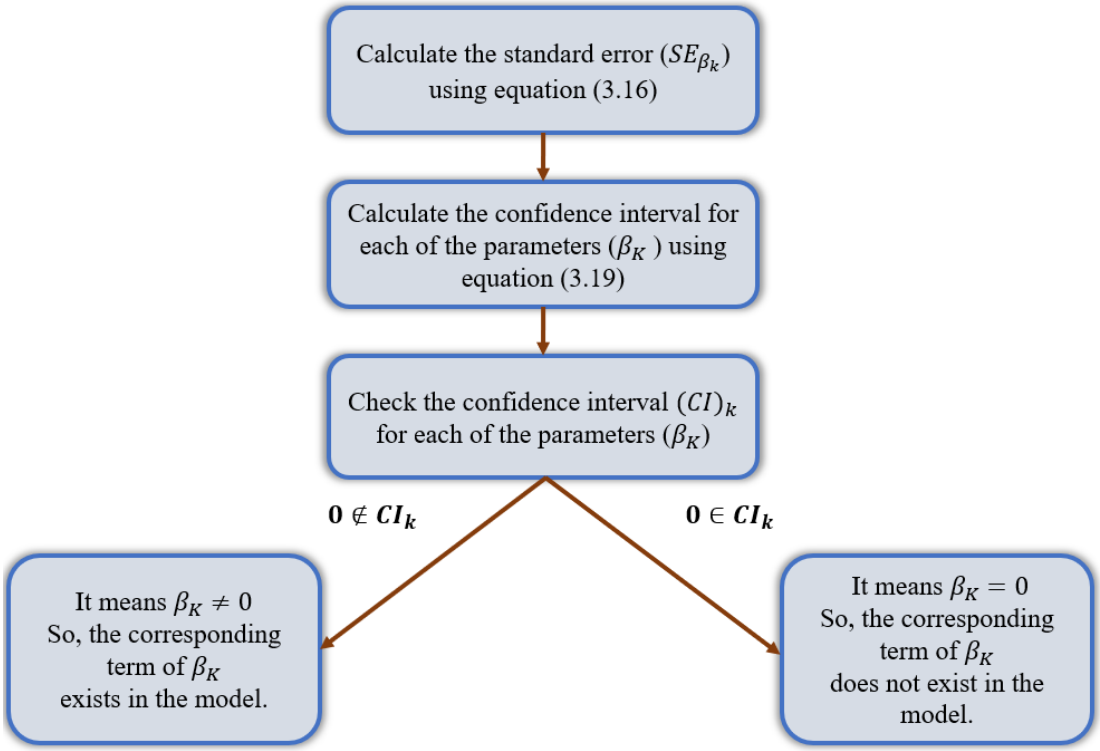


Figure 3.2 : The flowchart of the statistical confidence interval test.

3.4 Model Evaluation Criteria

Once the model and its parameters are identified, their performance in describing the underlying data has to be assessed using different performance criteria. In this study, two performance criteria, containing the coefficient of determination (R^2) and Mean Squared Errors (MSE) are employed to examine the performance of the models with determined parameters by the PSO and the NLSE algorithm. The R^2 and MSE are relative and absolute measures, respectively, that evaluate the fitness of the real and calculated outputs using identified models with determined parameters. In the following subsection, the formulation of R^2 and MSE and their importance in evaluating the models with identified parameters are explained.

3.4.1 Coefficient of determination (R^2)

The coefficient of determination (R^2) evaluates the variation that is described by a mathematical model of a physical system [162]. Particularly, it measures how well a model calculates the output of a physical system. The R^2 can have any value between 0 and 1, with 0 being the lowest and 1 being the highest. When the R^2 value is close to 1, it indicates that the model is capable of explaining more variations of the underlying

data around its mean. The values close to 0 indicate that the mathematical model does not sufficiently explain the data around the mean of the model. The R^2 is calculated for n samples of data using the Sums of Squares Errors (SSE) and Sums of Squares Total (SST) as equation (3.20) [162].

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{j=1}^{n}(Y_j - \hat{Y})^2}{\sum_{j=1}^{n}(Y_j - \bar{Y})^2} \tag{3.20}$$

where Y_j and $\hat{Y}_j$ represent the real output and the computed output of the identified models with determined parameters, respectively. The mean of the model is shown with $\bar{Y}$. The R^2 is generally represented by the percent values between 0 and 100.

3.4.2 Mean squared errors (MSE)

The Mean Square Error (MSE) is another criterion that is utilized to examine the performance of the identified models with determined parameters. The MSE value measures the magnitude of the error in calculating the actual output of a system [163]. It computes the average sums of the squared errors between the real outputs of the system and the computed outputs by obtained models with determined parameters. The smaller values of the MSE represent the accuracy of the identified model in computing the output of the system. The MSE value could be computed using equation (3.21).

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(Y_j - \hat{Y}_j)^2 \tag{3.21}$$

where Y_j and $\hat{Y}_j$ represent the real output and the computed output of the identified models with determined parameters, respectively.

4. BENCHMARK NONLINEAR MODELING PROBLEMS

The methodology explained in Chapter 3 will be utilized to solve the nonlinear modeling problem of different physical systems. For this purpose, four nonlinear modeling problems will be utilized as case studies in this study. In the first two benchmark problems, a nonlinear mathematical model of a DC motor and a solenoid actuator will be used to generate artificial data, and the suggested framework in Chapter 3 will be applied to identify and examine the mathematical models of the mechanical components. In the second group of case studies, real data is acquired from experimental setups to check the applicability of the framework for real-world nonlinear modeling problems of a ball and beam mechanism and a UR5 robotic manipulator.

4.1 Case Study 1: A Direct Current (DC) Motor

In the first benchmark problem, artificial data are generated from a simulation model of a Direct Current (DC) motor to apply the framework of nonlinear modeling and examine it using statistical analysis approaches. DC motors are electromechanical components that convert electrical energy into mechanical energy [164]. They are one of the most popular actuation elements in different industrial applications like automation systems, robotics, etc. because of their durability, mobility, and high efficiency. Building a realistic mathematical model that represents the accurate dynamical behavior of the motor is required for different purposes such as performance evaluation, controller design, etc. The DC motor's mathematical model incorporates both electrical and mechanical elements. Figure 4.1 depicts a schematic representation of a DC motor that includes both mechanical and electrical circuits.

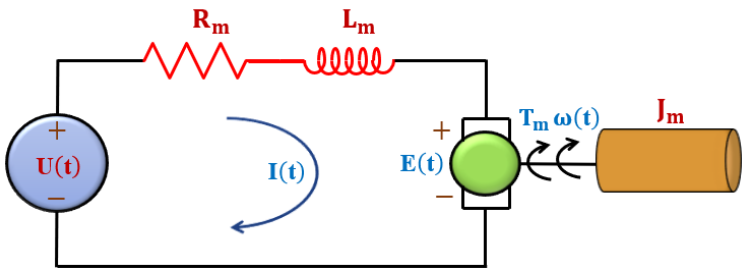


Figure 4.1 : A DC motor's schematic demonstration, which contains both mechanical and electrical circuits.

The electrical circuit's input is the voltage signal $\boldsymbol{U(t)}$ which is applied to the motor's armature [164]. The armature's inductance ($\boldsymbol{L_m}$) is connected in series with its resistance ($\boldsymbol{R_m}$) and the back electromotive force $\boldsymbol{E(t)}$. The electrical current $\boldsymbol{I(t)}$ passes through these elements. The mechanical circuit model's output is the rotor's angular velocity $\boldsymbol{\omega(t)}$. $\boldsymbol{T_m}$ and $\boldsymbol{J_m}$ represent the torque of the motor and the rotor's moment of inertia, respectively.

Using Kirchhoff's second law, the motor's electrical circuit model could be formulated as equation (4.1) [164]:

$$U(t) = R_m I(t) + L_m \frac{dI}{dt} + E(t) \tag{4.1}$$

The value of $\boldsymbol{E(t)}$ is computed by multiplying the constant ($\boldsymbol{K_m}$) by the $\boldsymbol{\omega(t)}$ as equation (4.2) [164].

$$E(t) = K_m \cdot \omega(t) \tag{4.2}$$

The torque of the motor $\boldsymbol{T_m(t)}$ is computed by Newton's second law as an equation (4.3) [164].

$$T_m(t) = J_m . \frac{d\omega}{dt} \tag{4.3}$$

The relationship between the $\boldsymbol{I(t)}$ and the $\boldsymbol{T_m(t)}$ could be represented as the equation (4.4) [164].

$$T_m(t) = K_m \, I(t) \tag{4.4}$$

Then, replacing equation (4.4) yields equation (4.5):

$$K_m \, I(t) = J_m . \frac{d\omega}{dt} \tag{4.5}$$

The common strategy in DC motor modeling is to disregard nonlinearities and create linear models using equation (4.1) - equation (4.5). The design of an effective controller for a DC motor necessitates a complete understanding of the dynamic of the system, comprising the system's linear and nonlinear characteristics.

Friction is an unavoidable feature of electromechanical systems, which is not a desirable phenomenon in the design of high-performance controllers. When a DC motor works at low speeds or operates in two directions, nonlinear frictions like Coulomb friction, static friction, viscous friction, etc. may occur which deteriorates the performance of the motor. Ignoring the effect of these phenomena leads to inaccuracies in the dynamic model of the motor. For developing a more precise mathematical model of the motor, the effect of these nonlinearities has to be considered in the general mathematical model. Therefore, the different types of nonlinear frictions that may exist in a DC motor are investigated. The general nonlinear friction ($\boldsymbol{T_f}$) model that may occur in the dynamic model of a DC is collected from the literature and is represented by the equation (4.6).

$$T_f = T_c \cdot sign(\omega) + (T_s - T_c) \cdot exp^{\left(\frac{|\omega|}{|\omega_s|}\right)} \cdot sign(\omega) + T_{v1}\omega + T_{v2} sign(\omega)\omega^2 + T_{v3}\omega^3 \quad (4.6)$$

where the static friction, Coulomb friction, and Steribeck velocity coefficients are demonstrated with $\boldsymbol{T_s}$, $\boldsymbol{T_c}$ and $\boldsymbol{\omega_s}$, respectively. The linear and nonlinear terms of viscous friction are represented with $\boldsymbol{T_{v1}}$, $\boldsymbol{T_{v2}}$, and $\boldsymbol{T_{v3}}$, respectively.

The nonlinear friction terms illustrated in equation (4.6) must be added to the right side of equation (4.5). Therefore, the general model of the mechanical circuit of a DC motor, which includes the nonlinear friction terms, could be represented as an equation (4.7).

$$K_m\, I(t) = J_m . \frac{d\omega}{dt} + T_f \quad (4.7)$$

In the next step, three different cases are examined for modeling the DC motor models.

In the first case, some linear models such as transfer function and state-space models along with nonlinear modeling methods including the NLSE and the PSO algorithm will be employed to establish the model of the motor. This is performed to examine the applicability of the linear system identification methods for nonlinear models.

In the next step, in order to develop different models of the DC motor with different combinations of nonlinear terms, two different models are developed and investigated. Precise modeling of the entire motor drive system, including nonlinear effects, is a

challenging problem. Some of the nonlinearities sometimes do not occur in the dynamic of the motor, and various combinations of the nonlinearities may exist in the dynamic model of the system.

In the second case, the model of the DC motor will incorporate all nonlinearities from equation (4.6). This indicates that in the actual mathematical model of the system, all nonlinear terms along with linear models of the DC motor will be taken into account. In this situation, equation (4.7) is rewritten as equation (4.8).

$$K_m\, I(t) = J_m.\frac{d\omega}{dt} + T_f \tag{4.8}$$

In this case, the NLSE and the PSO algorithms will be employed to determine the unknown parameters of the DC motor model including [$\boldsymbol{J_m}$, $\boldsymbol{K_m}$, $\boldsymbol{L_m}$, $\boldsymbol{R_m}$, $\boldsymbol{\omega_s^{(2)}}$, $\boldsymbol{T_c^{(2)}}$, $\boldsymbol{T_s^{(2)}}$, $\boldsymbol{T_{v1}^{(2)}}$, $\boldsymbol{T_{v2}^{(2)}}$, $\boldsymbol{T_{v3}^{(2)}}$].

In the third case, to assess the reliability of the algorithms in determining the right nonlinear components of the model, the viscous friction nonlinear terms (nonlinear terms corresponding to $\boldsymbol{T_{v2}}$ and $\boldsymbol{T_{v3}}$) are removed from the model represented by equation (4.8). In this case, equation (4.8) is rewritten as equation (4.9).

$$\begin{aligned} K_m\, I(t) = J_m.\frac{d\omega}{dt} + T_c^{(3)} \cdot sign(\omega) + \left(T_s^{(3)} - T_c^{(3)}\right) \cdot exp^{\left(\frac{|\omega|}{|\omega_s|}\right)} \\ \cdot sign(\omega) \; + T_{v1}^{(3)}\omega \end{aligned} \tag{4.9}$$

In this case, the NLSE and the PSO methods will be employed to determine the unknown model parameters including [$\boldsymbol{J_m}$, $\boldsymbol{K_m}$, $\boldsymbol{L_m}$, $\boldsymbol{R_m}$, $\boldsymbol{\omega_s^{(3)}}$,, $\boldsymbol{T_c^{(3)}}$, $\boldsymbol{T_s^{(3)}}$, $\boldsymbol{T_{v1}^{(3)}}$]. Then, the statistical analysis approaches will be applied to check the reliability of the determined parameters.

4.2 Case Study 2: A Solenoid Actuator

In the second benchmark problem, the artificial data along with experimental data are utilized to establish a mathematical model of a solenoid actuator using the methodology explained in Chapter 3. A solenoid actuator is a well-known electromagnetic component that converts electrical energy into magnetic energy [165]. It has been commonly employed in different industrial applications like refrigerators,

hydraulic valve mechanisms, fuel injection systems, the food industry, washing machines, etc. [166-168]. In addition, it has been the most preferred actuator in the research community due to its fast response time and its controllability characteristics [169, 170]. Developing an accurate mathematical model of a solenoid actuator is necessary for design, fault detection, controller design, etc. For this aim, an accurate mathematical model of a solenoid actuator that includes all possible nonlinearities, such as friction terms, is required.

Before developing a mathematical model of a solenoid actuator, the construction of a system and its working principles are investigated and explained as follows. The main construction of the system, including different parts such as a pole piece, plastic o-ring, anti-magnetic ring, working region, anti-magnetic part, coil, plunger, yoke, guider, and a push-pin, could be demonstrated in Figure 4.2.

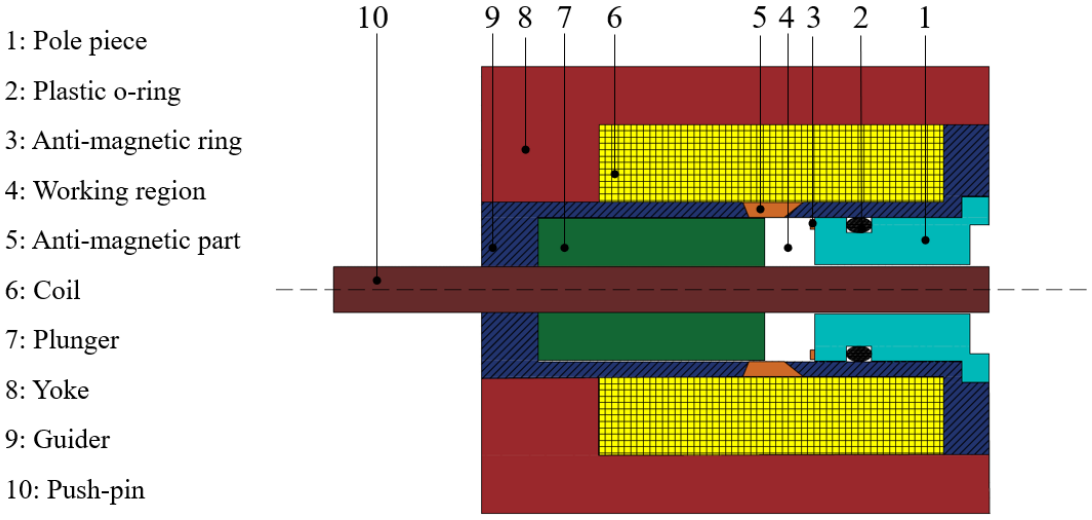


Figure 4.2 : A solenoid actuator's main construction.

The solenoid actuator working principle could be explained as follows: First, the electrical current passes through the winding of the coil, which creates a powerful magnetic field surrounding the actuator's coil [169, 170]. The developed magnetic field flows in the internal and external environments of the solenoid actuator. The strong magnetic field lines, which are normally located at the cross-section of the actuator, also pass through the guider part of the system. Then, the anti-magnetic part of the actuator's guider blocks and alters the route of the magnetic field lines on the surface of the plunger. The magnetic force is generated by magnetic lines that flow over the cross-section surface of the plunger. Then, the resulting magnetic force pushes the plunger through the working gap toward the ferromagnetic pole piece. Lastly, the

push-pin transfers the electromechanical force of the plunger to different components such as the hydraulic valve or other elements that are connected to the push-pin. When the energy of the coil is switched off, the plunger returns to its primary position by a spring located inside the guider of the system.

To compute the plunger position ($\boldsymbol{x_p}$), the second law of Newton was used as equation (4.10).

$$m_s\ddot{x}_p = F_{mag} - k_s x_p - F_f^s \tag{4.10}$$

where $\boldsymbol{m_s}$ and $\boldsymbol{k_s}$ are mechanical parameters which are representing the spring stiffness, and the moving parts' mass, respectively. $\boldsymbol{x_p}$ and $\boldsymbol{\ddot{x}_p}$ demonstrate the plunger position and the plunger acceleration, respectively. The magnetic force and friction force are shown with $\boldsymbol{F_{mag}}$ and $\boldsymbol{F_f^s}$, respectively.

The equation of the nonlinear friction force ($\boldsymbol{F_f^s}$) is expressed as equation (4.11).

$$F_f^s = F_c^s \cdot sign(\dot{x}_p) + (F_s^s - F_c^s) \cdot exp^{\left(\frac{|\dot{x}_p|}{|\dot{x}_s^s|}\right)} \cdot sign(\dot{x}_p) + F_v^s \dot{x}_p \tag{4.11}$$

where $\boldsymbol{F_c^s}$ and $\boldsymbol{F_c^s}$ are Coulomb friction constant and Static friction constant, respectively. The velocity of the plunger and Steribeck velocity are represented with $\boldsymbol{\dot{x}_p}$ and $\boldsymbol{\dot{x}_s^s}$, respectively. $\boldsymbol{F_v^s}$ denote the linear term of the viscous friction.

In a solenoid actuator, there is a nonlinear relationship between the F_{mag} and the I. In addition, the F_{mag} is affected by the x_p when the plunger moves inside the guider. Consequently, the value of F_{mag} is the function of two independent variables including I and x_p. To develop this nonlinear relationship, the following steps are performed.

First, the values of x_p and F_{mag} for a given input current of 2.7 A are acquired from the experimental setup. Then, the F_{mag} is calculated for the working region of the plunger utilizing the Finite Element Method (FEM) simulations. The result of the simulations is then validated with the experimental data acquired from the actuator. Then, using the numerical simulations, the F_{mag} is calculated for different coil current values between 0 and 2.7 A while the plunger moves inside the guider part. Using these data, a nonlinear relationship for calculating the value of F_{mag} using I and x_p is developed by the Curve Fitting toolbox of MATLAB.

Then, the NLSE and the PSO methods are employed to find the unknown model parameters of the solenoid actuator.

Finally, the statistical analysis approaches will be applied to examine the reliability of the determined parameters.

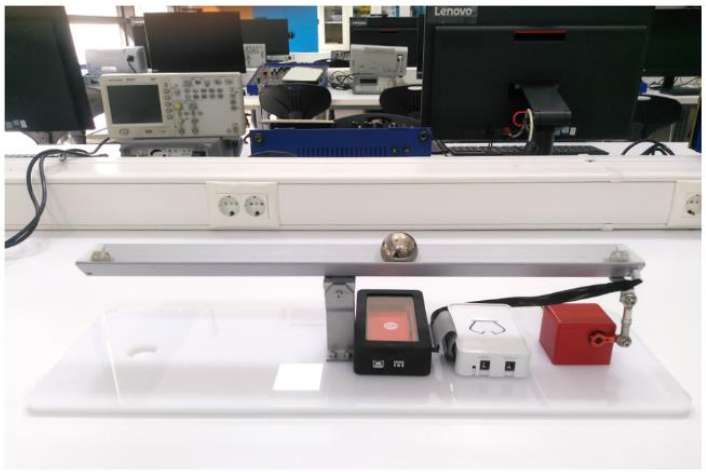
Once the methodology in this research is applied to the artificial data generated using simulation models of some mechanical systems such as a DC motor and a solenoid actuator, the experimental data from different mechanical mechanisms are acquired to check the applicability of the framework to physical systems modeling problems. For this purpose, the nonlinear model identification of a ball and beam mechanism and a UR5 robotic manipulator is investigated in the following subsections.

4.3 Case Study 3: A Ball and Beam Mechanism

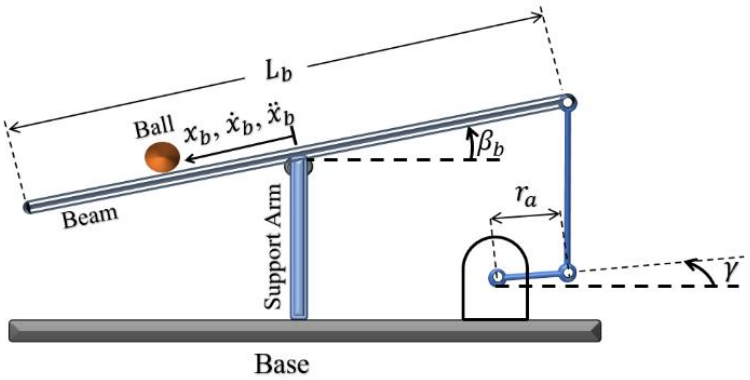
In the third benchmark problem, the real data are acquired to develop a mathematical model of a ball and beam mechanism using the methodology explained in Chapter 3.

In general, investigating the nonlinear phenomena in the modeling problems of real-world systems could not be performed in laboratory studies due to the size and complexity of the systems. For this purpose, the equivalent models or systems have to be employed to test and examine the performance of different modeling techniques. The ball and beam mechanism is an electromechanical system that is commonly employed in laboratory experiments to show the behavior of more complicated physical systems. The mechanism has various physical equivalents, such as the suspension systems of automobiles and missile balance control [171]. Because the system is very nonlinear, it is one of the most widely utilized mechanisms in control engineering to test various control schemes. [172-174]. In the literature, different research is conducted to model the system using various methods. In some of the papers, the mathematical models of the system are established utilizing the Lagrange-Euler method, and the model is linearized around the equilibrium point [175-177]. However, the relationship between the ball position and the angle of the motor could not be represented with the linear differential equations since the process is inherently nonlinear. Moreover, the friction effects are ignored in the mathematical models for simplicity. In the ball and beam mechanism, when the steel ball moves along the length of the aluminum beam, the friction force opposes the movement of the ball. Therefore,

to establish a more realistic model of the system, the friction terms have to be included in the model of the mechanism. Before developing a mathematical model of a ball and beam system, its main construction and working principles are explained as follows. The mechanism is including an aluminum beam, a steel ball, and a DC motor. The steel ball freely moves along the length (L_b) of the aluminum beam. For this purpose, the angle of the beam (β_b) is tuned using a DC motor which is connected to the beam by a connecting arm with the length of r_a. By changing the angle of the beam, the ball moves along the length of the beam. The displacement of the ball (x_b) is measured using a potentiometer which is attached to the beam. The beam angle is tuned by the angle of the gear disc (γ) which is connected to the DC motor. A ball and beam mechanism including its experimental setup (a) and its schematic representation (b) are represented in Figure 4.3.



(a)



(b)

Figure 4.3 : A ball and beam mechanism: experimental setup (a), the schematic demonstration of the system (b).

The complete equation of motion of the physical system could be derived utilizing the Euler–Lagrange approach. The Lagrange method is written as equation (4.12).

$$L_{bb} = K_{bb} - U_{bb} \tag{4.12}$$

where K_{bb} and U_{bb} represent the kinetic and potential energy of the mechanism.

The kinetic energy of the system (K_{bb}) could be written as equation (4.13).

$$K_{bb} = \frac{1}{2} J_{Beam} \dot{\beta}_b^2 + \frac{1}{2} J_b \left(\frac{\dot{x}_b}{R_b} + \dot{\beta}_b \right)^2 + \frac{1}{2} M_b (\dot{x}_b^2 + x_b^2 \dot{\beta}_b^2) \tag{4.13}$$

where R_b, and M_b represent the radius and mass of the ball, respectively. The values of R_b and M_b are measured experimentally and are assumed to be known parameters as 0.02 m and 0.261 kg, respectively. J_{Beam} represents the moment of inertia of the aluminum beam. The position and velocity of the ball are shown with x_b and $\dot{x}_b$, respectively. β_b is the angle of the beam. J_b is the moment of inertia of the ball which is considered a known parameter and is calculated using the radius and mass of the ball.

The system's potential energy (U_{BB}) could be written as equation (4.14).

$$U_{bb} = -M_b g x_b sin(\beta_b) \tag{4.14}$$

where g represents gravitational acceleration value is considered to be 9.81 m/s^2.

Then, the first-order Euler-Lagrange equation could be written as (4.15).

$$\frac{d}{dt}\left(\frac{\partial L_{bb}}{\partial \dot{x}_b}\right) - \frac{\partial L_{bb}}{\partial x_b} = -F_f^{bb} \tag{4.15}$$

where F_f^{bb} is the nonlinear friction force.

By employing equation (4.12) - equation (4.15) and performing calculations, equation (4.16) is obtained to represent the model of the ball and beam mechanism.

$$\left(\frac{J_b}{R_b^2} + M_b \right) \ddot{x}_b = M_b \mathrm{g} \sin(\beta_b) + M_b x_b \dot{\beta}_b^2 - F_f^{bb} \tag{4.16}$$

The friction force between the steel ball and the aluminum beam which is commonly ignored in the mathematical model of the physical system is shown with F_f^{bb}. The friction force is one of the most important nonlinear phenomena that occurs between the ball and the beam, and it has to be considered in the mathematical model of the physical system. The nonlinear friction F_f^{bb} including all types of friction terms which may exist in the dynamic model of the mechanism is represented by equation (4.17).

$$F_f^{bb} = F_c^{bb} \cdot sign(\dot{x}_b) + (F_s^{bb} - F_c^{bb}) \cdot exp^{\left(\frac{|\dot{x}_b|}{|\dot{x}_s^{bb}|}\right)} \cdot sign(\dot{x}_b) + F_{v1}^{bb}\dot{x}_b + F_{v2}^{bb} sign(\dot{x}_b)\dot{x}_b^2 + F_{v3}^{bb}\dot{x}_b^3 \tag{4.17}$$

where $\boldsymbol{\dot{x}_b}$ is the velocity of the ball. The Coulomb friction constant and static friction constant are shown with $\boldsymbol{F_c^{bb}}$ and $\boldsymbol{F_s^{bb}}$, respectively. The Stribeck velocity effect is demonstrated by $\boldsymbol{\dot{x}_s^{bb}}$. The viscous friction constants that capture the linear and nonlinear effects are represented by $\boldsymbol{F_{v1}^{bb}}$, $\boldsymbol{F_{v2}^{bb}}$, and $\boldsymbol{F_{v3}^{bb}}$.

Since the ball and beam mechanism represent a highly nonlinear behavior, other nonlinearities that possibly exist in the dynamics of the system have to be investigated thoroughly. For this purpose, the mathematical terms of the nonlinearities have to be included in the general model of the mechanism, and their existence in the model has to be checked. For instance, the other possible nonlinearity that may exist in the dynamic model system is the dead-zone. The dead-zone occurs when no sensible changes in the output are considered when the input of the system changes. It could happen due to the gear mechanism of the system. The non-symmetric dead-zone model is represented with equation (4.18) [178].

$$y(t) = \begin{cases} u(t) + U_l, & u(t) \le -U_l \\ 0, & -U_l < u(t) < U_r \\ u(t) - U_r, & u(t) \ge U_r \end{cases} \tag{4.18}$$

where the output and the input of the system are demonstrated by $y(t)$ and $u(t)$, respectively. U_l and U_r are constant values that represent the breakpoints.

The last phenomenon that has to be analyzed is the time-delay (T_d) that may happen in the dynamic model of the system due to the communication and digitization processes. Therefore, the presence of the time-delay has to be detected and considered in the mathematical model of the mechanism.

