Single Link Manipulator Trajectory Tracking using Nonlinear Control Algorithm

Musadaq A. Hadi[1*], Hazem I. Ali[2]

Author's affiliations:

1*) Department of Control and Systems Engineering, University of Technology, Baghdad-Iraq.
61361@student.uotechnology.edu.iq

2) Department of Control and Systems Engineering, University of Technology, Baghdad-Iraq.
60143@uotechnology.edu.iq

Abstract

A new robust control algorithm is proposed for a class of nonlinear systems represented by a Single Link Manipulator (SLM) system. This algorithm is based on new techniques and methods in order to obtain a controller for the SLM system. First of all, the system is simplified using Variable Transformation Technique (VTT) in order to fit the analysis procedure. Then, a new idea of designing a model reference for the multiple states (n=4) system is presented to correspond the control design. Next, the Lyapunov Stability Analysis (LSA) is used to figure out a proper controller that can compensate the stability and the performance of the SLM system. After that, the Most Valuable Player Algorithm (MVPA) is applied to find the optimal parameters of the proposed controller to accomplish the optimum performance improvement. Finally, it can be concluded that the proposed control algorithm has improved the stability and the performance of the SLM system. In addition, the simulation results show the remarkable effects of the proposed nonlinear controller on the SLM system.

Keywords: Single Link Manipulator, Nonlinear Control, Variable Transformation Technique, Lyapunov Stability Analysis, Most Valuable Player Algorithm.

<div dir="rtl">

المناور (الذراع) المنفرد متبع المسار باستخدام مسيطر غير خطي

مصدق احمد هادي , حازم ابراهيم علي

الخلاصة:

في هذا البحث تم اقتراح خوارزمية متينة لمنظومة غير خطية ممثلة بمنظومة المناور المنفرد. هذه الخوارزمية مبنية على تقنيات وطرق جديدة لغرض ايجاد مسيطر يمكن ان يسطر على منظومة المناور المنفرد. اولا يتم تبسيط المنظومة باستخدام تقنية المتغير المحول لجعل المنظومة متجانسة مع الخطوات التحليل. بعدها يتم تصميم نموذج مصدر ذو حالات متعددة لكي يتوافق مع منظومة المناور المنفرد. يتم تحليل المنظومة باستخدام تحليل ليابونوف للاستقرارية لمعرفة المسيطر المناسب الذي يعالج المنظومة. بعد ذلك يستخدم خوارزمية اللاعب الاغلى المثلى والتي هي خوارزمية رصينة مبنية على اساس البطولات الرياضية (كأس العالم) لجعل معاملات هذا المسيطر تمتلك قيم مثلى. واخيرا يتم استعراض نتائج المحاكاة التي توضح وتثبت كفاءة خوارزمية المسيطر الغير الخطي المقترح على منظومة المناور المنفرد.

</div>

1. Introduction

In previous works, nonlinear systems were the most important topic in control theory. Typically, nonlinear phenomenon represents the behavior of a real system. It is hard to maintain a good stability margins and desired performance properties for the closed loop systems due to the high uncertainty and nonlinearity of these systems [1]. Furthermore, the robot manipulators technology was an active topic in the literature of the control theory. Many manual processes, operations, works are replaced with manipulators and robots in order to achieve flexibility and accuracy in the industry, medical and military. In addition, all regular manipulators are substituted by flexible manipulators gradually. Many active researches were done by proposing different controllers for the manipulator control systems. μ-synthesis robust controller was proposed for a class of linearized single link manipulator system to handle the uncertainty of the system [2]. A fuzzy adaptive neural control was proposed for a flexible link manipulator and achieved the tracking in spite of the