The reliability of the methodology explained in Chapter 3 for identification of the parameters of the models could be checked with different experimental conditions. For this purpose, the mathematical model of the ball and beam mechanism is developed for different cases. In the first case, the experiments are performed when there is direct contact between the ball and the aluminum beam surface. In the second case, however, the adhesive paper tape is attached along the surface of the beam to change the friction parameters between. The experimental setup with the adhesive tape paper attached to the beam of the mechanism is represented in Figure 4.4.

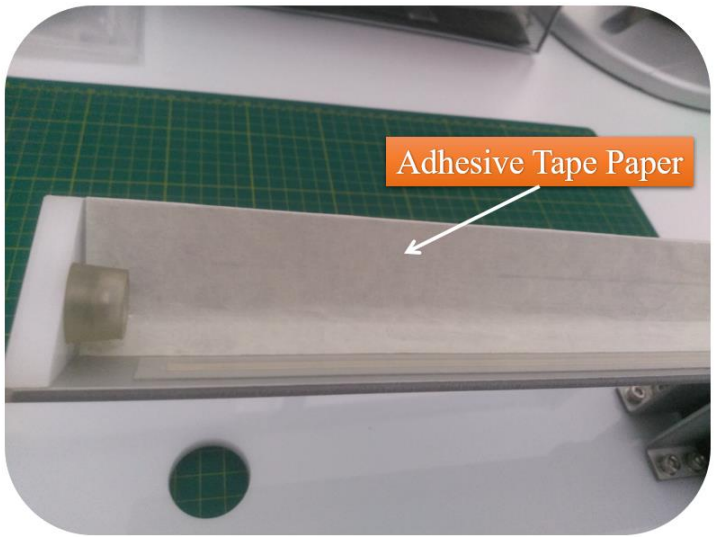


Figure 4.4 : The experimental setup with the adhesive tape paper attached to the beam.

In the next subsection, the fourth case study which is a UR5 manipulator will be investigated.

4.4 Case Study 4: A UR5 Robotic Manipulator

In the fourth benchmark problem, the real data are experimentally acquired to establish a dynamic model of a UR5 robotic arm utilizing the methodology explained in Chapter 3.

Developing dynamic models of robotic arms is crucial for different purposes such as simulation applications, motion planning, sensor-less control, etc. [179,180]. The dynamics model of the manipulators is commonly expressed as a second-order differential equation derived employing the LaGrange approach or the recursive Newton-Euler method [181,182]. In the literature, different methodologies are

suggested to develop dynamic models of robotic arms, such as a UR5 manipulator. In [183], Guo et al. utilized the gradient approach to find the parameters of the dynamic model of a UR5 arm. In their research, the UR5 manipulator is considered a system that operates only on a 2D plane, while its end-effector performs only a movement along the z-axis. In other work, a semi-parametric Deep Learning algorithm is employed to model the UR5 manipulator's inverse dynamics. In this work, the torque values of each joint of the manipulator are predicted using the joints' positions, velocities, and accelerations [184]. Panda et al. developed a mathematical model of a UR5 arm by a multi-layer artificial neural network model. In this paper, the simulation data is utilized to train and test the established ANN model [185]. In the reviewed studies, some black-box modeling methods such as ANN methods are utilized to develop a dynamic model of the UR5 manipulator, which does not represent physical laws or phenomena in the system. In addition, these models do not represent the possible nonlinearities that may occur in the manipulator's joints. Therefore, an accurate dynamic model of a UR5 robotic manipulator, which includes all nonlinearities such as friction terms in the joints of the robot, will be developed using the methodology explained in Chapter 3. However, before developing a dynamic model of the manipulator, its characteristics and working principles are explained as follows:

The UR5 arm is one of the UR series robots of Universal Robotics, which are widely utilized for educational and industrial purposes. The robot is quick, simple to program, adaptable, and safe [186]. Additionally, access to the robot controller is easily possible with low-level programming. The UR5 robotic manipulator is a 6-degree-of-freedom arm made up of six revolute joints divided into two groups. The first set of joints includes bigger motors for the first three joints comprising the base, shoulder, and elbow, however, the second set contains smaller motors for the remaining three wrist joints. Each joint of the robot is equipped with a brushless motor and mechanical strain wave gearing. The experimental setup of the UR5 robotic arm (a) and schematic representation of the system (b) are demonstrated in Figure 4.5.

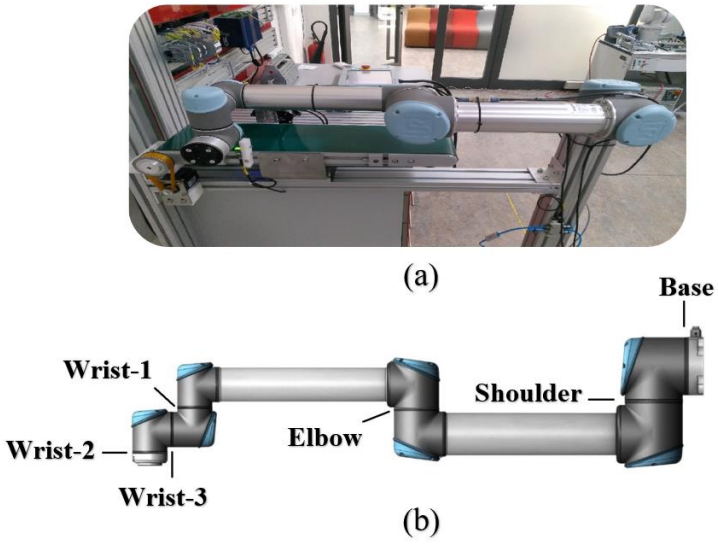


Figure 4.5 : A UR5 robotic arm: (a) experimental setup, (b) a schematic representation.

A process of system identification for a robot includes dynamic model development, generation of the trajectory, data acquisition from the robot, preprocessing of the data to remove noise, the finding of the unknown parameters of the dynamic model, and validation of the model with new trajectory data [187,188]. The performance of the robot is directly affected by the dynamic model of the manipulator. The explicit understanding of a robot's dynamic model is critical since it may be used for a range of applications such as model-based controller design, sensorless force estimation, and so on. The dynamic model of the manipulator has to be established to calculate the torques of the joints as a function of their positions, velocities, and accelerations. The joint torques ($\boldsymbol{\tau}$) of a robot could be computed by the Euler-LaGrange approach through the equation (4.19) by employing the coordinates of the joints as generalized coordinates [189].

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{4.19}$$

where $\mathrm{M(q)}$, $\mathrm{C(q,\dot{q})}$, and $\mathrm{G(q)}$ demonstrate the inertia matrix, Coriolis and centrifugal matrix, and gravity vector, respectively. The elements of these matrices are calculated using data provided by the manufacturer and are assumed to be known values. The joints' angles, velocities, and accelerations are shown as q, $\dot{\mathrm{q}}$ and $\ddot{\mathrm{q}}$, respectively.

To establish the dynamic models of the UR5 arm, the following steps are taken.

First, the Denavit-Hartenberg (DH) approach is employed to develop a kinematic model of the UR5 manipulator [189]. Using this approach, the tip-point position, and its orientation in accordance with the reference coordinate frame ($o_0x_0y_0z_0$) are computed parametrically in terms of joint angles (q_i). For this purpose, the coordinate frames ($o_ix_iy_iz_i$ where $i = 0 - 6$) are assigned the UR5 robotic manipulator, and the Denavit-Hartenberg convention is applied to represent the DH parameters of the manipulator (q_i, α_i, a_i, and d_i where $i = 1 - 6$). The schematic representation of the UR5 manipulator along with the assigned frames to its joints is demonstrated in Figure 4.6. The DH parameters of the UR5 robotic arm, which are given by the manufacturer (Universal Robotics), are shown in Table 4.1 [URL - 1].

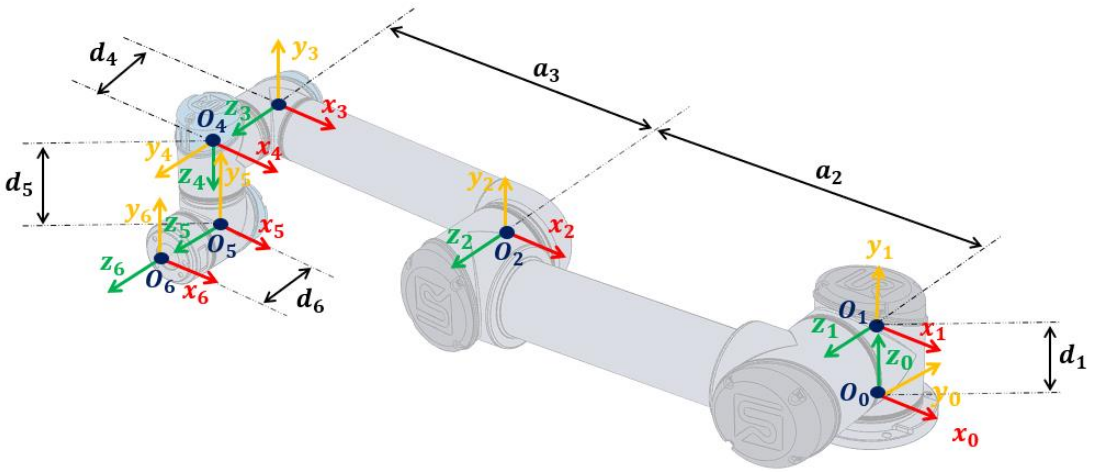


Figure 4.6 : The placement of the coordinate frames to each joint of the UR5 robotic manipulator along with its DH parameters.

Table 4.1 : The UR5 robotic arm's DH parameters [URL - 1].

	$\theta_i\ (rad)$	$d_i(m)$	$a_i(m)$	$\alpha_i(rad)$
1	q_1	0.089159	0	$\pi/2$
2	q_2	0	-0.425	0
3	q_3	0	-0.39225	0
4	q_4	0.10915	0	$\pi/2$
5	q_5	0.09465	0	$-\pi/2$
6	q_6	0.0823	0	0

In the next step, the transformation matrices of the UR5 manipulator, are obtained using equation (4.20) [189].

$$T_i^{i-1} = \begin{bmatrix} Cos(q_i) & -in\,(q_i)Cos(\alpha_i) & Sin(q_i)Sin(\alpha_i) & a_iCos(q_i) \\ Sin\,(q_i) & Cos(q_i)Cos(\alpha_i) & -Cos(q_i)Sin(\alpha_i) & a_iSin(q_i) \\ 0 & Sin(\alpha_i) & Cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_i^{i-1} & t_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (4.20)$$

Where R_i^{i-1} and t_i^{i-1} represent the rotation matrices and translation vectors between frame $i-1$ to frame i. Using equation (4.20) along with the information provided in Table 4.1, the transformation matrices of the UR5 manipulator could be calculated as equation (4.21) - equation (4.26).

$$T_1^0 = \begin{bmatrix} R_1^0 & t_1^0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Cos(q_1) & 0 & Sin(q_1) & 0 \\ Sin\,(q_1) & 0 & -Cos(q_1) & 0 \\ 0 & 1 & 0 & 0.089159 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.21)$$

$$T_2^1 = \begin{bmatrix} R_2^1 & t_2^1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Cos(q_2) & -Sin\,(q_2) & 0 & -0.425Cos(q_2) \\ Sin\,(q_2) & Cos(q_2) & 0 & -0.425Sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

$$T_3^2 = \begin{bmatrix} R_3^2 & t_3^2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Cos(q_3) & -Sin\,(q_3) & 0 & -0.39225Cos(q_3) \\ Sin\,(q_3) & Cos(q_3) & 0 & -0.39225Sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.23)$$

$$T_4^3 = \begin{bmatrix} R_4^3 & t_4^3 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Cos(q_4) & 0 & Sin(q_4) & 0 \\ Sin\,(q_4) & 0 & -Cos(q_4) & 0 \\ 0 & 1 & 0 & 0.10915 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.24)$$

$$T_5^4 = \begin{bmatrix} R_5^4 & t_5^4 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Cos(q_5) & 0 & -Sin(q_5) & 0 \\ Sin\,(q_5) & 0 & Cos(q_5) & 0 \\ 0 & -1 & 0 & 0.09465 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.25)$$

$$T_6^5 = \begin{bmatrix} R_6^5 & t_6^5 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Cos(q_6) & -Sin\,(q_6) & 0 & 0 \\ Sin\,(q_6) & Cos(q_6) & 0 & 0 \\ 0 & 0 & 1 & 0.0823 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.26)$$

Then, the equations for calculating the transformation matrices transforming between each coordinate frame i to the reference coordinate frame could be easily calculated as equation (4.27) - equation (4.32).

$$T_1^0 = T_1^0 = \begin{bmatrix} R_1^0 & t_1^0 \\ 0 & 1 \end{bmatrix} \tag{4.27}$$

$$T_2^0 = T_1^0 T_2^1 = \begin{bmatrix} R_2^0 & t_2^0 \\ 0 & 1 \end{bmatrix} \tag{4.28}$$

$$T_3^0 = T_1^0 T_2^1 T_3^2 = \begin{bmatrix} R_3^0 & t_3^0 \\ 0 & 1 \end{bmatrix} \tag{4.29}$$

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \begin{bmatrix} R_4^0 & t_4^0 \\ 0 & 1 \end{bmatrix} \tag{4.30}$$

$$T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 = \begin{bmatrix} R_5^0 & t_5^0 \\ 0 & 1 \end{bmatrix} \tag{4.31}$$

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 = \begin{bmatrix} R_6^0 & t_6^0 \\ 0 & 1 \end{bmatrix} \tag{4.32}$$

After calculating the kinematics of the UR5 arm at the position level, the robot's kinematics at the velocity level is obtained. For this purpose, the linear and the angular velocities of the tip-point are computed parametrically in terms of the velocities of the joints of the arm. The equations for obtaining the linear velocity of the tip-point (v_{tp}) and angular velocity of the tip-point (ω_{tp}) are represented as equation (4.33) and equation (4.34), respectively [189].

$$v_{tp} = J_{V_{tp}} \dot{q} \tag{4.33}$$

$$\omega_{tp} = J_{\omega_{tp}} \dot{q} \tag{4.34}$$

where $J_{V_{tp}}$ and $J_{\omega_{tp}}$ are geometric Jacobians of the tip-point where each of them is a (3×6) matrix for the UR5 manipulator. $\dot{q}$ is a (6×1) matrix that contains the velocities of the joints of the robot. The Jacobian matrix (J_{tp}) of the tip-point is computed utilizing equation (4.35) [189].

$$J_{tp} = \begin{bmatrix} J_{v_{tp}} \\ J_{\omega_{tp}} \end{bmatrix} = \begin{bmatrix} \frac{\partial t_6^0}{\partial q_i} \\ z_{i-1} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (t_6^0 - t_{i-1}^0) \\ z_{i-1} \end{bmatrix} \quad (4.35)$$

t_6^0 and t_{i-1}^0 are the position vectors which are extracted from the previously obtained equation (4.27)- equation (4.32). The vectors of z_{i-1} are calculated using equation (4.36) [189].

$$z_{i-1} = R_1^0 \cdots R_{i-1}^{i-2} Z_0 \quad (4.36)$$

where Z_0 is equal to the matrix $[0 \quad 0 \quad 1]^T$. R_1^0 and R_{i-1}^{i-2} represent the rotation matrices that are extracted using previously obtained equation (4.21) - equation (4.26). The z_{i-1} is a column vector with 3×1 elements that are calculated by equation (4.37) - equation (4.42).

$$z_0 = Z_0 \quad (4.37)$$

$$z_1 = R_1^0 Z_0 \quad (4.38)$$

$$z_2 = R_1^0 R_2^1 Z_0 \quad (4.39)$$

$$z_3 = R_1^0 R_2^1 R_3^2 Z_0 \quad (4.40)$$

$$z_4 = R_1^0 R_2^1 R_3^2 R_4^3 Z_0 \quad (4.41)$$

$$z_5 = R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 Z_0 \quad (4.42)$$

Finally, using the obtained information, the Jacobian matrix of the tip-point is developed using equation (4.43) and equation (4.44).

$$J_{v_{tp}} = [z_0 \times (t_6^0 - t_0^0) \quad z_1 \times (t_6^0 - t_1^0) \quad \cdots \quad z_5 \times (t_6^0 - t_5^0) \quad]_{3\times 6} \quad (4.43)$$

$$J_{\omega_{tp}} = [Z_0 \quad R_1^0 Z_0 \quad R_1^0 R_2^1 Z_0 \quad R_1^0 R_2^1 R_3^2 Z_0 \quad \cdots \quad R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 Z_0]_{3\times 6} \quad (4.44)$$

Then, using the Jacobian matrices ($J_{v_{tp}}$ and $J_{\omega_{tp}}$), the liner and the angular velocities of the tip-point are calculated parametrically in terms of the velocities of the joints of the robot ($\dot{q}_i$) as mentioned in equation (4.33) and equation (4.34), respectively.

Then, the Jacobian matrices are calculated for the mass center of the links of the UR5 manipulator, in the same way, using equation (4.45) [189].

$$J_{v_{mci}} = \begin{bmatrix} J_{v_{mci}} \\ J_{\omega_{mci}} \end{bmatrix} = \begin{bmatrix} \frac{\partial r^0_{mc_i}}{\partial q_i} \\ z_{i-1} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (r^0_{mc_i} - t^0_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{4.45}$$

where $r^0_{mc_i}$ represents the mass center of the link i with respect to the reference coordinate frame ($o_0 x_0 y_0 z_0$). The 3×1 column vector of the $r^0_{mc_i}$ for all the inks of the UR5 are calculated using equation (4.46) - equation (4.51).

$$r^{(0)}_{mc_1} = t^0_1 + R^0_1 mc_1 \tag{4.46}$$

$$r^{(0)}_{mc_2} = t^0_2 + R^0_2 mc_2 \tag{4.47}$$

$$r^{(0)}_{mc_3} = t^0_3 + R^0_3 mc_3 \tag{4.48}$$

$$r^{(0)}_{mc_4} = t^0_4 + R^0_4 mc_4 \tag{4.49}$$

$$r^{(0)}_{mc_5} = t^0_5 + R^0_5 mc_5 \tag{4.50}$$

$$r^{(0)}_{mc_6} = t^0_6 + R^0_6 mc_6 \tag{4.51}$$

where mc_i $(i = 1 - 6)$ is the mass center of the links which are provided by the manufacturer. The properties of the link i of the UR5 manipulator including mass (m_i), and mass center (mc_i) are shown in Table 4.2

Table 4.2 : The links properties of the UR5 robotic manipulator [URL - 1].

Link	$m_i\ (Kg)$	$mc_i\ (m)$
1	3.7	$[0, -0.02561, 0.00193]$
2	8.393	$[0.2125, 0, 0.11336]$
3	2.33	$[0.15, 0, 0.0265]$
4	1.219	[0, -0.0018, 0.01634]
5	1.219	[0, 0.0018,0.01634]
6	0.1879	[0, 0, -0.001159]

By inserting equation (4.37) - equation (4.42) and equation (4.46) - equation (4.51) into equation (4.45), the Jacobian matrices for the center of the masses of the links are calculated. After calculating the kinematics of the UR5 manipulator at both position

and velocity levels, the inverse dynamic model of the robotic arm is established. Then, the obtained kinematic information along with the information given by the robot's manufacturer is then utilized to establish a dynamic model of the UR5 manipulator.

Then, the dynamic model of the robotic arm is established to compute the torques of the joints (τ) using equation (4.52).

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{4.52}$$

where $M(q)$, $C(q,\dot{q})$, and $G(q)$ demonstrate the inertia matrix with 6×6 elements, the Coriolis and centrifugal matrix with 6×6, and a 6×1 gravity vector, respectively. The elements of these matrices are calculated using the kinematic data and the data provided by the manufacturer. The positions, velocities, and accelerations of the joints are 6×1 matrices that are represented as q, $\dot{q}$ and $\ddot{q}$, respectively. The values of q and $\dot{q}$ are attained from the measurements, and the values of $\ddot{q}$ are obtained from the numerical differentiation of the velocities of the joints. The joint torques of the robot (τ) are computed by the multiplication of the motor constants and gear ratios of each joint.

The inertia matrix ($M(q)$) of the UR5 manipulator is obtained using equation (4.53).

$$M(q) = \sum_{i=1}^{6} \left[m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^T R_i^0 I_i {R_i^0}^T J_{\omega_i} \right] \tag{4.53}$$

where J_{v_i} and J_{ω_i} are Jacobian matrices, which are previously calculated by equation (4.45). m_i represent the mass of the links ($i = 1 - 6$) of the UR5 robot, that is provided by the robot manufacturer. The transposition of the Jacobian matrices is demonstrated by $J_{v_i}^T$ and $J_{\omega_i}^T$. The rotation matrices R_i^0 are extracted from equation (4.27) - equation (4.32) and their transpose matrices are represented by ${R_i^0}^T$. The inertia tensors of the links of the UR5 manipulator are extracted from CAD models provided by the manufacturer and are shown by equation (4.54) - equation (4.59).

$$I_1^{(y)} = \begin{bmatrix} 0.0085 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0085 \end{bmatrix} \tag{4.54}$$

$$I_2^{(x)} = \begin{bmatrix} 0.0085 & 0 & 0 \\ 0 & 0.212 & 0 \\ 0 & 0 & 0.212 \end{bmatrix} \quad (4.55)$$

$$I_3^{(x)} = \begin{bmatrix} 0.0017 & 0 & 0 \\ 0 & 0.0473 & 0 \\ 0 & 0 & 0.0473 \end{bmatrix} \quad (4.56)$$

$$I_4^{(z)} = \begin{bmatrix} 0.0018 & 0 & 0 \\ 0 & 0.0018 & 0 \\ 0 & 0 & 0.0014 \end{bmatrix} \quad (4.57)$$

$$I_5^{(z)} = \begin{bmatrix} 0.0018 & 0 & 0 \\ 0 & 0.0018 & 0 \\ 0 & 0 & 0.0014 \end{bmatrix} \quad (4.58)$$

$$I_6^{(z)} = \begin{bmatrix} 0.00008 & 0 & 0 \\ 0 & 0.00008 & 0 \\ 0 & 0 & 0.00013 \end{bmatrix} \quad (4.59)$$

In the next step, the component c_{kj} of the Coriolis and centrifugal acceleration matrix ($C(q, \dot{q})$) of the UR5 manipulator is calculated using equation (4.60) [189].

$$c_{kj} = \sum_{i=1}^{6} \frac{1}{2} \left[\frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{kj}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k} \right] \dot{q}_i \quad (4.60)$$

The gravity vector ($G(q)$) of the UR5 manipulator is then calculated using equation (4.61) [189].

$$G(q) = \left(\frac{\partial u}{\partial q}\right)^T \quad (4.61)$$

Where u is the total potential energy of the UR5 robot which could be calculated using equation (4.62).

$$u(q) = \sum_{i=1}^{6} g^T r_{mc_i}^{(0)} m_i \quad (4.62)$$

where g is a vector that represents the direction of gravity with respect to the reference frame. For the UR5 manipulator, the vector could be represented by vector [0 0 -9.81].

$r_{mc_i}^{(0)}$ is the coordinate of the mass center of the link i with respect to the reference frame which is previously obtained by equation (4.46) - equation (4.51).

Finally, by inserting M(q), $C(q,\dot{q})$, and $G(q)$ into equation (4.52), the dynamic model of the robotic arm is completed. However, to develop an accurate dynamic model of the UR5 arm, the nonlinearities that affect the performance of the robot, have to be added to the model. Friction is the most well-known nonlinear term that adversely affects the performance of the robot. Different frictions may occur in the motor and strain wave gearing mechanism of each of the joints of the robot. For this purpose, an extra term describing the friction terms (τ_f) have to be added to equation (4.52) which is shown as equation (4.63).

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f \tag{4.63}$$

Then, friction τ_f that may occur in the robot joints could be represented as the equation (4.64).

$$\tau_f = \tau_c \cdot sign(\dot{q}) + (\tau_s - \tau_c) \cdot exp^{\left(\frac{|\dot{q}|}{|\dot{q}_s|}\right)} \cdot sign(\dot{q}) + \tau_{v1}\dot{q} + \tau_{v2} sign(\dot{q})\dot{q}^2 + \tau_{v3}\dot{q}^3 \tag{4.64}$$

Because multiple nonlinear frictional force configurations such as Coulomb, static, Steribeck, linear, and nonlinear viscous frictions may arise when the robot is working, identifying the specific type of friction that exists in the joint model is critical. An accurate dynamics model of the robots has often required the determination of accurate parameters for these nonlinear phenomena. To examine the applicability of the methodology explained in Chapter 3 for real systems, the unknown parameters of the fictional parameters for each joint of the robot will be determined and examined.

To establish a dynamic model of the manipulator, the real data are acquired from the experimental setup. The procedure of data acquisition using an experimental setup could be described as follows. The UR-ROS driver, supplied by the ROS (Robot Operating System), is used to connect to the UR5 robotic manipulator. The trajectories with the random frequencies are designed in MATLAB at waypoints which are sampled at 100 Hz. Two trajectories are generated for model identification and validation stages. In the next step, the Joint Trajectory Action client is used to send the

trajectories including waypoints to the manipulator. As the manipulator follows the provided trajectories, motor variables, in conjunction with the current values of the motors, are obtained by the ROS driver. Then numerical differentiation of the joints' velocities is conducted to calculate the acceleration profiles of the joints. To compute the torque values of the joints of the robotic arm, the current obtained from the motors is multiplied by the motors' torque constant and their gear ratio. The data acquisition from the UR5 manipulator is schematically demonstrated in Figure 4.7.

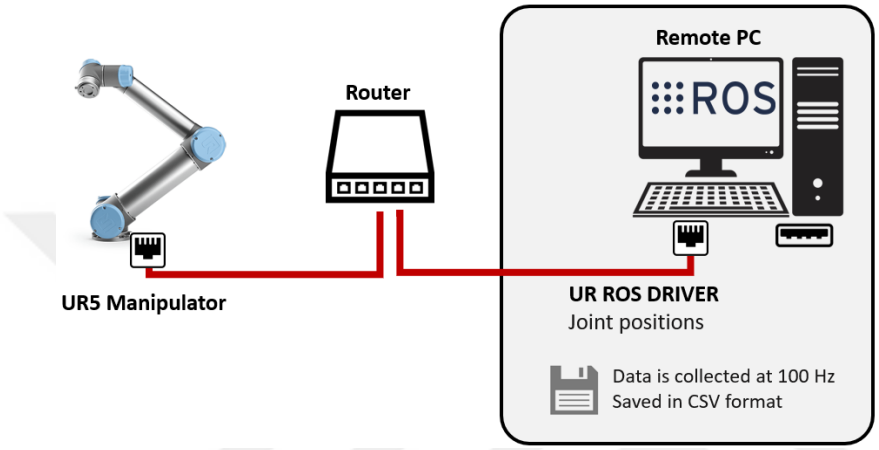


Figure 4.7 : The data acquisition setup’s schematic representation for the UR5 manipulator.

The result and discussion of this study are provided in Chapter 5.

5. RESULTS AND DISCUSSION

The results of the modeling of the real-world mechanical systems along with their interpretations are provided in this chapter. First, the results of the modeling of the DC motor for different cases are provided, and their comparisons are performed in sub-section 5.1. Second, the results of the modeling of the solenoid actuator are provided and discussed in sub-section 5.2. Third, the results of the modeling of the ball and beam mechanism are presented and evaluated in sub-section 5.3. Finally, the results of the modeling of the UR5 robotic manipulator are provided and discussed in sub-section 5.4.

5.1 Case Study 1: Modeling of a DC Motor

To establish a DC motor's mathematical model, the Simulink environment of the MATLAB toolbox is employed. The general Simulink model using equation (4.1) - equation (4.9) including friction nonlinearities is developed and is demonstrated in Figure 5.1.

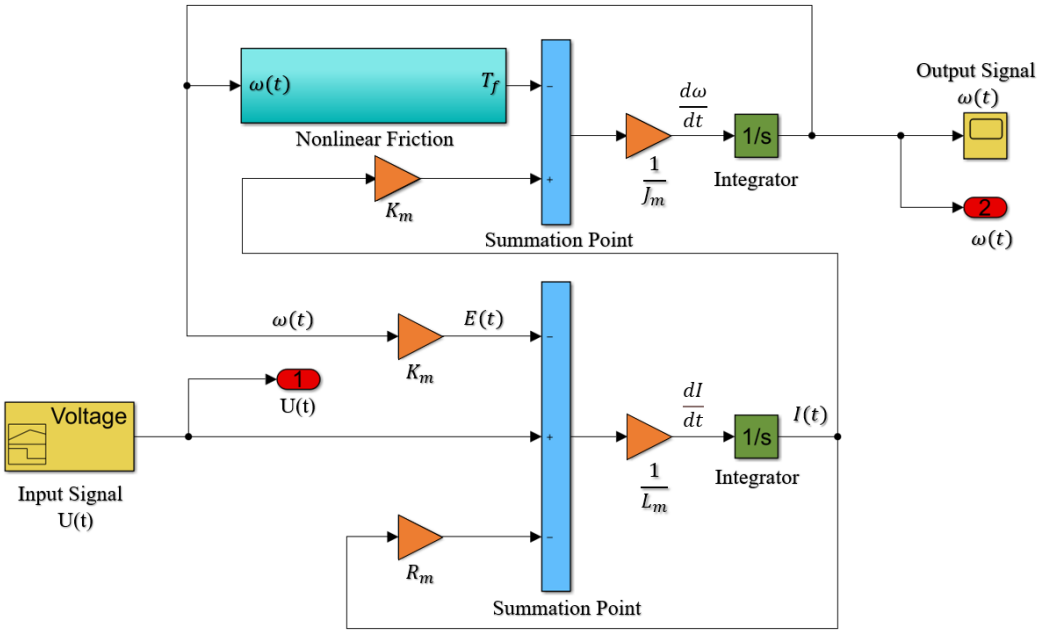


Figure 5.1 : The Simulink model of a DC motor with friction nonlinearities.

Once the mathematical model of a DC motor is established, the simulation data including the input and output datasets are generated through the developed model. In order to make the output signals more realistic, an appropriate white Gaussian noise is added to the output signal of the Simulink model. Consequently, the artificial data and

simulation model of a DC motor along with the parameterization algorithms and statistical methods will be employed to find the model parameters.

The results of the modeling of the DC motor for three cases are provided and discussed as follows. In the first case, the results of different linear and nonlinear modeling methods for representing the output of the DC motor are provided and discussed. In the second case, the results of different nonlinear modeling methods are provided under the condition that all the nonlinearities exist in the model of the DC motor. In the third case, the results of the nonlinear modeling of the DC motor are provided and discussed under the condition that some of the nonlinearities exist in the model.

5.1.1 Case one

In case one, some linear modeling methods, such as the transfer function, the state-space model, and nonlinear modeling methods, including the NLSE and the PSO algorithm are provided. This case is performed to examine the applicability of the linear system identification methods for modeling nonlinear systems. Initially, a Pseudo-Random-Multilevel-Sequence (PRMS) signal with different levels of voltage is generated as an input signal to the model. A PRMS signal between -5 and 5 V that covers all operating regions of the DC motor is generated and represented in Figure 5.2.

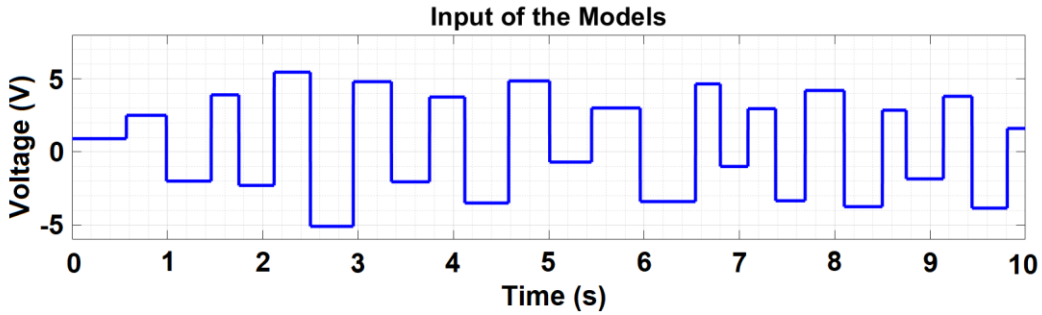


Figure 5.2 : The PRMS voltage signal for identification of the model of the DC motor.

Then, a simulation model that contains different nonlinear friction terms is employed to generate the output signal for the model. In order to provide a more realistic output signal, the white Gaussian noise (28 $\boldsymbol{dbW}$ SNR) is added to the $\boldsymbol{\omega(t)}$. Two linear models, including the transfer function and the state-space models, are employed for modeling the DC motor using artificially generated input-output data. For this purpose, the System Identification Toolbox of MATLAB is used. Since the model of the DC

motor is a second-order model, a transfer function model ($\boldsymbol{G(s)}$) is obtained as equation (5.1).

$$G(s) = \frac{31.9}{s^2 + 34.41\,s + 39.77} \tag{5.1}$$

A second-order state-space model is obtained and is represented by equation (5.2) and equation (5.3).

$$\dot{x}(t) = \begin{bmatrix} 0.187 & -3.406 \\ 6.686 & -14.79 \end{bmatrix} x(t) + \begin{bmatrix} 0.0015 \\ -0.068 \end{bmatrix} u(t) \tag{5.2}$$

$$y(t) = [57.31 \quad 1.267]x(t) \tag{5.3}$$

Then, the NLSE and the PSO algorithms are employed to determine the nonlinear model parameters of the system.

The real output of the mathematical model and the outputs calculated using the linear models (transfer function and state-space models) and the nonlinear models identified using the NLSE and the PSO algorithms, as well as their absolute errors, are computed and demonstrated in Figure 5.3.

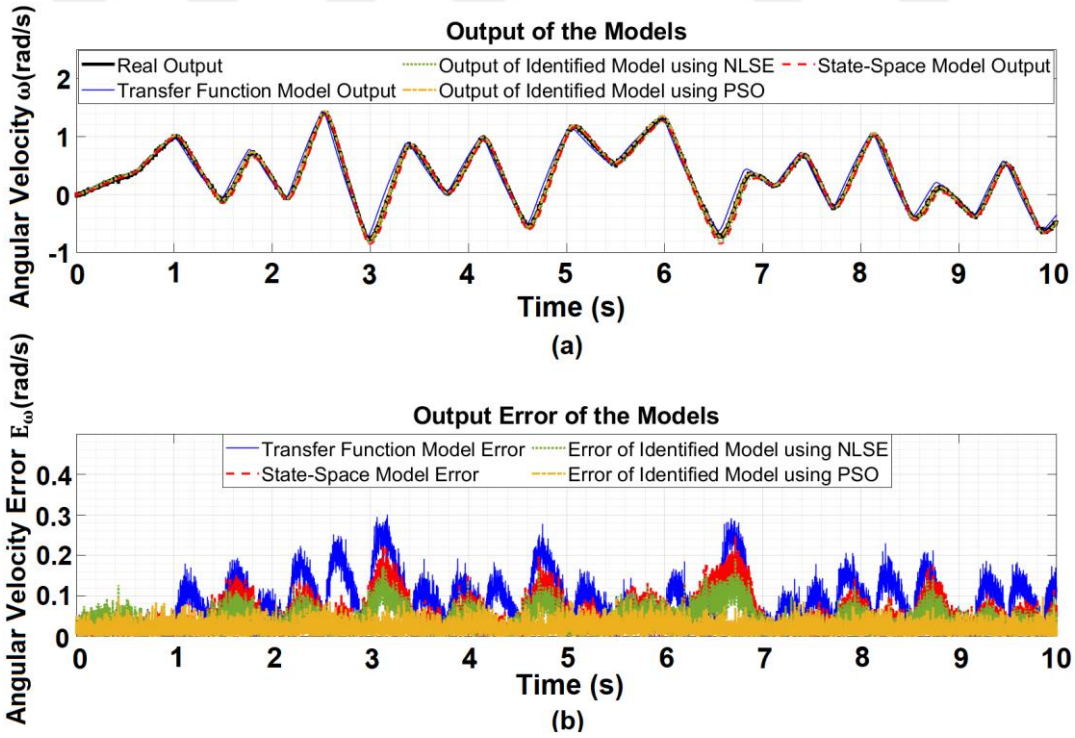


Figure 5.3 : The real output of the DC motor model and the output of linear and nonlinear models (case one) (a), and the absolute error of the output of the models (case one) (b).

The Mean Squared Errors (MSE) and $\boldsymbol{R^2}$ values of the models are computed and represented in Table 5.1.

Table 5.1 : The MSE and R^2 values for the output of the models of DC motor (Case one).

Model	MSE (rad/s)	R^2 (%)
Transfer function	0.0216	78.8
State-space	0.0086	82.61
Identified model using NLSE	0.0049	91.3
Identified model using PSO	0.0015	97.1

According to Table 5.1, the MSE value for linear models including the transfer function and state-space model is obtained as 0.0108 rad/s and 0.0083 rad/s, respectively. The $\boldsymbol{R^2}$ values of the transfer function and state-space models are calculated to be 78.8 % and 82.61 %, respectively. On the other hand, the MSE values for the identified models by the NLSE and the PSO algorithms are obtained to be 0.0024 rad/s and 0.0007 rad/s, respectively. The $\boldsymbol{R^2}$ values of the identified models using the NLSE and the PSO algorithms are computed to be 91.3 % and 97.1 %, respectively. Due to the results, as expected, since the DC motor model includes nonlinearities, the linear models including transfer function and state-space models do not provide accurate results. However, the identified nonlinear model using the NLSE and the PSO algorithm provides more accurate results when compared to the linear models. Consequently, for the rest of the study, the NLSE and the PSO algorithms will be employed to identify the nonlinear models of the mechanical systems.

5.1.2 Case two

In the second case, the DC motor's mathematical model comprises all the nonlinear components explained in equation (4.6), and artificial data is created. For model identification and testing the models, two distinct input signals are designed. The input signals cover various ranges of voltage values as input signals to the DC motor model. For this purpose, the PRMS signal, which was depicted in Figure 5.2, is employed as an input signal. In the next step, the output data of the model ($\omega(t)$) is extracted from the Simulink model of the DC motor. In order to develop output data, the model parameters are assumed to be unknown. In order to provide a more realistic output signal, the white Gaussian noise (28 dbW SNR) is added to the $\omega(t)$. Once the artificial input-output data of the system is prepared, the PSO algorithm is coded and run to

determine the parameters of the Simulink model. For this purpose, the parameters of the PSO algorithm are tuned by the trial-and-error method and are shown in Table 5.2.

Table 5.2 : The tuning parameters of the PSO algorithm.

Parameter	Value
Number of dimensions	10
Number of particles for each dimension	10
Number of iterations	1000
χ_{min} and χ_{max}	0.4 and 0.9
d_1 and d_2	2

Then, using the model parameters that are found by the PSO algorithm, the output of the model of the DC motor is calculated. Then, the statistical Hypotheses test is conducted to examine the credibility of the determined parameters. In addition to the PSO algorithm, the NLSE method is employed to determine the model parameters of the DC motor. Then, using parameters identified through the NLSE method, the model output is calculated. The parameters determined by the PSO and the NLSE methods, determination error ($\boldsymbol{E_{PSO}}$ and $\boldsymbol{E_{NLSE}}$) and the statistical test results are demonstrated in Table 5.3.

Table 5.3 : The parameters determined by the PSO and NLSE methods, the errors, and statistical test results for the DC motor model (Case two).

Parameter (Unit)	Real Value	Determined using PSO	E_{PSO} (%)	Determined using NLSE	E_{NLSE} (%)	p_k
J ($Kg.m^2$)	0.1	0.095	5	0.112	12	≈ 0
K (N.m/A)	0.2	0.19	5	0.164	18	≈ 0
L (H)	0.1	0.099	1	0.07	30	≈ 0
R (Ω)	1.5	1.485	1	1.68	12	≈ 0
$T_s^{(2)}$(N)	0.1	0.095	5	0.067	33	≈ 0
$T_c^{(2)}$(N)	0.2	0.19	5	0.23	15	≈ 0
$T_{v1}^{(2)}(N.s/rad)$	0.05	0.05	0	0.055	11	≈ 0
$T_{v2}^{(2)}(N.s^2/rad^2)$	0.05	0.05	0	0.041	18	≈ 0
$T_{v3}^{(2)}(N.s^3/rad^3)$	0.02	0.019	5	0.061	41	≈ 0
$\omega_s^{(2)}(rad/s)$	0.01	0.01	0	0.007	23	≈ 0

According to Table 5.3, the determination error of the parameters using the PSO algorithm is between 0 % and 5 %, while it is between 11 % and 41 % for the NLSE algorithm. In addition, the p_k values of all model parameters are computed to be approximately equal to zero. Since their p_k values are smaller than the significance level ($\alpha = 0.1$), it could be inferred that they are non-zero and exist in the model of the

system. Consequently, it could be inferred that the PSO algorithm determined the model parameters more accurately than the NLSE algorithm, and the statistical hypotheses test also approved the results.

In the next step, the output of the models with determined parameters by the PSO and NLSE methods is calculated using the Simulink model of the system. The output of the identified models with determined parameters utilizing both methods, as well as their absolute error is represented in Figure 5.4.

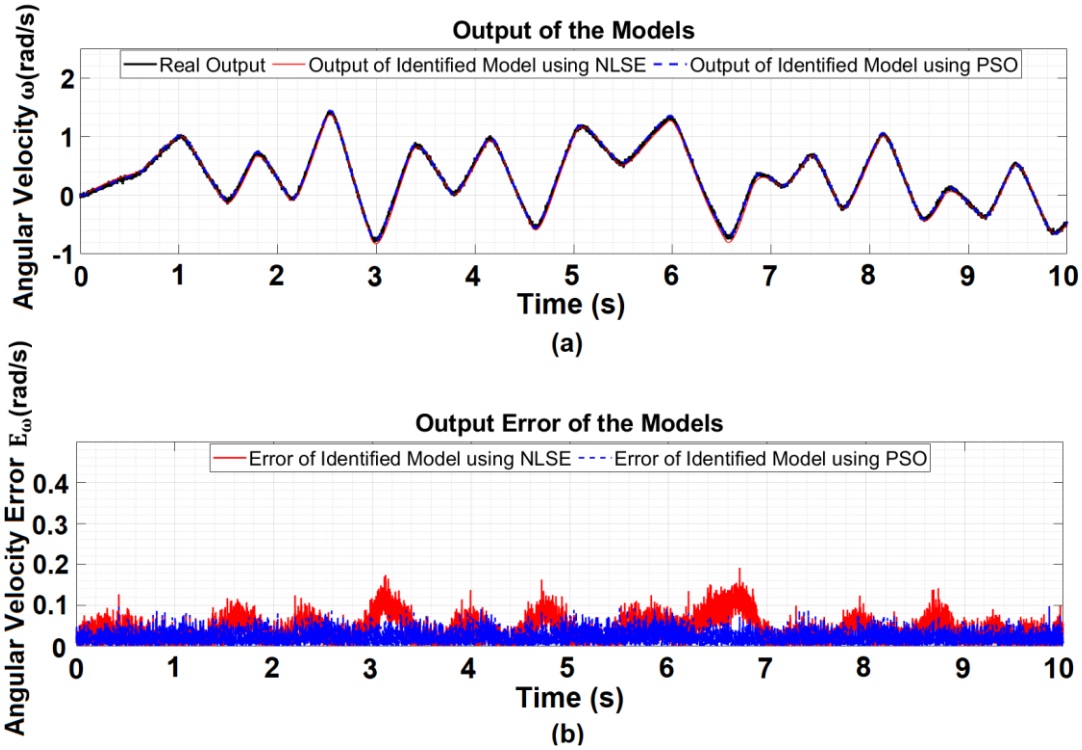


Figure 5.4 : The real output of the DC motor model and the calculated output of identified models with parameters determined by the PSO and NLSE methods (case two) (a), and the output error of the models (b).

According to Figure 5.4, the maximum absolute error corresponding to the PSO and the NLSE methods are 0.09 rad/s and 0.19 rad/s, respectively. It shows that the PSO algorithm performs more accurately than the PSO algorithm in calculating the output of the model of the system. For further evaluation of the identified models, new sets of input-output data are generated, and the identified models are tested. The input PRMS signal used for testing the models is designed between -5 V and 5 V and is represented in Figure 5.5.

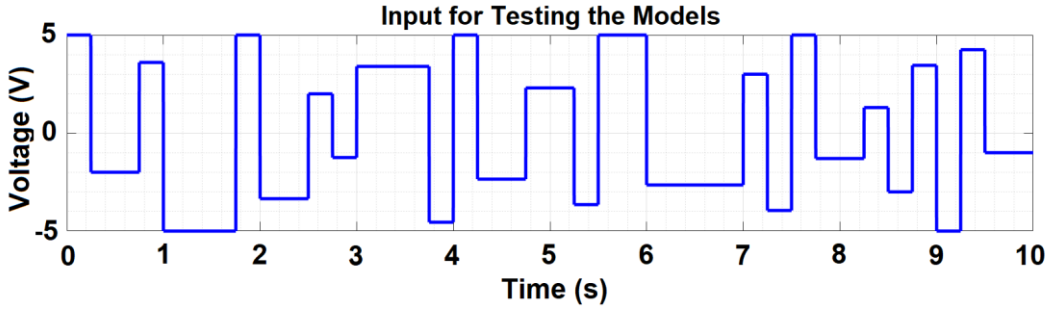


Figure 5.5 : The input PRMS voltage signal for testing the identified DC motor models.

The output of the models with parameters determined by the PSO and the NLSE methods is computed using the Simulink model with the new PRMS input signal which was depicted in Figure 5.4. The output of the identified models with determined parameters via both of the algorithms along with the modeling error for the testing signal is demonstrated in Figure 5.6.

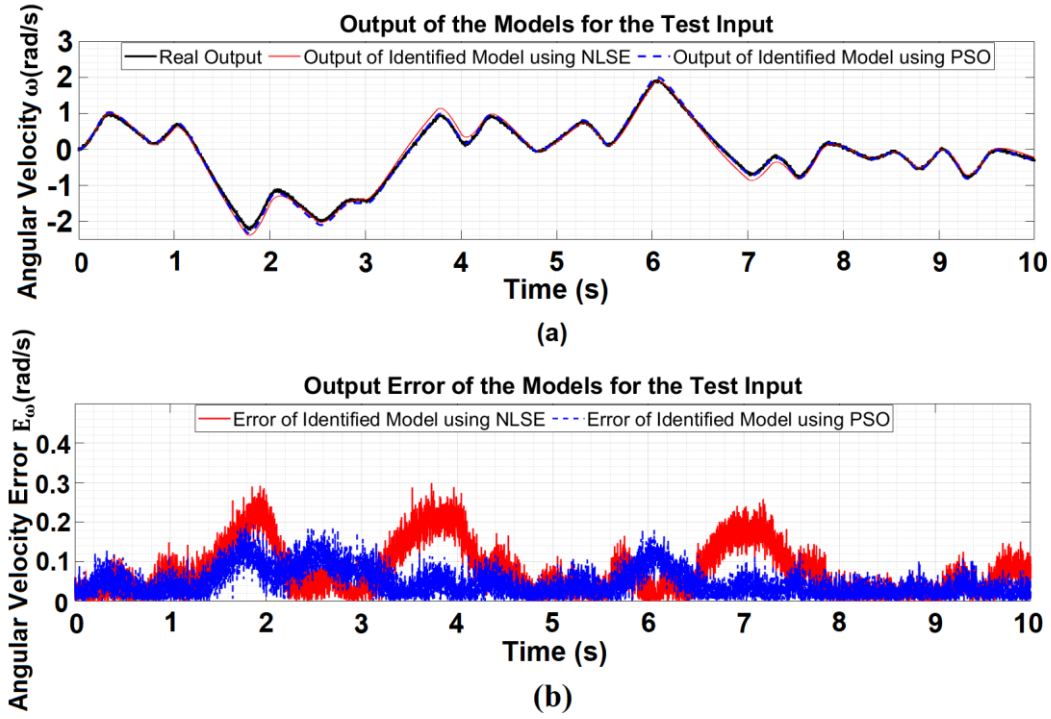


Figure 5.6 : The real output of the DC motor model and the calculated output of identified models by the PSO and NLSE methods for testing input (case two) (a), and the output error of the models (b).

According to Figure 5.6, the maximum absolute error corresponding to the PSO and the NLSE methods is 0.19 rad/s and 0.29 rad/s, respectively. It shows that the PSO algorithm performs more accurately than the PSO algorithm in calculating the output of the model of the system. The MSE and $\mathbf{R^2}$ values of the test data are calculated for

the outputs of the identified models with the determined parameters by the PSO and the NLSE methods and are given in Table 5.4.

Table 5.4 : The MSE and R^2 values of the identified models with determined parameters by the PSO and NLSE methods for a DC motor (Case two).

Model	MSE (rad/s)	R^2(%)
Identified model using PSO	0.006	97.1
Identified model using NLSE	0.019	88.6

According to Table 5.4, the MSE values for the identified models with determined parameters by the PSO and the NLSE methods are 0.006 rad/s and 0.019 rad/s, respectively. The $\mathbf{R^2}$ values of the identified models by the PSO and the NLSE algorithms are computed to be 97.1 % and 88.6 %, respectively. Due to the results, the identified model with determined parameters by the PSO algorithm can describe more variations in the real output of the model around its means when compared to the identified model with determined parameters by the NLSE algorithm.

5.1.3 Case three

In the third case, the DC motor**'s** mathematical model comprises some of the nonlinear components which are shown in equation (4.9). For parameter determination and model validation, two distinct input signals covering the operating regions of the system are employed. The input signals cover various ranges of voltage values as input signals to the system. For this purpose, the same PRMS signal, which was depicted in Figure 5.1, is employed as an input signal. In the next step, the output data of the model ($\omega(t)$) is obtained from the Simulink model of the DC motor. In order to develop output data, the model parameters (electrical and mechanical parameters of the DC motor) are assumed to be known. The white Gaussian noise (25 dbW SNR) is added to the ($\omega(t)$). Once the artificial input-output data of the system is prepared, the PSO algorithm is coded and run to determine the parameters of the Simulink model. Then, using the model parameters that are found by the PSO algorithm, the output of the model is calculated. Then, the statistical hypotheses test is performed to examine the accuracy of the determined parameters. In addition to the PSO algorithm, the NLSE method is employed to determine the parameters of the system. Then, using parameters determined through the NLSE algorithm, the output of the system is calculated. The parameters determined by the PSO and the NLSE methods, determination error (E_{PSO} and E_{NLSE}) and the statistical test results are demonstrated in Table 5.5.

Table 5.5 : The parameters determined by the PSO and NLSE methods, the determination errors, and the statistical test results for the DC motor (case three).

Parameter (Unit)	Real Value	Determined using PSO	E_{PSO} (%)	Determined using NLSE	E_{NLSE} (%)	p_k
J ($Kg.m^2$)	0.1	0.1	0	0.138	18	≈ 0
K (N.m/A)	0.2	0.19	5	0.18	10	≈ 0
L (H)	0.1	0.1	0	0.121	21	≈ 0
R (Ω)	1.5	1.52	1.3	1.92	28	≈ 0
$T_s^{(3)}$(N)	0.1	0.1	0	0.138	38	≈ 0
$T_c^{(3)}$(N)	0.2	0.2	0	0.27	33	≈ 0
$T_{v1}^{(3)}(N.s/rad)$	0.05	0.05	0	0.037	26	≈ 0
$T_{v2}^{(3)}(N.s^2/rad^2)$	0	10×10^{-7}	≈ 0	0.075	<100	≈ 1
$T_{v3}^{(3)}(N.s^3/rad^3)$	0	10×10^{-9}	≈ 0	0.029	<100	≈ 1
$\omega_s^{(3)}(rad/s)$	0.01	0.01	0	0.0131	31	≈ 0

According to Table 5.5, the determination error of the parameters using the PSO algorithm is between 0 % and 5 %, while it is between 10 % and 100 % for the NLSE algorithm. In addition, the $\mathbf{p_k}$ values of all parameters except $\boldsymbol{T_{v2}^{(3)}}$ and $\boldsymbol{T_{v3}^{(3)}}$, are calculated to be approximately equal to zero. Since their $\mathbf{p_k}$ values are smaller than the significance level ($\boldsymbol{\alpha = 0.01}$), it could be inferred that they are non-zero and exist in the model of the system. On the contrary, the $\mathbf{p_k}$ values of the $\boldsymbol{T_{v2}^{(3)}}$ and $\boldsymbol{T_{v3}^{(3)}}$ are obtained as being approximately equal to 1. Since their $\mathbf{p_k}$ values are higher than the significance level ($\boldsymbol{\alpha = 0.01}$), it could be inferred that their value is equal to zero and they do not exist in the model of the system. Therefore, since the real values $\boldsymbol{T_{v2}^{(3)}}$ and $\boldsymbol{T_{v3}^{(3)}}$ are assumed to be 0, it could be inferred that the PSO algorithm determined the model parameters more accurately and the statistical hypotheses test approved its results.

In the next step, the same PRMS signal, which was depicted in Figure 5.5, is employed as an input signal for testing the models. The output of the models with parameters determined utilizing the PSO and the NLSE methods is computed using the Simulink model with the new PRMS input signal. The identified models output with determined parameters by both of the algorithms along with the modeling error for the testing signal is demonstrated in Figure 5.7.

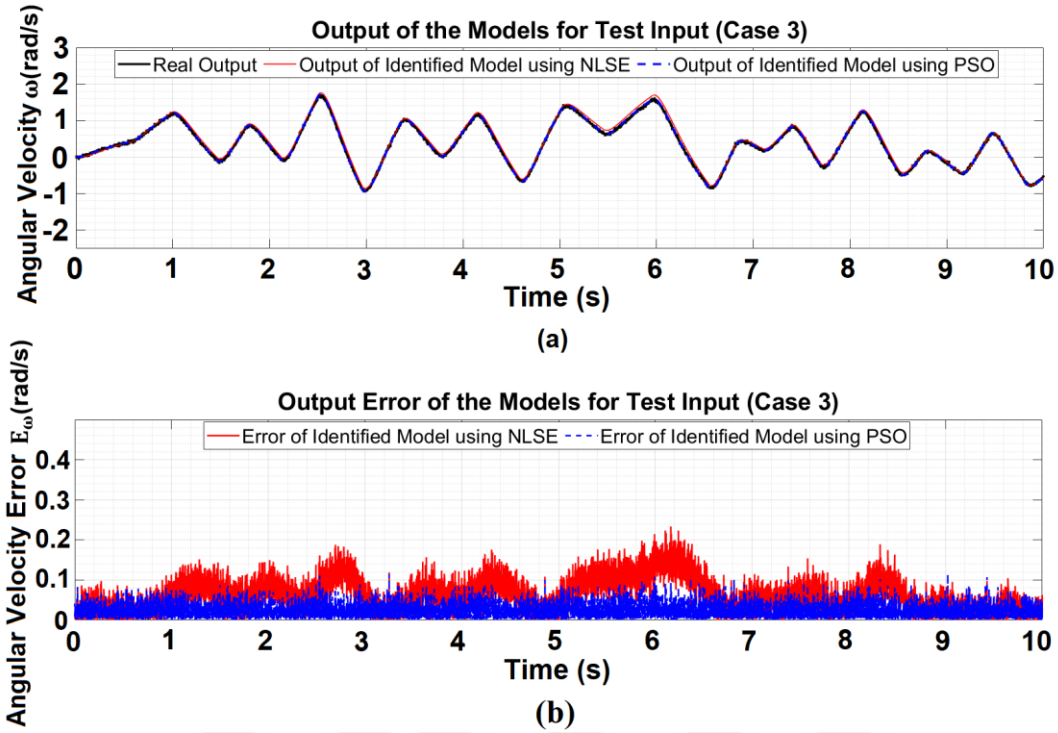


Figure 5.7 : The real output of the DC motor and the output of identified models by PSO and NLSE methods for testing input (case three) (a), and the output error of the models (b).

According to Figure 5.7, the maximum absolute error corresponding to the PSO and the NLSE methods is 0.11 rad/s and 0.23 rad/s, respectively. It shows that the PSO algorithm performs more accurately than the PSO algorithm in calculating the output of the system. For further evaluation of the identified models with determined parameters by the PSO and NLSE methods, new sets of input-output data are generated, and the identified models are tested. The MSE and $\mathbf{R^2}$ values of the outputs of the identified models with the determined parameters by the PSO and the NLSE methods (case three) and provided in Table 5.6.

Table 5.6 : The MSE and R^2values of the identified models with determined parameters by the PSO and NLSE methods for the DC motor (case three).

Model	MSE (rad/s)	R^2(%)
Identified model using PSO	0.007	96.32
Identified model using NLSE	0.01	87.41

According to Table 5.6, the MSE values for the identified models with determined parameters utilizing the PSO and the NLSE methods are 0.007 rad/s and 0.01 rad/s, respectively. The R^2 values of the identified models utilizing the PSO and the NLSE methods are computed to be 96.32 % and 87.41 %, respectively. Due to the results, the

identified model with determined parameters by the PSO algorithm can describe more variations in the real output of the model around its means when compared to the identified model with determined parameters by the NLSE algorithm.

5.2 Case Study 2: A Solenoid Actuator

The Simulink model of the solenoid actuator using this information along with equation (4.10) and equation (4.11) is developed and is represented in Figure 5.8.

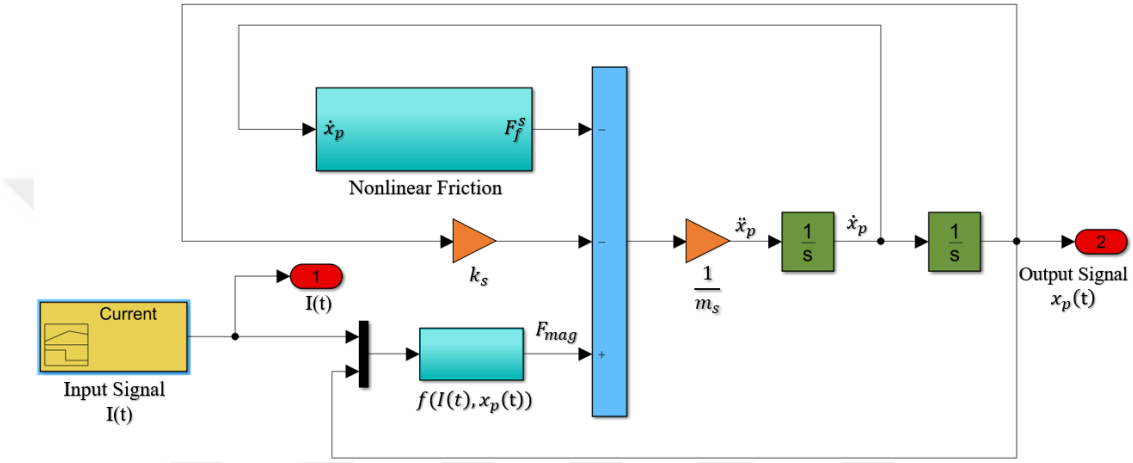


Figure 5.8 : The Simulink model of a solenoid actuator.

Once the solenoid actuator's mathematical model is established, the artificial data including the input and output datasets (coil current ($I(t)$) as an input and plunger position ($x_p(t)$) as an output signal) are generated through the developed model. In order to make the output signals more realistic, an appropriate white Gaussian noise is added to the values of the $x_p(t)$. Consequently, the artificial data and simulation model of a solenoid actuator along with the parameter determination algorithms will be employed to determine the unknown parameters [m_s, k_s, F_c^s, F_s^s, F_v^s] of the model. Furthermore, the reliability of the determined parameters will be tested using statistical tests. Finally, using model evaluation criteria including R^2 and MSE values, the quality of the model with determined parameters in calculating the output of the solenoid actuator will be evaluated.

In addition, the geometrical design parameters of the solenoid actuator are optimized using a multi-objective PSO algorithm which is explained in Appendix A.

The results of the modeling of a solenoid actuator are provided and discussed below. First, the FEM simulation of the actuator is conducted to compute the magnetic force

values of the real-world actuator. For this aim, as described in Appendix A, the design of different parts of the system is completed and their materials along with their characteristics are assigned using simulation software. Then, the meshing of different parts of the model is completed by triangle meshes. Finally, magnetic field analysis is performed to calculate the magnetic force generated in each position of the plunger along the guider part of the actuator. The magnetic force is computed for the working region of the plunger between 0 and 0.005 m. Using the trial-and-error method of choosing different mesh properties, the experimental and simulation results of the original solenoid actuator are closely matched (Appendix A). Then, the performance of the actuator for different current values between 0 and 2.7 A is evaluated using simulations. The magnetic force (F_{mag}) values versus plunger position (x_p) for different coil current (I) values are demonstrated in Figure 5.9.

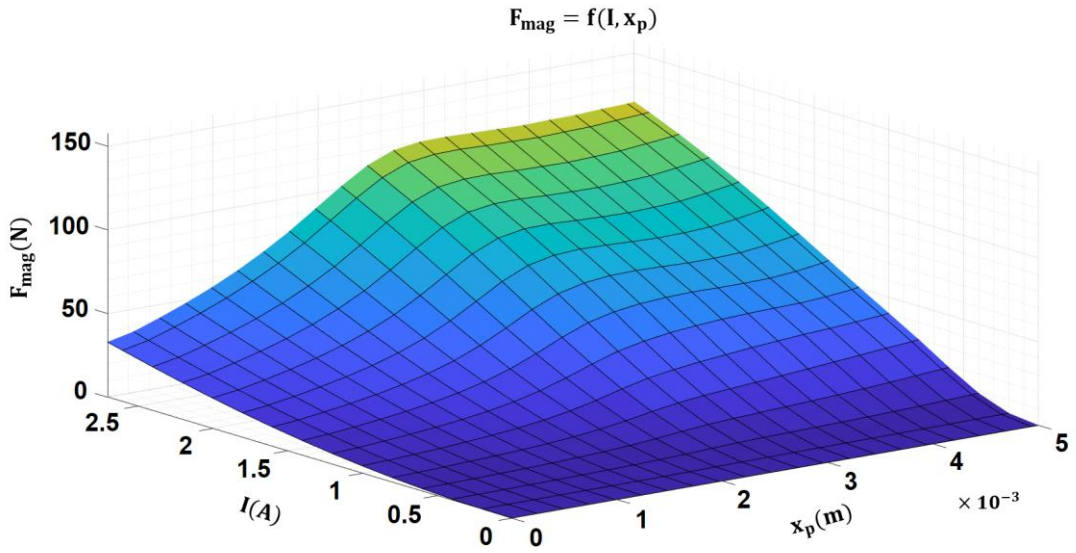


Figure 5.9 : The magnetic force (F_{mag}) values versus plunger position (x_p) for different values of coil current (I).

Then, the third-order nonlinear model represents the relationship between the dependent variable F_{mag} and independent variables x_p and I is obtained and replaced in the model of the system (Figure 5.8). Then, the mechanical parameters of the model are obtained using CAD information along with the experimental data. The mass of the moving part is obtained to be 0.064 kg using the CAD design of the solenoid actuator. The spring constant (k_s) is calculated using Hooke's law from the acquired experimental data. Equation (5.4) is employed to compute the approximate value of the stiffness of the spring.

$$k_s = \left|\frac{\Delta F}{\Delta X_S}\right| = \left|\frac{10(N)}{-0.00146(m)}\right| \approx 6850\ (N/m) \tag{5.4}$$

Then, the friction parameters are assumed to be known to generate the output data using the Simulink model. Once the Simulink model of the solenoid actuator is completed, the sum of two sinusoidal signals with frequencies of 1 Hz and 4 Hz is designed as the input current signal. For identification of the model and its validation, two distinct sinusoidal input signals covering the operating regions of the solenoid actuator are employed. The current (I) signal ranging between 0 and 2.7 A is employed as an input signal, which is shown in Figure 5.10.

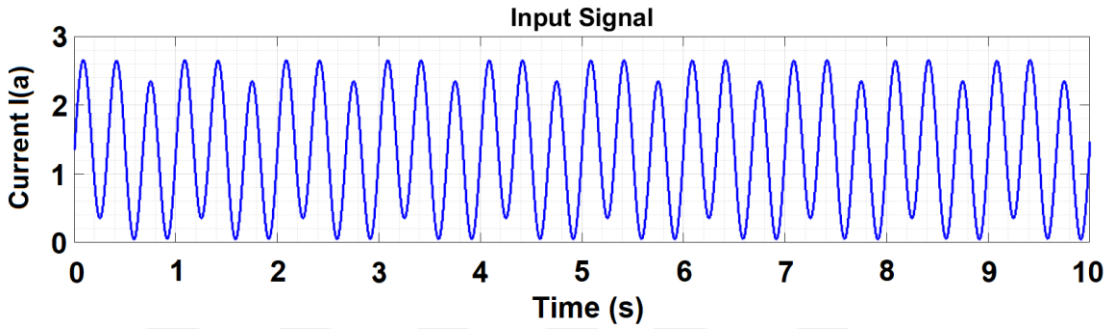


Figure 5.10 : The current (I) input signal ranges between 0 and 2.7 A for model identification.

In the next step, the output data of the model (x_p) is obtained from the Simulink model of the system. Then, in order to provide a more realistic output signal, the white Gaussian noise (25 dbW SNR) is added to the x_p. Once the simulation input-output data of the system is prepared, the PSO algorithm is coded and run to determine the model parameters. For this aim, the PSO algorithm's parameters are tuned by the trial-and-error method and are shown in Table 5.7.

Table 5.7 : The tuning parameters of the PSO algorithm for modeling a solenoid actuator.

Parameter	Value
Number of dimensions	6
Number of particles for each dimension	10
Number of iterations	2000
$\chi_{\min}$ and $\chi_{\max}$	0.4 and 0.9
d_1 and d_2	2

Then, using the model parameters that are found by the PSO algorithm, the model output is calculated. Then, the statistical test is conducted to examine the accuracy of

the determined parameters. In addition to the PSO algorithm, the NLSE method is employed to determine the model parameters of the solenoid actuator. Then, using parameters determined through the NLSE method, the model output is computed. The real values of the parameters, parameters determined by the PSO and the NLSE methods, determination error (E_{PSO} and E_{NLSE}) and the statistical test results are demonstrated in Table 5.8.

Table 5.8 : The parameters determined by the PSO and NLSE methods, the determination errors, and the statistical tests for a solenoid actuator.

Parameter (Unit)	Real Value	Determined using PSO	E_{PSO} (%)	Determined using NLSE	E_{NLSE} (%)	p_k
$k_s(N/m)$	6850	6966	1.6	7161	4.54	≈ 0
m_s (Kg)	0.064	0.05504	14.5	0.022476	65.62	≈ 0
$F_c^s(N)$	2	1.7747	11.26	8.5638	< 100	≈ 0
$F_s^s(N)$	4	3.5153	12.11	6.3546	58.75	≈ 0
$F_v^s(N.s/m)$	5	4.5546	8.9	8.9847	79.6	≈ 0
$\dot{x}_s^s(m/s)$	0.1	0.10097	≈ 0	2.8344	< 100	≈ 0

According to Table 5.8, the determination error of the parameters using the PSO algorithm is between 0% and 14.5%, while it is between 4.54% and 79.6%, and even for some values, it is over 100% for the NLSE algorithm. In addition, the p_k values of all parameters are calculated to be approximately equal to zero. Since their p_k values are smaller than the significance level ($\alpha = 0.1$), it could be inferred that they are non-zero and exist in the system's mathematical model. Therefore, it could be inferred that the PSO algorithm determined the model parameters with fewer errors when compared to the NLSE algorithm, and the statistical hypotheses test also approved the results.

In the next step, the model output with determined parameters by the PSO and NLSE methods is calculated using the Simulink model of the system. The output of the identified models with determined parameters by both of the algorithms, as well as their absolute error, is represented in Figure 5.11.

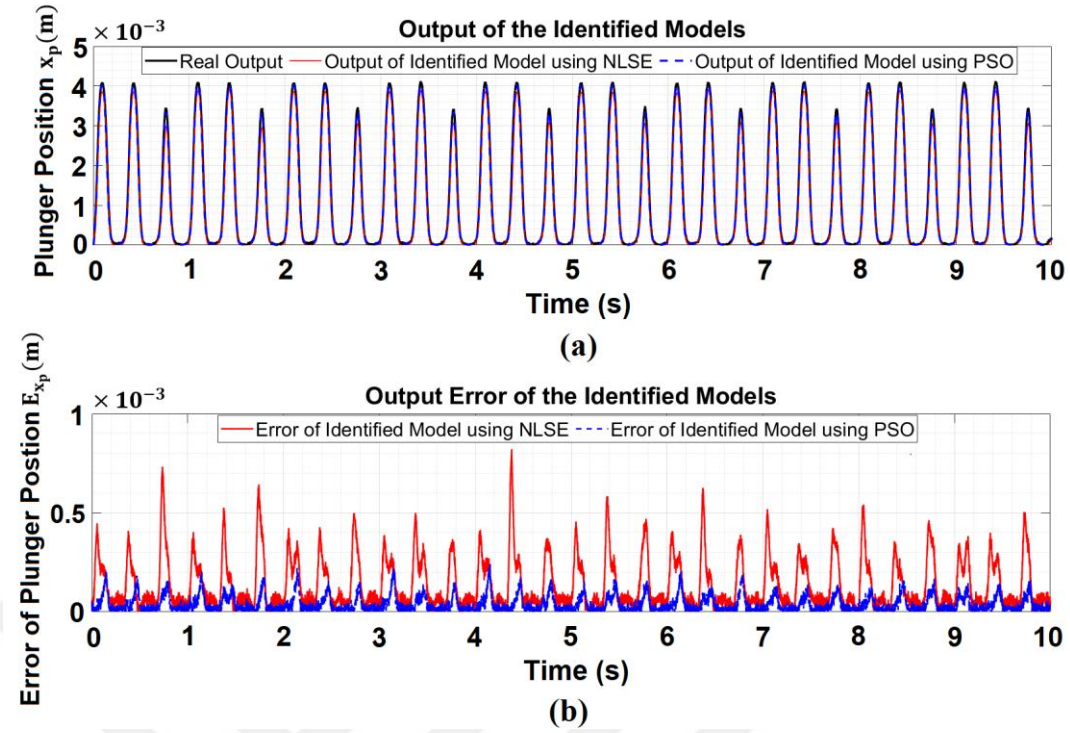


Figure 5.11 : The real output of the model and the calculated output of identified models by the PSO and NLSE methods (a), and the output error of the models (b).

According to Figure 5.11, the maximum absolute error of the output corresponding to the models identified by the PSO and the NLSE methods is 2×10^{-4} m and 8×10^{-4} m, respectively. It shows that the PSO algorithm performs more accurately than the NLSE algorithm in calculating the output of the model of the solenoid actuator. For further evaluation of the identified models, a new sinusoidal input signal is designed and represented in Figure 5.12.

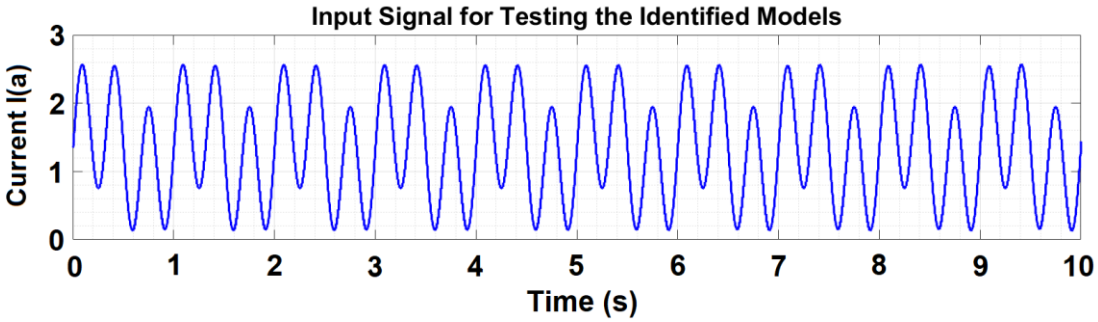


Figure 5.12 : The current (I) input signal for testing the identified models of the solenoid actuator.

Then, the identified models' output with parameters determined by the PSO and NLSE algorithms is calculated using the new input signal represented in Figure 5.12. The real output of the system and the identified models' output with determined parameters by

both of the algorithms along with the modeling error for the testing input signal are represented in Figure 5.13.

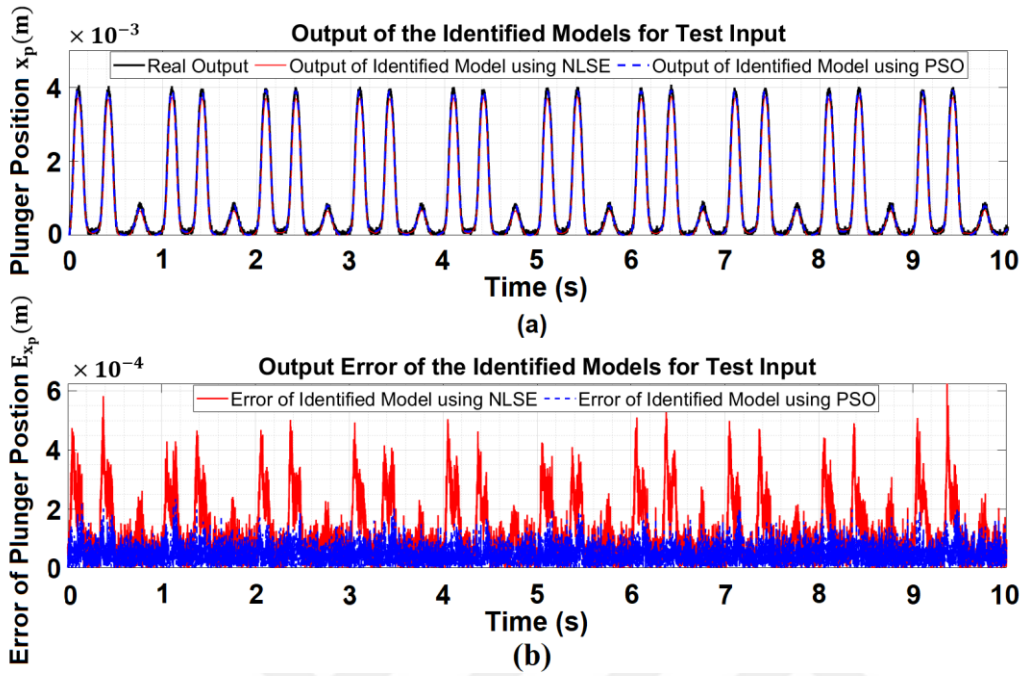


Figure 5.13 : The real output of the solenoid actuator and the calculated output of identified models using the PSO and NLSE algorithms (test input) (a), and the output error of the models (b).

The MSE and R^2 values of the identified model outputs with the determined parameters by the PSO and NLSE methods with the test input signal are computed and provided in Table 5.9.

Table 5.9 : The MSE and R^2 values of the identified models of the solenoid actuator with determined parameters by PSO and NLSE methods for a solenoid actuator.

Model	MSE (m)	R^2(%)
Identified model using PSO	6×10^{-5}	94.7
Identified model using NLSE	1.8×10^{-4}	89.1

According to Table 5.9, the MSE values for the identified models with determined parameters by the PSO and the NLSE are 6×10^{-5} m and 1.8×10^{-4} m, respectively. The R^2 values of the identified models by the PSO and NLSE algorithms are calculated to be 94.7% and 89.1%, respectively. Due to the results, the identified model with determined parameters by the PSO algorithm can describe more variations in the real output of the model around its means when compared to the identified model with determined parameters by the NLSE algorithm.

5.3 Case Study 3: A Ball and Beam Mechanism

In this subsection, the simulation model of the ball and beam, including all nonlinear terms, is developed using MATLAB/Simulink, and the PSO and NLSE algorithms are employed to determine the parameters of the model using experimental data. The Simulink model of a ball and beam mechanism is developed and is demonstrated in Figure 5.14.

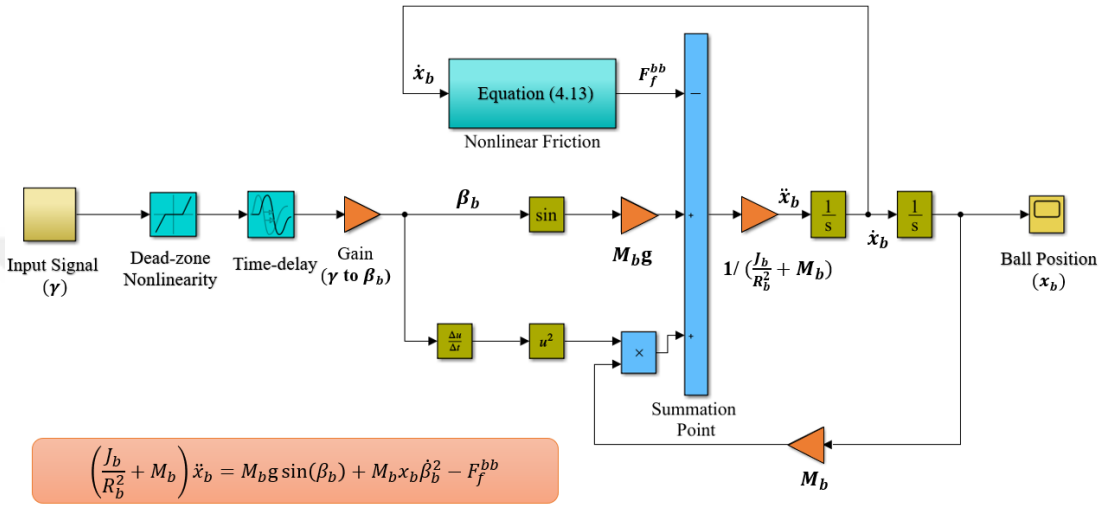


Figure 5.14 : The Simulink model of a ball and beam mechanism.

In the next step, the experimental data are acquired from the ball and beam mechanism to determine the model parameters using the methodology explained in Chapter 3. To acquire experimental data, the following procedure is conducted. First, the ball and beam system is connected to an Arduino Mega microcontroller, which is itself connected to a desktop computer via serial communication. Then, the Simulink Support Package for Arduino hardware is utilized to access Arduino utilities within the Simulink environment. Then, the output of the system (x_b) is obtained by interpreting the voltage across the linear potentiometer. This value is read by using the "Analog Read" function/block. The values are sampled at a rate of 100 Hz with 8-bit resolution. Similarly, the commands for the servo motor (γ) are generated within the Simulink environment by combining sinusoidal sources and sent to the motor using the "Servo Write" function/block. Both signals including γ and x_b are saved into the MATLAB workspace time series which correspond to the input and output of the mechanism, respectively.

Then, statistical tests are conducted to examine the credibility of the obtained results. The results of the modeling of a ball and beam mechanism for two cases are provided and discussed as follows.

5.3.1 Case one

In the first case, there is direct contact between the surface of the steel ball and the aluminum beam. In this case, two different input signals (motor angle) are designed to acquire real data for the identification and testing of the models. The first signal, which is utilized for the identification of the model, is the sum of two sinusoidal signals with distinct frequencies (0.23 Hz and 0.31 Hz). The input signal which is supplied to the ball and beam mechanism is depicted in Figure 5.15.

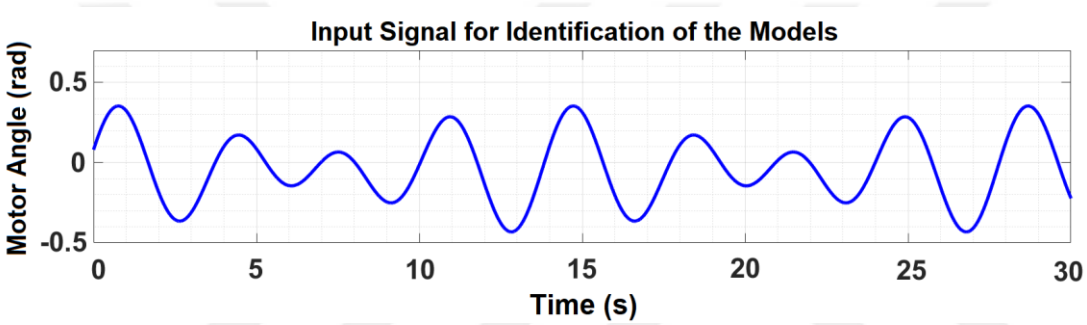


Figure 5.15 : The input signal for model identification of the ball and beam mechanism.

In the next step, the output data of the ball and beam mechanism (x_b) is collected from the system. Once the experimental data of the system is prepared, the PSO algorithm is coded and run to find the parameters of the model of the ball and beam mechanism. Then, using the model parameters that are found by the PSO algorithm, the output of the mechanism is calculated. The algorithm consists of 11 parameters to be identified, and 10 particles are considered for each of these parameters (90 particles in total). Then, the statistical hypotheses tests including hypotheses and confidence intervals, are performed to examine the credibility of the determined parameters. In addition to the PSO algorithm, the NLSE method is employed to determine the model parameters of the ball and beam mechanism. The determined parameters by the PSO and NLSE methods and the results of the statistical analysis are demonstrated in Table 5.10. The time-delay (T_d) in the system is obtained to be approximately 0.05 s by the PSO and NLSE methods, which is occurring due to the digitization and communication processes.

Table 5.10 : The determined parameters by the PSO and NLSE methods, the determination errors, and the statistical test results for the ball and beam mechanism (case one).

Parameter (Unit)	Determined using PSO	Determined using NLSE	p_k	Confidence Interval
$F_c^{bb}(N)$	0.09	2.58	≈ 0	[0.053 0.127]
$F_s^{bb}(N)$	0.15	0.93	≈ 0	[0.14 0.169]
$F_{v1}^{bb}(N.s/m)$	0.26	1.19	≈ 0	[0.251 0.278]
$F_{v2}^{bb}(N.s^2/m^2)$	4.6×10^{-5}	0.21	≈ 1	$[-1.58 \times 10^{-4}\ 2.5 \times 10^{-4}]$
$F_{v3}^{bb}(N.s^3/m^3)$	4.6×10^{-5}	0.13	≈ 1	$[-1.58 \times 10^{-4}\ 2.5 \times 10^{-4}]$
$\dot{x}_s^{bb}(m/s)$	0.0017	1.18	≈ 0	$[5.7 \times 10^{-7}\ 2.5 \times 10^{-4}]$
$U_l(rad)$	3.8×10^{-5}	0.04	≈ 1	$[-1.4 \times 10^{-4}\ 2.4 \times 10^{-4}]$
$U_r(rad)$	4.6×10^{-5}	0.04	≈ 1	$[-1.5 \times 10^{-4}\ 2.5 \times 10^{-4}]$

According to Table 5.10, friction parameters including F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$ are determined as non-zero values by the PSO algorithm which conveys that the corresponding terms of the determined parameters exist in the mathematical model of the system. However, the nonlinear viscous friction constants (F_{v2}^{bb} and F_{v3}^{bb}) and dead-zone parameters (U_l and U_r) values are identified as approximately equal to 0. It means that the nonlinear terms of the viscous friction and the dead-zone nonlinearity could be neglected in the mathematical model of the actuator. Furthermore, the parameters of these nonlinear terms include the zero value in their confidence interval. It implies that these terms do not exist in the model of the system since they are statistically insignificant. However, the remaining friction parameters (F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$) exist in the model of the system as they do not contain zero values in their confidence intervals. The p_k values of the friction parameters (F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$) are obtained less than the significance level ($p_k < 0.01$). It statistically verifies the significance of the friction parameters, which means that they all exist in the dynamic model of the system. However, the p_k values of the parameters of the nonlinear viscous friction terms (F_{v2}^{bb} and F_{v3}^{bb})) and the parameters of the dead-zone (U_l and U_r) are greater than the value of α ($p_k > 0.01$). It infers that the existence of these parameters is statistically rejected.

In the next step, the determined parameters by the PSO and NLSE methods are replaced in the Simulink model of the mechanism. Then, the output of the system (x_b) is calculated for both of the identified models. The real position of the ball and the computed position of the ball using the identified models with parameters

determined by both algorithms, as well as their absolute errors, are represented in Figure 5.16.

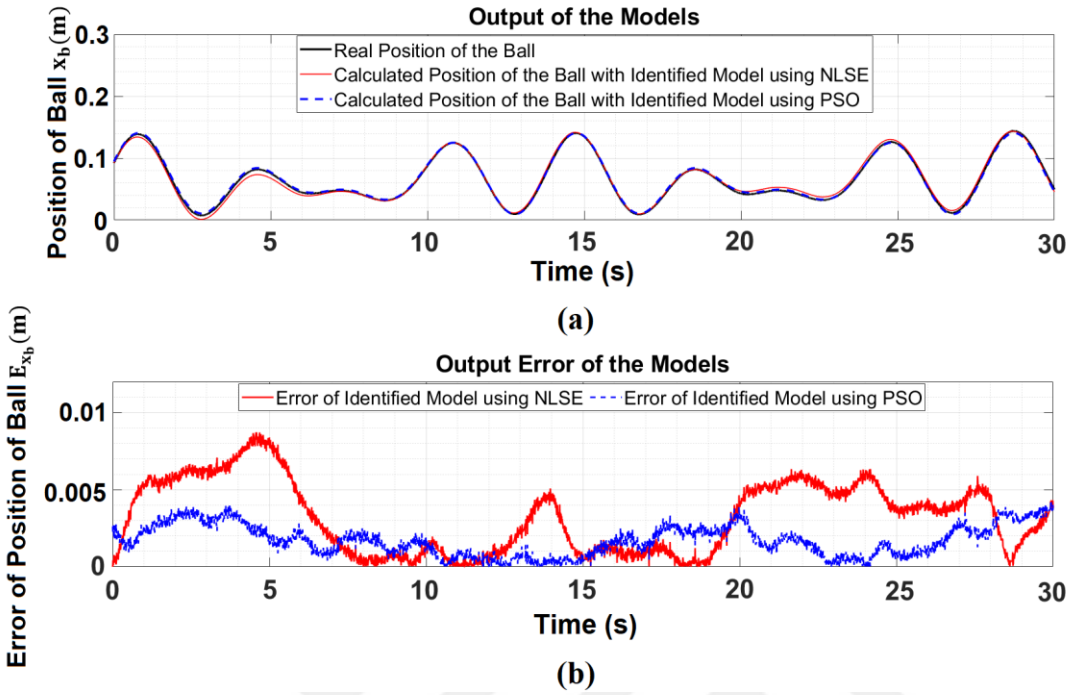


Figure 5.16 : The real (measured) position of the ball and the calculated position of the ball using identified models by the PSO and `NLSE methods (a), and the error of the ball position (b).

For further evaluation of the identified models, a new sinusoidal input signal is designed. The input of the system which is the sum of two Sine signals with frequencies of 0.28 Hz and 0.39 Hz, is demonstrated in Figure 5.17.

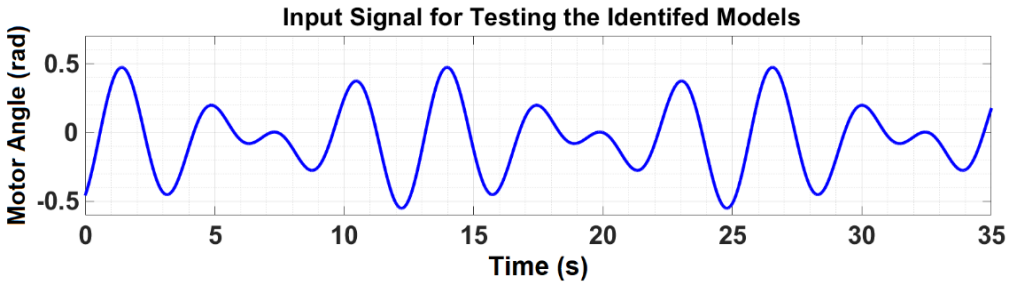


Figure 5.17 : The new input signal for testing the identified models.

In the next step, the output $(\boldsymbol{x_b})$ is measured from the system, and the input-output experimental dataset is constructed. Then, the output of the system is computed utilizing the identified models with determined parameters by the PSO and NLSE methods. The real position of the ball and the calculated position of the ball using

identified models with determined parameters by both methods, along with the modeling error for the testing input signal, are represented in Figure 5.18.

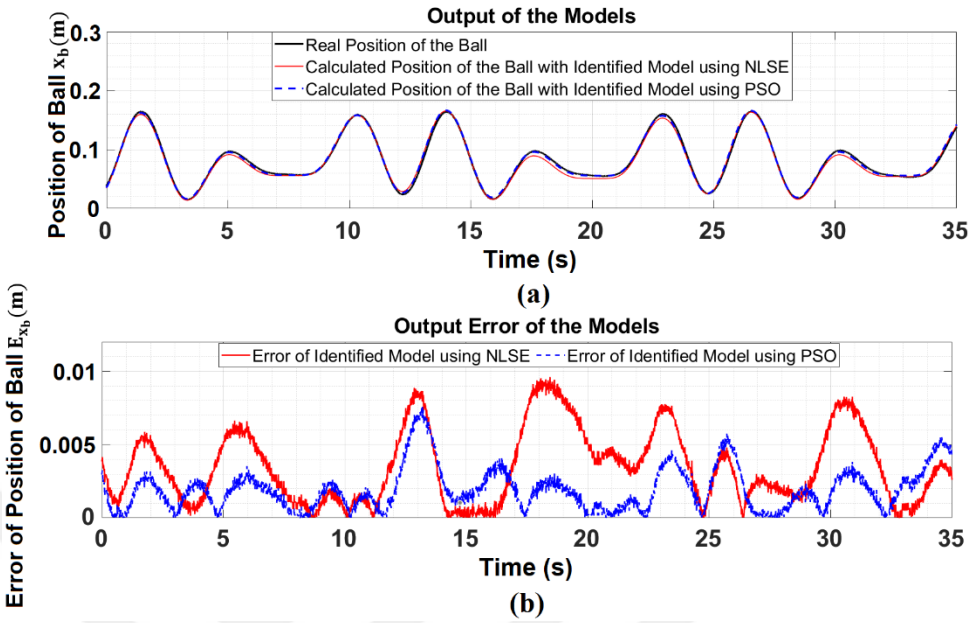


Figure 5.18 : The real position of the ball and the calculated position of the ball by identified models by PSO and NLSE methods (test input) (a), and the error of the ball position (b).

The MSE and R^2 values of the identified model outputs with the determined parameters by the PSO and NLSE algorithms with the test input signal are given in Table 5.11.

Table 5.11 : The MSE and R^2 values of the identified models of the ball and beam mechanism with determined parameters using the PSO and NLSE algorithms (case one).

Model	MSE (m)	R^2 (%)
Identified model using PSO	1.18×10^{-5}	96.7
Identified model using NLSE	1.04×10^{-4}	91.9

According to Table 5.11, the MSE values for the identified models with determined parameters by the PSO and the NLSE are 1.18×10^{-5} m and 1.04×10^{-4} m, respectively. The R^2 values of the identified models by the PSO and NLSE algorithms are calculated to be 96.7 % and 91.9 %, respectively. Due to the results, the identified model with determined parameters by the PSO algorithm can describe more variations in the real output of the model around its means when compared to the identified model with determined parameters by the NLSE method.

5.3.2 Case two

In the second case, the functionality of the PSO in the determination of the model parameters when they are being varied due to experimental conditions is checked. For this purpose, the adhesive paper tape is attached to the surface of the beam to change the value of the different frictions between the ball and the beam. In fact, there is no direct contact between the steel ball and the aluminum beam in this case. In this case, two different input signals are designed to acquire real data for the identification and testing of the models. In the next step, the output (x_b) is collected from the system. Once the experimental input-output data of the system is prepared, the PSO and NLSE methods are used to find the parameters of the mathematical model of the system. Then, the statistical tests including hypotheses and confidence interval tests are performed to check the credibility of the determined parameters. The determined parameters by the PSO and NLSE methods and the statistical test results for case two are demonstrated in Table 5.12.

Table 5.12 : The determined parameters by the PSO and NLSE algorithms, the determination errors, and the statistical test results for the ball and beam mechanism (case two).

Parameter (Unit)	Determined using PSO	Determined using NLSE	p_k	Confidence Interval
$F_c^{bb}(N)$	0.15	5.26	≈ 0	[0.13 0.17]
$F_s^{bb}(N)$	0.84	3.14	≈ 0	[0.82 0.86]
$F_{v1}^{bb}(N.s/m)$	0.16	0.18	≈ 0	[0.14 0.18]
$F_{v2}^{bb}(N.s^2/m^2)$	2.4×10^{-5}	0.3	≈ 1	$[-1.32 \times 10^{-4}\ 2.34 \times 10^{-4}]$
$F_{v3}^{bb}(N.s^3/m^3)$	2.4×10^{-5}	0.06	≈ 1	$[-1.32 \times 10^{-4}\ 2.34 \times 10^{-4}]$
$\dot{x}_s^{bb}(m/s)$	0.001	3.07	≈ 0	$[5.4 \times 10^{-7}\ 0.004]$
$U_l(rad)$	2×10^{-5}	0.08	≈ 1	$[-2.2 \times 10^{-4}\ 2.6 \times 10^{-4}]$
$U_r(rad)$	8×10^{-5}	0.07	≈ 1	$[-1.5 \times 10^{-4}\ 5.7 \times 10^{-4}]$

Due to Table 12, the friction parameters such as F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$ are determined as non-zero values by the PSO algorithm which conveys that the corresponding terms of these parameters exist in the model of the system. However, the parameters of viscous friction parameters (F_{v2}^{bb} and F_{v3}^{bb}) and dead-zone parameters (U_l and U_r) values are determined as almost equal to 0 by the PSO algorithm. It could be concluded that the nonlinear terms of the viscous friction and the dead-zone nonlinearity could be neglected in the mathematical model of the system. Furthermore, the parameters of these nonlinear terms include zero value in their confidence interval [LB UB]. It

implies that these terms do not exist in the model of the system since they are statistically insignificant. However, the remaining friction parameters (F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$) exist in the model of the system as they do not contain zero values in their confidence intervals. The p_k values of the friction parameters (F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$) are obtained less than the significance level ($p_k < 0.01$). It statistically verifies the significance of the friction parameters, which means that they all exist in the mathematical model of the system. However, the p_k values of the parameters such as F_{v2}^{bb}, F_{v3}^{bb}, U_l and U_r are greater than the value of $\propto$ ($p_k > 0.01$). It means that the existence of these parameters is statistically rejected.

In addition, when the determined parameters utilizing the PSO algorithm for cases one and two are compared (Table 5.10 and Table 5.12), it could be seen that some friction parameters including F_c^{bb}, F_s^{bb}, F_{v1}^{bb} and $\dot{x}_s^{bb}$ are varied significantly. It means that by adding some material between the steel ball and the aluminum beam, the friction parameters are varied as expected. By adding adhesive paper tape, the Coulomb friction $F_c^{b\&b}$, the static friction F_s^{bb} and $\dot{x}_s^{bb}$ are increased from 0.09 and 0.15 to 0.15 and 0.84, respectively, while the constant of the linear term of the viscous friction F_{v1}^{bb} is decreased from 0.26 to 0.16.

In order to examine the determined models, a new sinusoidal input signal is designed. The input signal which is the sum of two Sine signals with frequencies of 0.28 Hz and 0.39 Hz, is demonstrated in Figure 5.19.

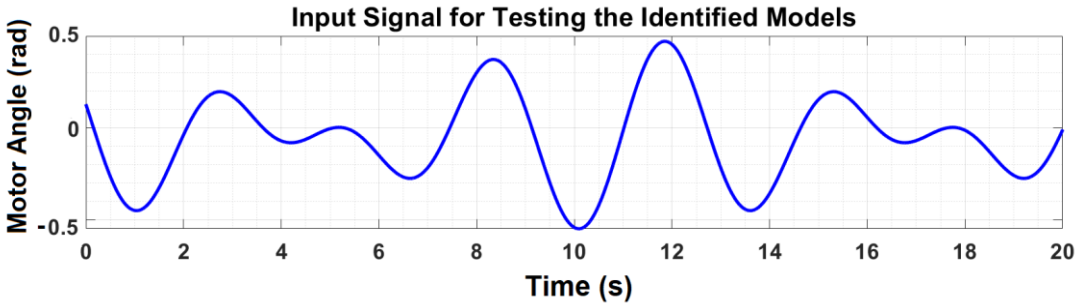


Figure 5.19 : The new input signal for testing the identified models of the ball and beam mechanism (case 2).

Then, the output ($\boldsymbol{x_b}$) is calculated for both of the identified models utilizing the PSO and NLSE methods. The real position of the ball and the calculated position of the ball using the identified models with parameters determined by both algorithms, as well as their absolute errors, are represented in Figure 5.20.

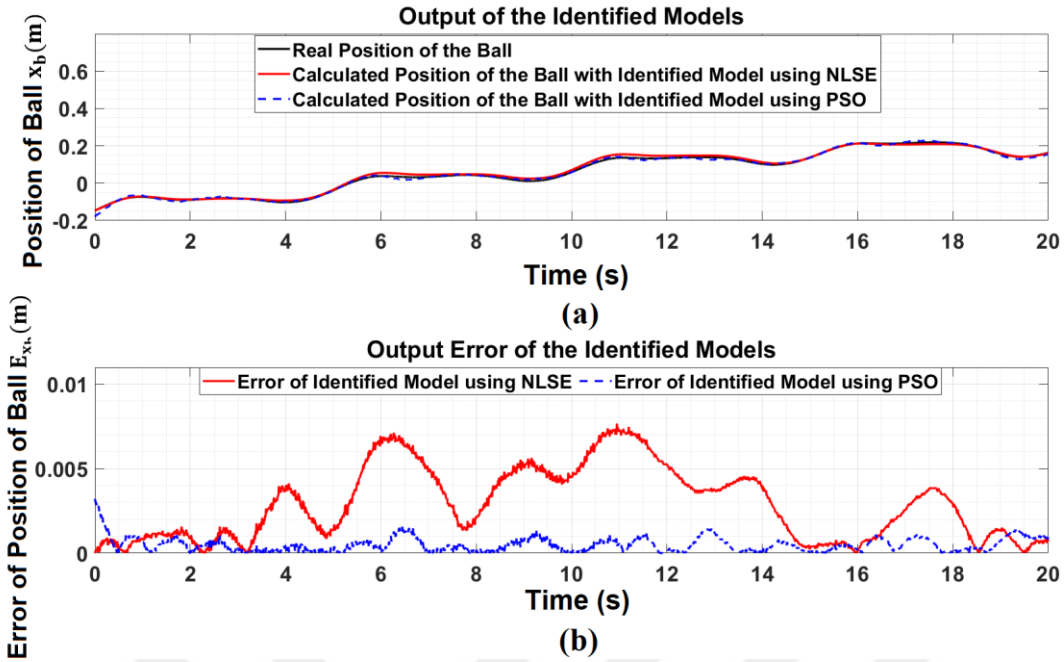


Figure 5.20 : The real position of the ball and the calculated position of the ball by identified models by PSO and NLSE methods (test input) (a), and the error of the ball position (b).

According to Figure 5.20, the maximum absolute error of the output corresponding to the models identified by the PSO and the NLSE algorithms is 0.003 m and 0.0077 m, respectively. It could be inferred that the PSO algorithm performs more accurately in calculating the position of the ball.

The MSE and R^2 values of the identified models outputs with the determined parameters both of the algorithm with the test input signal are provided in Table 5.13.

Table 5.13 : The MSE and R^2 values of the identified models of the ball and beam mechanism with determined parameters using the PSO and NLSE algorithms (case two).

Model	MSE (m)	R^2 (%)
Identified model using PSO	6.47×10^{-5}	95.3
Identified model using NLSE	8.38×10^{-4}	88.4

According to Table 5.13, the MSE values for the identified models with determined parameters by the PSO and the NLSE are 6.47×10^{-5} m and 8.38×10^{-4} m, respectively.

The R^2 values of the identified models output by the PSO and NLSE methods are computed to be 95.3% and 88.4%, respectively. Due to the results, the identified model with determined parameters by the PSO algorithm can describe more variations in the

real output of the model around its means when compared to the identified model with determined parameters by the NLSE method. Consequently, it is shown that the PSO algorithm could successfully determine the model parameters more accurately than the NLSE method even when some of the parameters were changed due to the experimental condition.

5.4 Case Study 4: A UR5 Robotic Manipulator

The joint variables along with their torques are obtained experimentally as explained in Chapter 4.

The input signals of the joints of the UR5 manipulator, including position (q_i), velocity ($\dot{q}_i$) and acceleration ($\ddot{q}_i$) of the joints (i=1-6) for model identification are represented in Figure 5.21.

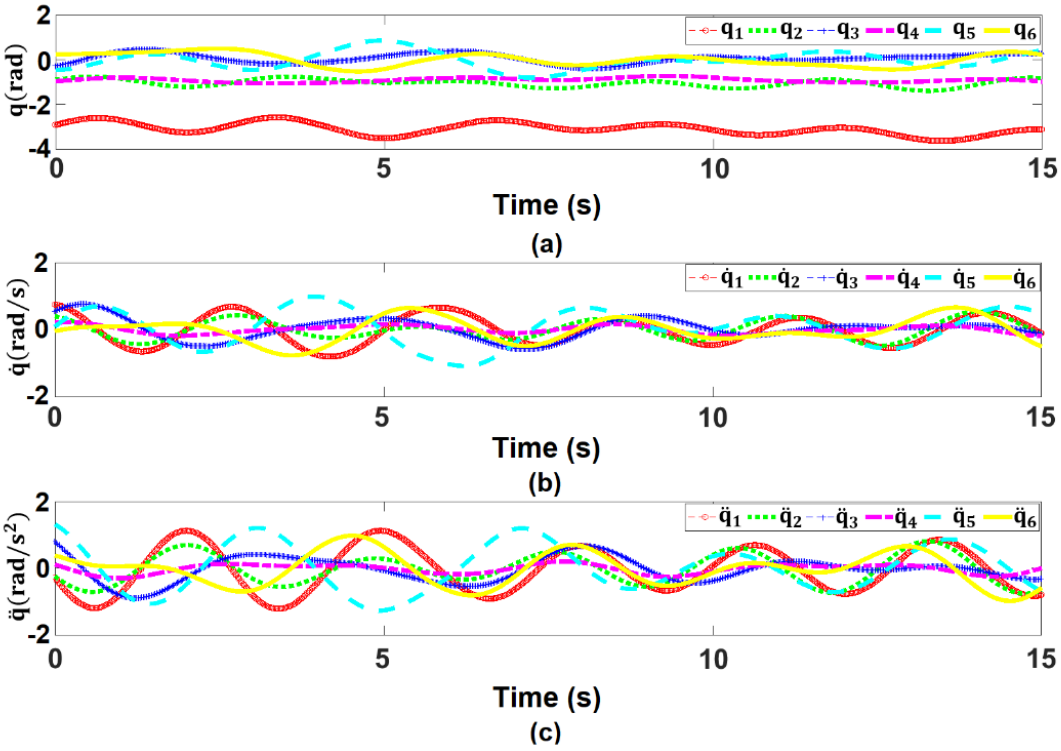


Figure 5.21 : The joint variables, including q_i, $\dot{q}_i$ and $\ddot{q}_i$ where (i=1-6) for model identification.

The trajectory of the corresponding signal to Figure 5.21 in cartesian space is represented in Figure 5.22.

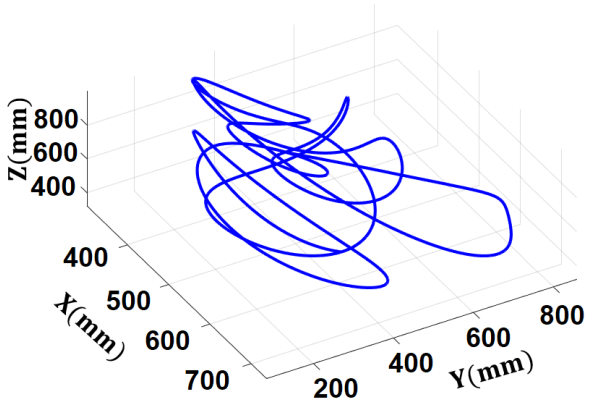


Figure 5.22 : The trajectory for model identification.

Once the experimental input-output data of the UR5 manipulator is prepared, the PSO algorithm is coded and run to find the unknown joint parameters of the manipulator. Then, using the model parameters that are found by the PSO algorithm, the joints' torque values are computed. In addition to the PSO algorithm, the NLSE method is employed to determine the joint parameters of the UR5 manipulator. The determined parameters of each joint of the robot are represented with $\tau_s^{(i)}$, $\tau_c^{(i)}$, $\tau_{v1}^{(i)}$, $\tau_{v2}^{(i)}$, $\tau_{v3}^{(i)}$ and $\dot{q}_s^{(i)}$ where i=1:6. The determined parameters by the PSO and NLSE methods, and the statistical test results are demonstrated in Table 5.14.

Table 5.14 : The determined parameters by the PSO and NLSE methods, as well as the statistical test results for the joints of the UR5 manipulator.

Joint	Parameter	Determined using PSO	Determined using NLSE	p_k	Confidence Interval
1	$\tau_s^{(1)}$	0.64	8.3	≈0	[0.6 0.67]
	$\tau_c^{(1)}$	0.91	-14.1	≈0	[0.88 0.93]
	$\tau_{v1}^{(1)}$	0.9	4.49	≈0	[0.88 0.92]
	$\tau_{v2}^{(1)}$	1.6×10^{-8}	-0.41	≈1	$[-9.8\times 10^{-6}\ 9.8\times 10^{-6}]$
	$\tau_{v3}^{(1)}$	2.4×10^{-8}	-0.79	≈1	$[-1.1\times 10^{-5}\ 1.2\times 10^{-6}]$
	$\dot{q}_s^{(1)}$	0.045	3.77	≈0	[0.029 0.062]
2	$\tau_s^{(2)}$	0.5	9.08	≈0	[0.46 0.53]
	$\tau_c^{(2)}$	0.95	1.6	≈0	[0.37 0.44]
	$\tau_{v1}^{(2)}$	0.4	-4.008	≈0	[0.37 0.44]
	$\tau_{v2}^{(2)}$	1.7×10^{-6}	-6.42	≈1	$[-9.9\times 10^{-5}\ 1.2\times 10^{-4}]$
	$\tau_{v3}^{(2)}$	1.8×10^{-6}	-0.56	≈1	$[-1\times 10^{-4}\ 1.05\times 10^{-4}]$
	$\dot{q}_s^{(2)}$	0.039	48.9	≈0	[0.024 0.055]

Table 5.14 (continued): The determined parameters by the PSO and NLSE methods, as well as the statistical test results for the joints of the UR5 manipulator.

Joint	Param.	Determined using PSO	Determined using NLSE	p_k	Confidence Interval
3	$\tau_s^{(3)}$	0.54	5.94	≈0	[0.5 0.5]
	$\tau_c^{(3)}$	0.59	1.02	≈0	[0.55 0.63]
	$\tau_{v1}^{(3)}$	0.89	0.15	≈0	[0.87 0.91]
	$\tau_{v2}^{(3)}$	1.1×10^{-8}	-8.7	≈1	$[-8.1 \times 10^{-6}\ 8.2 \times 10^{-6}]$
	$\tau_{v3}^{(3)}$	3.2×10^{-14}	6.81	≈1	$[-1.3 \times 10^{-8}\ 1.3 \times 10^{-8}]$
	$\dot{q}_s^{(3)}$	0.04	8.42	≈0	[0.03 0.06]
4	$\tau_s^{(4)}$	0.23	2.69	≈0	[0.19 0.26]
	$\tau_c^{(4)}$	0.2	38.53	≈0	[0.17 0.24]
	$\tau_{v1}^{(4)}$	0.16	-79.85	≈0	[0.14 0.19]
	$\tau_{v2}^{(4)}$	2.9×10^{-13}	-20.9	≈1	$[-4.2 \times 10^{-8}\ 4.2 \times 10^{-8}]$
	$\tau_{v3}^{(4)}$	2.8×10^{-12}	-0.35	≈1	$[-1.3 \times 10^{-7}\ 1.3 \times 10^{-7}]$
	$\dot{q}_s^{(4)}$	0.025	4.8	≈0	[0.013 0.037]
5	$\tau_s^{(5)}$	0.2	0.25	≈0	[0.17 0.24]
	$\tau_c^{(5)}$	0.17	-8.51	≈0	[0.14 0.2]
	$\tau_{v1}^{(5)}$	0.17	3.089	≈0	[0.14 0.20]
	$\tau_{v2}^{(5)}$	7.3×10^{-4}	-0.024	≈1	$[-1.3 \times 10^{-3}\ 2.8 \times 10^{-3}]$
	$\tau_{v3}^{(5)}$	1.1×10^{-12}	1.5×10^{-3}	≈1	$[-8.1 \times 10^{-8}\ 8.1 \times 10^{-8}]$
	$\dot{q}_s^{(5)}$	0.04	1.5×10^{-3}	≈0	[0.03 0.062]
6	$\tau_s^{(6)}$	0.29	0.23	≈0	[0.26 0.33]
	$\tau_c^{(6)}$	0.2	5.58	≈0	[0.17 0.24]
	$\tau_{v1}^{(6)}$	0.12	6.29	≈0	[0.097 0.148]
	$\tau_{v2}^{(6)}$	3×10^{-3}	0.31	≈1	$[-1.2 \times 10^{-3}\ 7.3 \times 10^{-3}]$
	$\tau_{v3}^{(6)}$	2.2×10^{-14}	0.37	≈1	$[-1.1 \times 10^{-8}\ 1.1 \times 10^{-8}]$
	$\dot{q}_s^{(6)}$	0.033	4.18	≈0	[0.019 0.047]

Due to Table 5.14, the parameters including $\tau_s^{(i)}$, $\tau_c^{(i)}$, $\tau_{v1}^{(i)}$, and $\dot{q}_s^{(i)}$ (i=1:6) are determined non-zero values by the PSO algorithm which indicates that the corresponding parameter exists in the model of the joint of the manipulator. On the other hand, the values of the parameters $\tau_{v2}^{(i)}$ and $\tau_{v3}^{(i)}$ (i=1:6) are obtained approximately equal to zero by the PSO algorithm which conveys that these parameters should not be included in the model of the joints. Meanwhile, these parameters ($\tau_{v2}^{(i)}$ and $\tau_{v3}^{(i)}$ where (i=1:6)) are found as non-zero values for all joints of the manipulator except for joint 5. It indicates that these parameters have to be included

in the joint's models, except for joint 5. In addition, the remaining parameters including $\tau_s^{(i)}$, $\tau_c^{(i)}$, $\tau_{v1}^{(i)}$, and $\dot{q}_s^{(i)}$ $(i = 1{:}\,6)$ are determined to be non-zero values by the NLSE algorithm. It indicates that these parameters should be added to the model of the joints. In addition, due to Table 5.14, the p_k values for the parameters $\tau_s^{(i)}$, $\tau_c^{(i)}$, $\tau_{v1}^{(i)}$, and $\dot{q}_s^{(i)}$ $(i = 1{:}\,6)$ are obtained approximately equal to zero, which indicates that the corresponding term of the parameter is statistically significant and have be added to the joints' models. On the other hand, the p_k values of the parameters $\tau_{v2}^{(i)}$ and $\tau_{v3}^{(i)}$ $(i = 1{:}\,6)$ for all joints are calculated to be approximately equal to 1, which indicates that the corresponding term of the parameters is statistically insignificant and could be dropped out from the joint models. Hence, due to statistical hypothesis test results, the PSO algorithm could accurately find the parameters which exist in the joint models. To further discuss the results of the algorithms, the confidence interval test results could be evaluated as below.

The parameters including $\tau_s^{(i)}$, $\tau_c^{(i)}$, $\tau_{v1}^{(i)}$, and $\dot{q}_s^{(i)}$ $(i = 1{:}\,6)$ for all joints of the manipulator do not contain zero values in their confidence intervals ($0 \notin$ confidence interval). It indicates that these parameters are statistically meaningful and have to be added to the joint model. On the other hand, the confidence intervals of the parameters $\tau_{v2}^{(i)}$ and $\tau_{v3}^{(i)}$ for all joints of the robotic arm comprise zero values ($0 \in$ confidence interval). It indicates that these parameters are statistically significant and could be dropped from the model of the joints. Therefore, due to the statistical confidence interval test, the PSO could accurately find the parameters which exist in the joint model.

In the next step, the output joints' torque of the UR5 manipulator with models with determined parameters by the PSO and NLSE methods is calculated. To represent the joint torques, the joints are classified into two groups with big and small motors. The first group comprises the base, the shoulder, and the elbow of the manipulator which have big motors. The second group comprises the 3 wrists of the manipulator, which has small motors. The measured torque of joints along with the calculated toque using models identified with determined parameters by the PSO and NLSE methods are shown for the first and second groups of joints in Figure 5.23 and Figure 5.24, respectively.

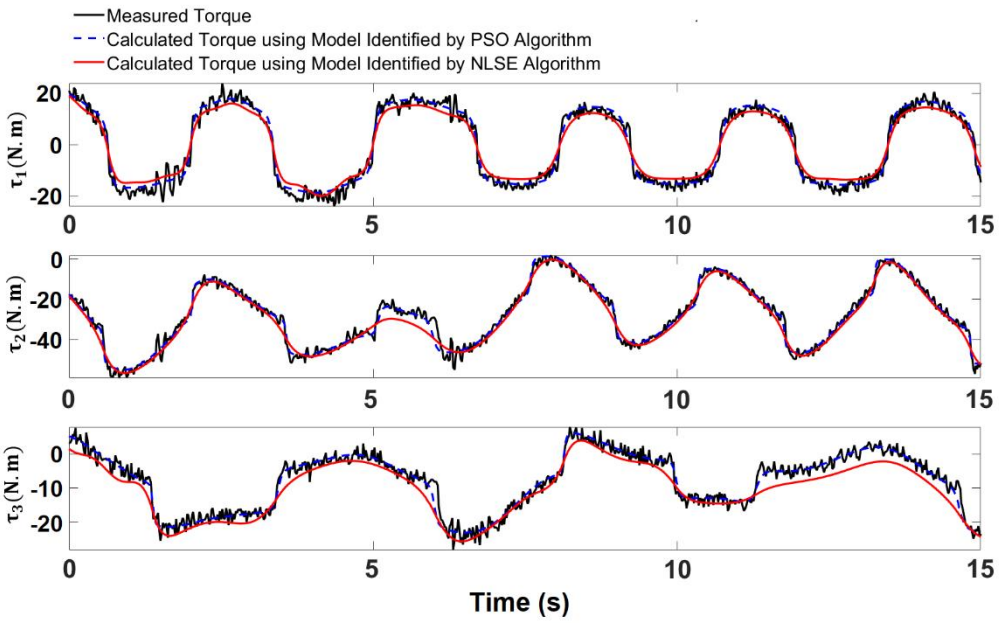


Figure 5.23 : The measured torque of joints along with the calculated toques through models identified with determined parameters by the PSO and NLSE methods (first group of joints).

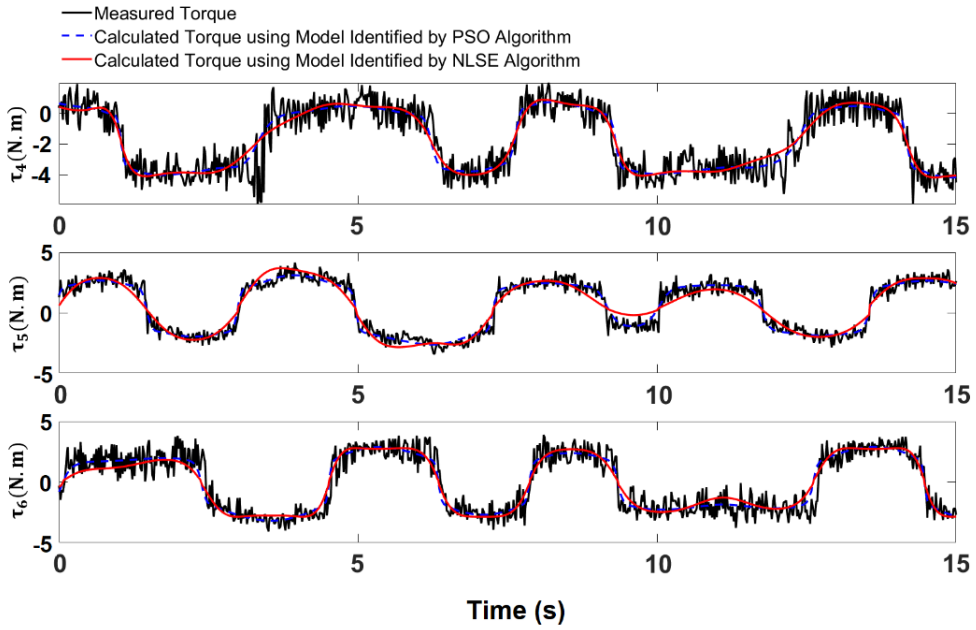


Figure 5.24 : The measured torque of joints along with the calculated toques through models identified with determined parameters by the PSO and NLSE methods (second group of joints).

The joint variables of the robotic arm, including q_i, $\dot{q}_i$ and $\ddot{q}_i$ $(i = 1 - 6)$ for testing, the identified models are represented in Figure 5.25 and the trajectory of the corresponding signal in cartesian space is shown in Figure 5.26.

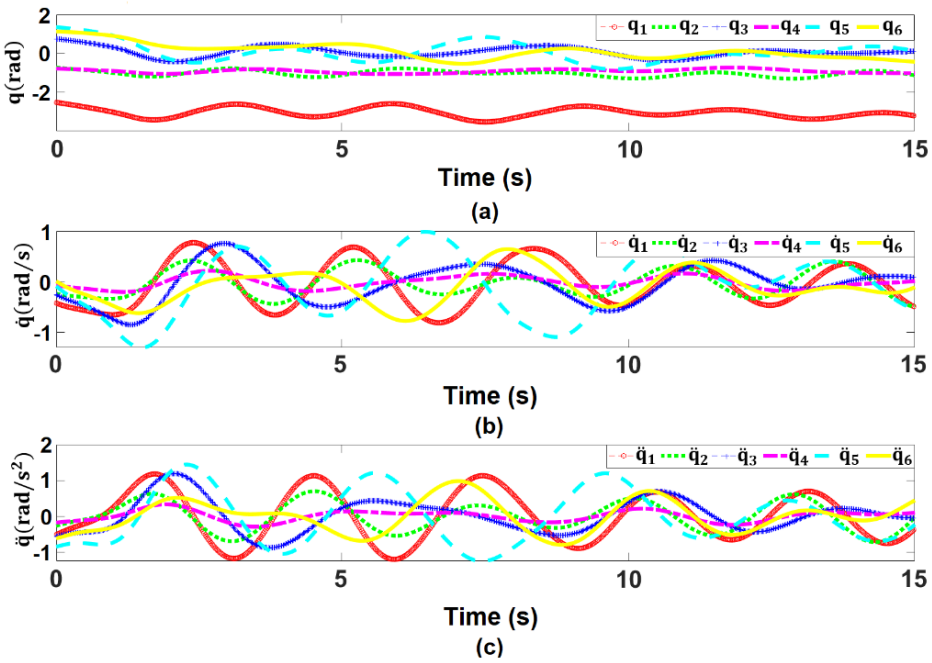


Figure 5.25 : The joint variables including $\boldsymbol{q_i}$, $\boldsymbol{\dot{q}_i}$ and $\boldsymbol{\ddot{q}_i}$ where for testing the identified models where $i = 1 - 6$.

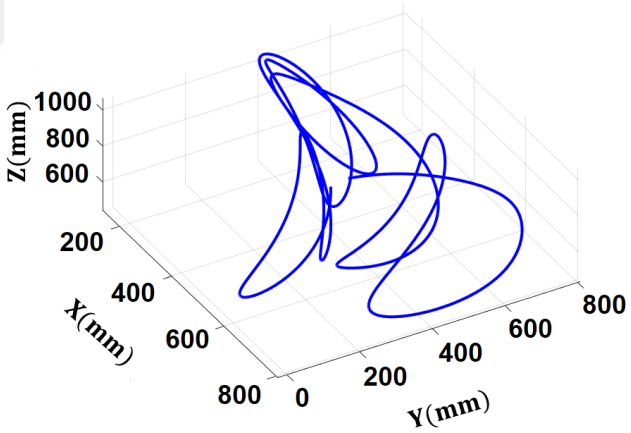


Figure 5.26 : The trajectory for testing the identified models.

In the next step, the torque values of the manipulator are computed utilizing the identified models with parameters determined by the PSO and NLSE methods. The measured torque of joints along with the calculated toque using models identified with determined parameters by the PSO and NLSE methods, as well as absolute torque errors for the first group of joints are demonstrated in Figure 5.27 and Figure 5.28, respectively.

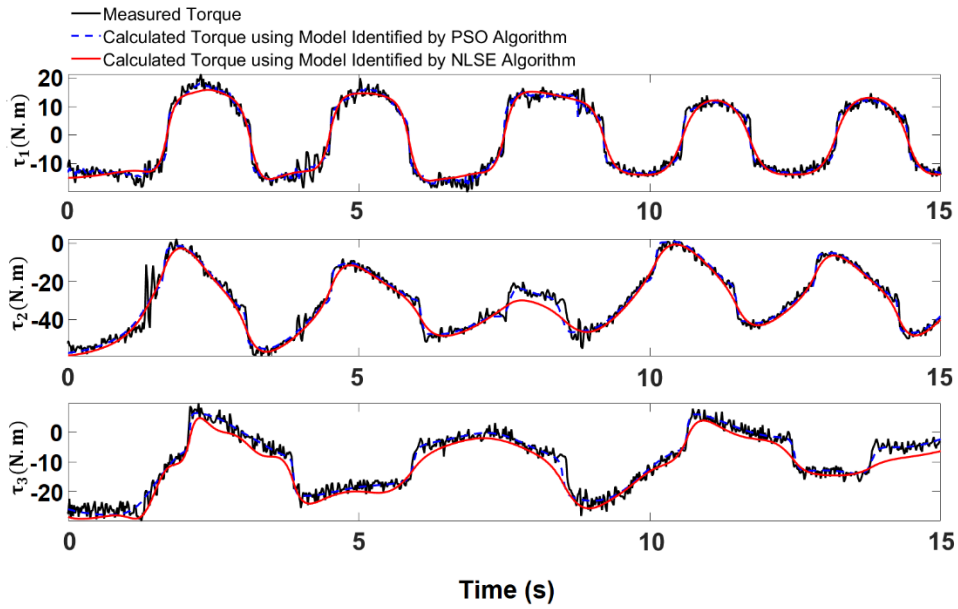


Figure 5.27 : The measured torque of joints along with the calculated toques through models identified with determined parameters by the PSO and NLSE methods (first group of joints).

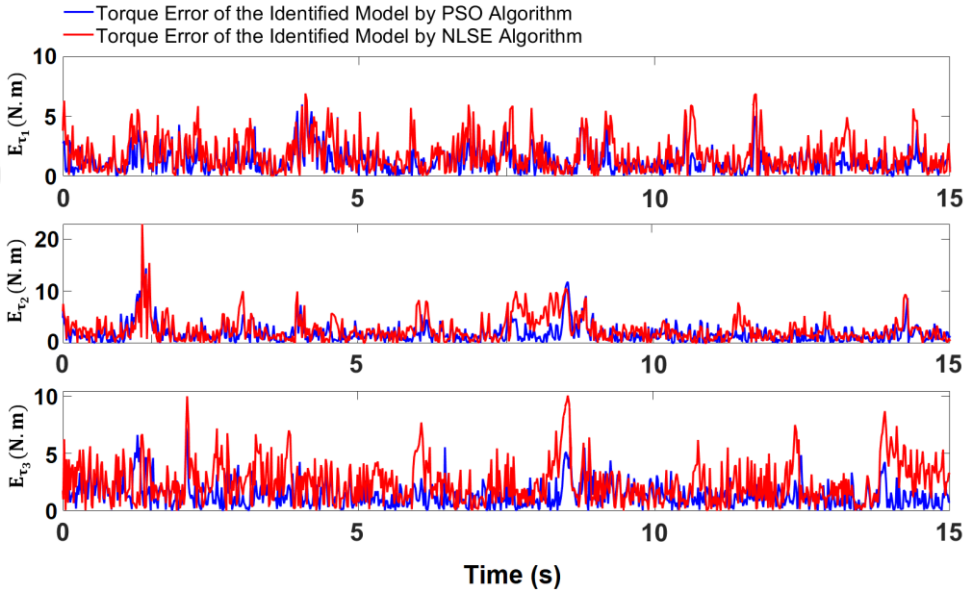


Figure 5.28 : The torque errors of joints through models identified with determined parameters by the PSO and NLSE methods (first group of joints).

The measured torque of joints along with the calculated toque using models identified with determined parameters by the PSO and NLSE methods, as well as absolute torque errors for the second group of joints, are demonstrated in Figure 5.29 and Figure 5.30, respectively.

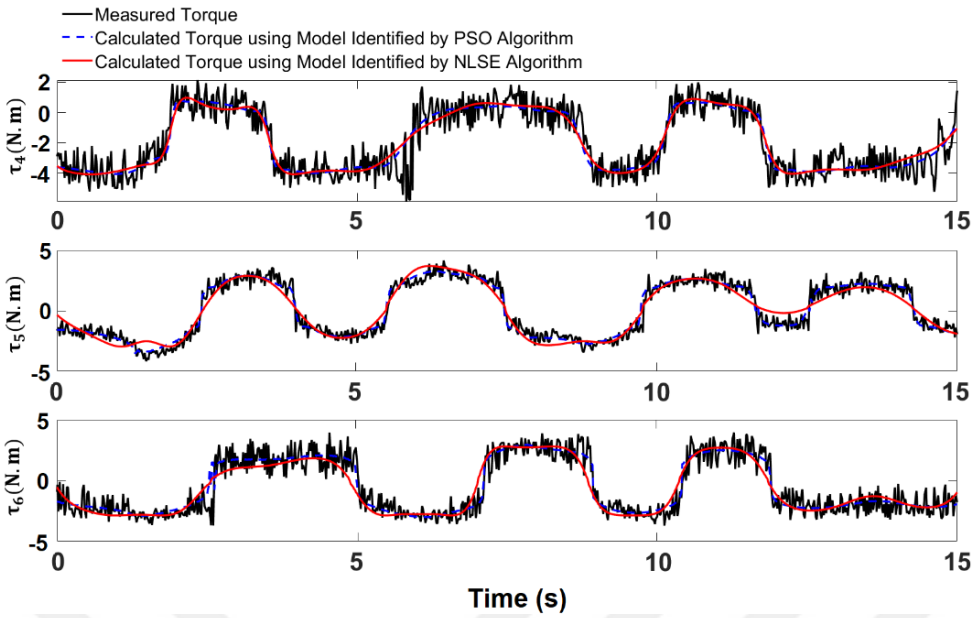


Figure 5.29 : The measured torque of joints along with the calculated toques through models identified with determined parameters by the PSO and NLSE methods (second group of joints).

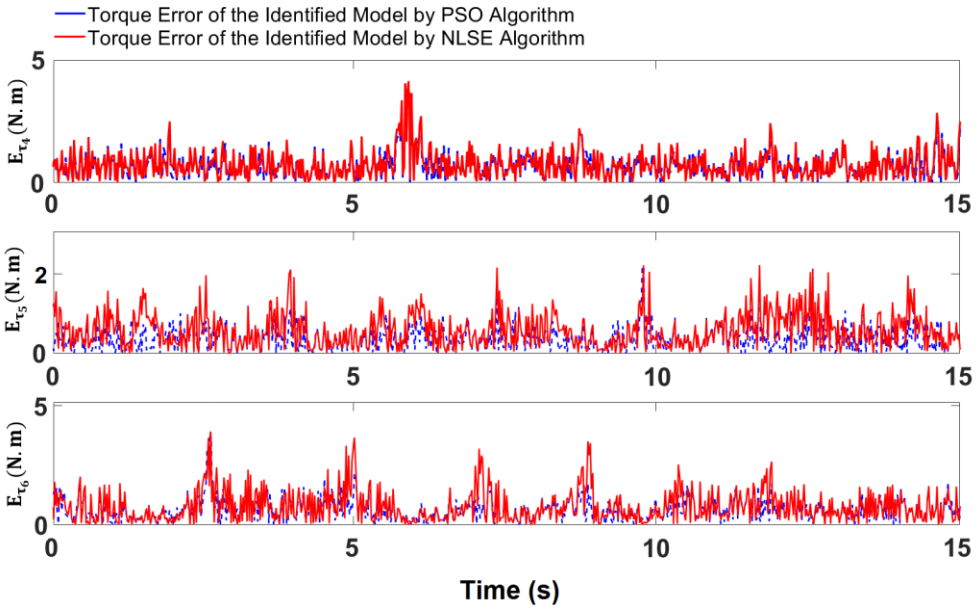


Figure 5.30 : The torque errors of joints through models identified with determined parameters by the PSO and NLSE methods (second group of joints).

Finally, the MSE and R^2 values of the identified model outputs with the determined parameters by the PSO and NLSE algorithms with the test input signal are computed and provided in Table 5.15.

Table 5.15 : The MSE and R^2 values for the output of the identified model with determined parameters by the PSO and NLSE methods for the UR5 robotic arm.

i^{th} Joint	PSO		NLSE	
	MSE (N.m)	R^2 (%)	MSE (N.m)	R^2 (%)
1	3.13	97.55	19.98	90.27
2	5.43	97.47	12.6	90.49
3	2.19	96.81	13.32	88.99
Average (1)	3.64	97.27	15.3	89.91
4	0.64	87.28	0.84	83.05
5	0.24	93.53	0.68	87.97
6	0.57	88.25	0.96	83.91
Average (2)	0.48	89.68	0.82	84.97

Due to Table 5.15, the MSE values for the first set of joints, including the base, shoulder, and elbow ($i = 1:3$) which are obtained by the PSO algorithm, are greater than the ones obtained by the NLSE method. The average values of the MSE algorithm for the first set of joints are computed to be 3.64 N.m and 15.3 N.m for the PSO and NLSE methods, respectively. For the second set of joints, including wrist i ($i = 4:6$), the MSE values of the models identified by the PSO algorithm are greater than the MSE values computed for the models identified by the NLSE method. The average MSE values for the second set of joints are computed to be 0.48 N.m and 0.82 N.m for the PSO and NLSE methods, respectively. Comparing the results of the MSE values for both sets of joints shows that the model output with the parameters determined by the PSO algorithm has smaller MSE values than the ones computed for the output of the models identified with the parameters determined by the NLSE algorithm.

For further evaluation of the output of the identified models using both algorithms, the R^2 values for the joints of the robotic arm could be evaluated as follows. The average R^2 values for the first set of joints ($i = 1:3$) are calculated to be 97.27 % and 89.91 % for the output of the determined models by the PSO and NLSE methods, respectively. The average R^2 values for the second set of joints ($i = 4:6$) are calculated to be 89.68 % and 84.97 % for the output of the determined models by the PSO and NLSE methods, respectively. Consequently, the identified models with determined parameters by the PSO algorithm could compute the output torque of the joints more accurately when compared to the identified models with determined parameters by the NLSE algorithm.

6. CONCLUSIONS

In this chapter, the conclusion and future studies are presented in sub-sections 6.1 and 6.2, respectively.

6.1 Conclusion

One of the primary goals of science is to identify and describe the structure and the physical laws in nature. When the data corresponding to the input and output of a system is available, but the underlying rules and the structure of the system are unknown, it is essential to employ various approaches to determine these rules and structures. Determination of the underlying rules and structure of a system, particularly in some response regions, is a difficult task because of the presence of some nonlinear phenomena in the structure of the model. Therefore, choosing a reliable approach to identify the model structure in the different working regions of the system is crucial. If the structure of the model containing different nonlinear equations such as nonlinear frictions or dead-zone is available for any mechanical system, any conventional or intelligent parameter estimation approach may be utilized to find the parameters of the mathematical models of the physical system. Finding a non-zero parameter, on the other hand, does not imply the presence of the nonlinear term in the mathematical model of the physical system or vice versa. As a result, a thorough statistical investigation of the verification methods is needed to assure the identification process's trustworthiness. In this thesis, different nonlinearities that may exist in the mathematical models of mechanical systems are tried to be detected and determined accurately, and their existence is also investigated statistically. The suggested framework in this study could be summarized as follows: In the first step, the Particle Swarm (PSO) algorithm was coded to determine the parameters of the general model of the systems by both artificial and experimental input-output datasets. Second, to compare the results of the PSO algorithm in the determination of the parameters of the model of the systems, the conventional Nonlinear Least Squared Errors (NLSE) estimation method was utilized to determine the parameters of the systems' model with

nonlinearities. Third, different statistical analysis methods, such as the null hypothesis test and confidence interval test, were conducted to examine the credibility of the identification results. Finally, the capability of the identified models in calculating the output of real systems is evaluated using model evaluation criteria such as MSE and R^2 values. The suggested framework in this thesis is applied to four different real-world case studies. The application of the suggested framework in this study for the case studies and the contribution of this study are discussed as follows: First, two case studies are chosen to implement the suggested framework using the simulation input-output data. For this purpose, a DC motor and a solenoid actuator are chosen, and their simulation models with different combinations of nonlinear terms are constructed. Second, the experimental data were acquired from other case studies including a ball and beam mechanism and a UR5 manipulator, which is then employed to check the reliability of the suggested framework in this research.

In the first case study (a DC motor), the capability of the suggested framework in this study along with the linear modeling methods in representing the behavior of the system with different nonlinearities such as static, Coulomb, viscous, and Steribeck frictions are investigated. It was shown that the linear system identification methods such as transfer function and state-space models are unable to represent the exact response of the DC motor when it contains some nonlinearities in its dynamic model. Therefore, nonlinear system identification methods, including the PSO and the NLSE methods, are chosen to establish mathematical models of the physical systems for the rest of the studies. In addition, the performance of the PSO and NLSE algorithms in determining an exact parameter of the DC motor model is tested using models with different combinations of nonlinearities. It was shown that the PSO algorithm could determine the parameters of the model with less error than the NLSE algorithm. Then, the statistical hypothesis tests were performed, and the results obtained from the PSO algorithm were validated using these tests. It was shown that the determined parameters by the PSO algorithm are statistically significant or insignificant which shows their existence or nonexistence in the nonlinear model of the DC motor. Finally, model evaluation criteria including absolute error, MSE, and R^2 are calculated when new testing input data was supplied to the system. It was shown that the DC motor model with parameters determined by the PSO algorithm computes the output of the system with smaller errors compared to ones calculated by the NLSE algorithm.

In the second case study (a solenoid actuator), the FEM simulation of the solenoid actuator was conducted to compute the magnetic force values of the real-world actuator. The magnetic field analysis was performed to calculate the magnetic force generated in each position of the plunger inside the guider part of the solenoid actuator. Once the simulation data were verified using experimental data, the simulation studies were completed for different current inputs of the solenoid actuator, and their corresponding magnetic forces, along with the position of the solenoid actuator, were calculated. Once the complete input-output data of the solenoid actuator was constructed, the model of the solenoid actuator was simulated using Simulink/MATLAB. Then, the PSO and NLSE algorithms are used to find the unknown parameters of the simulation model. It was shown that the PSO algorithm determines the unknown parameters of the nonlinear model of the system with smaller errors compared to the ones determined by the NLSE algorithm. In addition, in order to examine the credibility of the PSO algorithm in determining the exact parameters of the model of the solenoid actuator, the statistical hypothesis test was performed. The test results showed that the PSO algorithm accurately determined the parameters of the nonlinear model, which exactly exist or do not exist in the model of the actuator. Finally, for further evaluation of the output of the model, the MSE and R^2 values are calculated when the new testing input signal was provided to the system. It was shown that the identified model of the DC motor with parameters determined by the PSO algorithm computes the output of the system with smaller errors compared to those computed by the NLSE algorithm. In addition, the PSO algorithm with multiple cost functions is coded to find the optimal geometry of the real-world solenoid actuator. The solenoid actuator with optimal design parameters was fabricated, and the results were compared. It was shown that, despite the simulation results, the fabricated solenoid actuator only satisfied one of the goals of the optimization. The reasons for the inconsistency between the fabricated and simulation results of the optimized solenoid actuator could be investigated as follows: The magnetic force values was greatly affected by the characteristics of the ferromagnetic materials of the component. Because the B-H curve of the materials employed in the fabrication of the actuator was not accessible, its precise magnetic characteristics couldn’t be utilized for the simulations studies. Another possibility for the inconsistency between the fabricated and simulation results was the errors that occur in the machining process of the manufacturing.

In the third case study (a ball and beam mechanism), experimental data were acquired to check the applicability of the suggested framework in this study to somewhat complicated real-world mechanisms. For this purpose, the experimental input-output data were acquired from the system. Then, the dynamic model of the system, including all nonlinear terms, was established using MATLAB/Simulink, and the PSO and NLSE algorithms were employed to determine unknown parameters of the friction, dead-zone, and time-delay of the physical system. The PSO algorithm results showed that different friction terms such as static, Coulomb, Steribeck, and linear parts of the viscous friction and time-delay exist in the model of the system. On the contrary, due to the results of the PSO algorithm, the parameters of the nonlinear part of the viscous friction and the parameters of dead-zone nonlinearity are obtained to be zero. Then, statistical tests, including hypothesis testing and confidence intervals, were performed to examine the credibility of the results of the PSO algorithm. It was demonstrated that the PSO algorithm could reliably determine the parameters of the nonlinearities that exist or do not exist in the model of the system. Furthermore, the functionality of the PSO in the determination of the parameters of the nonlinear model when they are being varied due to experimental conditions was checked. For this purpose, the adhesive paper tape was attached to the surface of the beam to change the value of the different frictions between the ball and the beam. In this case, there was no direct contact between the steel ball and the aluminum beam. Then, the PSO algorithm was employed to find the unknown varied parameters of the nonlinear model. The PSO algorithm results showed that different friction terms such as static, Coulomb, Steribeck, and the linear part of the viscous friction, which was varied due to the new experiment condition, were accurately identified. Then, the statistical tests, including hypothesis testing and confidence intervals, were PERFORMED to examine the credibility of the results of the PSO algorithm. It was demonstrated that the PSO algorithm could reliably determine the parameters of the nonlinearities that were varied in the model of the system.

In the fourth case study (a UR5 robotic manipulator), experimental data were acquired to check the applicability of the suggested framework in this study to more complicated real-world problems. For this aim, the experimental input-output data were collected from the UR5 robotic arm, and the kinematic and dynamic analysis of the UR5 manipulator was performed and developed using MATLAB. The process of

analytical modeling of the UR5 arm is explained as follows: First, the Denavit-Hartenberg (DH) approach is employed to develop the robot's kinematics at both position and velocity levels. Using this approach, the tip-point position, and its orientation in accordance with the reference coordinate frame are computed parametrically in terms of joint angles. In addition, the transformation matrices of each link of the robot are calculated. After calculating the kinematics of the UR5 manipulator at the position level, the kinematics of the robot at the velocity level is obtained. For this purpose, Then, using the tip-point Jacobian, the liner and the angular velocities of the tip-point are computed parametrically in terms of the joint velocities. In the next step, the Jacobian matrices are calculated for the mass center of the links of the UR5 manipulator. After calculating the kinematics of the UR5 manipulator at both position and velocity levels, the inverse dynamic model of the manipulator is established. The information obtained at the kinematic level, along with the values of the mass and mass centers of the links, is then utilized to calculate the components of the inertia matrix, the Coriolis and centrifugal matrices, and the gravity vector in the dynamic model of the robot. Then, to develop a precise dynamic model of the UR5 manipulator, the nonlinearities that affect the performance of the manipulator were added to the model of the joints. For this purpose, different nonlinear frictions that may occur in the motor and strain wave gearing mechanisms of each of the joints of the robot were added to the models of the joints. Then, the PSO and NLSE methods were employed to determine the unknown friction parameters in all joints of the robot. The results of the PSO algorithm showed that different friction terms such as static, Coulomb, Steribeck, and linear parts of the viscous friction exist in the model of the joint of the manipulator. However, constants of the nonlinear viscous friction are found to be zero using the PSO algorithm, which indicates the nonexistence of these parameters in the model of the joints. Then, statistical analysis methods, such as hypothesis tests and confidence interval tests, were conducted to check the reliability of the results of the PSO algorithm. It was demonstrated that the PSO algorithm could reliably determine the parameters of the nonlinearities that exist or do not exist in the mathematical model of the joints of the robot. Finally, for further evaluation of the output torques of the robot joints, the MSE and R^2 values are calculated. It was demonstrated that the identified model of the robot with parameters determined by the PSO algorithm computes the torque of the joints with smaller errors when compared to ones computed by the NLSE method.

After a brief explanation of the studies conducted in this research, the contributions of the study could be listed as follows:

- A PSO algorithm with multiple cost functions is coded and utilized to find the optimal parameters of the geometry of the real-world solenoid actuator. The solenoid actuator with optimized parameters using this framework was built and tested.
- It was shown that linear modeling methods such as transfer function or state-space models are unable to represent the exact behavior of the models with different nonlinearities. Therefore, nonlinear modeling methods were chosen to develop exact mathematical models of the systems with different nonlinearities.
- Mathematical models of physical systems from simple to more complicated mechanisms, where each term of the model is physically interpretable, were developed accurately.
- The credibility of the obtained results was tested using statistical analysis methods. For this purpose, the identified nonlinear model parameters in the real-world components are validated using null hypotheses and confidence interval tests. In the literature, these methods have not been utilized for these purposes previously.
- The phenomena such as friction, dead-zone, and time-delay which are physically interpretable and adversely affect the performance of the mechanical systems, were identified and verified accurately utilizing the PSO algorithm and the statistical analysis methods.
- The functionality of the model identification and statistical verification algorithms was tested by conducting different experiments by changing the characteristics of the nonlinear terms in a real-world system. For instance, the friction between the ball and beam mechanism was changed by adding an adhesive tape band between the contact surface of the ball and the beam of the mechanism. Then, the framework was repeated to examine the applicability of the method when the nonlinearities of the mechanism were changed. It was shown that using the framework suggested in this study, the unknown model

parameters which were varied due to the experimental conditions could be successfully determined.

- A conventional estimation method Nonlinear Least Squares Errors (NLSE) estimation method was employed to solve the nonlinear modeling problems of the different case studies which have not been utilized for this purpose previously.

6.2 Future Works

The future studies could be explained below:

First, in order to make the PSO algorithm more applicable for real-world physical systems where the parameters of the model change due to the working conditions of the systems, the time required to find optimal parameters could be reduced by examining some of the following solutions: First, the inertia weight of the algorithm could be tuned adaptively such that in every iteration, the inertia weight is tuned according to the position of the particle concerning the positions of other particles. The other solution is to combine the PSO algorithm with other population-based optimization algorithms, such as Differential Evolution, to increase the convergence time. When some particles get stuck in local optima and cause more computation time, these methods could be used to increase the efficiency of the algorithms. Other techniques could be investigated to decrease the time required to find optimal values through the PSO algorithm.

Second, other evolutionary approaches such as Simulated Annealing, and Genetic Algorithm algorithms could be used to check the applicability of these methods to solve nonlinear modeling problems. The performance of these methods could then be compared to the PSO algorithm.

Third, other physically interpretable nonlinear terms like flexibility, saturation, and so on that occur in mechanical systems would be obtained from the literature and integrated with the nonlinear mathematical models for further development of the algorithm.

Fourth, the nonlinear model identification framework employed here may be used for a variety of systems with nonlinear models like fluidic or thermal systems and electrical or mechanical systems. Therefore, the framework used in this study would

give new insight into solving both industrial application problems and academic research areas.

Fifth, the proposed methodology might be utilized for different purposes such as fault diagnosis and model-based controller design for the utilized case studies and other real-world problems such as hydraulic, fluidic, and thermal systems.

Finally, other alternative approaches like the Monte Carlo Markov Chain (MCMC) with the Bayesian approach, might be used to identify the parameters of the nonlinear models of the mechanical systems to compare with the findings of this study.

REFERENCES

[1] **Billings, S. A.** (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. London, England: John Wiley & Sons.

[2] **Billings, S. A.** (1980). Identification of nonlinear systems–a survey. *IEE Proceedings D— Control Theory and Applications*, *127*(6), 272–285, November 1980.

[3] **Keesman, K. J., & Keesman, K. J.** (2011). *System identification: an introduction*. London: Springer.

[4] **Tangirala, A. K.** (2018). *Principles of system identification: theory and practice*. Boca Raton: CRC Press.

[5] **Belegundu, A. D., & Chandrupatla, T. R.** (2019). *Optimization concepts and applications in engineering*. New York, USA: Cambridge University Press.

[6] **Pintelon, R., & Schoukens, J.** (2012). *System identification: a frequency domain approach*. London, England: John Wiley & Sons.

[7] **Lai, C. Y., Xiang, C., & Lee, T. H.** (2011). Data-based identification and control of nonlinear systems via piecewise affine approximation. *IEEE transactions on neural networks*, *22*(12), 2189-2200.

[8] **Socha, L.** (2007). *Linearization methods for stochastic dynamic systems (Vol. 730)*. Heidelberg: Springer Science & Business Media.

[9] **Julier, S. J., & Uhlmann, J. K.** (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401-422.

[10] **Stojanovic, V., & Filipovic, V.** (2014). Adaptive input design for identification of output error model with constrained output. *Circuits, Systems, and Signal Processing*, *33*(1), 97-113.

[11] **Chen, B., Zhu, Y., Hu, J., & Principe, J. C.** (2011). A variable step-size SIG algorithm for realizing the optimal adaptive FIR filter. *International Journal of Control, Automation and Systems*, *9*(6), 1049-1055.

[12] **Ogata, K.** (2010). *Modern control engineering*. Upper Saddle River, NJ: Prentice Hall.

[13] **Schoukens, J., & Pintelon, R.** (1991). *Identification of linear systems: a practical guideline to accurate modeling*. Oxford, UK: Pergamon Press.

[14] **Van den Bosch, P. P., & Van der Klauw, A. C.** (2020). *Modeling, identification and simulation of dynamical systems*. London, England: CRC Press.

[15] **Wong, C. X., & Worden, K.** (2007). Generalised NARX shunting neural network modelling of friction. *Mechanical Systems and Signal Processing*, *21*(1), 553-572.

[16] **Ge, X., Luo, Z., Ma, Y., Liu, H., & Zhu, Y.** (2019). A novel data-driven model based parameter estimation of nonlinear systems. *Journal of Sound and Vibration*, *453*, 188-200.

[17] **Wang, B. M., Xu, J. X., Zhu, F., & Wu, Z. X.** (2013). Analysis and Models for the Friction Coefficient of a Sliding Bearing. *Applied Mechanics and Materials, 395*, 826-830.

[18] **Wan, Y., Wong, C. X., Dodd, T. J., & Harrison, R. F.** (2005). Application of a kernel method in modeling friction dynamics. *IFAC Proceedings Volumes, 38*(1), 698-703.

[19] **Naerum, E., Cornella, J., & Elle, O. J.** (2008, May). Wavelet networks for estimation of coupled friction in robotic manipulators. In *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 19-23 May.

[20] **Loyola-Gonzalez, O.** (2019). Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access*, 7, 154096-154113.

[21] **Marques, F., Flores, P., Pimenta Claro, J. C., & Lankarani, H. M.** (2016). A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems. *Nonlinear Dynamics, 86*(3), 1407-1443.

[22] **Tadić, B., Kočović, V., Matejić, M., Brzaković, L., & Mijatović, M.** (2016). Static coefficient of rolling friction at high contact temperatures and various contact pressure. *Tribology in Industry, 38*(1), 83-89.

[23] **De Wit, C. C., Olsson, H., Astrom, K. J., & Lischinsky, P.** (1995). A new model for control of systems with friction. *IEEE Transactions on automatic control, 40*(3), 419-425.

[24] **Olsson, H., Åström, K. J., De Wit, C. C., Gäfvert, M., & Lischinsky, P.** (1998). Friction models and friction compensation. European Journal of Control, *4*(3), 176-195.

[25] **Armstrong-Helouvry, B.** (1990). Stick-slip arising from stribeck friction. In Proceedings., *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA, 13-18 May, 1377-1382.

[26] **Haessig Jr, D. A., & Friedland, B.** (1991). On the modeling and simulation of friction. *Journal of Dynamic Systems, Measurement, and Control, 113*(3), 354-362

[27] **Makkar, C., Dixon, W. E., Sawyer, W. G., & Hu, G.** (2005). A new continuously differentiable friction model for control systems design. In Proceedings, *2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Monterey, CA, 24-28 July, 600-605.

[28] **Quiñones-Cisneros, S. E., Zéberg-Mikkelsen, C. K., & Stenby, E. H.** (2000). The friction theory (f-theory) for viscosity modeling. *Fluid Phase Equilibria, 169*(2), 249-276.

[29**] Wolf, S., & Iskandar, M.** (2018). Extending a dynamic friction model with nonlinear viscous and thermal dependency for a motor and harmonic drive gear. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 21-25 May, 783-790.

[30] **Gautier, M.** (1997). Dynamic identification of robots with power model. In *Proceedings of international conference on robotics and automation*, Albuquerque, NM, USA, 25-25 April, 1922-1927.

[31] **Gautier, M., & Poignet, P.** (2001). Extended Kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice, 9*(12), 1361-1372.

[32] **Swevers, J., Verdonck, W., & De Schutter, J.** (2007). Dynamic model identification for industrial robots. *IEEE control systems magazine, 27*(5), 58-71.

[33] **Bittencourt, A. C., & Axelsson, P.** (2013). Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data. *IEEE/ASME transactions on mechatronics, 19*(5), 1694-1706.

[34] **Huber, P. J.** (1972). The 1972 wald lecture robust statistics: A review. *The Annals of Mathematical Statistics, 43*(4), 1041-1067.

[35] **Norton, J. P.** (2009). *An introduction to identification*. New York, United States: Courier Corporation.

[36] **Shaw, S. R., & Laughman, C. R.** (2006). A method for nonlinear least squares with structured residuals. *IEEE transactions on automatic control, 51*(10), 1704-1708.

[37] **Mata, V., Benimeli, F., Farhat, N., & Valera, A.** (2005). Dynamic parameter identification in industrial robots considering physical feasibility. *Advanced Robotics, 19*(1), 101-119.

[38] **Olsen, M. M., Swevers, J., & Verdonck, W.** (2002). Maximum likelihood identification of a dynamic robot model: Implementation issues. *The International Journal of robotics research, 21*(2), 89-96.

[39] **Olsen, M. M., & Petersen, H. G.** (2001). A new method for estimating parameters of a dynamic robot model. *IEEE transactions on robotics and automation, 17*(1), 95-100.

[40] **Linderoth, M., Stolt, A., Robertsson, A., & Johansson, R.** (2013). Robotic force estimation using motor torques and modeling of low velocity friction disturbances. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 3-7 November, 3550-3556.

[41] **Bittencourt, A. C., & Axelsson, P.** (2013). Modeling and experiment design for identification of wear in a robot joint under load and temperature uncertainties based on friction data. *IEEE/ASME transactions on mechatronics, 19*(5), 1694-1706.

[42] **Ge, W., Wang, B., & Mu, H.** (2019). Dynamic parameter identification for reconfigurable robot using adaline neural network. In *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, Tianjin, China, 4-7 August, 319-324.

[43] **Li, X., Gu, J., Sun, X., Li, J., & Tang, S.** (2022). Parameter identification of robot manipulators with unknown payloads using an improved chaotic sparrow search algorithm. *Applied Intelligence, 52*(9), 10341–10351.

[44] **Feng, Y., & Zhang, K.** (2022). Dynamic parameter identification and nonlinear friction compensation method for safety perception of heavy explosion-proof robots. *Journal of Control and Decision, 9*(4), 455-464.

[45] **Wu, R. H., & Tung, P. C.** (2002). Studies of stick-slip friction, presliding displacement, and hunting. *Journal of Dynamic Systems, Measurement, and Control, 124*(1), 111-117.

[46] **Sorensen, K. L., Cross, P. W., Singhose, W. E., & Prakash, S.** (2011). Vibration analysis and mitigation of dead-zone on systems using two-impulse zero-vibration input shaping. *Journal of Computational and Nonlinear Dynamics, 6*(1), 011011.

[47] **Li, X., & Song, S.** (2016). Stabilization of delay systems: delay-dependent impulsive control. *IEEE Transactions on Automatic Control, 62*(1), 406-411.

[48] **Ljung, L., & Glad, T.** (1994). *Modeling of dynamic systems*. New Jersey, United States, Prentice-Hall, Inc.

[49] **Wang, W., & Wong, A. K.** (2002). Autoregressive model-based gear fault diagnosis, *Journal of Vibration and Acoustics, 124*(2), 172-179.

[50] **Galrinho, M., Everitt, N., & Hjalmarsson, H.** (2017). ARX modeling of unstable linear systems. *Automatica, 75*, 167-171.

[51] **Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M.** (2015). *Time series analysis: forecasting and control*. Hoboken, NJ: John Wiley & Sons.

[52] **Rachad, S., Nsiri, B., & Bensassi, B.** (2015). System identification of inventory system using ARX and ARMAX models. *International Journal of Control and Automation, 8*(12), 283-294.

[53] **Stojanovic, V., & Filipovic, V.** (2014). Adaptive input design for identification of output error model with constrained output. *Circuits, Systems, and Signal Processing, 33*(1), 97-113.

[54] **Chen, B., Zhu, Y., Hu, J., & Principe, J. C.** (2011). A variable step-size SIG algorithm for realizing the optimal adaptive FIR *filter. International Journal of Control, Automation and Systems, 9*(6), 1049-1055.

[55] **Ogata, K.** (2010). *Modern control engineering*. Upper Saddle River, New Jersey: Prentice Hall.

[56] **Schoukens, J., & Pintelon, R.** (2014). *Identification of linear systems: a practical guideline to accurate modeling*. London, United Kingdom: Elsevier.

[57] **McCullagh, P., & Nelder, J. A.** (2019). *Generalized linear models.* New York, United States: Routledge.

[58] **Melsa, J. L. (Ed.).** (1971). *System identification*. New York, United States: Academic Press.

[59] **Lai, C. Y., Xiang, C., & Lee, T. H.** (2011). Data-based identification and control of nonlinear systems via piecewise affine approximation. *IEEE transactions on neural networks, 22*(12), 2189-2200.

[60] **Socha, L.** (2007). *Linearization methods for stochastic dynamic systems* (Vol. 730). Berlin, Germany: Springer Science & Business Media.

[61] **Volterra, V., & Whittaker, E. T.** (1959). *Theory of functionals and of integral and integro-differential equations*. New York, United States: Dover publications.

[62] **Cheng, C. M., Peng, Z. K., Zhang, W. M., & Meng, G.** (2017). Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review. *Mechanical Systems and Signal Processing, 87*, 340-364.

[63] **Yu, Y., Zhang, C., Han, Z., & Zhou, M.** (2021). Hysteresis Modeling of Magnetic Shape Memory Alloy Actuator Based on Volterra Series. *IEEE Transactions on Magnetics, 57*(7), 1-4.

[64] **Chatterjee, A., & Chintha, H. P.** (2021). Identification and parameter estimation of asymmetric nonlinear damping in a single-degree-of-freedom system using Volterra series. *Journal of Vibration Engineering & Technologies, 9*(5), 817-843.

[65] **Favier, G., Kibangou, A. Y., & Bouilloc, T.** (2012). Nonlinear system modeling and identification using Volterra-PARAFAC models. International *Journal of Adaptive Control and Signal Processing, 26*(1), 30-53.

[66] **Pearson, R. K., & Pottmann, M.** (2000). Gray-box identification of block-oriented nonlinear models. Journal of Process Control, *10*(4), 301-315.

[67] **Khalfi, J., Boumaaz, N., Soulmani, A., & Laadissi, E. M.** (2021). Nonlinear Modeling of Lithium-Ion Battery Cells for Electric Vehicles using a Hammerstein–Wiener Model. *Journal of Electrical Engineering & Technology, 16*(2), 659-669.

[68] **Lacy, S. L., & Bernstein, D. S.** (2002). Identification of an electromagnetic actuator. In *IEEE CONFERENCE ON DECISION AND CONTROL*, Las Vegas, NV, USA, 10-13 December, 4521-4526.

[69] **Yan, J., Li, B., Guo, G., Zeng, Y., & Zhang, M.** (2013). Nonlinear modeling and identification of the electro-hydraulic control system of an excavator arm using BONL model. *Chinese Journal of Mechanical Engineering, 26*(6), 1212-1221.

[70] **Wu, Q., Dong, S., Zhang, W. A., & Yu, L.** (2021). Online Modeling of the CNC Engraving System With Dead-Zone Input Nonlinearity. *IEEE Transactions on Industrial Electronics, 69*(1), 774-782.

[71] **Schoukens, M., & Tiels, K.** (2017). Identification of block-oriented nonlinear systems starting from linear approximations: A survey. *Automatica, 85*, 272-292.

[72] **Rahrooh, A., & Shepard, S.** (2009). Identification of nonlinear systems using NARMAX model. *Nonlinear Analysis: Theory, Methods & Applications, 71*(12), 1198-1202.

[73] **Chiras, N., Evans, C., & Rees, D.** (2001). Nonlinear gas turbine modeling using NARMAX structures. *IEEE Transactions on Instrumentation and Measurement, 50*(4), 893-898.

[74] **Kukreja, S. L., & Brenner, M. J.** (2006). Nonlinear aeroelastic system identification with application to experimental data. *Journal of guidance, control, and dynamics*, *29*(2), 374-381.

[75] **Zito, G., & Landau, I. D.** (2005). A methodology for identification of narmax models applied to diesel engines. *IFAC Proceedings Volumes*, *38*(1), 374-379.

[76] **Noël, J. P., & Kerschen, G.** (2017). Nonlinear system identification in structural dynamics: 10 more years of progress. *Mechanical Systems and Signal Processing*, *83*, 2-35.

[77] **Ma, Y., Liu, H., Zhu, Y., Wang, F., & Luo, Z.** (2017). The NARX model-based system identification on nonlinear, rotor-bearing systems. *Applied Sciences*, *7*(9), 911.

[78] **Barbosa, M. P. S., da Costa, D. P., & Ayala, H. V. H.** (2020). Evaluation of Nonlinear System Identification to Model Piezoacoustic Transmission. *IFAC-PapersOnLine*, *53*(2), 8802-8807.

[79] **Pawlus, W., Karimi, H. R., & Robbersmyr, K. G.** (2013). Data-based modeling of vehicle collisions by nonlinear autoregressive model and feedforward neural network. *Information Sciences*, *235*, 65-79.

[80] **Demiris, E. N., Likothanassis, S. D., Beligiannis, G. N., & Adamopoulos, A.** (2000). Nonlinear AR model identification with unknown process order. In *Proceedings IEEE international symposium intelligent signal processing and communication systems (ISPACS)*, Hilton Hotel and Convention Center, Istanbul, Turkey, 5-9 June, 777-782.

[81] **Amrit, R., & Saha, P.** (2007). Stochastic identification of bioreactor process exhibiting input multiplicity. *Bioprocess and biosystems engineering*, *30*(3), 165-172.

[82] **Piroddi, L., & Spinelli, W.** (2003). An identification algorithm for polynomial NARX models based on simulation error minimization. *International Journal of Control*, *76*(17), 1767-1781.

[83] **Chiu, H. L.** (2022). Identification Approach for Nonlinear MIMO Dynamics of Closed-Loop Active Magnetic Bearing System. *Applied Sciences*, *12*(17), 8556.

[84] **Ljung, L.** (1998). System identification. In *Signal analysis and prediction*. Birkhäuser, Boston, MA.

[85] **Fomby, T. B., Johnson, S. R., & Hill, R. C.** (1984). Review of ordinary least squares and generalized least squares. In *Advanced econometric methods*. Springer, New York, NY.

[86] **Cantrell, C. A.** (2008). Review of methods for linear least-squares fitting of data and application to atmospheric chemistry problems. *Atmospheric Chemistry and Physics*, *8*(17), 5477-5487.

[87] **Hammarling, S., & Lucas, C.** (2008). Updating the QR factorization and the least squares problem. *MIMS EPrint 2008*, University of Manchester, Manchester, UK.

[88] **Gibbs, B. P.** (2011). *Advanced Kalman filtering, least-squares and modeling: a practical handbook*, Hoboken, United States: John Wiley & Sons.

[89] **C., Pereyra, V., & Scherer, G.** (2013). *Least squares data fitting with applications*. Maryland, United States: Johns Hopkins University Press.

[90] **Markovsky, I., & Van Huffel, S.** (2007). Overview of total least-squares methods. *Signal processing, 87*(10), 2283-2302.

[91] **Yang, B., & Chon, K. H.** (2010). Estimating time-varying nonlinear autoregressive model parameters by minimizing hypersurface distance. *IEEE Transactions on Biomedical Engineering*, *57*(8), 1937-1944.

[92] **Worden, K., & Tomlinson, G. R.** (2019). *Nonlinearity in structural dynamics: detection, identification and modelling*. Boca Raton: CRC Press.

[93] **Kay, S. M.** (1993). *Fundamentals of statistical signal processing: estimation theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

[94] **Marquardt, D. W.** (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics, 11*(2**)**, 431-441.

[95] **Eriksson, J.** (1996). *Optimization and regularization of nonlinear least squares problems*. (Ph.D. thesis), Department of Computing Science, Umea University, Sweden.

[96**] Bonate, P. L.** (2006). Nonlinear models and regression. In *Pharmacokinetic-Pharmacodynamic Modeling and Simulation*, Springer, Boston, MA. 93-124.

[97] **Hansen, P. C., Pereyra, V., & Scherer, G.** (2013). *Least squares data fitting with applications*. Maryland, United States: JHU Press.

[98] **Corriou, J. P.** (2021). Numerical Methods of Optimization. In *Numerical Methods and Optimization*. Cham, Switzerland: Springer, 505-574.

[99] **Gavin, H. P.** (2019). The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering, Duke University, 1-19.

[100] **Cao, J., Huang, J. Z., & Wu, H.** (2012). Penalized nonlinear least squares estimation of time-varying parameters in ordinary differential equations. *Journal of computational and graphical statistics*, *21*(1), 42-56.

[101] **Zhou, W., & Zhang, L.** (2010). Global convergence of a regularized factorized quasi-Newton method for nonlinear least squares problems. *Computational & Applied Mathematics*, *29*, 195-214.

[102] **Lai, W. H., Kek, S. L., & Tay, K. G.** (2017). Solving nonlinear least squares problem using Gauss-Newton method. *International Journal of Innovative Science, Engineering & Technology*, *4*, 258-262.

[103] **Wei, B., & Xie, N.** (2021). Parameter estimation for grey system models: A nonlinear least squares perspective. *Communications in Nonlinear Science and Numerical Simulation*, *95*, 105653.

[104] **Gavin, H. P.** (2019). The Levenberg–Marquardt method for nonlinear least squares curve-fitting problems. Dept. Civil and Environmental Engineering Duke Univ, 19.

[105] **Bellavia, S., Gratton, S., & Riccietti, E.** (2018). A Levenberg–Marquardt method for large nonlinear least-squares problems with dynamic accuracy in functions and gradients. *Numerische Mathematik, 140*(3), 791-825.

[106] **Barati, R.** (2011). Parameter estimation of nonlinear Muskingum models using Nelder-Mead simplex algorithm. *Journal of Hydrologic Engineering, 16*(11), 946-954.

[107] **Myung, I. J.** (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology, 47*(1), 90-100.

[108] **Hey, J., & Nielsen, R.** (2007). Integration within the Felsenstein equation for improved Markov chain Monte Carlo methods in population genetics. *Proceedings of the National Academy of Sciences, 104*(8), 2785-2790.

[109] **Doornik, J. A., & Ooms, M.** (2003). Computational aspects of maximum likelihood estimation of autoregressive fractionally integrated moving average models. *Computational Statistics & Data Analysis, 42*(3), 333-348.

[110] **Lele, S. R., Dennis, B., & Lutscher, F.** (2007). Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods. *Ecology Letters, 10*(7), 551-563.

[111] **Maiti, R., Sharma, K. D., & Sarkar, G.** (2018). PSO based parameter estimation and PID controller tuning for 2-DOF nonlinear twin rotor MIMO system. *Int. J. Autom. Control., 12*(4), 582-609.

[112] **Katoch, S., Chauhan, S. S., & Kumar, V.** (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80*(5), 8091-8126.

[113] **Carey, M., & Ramsay, J. O.** (2021). Fast stable parameter estimation for linear dynamical systems. *Computational Statistics & Data Analysis, 156*, 107124.

[114] **Gharehchopogh, F. S., & Gholizadeh, H.** (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation, 48*, 1-24.

[115] **Dubey, M., Kumar, V., Kaur, M., & Dao, T. P.** (2021). A systematic review on harmony search algorithm: theory, literature, and applications. *Mathematical Problems in Engineering, 2021*, 1-22.

[116] **Maiti, R., Sharma, K. D., & Sarkar, G.** (2018). PSO based parameter estimation and PID controller tuning for 2-DOF nonlinear twin rotor MIMO system. *Int. J. Autom. Control, 12*(4), 582-609.

[117] **Le Guisquet, S., & Amabili, M.** (2021). Identification by means of a genetic algorithm of nonlinear damping and stiffness of continuous structures subjected to large-amplitude vibrations. Part I: single-degree-of-freedom responses. *Mechanical Systems and Signal Processing, 153*, 107470.

[118] **K.M. El-Naggar, M.R. AlRashidi, M.F. AlHajri, A.K. Al-Othman** (2012 Simulated Annealing algorithm for photovoltaic parameters identification. *Sol Energy, 86*(1), 266-274.

[119] **Rana, N., Latiff, M. S. A., Abdulhamid, S. I. M., & Chiroma, H.** (2020). Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments. *Neural Computing and Applications, 32*(20), 16245-16277.

[120] **Ala'a, A., Alsewari, A. A., Alamri, H. S., & Zamli, K. Z.** (2019). Comprehensive review of the development of the harmony search algorithm and its applications. *IEEE Access, 7*, 14233-14245.

[121**] Cohen, A. C., & Whitten, B. J.** (2020). *Parameter estimation in reliability and life span models*. Boca Raton: CRC Press.

[122] **Khalil, M., Sarkar, A., & Adhikari, S.** (2009). Nonlinear filters for chaotic oscillatory systems. *Nonlinear Dynamics, 55*(1), 113-137.

[123] **Khalil, M., Poirel, D., & Sarkar, A.** (2013). Probabilistic parameter estimation of a fluttering aeroelastic system in the transitional Reynolds number regime. *Journal of Sound and Vibration, 332*(15), 3670-3691.

[124] **Overschee, P. V.** (1998). Data Processing for System Identification. In *Advanced Instrumentation, Data Interpretation, and Control of Biotechnological Processes* (pp. 191-209). Springer, Dordrecht.

[125] **Nowak, R. D., & Van Veen, B. D.** (1994). Random and pseudorandom inputs for Volterra filter identification. *IEEE Transactions on Signal Processing, 42*(8), 2124-2135.

[126] **Nelles, O.** (2020). Nonlinear Dynamic System Identification. In *Nonlinear System Identification* (pp. 831-891). Springer, Cham.

[127] **Yadav, E. S., & Indiran, T.** (2019). PRBS based model identification and GPC PID control design for MIMO Process. *Materials Today: Proceedings, 17*, 16-25.

[128] **Zhao, Y., Wu, J., Zeng, C., & Huang, Y.** (2022). Identification of hydrodynamic coefficients of a ship manoeuvring model based on PRBS input. *Ocean Engineering, 246*, 110640.

[129] **Nowak, R. D., & Van Veen, B. D.** (1994). Random and pseudorandom inputs for Volterra filter identification. *IEEE Transactions on Signal Processing, 42*(8), 2124-2135.

[130] **Toker, O., & Emara-Shabaik, H. E.** (2004). Pseudo-random multilevel sequences: Spectral properties and identification of Hammerstein systems. *IMA Journal of Mathematical Control and Information, 21*(2), 183-205.

[131] **Medina-Ramos, C., Betetta-Gómez, J., Carbonel-Olazabal, D., & Pilco-Barrenechea, M.** (2013, October). System identification and MPC based on the Volterra-Laguerre model for improvement of the laminator systems performance. In *2013 IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE)* Munich, Germany, 17-19 October.

[132] **R Roman, M., M Shaaban, S., G Rabie, M., & H Aly, M.** (2021). System identification and sliding mode tracking control for electro-hydraulic steer-by-wire system. *Engineering Research Journal, 172*, 121-130.

[133] **Annus, P., Land, R., Min, M., & Ojarand, J.** (2012). Simple signals for system identification. *Fourier Transform-Signal Processing*. InTech: Rijeka, 257-276.

[134] **Xu, L.** (2016). The damping iterative parameter identification method for dynamical systems based on the sine signal measurement. *Signal Processing, 120*, 660-667.

[135] **Brouri, A., Kadi, L., & Lahdachi, K.** (2022). Identification of nonlinear system composed of parallel coupling of Wiener and Hammerstein models. *Asian Journal of Control, 24*(3), 1152-1164.

[136] **Pintelon, R., & Schoukens, J.** (2012). System identification: a frequency domain approach, Hoboken, NJ: John Wiley & Sons.

[137] **Mendez, E. M. A. M., & Billings, S. A.** (2001). An alternative solution to the model structure selection problem. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 31*(6), 597-608.

[138] **Haber, R., & Keviczky, L.** (1999). Nonlinear system identification. 2. *Nonlinear system structure identification* (Vol. 7). Springer Science & Business Media.

[139] **Ling, T. G., Rahmat, M. F., Husain, A. R., & Ghazali, R.** (2011). System identification of electro-hydraulic actuator servo system. In *2011 4th International Conference on Mechatronics (ICOM)*, Kuala Lumpur, Malaysia, 17-19 May.

[140] **Shariff, H. M., Rahiman, M. H. F., & Tajjudin, M.** (2013). Nonlinear system identification: Comparison between PRBS and Random Gaussian perturbation on steam distillation pilot plant. In *2013 IEEE 3rd International Conference on System Engineering and Technology*, Shah Alam, Malaysia, 19-20 August.

[141] **Al-Waeli, A. H., Kazem, H. A., Yousif, J. H., Chaichan, M. T., & Sopian, K.** (2019). Mathematical and neural network models for predicting the electrical performance of a PV/T system. *International Journal of Energy Research, 43*(14), 8100-8117.

[142] **Wang, Y., Li, M., & Chen, Z.** (2020). Experimental study of fractional-order models for lithium-ion battery and ultra-capacitor: Modeling, system identification, and validation. *Applied Energy, 278*, 115736.

[143] **Legowo, A., Sulaeman, E., & Rosli, D.** (2019). Review on system identification for quadrotor unmanned aerial vehicle (UAV). In *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, United Arab Emirates, 26 March - 10 April.

[144] **Cadenas, E., Rivera, W., Campos-Amezcua, R., & Heard, C**. (2016). Wind speed prediction using a univariate ARIMA model and a multivariate NARX model. *Energies, 9*(2), 109.

[145] **Ali, N., Viggiano, B., Tutkun, M., & Cal, R. B.** (2021). Data-driven machine learning for accurate prediction and statistical quantification of two

phase flow regimes. *Journal of Petroleum Science and Engineering, 202*, 108488.

[146] **Kennedy, J., & Eberhart, R.** (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Perth, WA, Australia, November 27 - December 1.

[147] **Jelali, M., & Huang, B. (Eds.).** (2010). *Detection and diagnosis of stiction in control loops: state of the art and advanced methods* (p. 07). London: Springer.

[148] **Kecskés, I., Székács, L., Fodor, J. C., & Odry, P.** (2013). PSO and GA optimization methods comparison on simulation model of a real hexapod robot. In *2013 IEEE 9th International Conference on Computational Cybernetics (ICCC)*, Tihany, Hungary, 8-10 July.

[149] **Gaing, Z. L.** (2003). Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE transactions on power systems, *18*(3), 1187-1195.

[150] **Verma, H., Jain, C., Rathore, A., & Gupta, P.** (2012). A comparative study of GA, PSO and big bang-big crunch optimization techniques for optimal placement of SVC's. *Int. J. Electron. Commun. Comput. Eng.*, 3, 263-269.

[151] **Eberhart, R. C., & Shi, Y.** (1998). Comparison between genetic algorithms and particle swarm optimization. In *International conference on evolutionary programming*, San Diego, CA, USA, 25-27 March

[152] **Clerc, M., & Kennedy, J.** (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation, 6*(1), 58-73.

[153] **Eberhart, R. C., & Shi, Y.** (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 congress on evolutionary computation*, La Jolla, CA, USA, 16-19 July

[154] **Li, M. Y.** (2018). Parameter estimation and nonlinear least-squares methods. In *An Introduction to Mathematical Modeling of Infectious Diseases* (pp. 103-124). Springer, Cham.

[155] **Levenberg, K.** (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics, 2*(2), 164-168.

[156] **Gill, P. E., & Murray, W.** (1978). Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal on Numerical Analysis, 15*(5), 977-992.

[157] **Marquardt, D. W.** (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics, 11*(2), 431-441.

[158] **Berk, K.** (1991). Nonlinear regression. *American Statistical Association, 33*(3), 362-363.

[159] **Ramachandran, K. M., & Tsokos, C. P.** (2020). *Mathematical statistics with applications in R*. MA, USA: Academic Press.

[160] **Montgomery, D. C., & Runger, G. C.** (1994). *Applied statistics and probability for engineers*. New York: John Wiley & Sons.

[161] **Gravetter, F. J., Wallnau, L. B., Forzano, L. A. B., & Witnauer, J. E.** (2020). Essentials of statistics for the behavioral sciences. Belmont, CA: Wadsworth.

[162] **Draper, N. R., & Smith, H.** (1998). *Applied regression analysis* (Vol. 326). New York: John Wiley & Sons.

[163] **Sammut, C., & Webb, G. I. (Eds.).** (2011). *Encyclopedia of machine learning*. Boston, MA, USA: Springer Science & Business Media.

[164] **Janocha, H.** (2004). *Actuators. Berlin*: Springer.

[165] **Gomis-Bellmunt, O., & Campanile, L. F.** (2009). *Design rules for actuators in active mechanical systems*. London: Springer Science & Business Media.

[166] **Mutschler, K., Dwivedi, S., Kartmann, S., Bammesberger, S., Koltay, P., Zengerle, R., & Tanguy, L.** (2014). Multi physics network simulation of a solenoid dispensing valve. *Mechatronics*, *24*(3), 209-221.

[167] **Yang, X., & Liang, K.** (2021). Measurement and modelling of a linear electromagnetic actuator driven camless valve train for spark ignition IC engines under full load condition. *Mechatronics*, *77*, 102604.

[168] **Lu, H., Zhu, J., Lin, Z., & Guo, Y.** (2009). An inchworm mobile robot using electromagnetic linear actuator. *Mechatronics*, *19*(7), 1116-1125.

[169] **Eyabi, P., & Washington, G.** (2006). Modeling and sensorless control of an electromagnetic valve actuator. *Mechatronics*, *16*(3-4), 159-175.

[170] **Yang, M., Zhang, J., & Xu, B.** (2018). Experimental study and simulation analysis on electromagnetic characteristics and dynamic response of a new miniature digital valve. *Advances in Materials Science and Engineering*, 2018. 1-8.

[171] **Mehdi, N., Rehan, M., Malik, F. M., Bhatti, A. I., & Tufail, M.** (2014). A novel anti-windup framework for cascade control systems: An application to underactuated mechanical systems. *ISA transactions*, *53*(3), 802-815.

[172] **Al-Dujaili, A. Q., Humaidi, A. J., Pereira, D. A., & Ibraheem, I. K.** (2021). Adaptive backstepping control design for ball and beam system. *International Review of Applied Sciences and Engineering*, *12*(3), 211-221.

[173] **Danilo Montoya, O., Gil-González, W., & Ramírez-Vanegas, C.** (2020). Discrete-inverse optimal control applied to the ball and beam dynamical system: a passivity-based control approach. *Symmetry*, *12*(8), 1359.

[174] **Mehedi, I. M., Al-Saggaf, U. M., Mansouri, R., & Bettayeb, M.** (2019). Two degrees of freedom fractional controller design: Application to the ball and beam system. *Measurement*, *135*, 13-22.

[175] **Ding, M., Liu, B., & Wang, L.** (2019). Position control for ball and beam system based on active disturbance rejection control. *Systems Science & Control Engineering, 7*(1), 97-108.

[176] **Ali, E. T., Abdullah, S., Amir, M., & Adeel, E. M.** (2019). Stability control of ball and beam system using heuristic computation based PI-D and PI-PD controller. *Technical Journal, 24*(01), 21-29.

[177] **Castillo, O., Lizárraga, E., Soria, J., Melin, P., & Valdez, F.** (2015). New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system. *Information Sciences, 294*, 203-215.

[178] **Tao, G., & Kokotovic, P. V.** (1996). *Adaptive control of systems with actuator and sensor nonlinearities*. New York: John Wiley & Sons, Inc.

[179] **Cheraghpour, F., Moosavian, S. A. A., & Nahvi, A.** (2009). Multiple aspect grasp performance index for cooperative object manipulation tasks. In *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Suntec International Convention and Exhibition Center, Singapore, 14-17 July.

[180] **Karimi, M., & Moosavian, S. A. A.** (2010). Modified transpose effective jacobian law for control of underactuated manipulators. *Advanced Robotics, 24*(4), 605-626.

[181] **Hollerbach, J. M.** (1980). A recursive lagrangian formulation of maniputator dynamics and a comparative study of dynamics formulation complexity. *IEEE Transactions on Systems, Man, and Cybernetics, 10*(11), 730-736.

[182] **Luh, J. Y., Walker, M. W., & Paul, R. P.** (1980). On-line computational scheme for mechanical manipulators. *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*. *102*, 69-76.

[183] **Guo, M., Zhang, H., Feng, C., Liu, M., & Huo, J.** (2018). Manipulator residual estimation and its application in collision detection. *Industrial Robot: An International Journal,45*(3), 354-362.

[184] **Liu, N., Li, L., Hao, B., Yang, L., Hu, T., Xue, T., Wang, S., & Shao, X.** (2020). Semiparametric deep learning manipulator inverse dynamics modeling method for smart city and industrial applications. *Complexity,* 2020, 1-11.

[185] **Panda, A. S., Prakash, R., Behera, L., & Dutta, A.** (2020). Combined Online and Offline Inverse Dynamics Learning for a Robot Manipulator. In *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom, 19-24 July.

[186] **Universal Robots.** (2013). User Manual: UR5.

[187] **Wu, J., Wang, J., & You, Z.** (2010). An overview of dynamic parameter identification of robots. *Robotics and computer-integrated manufacturing, 26*(5), 414-419.

[188] **Argin, O. F., & Bayraktaroglu, Z. Y.** (2021). Consistent dynamic model identification of the Stäubli RX-160 industrial robot using convex optimization method. *Journal of Mechanical Science and Technology, 35*(5), 2185-2195.

[189] **Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G.** (2010). *Robotics: Modelling, Planning and Control*. London, UK: Springer Publishing Company, Incorporated.

[190] URL - 1 < https://www.universal-robots.com >, date retrieved 25.11.2021.

APPENDICES

APPENDIX A: The optimization procedure of the solenoid actuator.

APPENDIX A: The optimization procedure of the solenoid actuator

The solenoid actuator optimization procedure is described below:

First, the Finite Element Method (FEM) is employed to simulate a real-world existing actuator. The analysis of the electrical and magnetic fields is carried out using the solenoid actuator's 2D axisymmetric simulation model. The materials and their properties which are utilized for different parts of the actuator in the simulation studies are shown in Table A.1. The B-H characteristic of the 11SMN30 steel, which highly affects the magnetic field generated around the copper coil, is shown in Figure A.1.

Table A.1 : The properties of the material employed for the simulation studies.

Material	Electrical Conductivity (S/m)	Relative Permeability	Relative Permittivity
Copper	5.998×10^{-7}	1	1
Stainless Steel	1.45×10^{-6}	1	1
11SMN30	1428570	1	5000

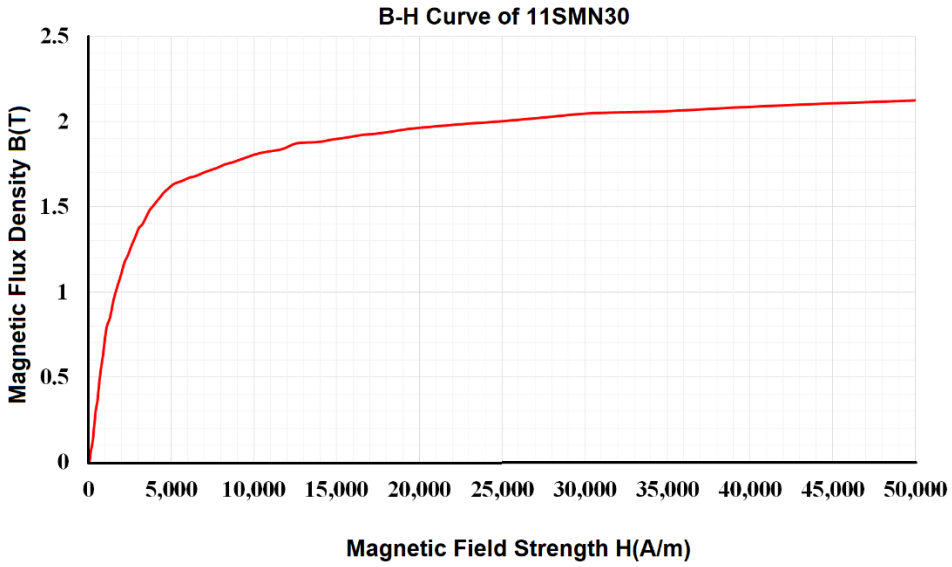


Figure A.1 : The magnetic flux density function of the magnetic field strength for the 11SMN30 steel.

The coil characteristics utilized for the simulations are as follows: the number of windings, the diameter, and the current values of the coil are set to 450 turns, 0.7 m, and 2.7 A, respectively.

Then, magnetic field analysis is performed to calculate the magnetic force produced in each position of the plunger inside the guider part of the actuator. The magnetic force is computed for the plunger position operation region between 0 and 5 mm. The magnetic force versus plunger position obtained using experiments and simulation models is demonstrated in Figure A.2.

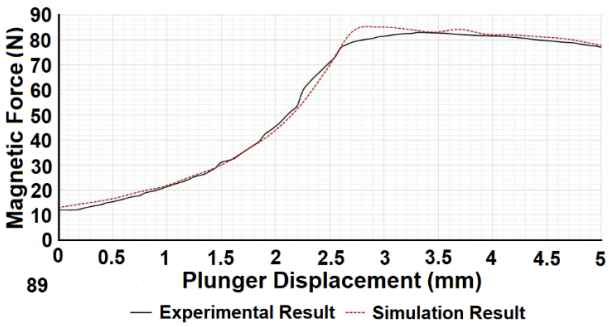


Figure A.2 : The magnetic force function of the plunger position for experimental and simulation studies.

Then, various geometrical changes were performed to facilitate the manufacturing process of the actuator. First, the conical angles on both sides of the guider anti-magnetic part were removed and set to zero degrees. Second, because the plunger's tubular shape allows for greater control of the magnetic force, the plunger's flat tubular shape is modified to a conical tubular shape. Third, the push-pin which does not affect the values of the magnetic force was omitted from the simulation process for simplicity. Once the new design of the actuator is developed, the FEM simulations are performed to obtain the most important geometrical parameters. Figure A.3 depicts the new actuator design as well as the most significant design parameters ($R, L1, L2, W$, and θ) which have a high impact on the magnetic force values.

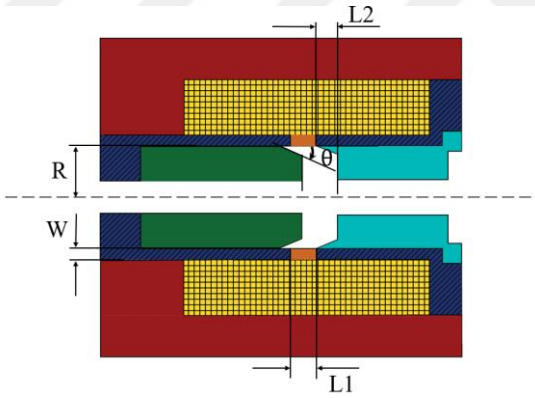


Figure A.3 : The new design of the solenoid actuator along with the most significant design parameters ($R, L1, L2, W$, and θ).

The geometrical parameters along with their bound and median values are represented in Table A.2.

Table A.2 : The geometrical parameters of the solenoid actuator.

Parameter (Unit)	Values
R (mm)	(8.35, 9.35, 9.85, 10.35, 11.35)
L1 (mm)	(2.7, 3.7, 4.2, 4.7, 5.7)
L2 (mm)	(4, 4.5, 5, 5.5, 6)
W (mm)	(1.5, 2, 2.5, 3, 3.5)
θ (°)	(0, 4, 8, 12, 16)

In the next step, simulations are performed using FEM for each value of the optimization parameters. Figure A.4 – Figure A.8 represent the magnetic force values obtained for different values of $R, L1, L2, W$, and θ using simulations, respectively.

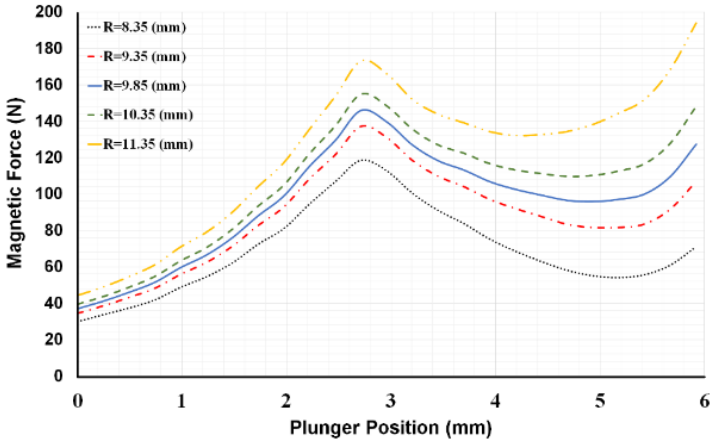


Figure A.4 : The magnetic force for different values of design parameter R.

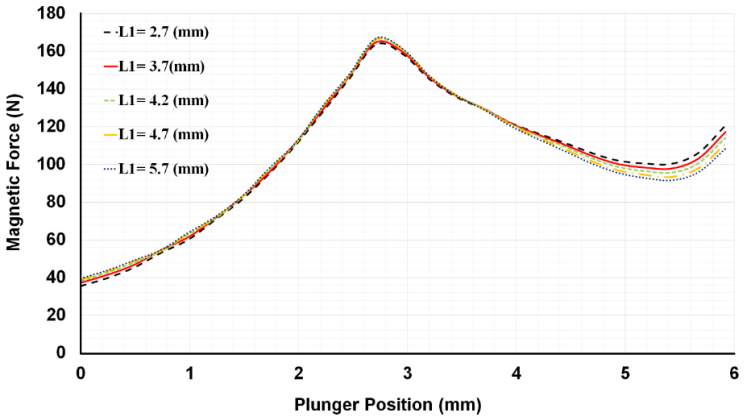


Figure A.5 : The magnetic force for different values of design parameter $L1$.

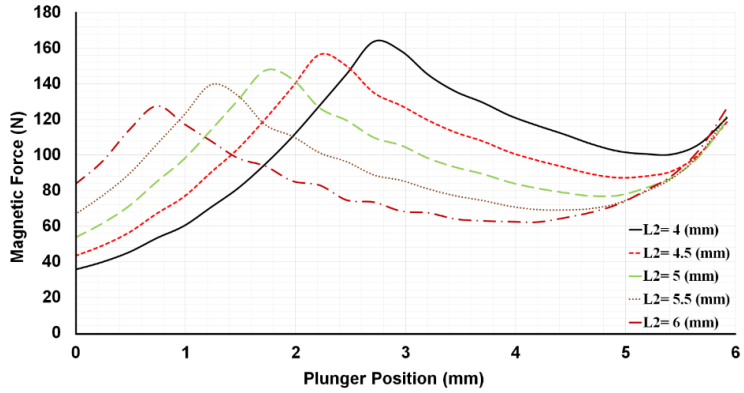


Figure A.6 : The magnetic force for different values of design parameter $L2$.

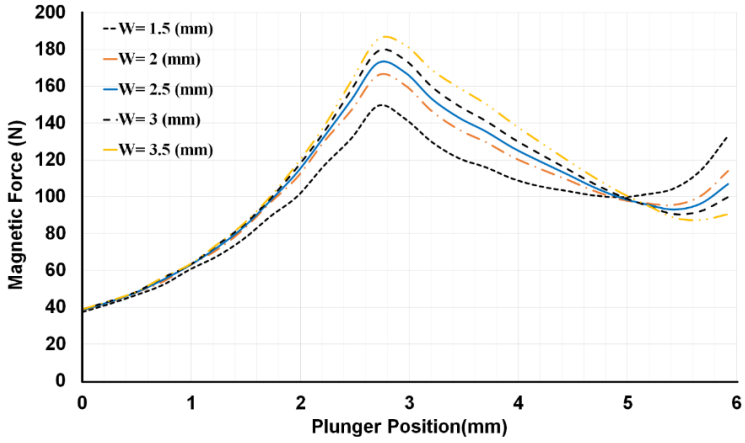


Figure A.7 : The magnetic force for different values of design parameter W.

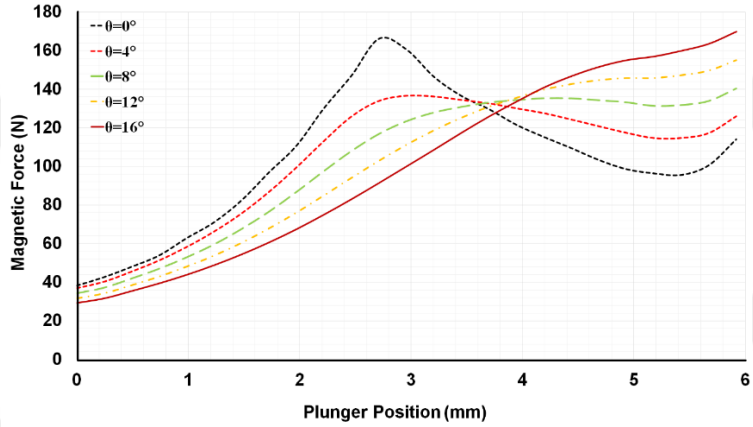


Figure A.8 : The magnetic force for different values of design parameter θ.

Then. to build a complete database, 5^5 FEM simulations were performed for different combinations of five values of the five optimization parameters. In the next step, the complete data regarding the median values corresponding to the bounds of the parameters are obtained using the interpolation approach through MATLAB.

Then, this optimization problem is considered to have two contradictory cost functions. The first cost function is to maximize the magnetic force, while the other cost function is to minimize the DF_{mag}/Dx_p . To unify these two cost functions into a single cost function, (DF_{mag}/Dx) is reversed to simplify the problem of the optimization of the solenoid actuator to a maximization problem with a single cost function. The cost function (CF) of the optimization problem that is formulated to be utilized in the PSO algorithm could be shown as equation (A.1).

$$CF = k_1 CF_1 + k_2 CF_2 \tag{A.1}$$

where the weights of each of the cost functions are represented with k_1 and k_2. The values of the weights might be readily changed between 0 and 1 based on the design goals. In this study, since there is a trade-off between the two cost functions, the values of k_1 and k_2 are assumed to be 1. CF_1 and CF_2 are cost functions of the problem. The

cost functions are normalized to equalize their contribution to the general cost function which is demonstrated by equation (A.2) and equation (A.3).

$$CF_1 = Maximum((F_{mag})_N) = Maximum\frac{F_{mag} - (F_{mag})_{min}}{(F_{mag})_{max} - (F_{mag})_{min}} \quad \text{(A.2)}$$

$$CF_2 = Maximum((1/DF_{mag}/Dx_p)_N)$$
$$= Maximum(\frac{(1/DF_{mag}/Dx_p) - (1/DF_{mag}/Dx_p)_{min}}{(1/DF_{ma}/Dx_p)_{max} - (1/DF_{mag}/Dx_p)_{min}}) \quad \text{(A.3)}$$

The optimal values of the design parameters are found and compared with the original values as shown in Table A.3.

Table A.3 : The original and optimal parameters found by the PSO algorithm.

Design Parameter (Unit)	The Original Value	The Optimal Value
R (mm)	9.35	10
L1 (mm)	4.7	3.7
L2 (mm)	4	4
W (mm)	2.5	2
θ (°)	0	4.5

Then, using the optimal values obtained using the PSO algorithm, the FEM simulation is repeated to calculate magnetic force versus plunger position. In addition, the solenoid actuator with optimal values is fabricated to compare the results. Figure A. 9 depicts a fabricated solenoid actuator attached to a hydraulic valve.

Figure A.9 : The fabricated solenoid actuator along with a hydraulic valve experimental setup.

The real data, including magnetic force versus plunger position, is acquired from the experimental setup. Finally, the simulation results of the original and optimized actuator, along with the experimental results of the optimized design, are represented in Figure A.10.

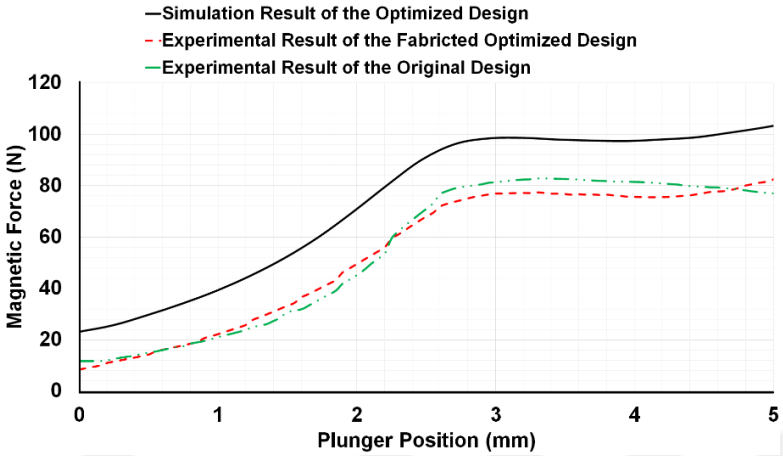


Figure A.10 : The magnetic force function of the position of the plunger for simulation results of the optimized design, experimental results of the fabricated optimized design, and the original design.

A comparison is made between the experimental result of the fabricated optimized design and the simulation result of the original design of the actuator. The maximum magnetic force ($(F_{mag})_{max}$) of the fabricated optimized solenoid actuator (experiment result) and the simulation model of the original solenoid actuator are obtained as 71.9 N and 78.9 N, respectively. Therefore, the improvement in maximum magnetic force is not achieved due to the results of the fabricated optimized solenoid actuator. On the other hand, the value of DF_{mag}/Dx_p is obtained as 0.66 N/mm and -3.86 N/mm for the fabricated solenoid actuator and the simulation model of the original design of the actuator, respectively. It shows that the fabricated optimized solenoid actuator represents about a 25 % improvement in the value of DF_{mag}/Dx_p. Consequently, evaluating the results of the fabricated optimized solenoid actuator demonstrates that only one of the objectives of the optimization is fulfilled.

The reasons for the inconsistency between the fabricated and simulation results of the optimized solenoid actuator could be investigated as follows: The generated magnetic force is greatly affected by the material's electromagnetic properties of the ferromagnetic elements of the actuator. Because the electromagnetic characteristics of the materials employed in the fabrication of the actuator were not accessible, its precise B-H characteristics could not be utilized for the simulation studies. Another possibility for the lower magnetic force in the fabricated solenoid actuator is the errors that occur in the fabrication process. Because the edge of the plunger is conic, it necessitates a delicate machining process to obtain exact angles.

The sum up, the framework employed for the optimization of the solenoid actuator is represented in Figure A.11.

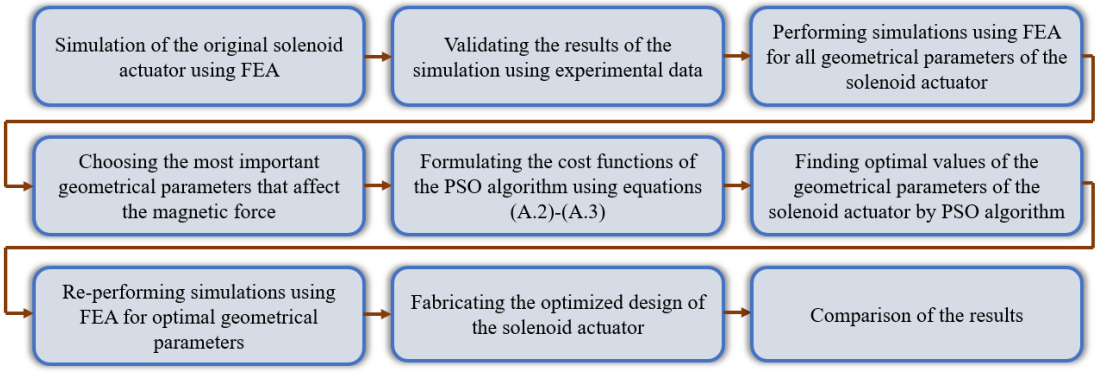


Figure A.11 : The representation of the framework used for the optimization of the solenoid actuator.

CURRICULUM VITAE

Name Surname **:** Masoud ABEDINIFAR

EDUCATION **:**

- **B.Sc.** **:** 2011, Bonab Azad University, Faculty of Engineering, Department of Mechanical Engineering
- **M.Sc.** **:** 2013, Tabriz Azad University, Faculty of Engineering, Department of Mechatronics Engineering

PROFESSIONAL EXPERIENCE:

- 2013-2014 Mechanical Engineer, Maral Trailer, Urmia/Iran.
- 2014-2016 Production Planning Manager, Fox Axle Manufacturing, Urmia/Iran.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Abedinifar, M.,** Ertuğrul, Ş., and Argüz, S. H. (2022)**.** Nonlinear Model Identification and Statistical Verification using Experimental Data with a Case Study of the UR5 Manipulator Joint Parameters, *Robotica*, 1-23.
- **Abedinifar, M.,** Ertuğrul, Ş., and Tayyar, G. T. (2022)**.** Design Optimization of a Solenoid Actuator using Particle Swarm Optimization Algorithm with Multiple Objectives, *Advances in Mechanical Engineering*, 14(11), 1-12.
- **Abedinifar, M.,** Ertuğrul, Ş. (2021)**.** Comparative Study of Identification Using Nonlinear Least Squares Errors and Particle Swarm Optimization Algorithms for a Nonlinear DC Motor Model, International Conference on Intelligent and Fuzzy Systems, August 24-26, 2021 Istanbul, Turkey.
- **Abedinifar, M.,** Argin, Ö. F., and Ertuğrul, Ş. (2020)**.** Identification of a Nonlinear DC Motor Model with Particle Swarm Optimization, 3rd International Conference on Artificial Intelligence towards Industry 4.0, November 12-14, 2020 Izmir, Turkey.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Abedinifar, M.,** Ertuğrul, Ş., and Argüz, S. H. (2023)**.** Nonlinear Model Identification of a Ball and Beam Mechanism using Experimental Data, Control, Automation, Robotics, Optimization, Decision, Cybernetics, Computer Science and Information Technologies (CODIT'23), July 03-06, 2023 Rome, Italy, (Under review).

- Kaya, O., **Abedinifar, M.,** Feldhaus, D., Diaz, F., Ertuğrul, Ş., and Friedrich., B. (2022)**.** System Identification and Artificial Intelligent (AI) Modeling of the Molten Salt Electrolysis for Prediction, *Computational Materials Science*, (Under revision).
- Mihçin, Ş., and **Abedinifar, M.** 2022**.** Lower Body Kinematics and Kinetics For Estimation of Boundary Conditions Over West Asian Population, *Journal of Prosthetics and Orthotics*, (Under review).
- **Abedinifar, M.,** Ertuğrul, Ş., and Sütcü, B. 2020**.** Modeling and System Identification of Solenoid Actuators: A Survey, 9th International Conference on Advanced Technologies, August 10-12, 2020 Istanbul, Turkey.
- **Abedinifar, M.,** 2016**.** Hybrid Intelligent Controlling Action in Cold Fluid Outflow Temperature of Shell and Tube Heat Exchangers, *International Journal of Advanced Research*, 4(11), 784-790.