existence of dynamic payload [3]. Adaptive robust control was designed for a nonlinear joint manipulator with mismatched uncertainty [4]. An adaptive fuzzy control with indirect method was proposed for a class of driven flexible link robot manipulators using a new estimation technique to estimate the uncertainties [5]. An adaptive sliding mode control was presented to accomplish tracking property of the position of the robot manipulator [6]. An adaptive control with backstepping technique was proposed for a flexible link robotic manipulator with the aid of Lyapunov analysis [7]. A robust control based on Lyapunov was proposed for a rigid flexible joint manipulator in order to compensate the uncertainties included in the system [8]. An adaptive control using stochastic switching was proposed for one link manipulator in order to achieve the stability and improve the performance [9]. An optimal adaptive fuzzy control was proposed to deal with uncertain class of flexible manipulator [10]. Dual terminal sliding mode controller was presented to achieve the tracking property of the single link manipulator system [11]. A finite time sliding mode controller based on disturbance observer was introduced to compensate the stability and the holomonic constraint of a flexible link joint manipulator [12]. A fixed time super twisting sliding mode control using leader follower was proposed for multiple flexible robot manipulators in order to achieve position tracking for the system [13]. In addition, an optimal nonlinear model reference controller was proposed for ball and plate system [14]. Furthermore, the optimal control law was proposed for a class of nonlinear system representing by marine surface vessels to compensate this system [15].

In this paper, a new nonlinear control design is proposed for a class of nonlinear system represented by a Single Link Manipulator (SLM) system. The contribution of this work is to construct a proper controller using the proposed nonlinear control that does the followings: *i)* improves the stability the SLM system *ii)* deals with nonlinearity of the SLM system. *iii)* compensates the SLM system to obtain a desirable performance. *iv)* optimizes the parameters of the proposed control algorithm using the Most Valuable Player Algorithm (MVPA). The following is the procedure of the proposed control algorithm:

Step 1: Develop the mathematical model of the nonlinear system (SLM).

Step 2: Simplify the model using Variable Transformation Technique (VTT) to lump the bad term that need to be eliminate by the controller.

Step 3: Design and optimize a stable model reference to fit the analysis procedure.

Step 4: Analysis the nonlinear system using Lyapunov Stability Analysis (LSA) in order to obtain a proper controller that can stabilized the system.

Step 5: Select a proper cost function (ISE) to be used in the optimization method to minimize the error.

Step 6: Optimize the parameters of the obtained controller using the MVPA with the aid of the selected cost function in Step 5.

2. The Most Valuable Player Algorithm

Most Valuable Player Algorithm (MVPA) is new optimization method based on the metaphor of sport where a population of players competes collectively in teams in order to win the leagues' championship, and they compete individually in order to win the MVP trophy. Like other metaheuristic methods. The population is represented as a group of players that have skills, which is analogous to the design variables where the number of players' skills corresponds to the dimension of the problem. Before explaining the MVPA, some sport related terms should be defined [16]:

- *Player*: a person who participate in a game or sport.
- *Team:* a group of players who play a certain game/sport versus another group of people.
- *Franchise player*: the best/the most valuable player on a professional sports team.
- *Most valuable player*: the award given to the best player in a game or series of games during a particular season.
- *League*: a group of sports teams who all play each other to gain points and discover which team is best.
- *Championship*: a competition to figure out the best player or team in a specific sport.
- *Fixture*: a sports event that has been prepared to take place on a specific date and at a specific place. In this algorithm, a player and a team which is a group of players both are represented as follows [16]:

$$Player_k = [S_{k,1} \quad S_{k,2} \quad . \quad . \quad S_{k,ProblemSize}] \qquad ...(1)$$

$$Team_i = \begin{bmatrix} S_{1.1} & S_{1.2} & . & S_{1,ProblemSize} \\ S_{2.1} & S_{2.2} & . & S_{1,ProblemSize} \\ \vdots & \vdots & . & \vdots \\ S_{PlayerSize,1} & S_{PlayerSize,2} & . & S_{PlayerSize,ProblemSize} \end{bmatrix} .(2)$$

where *PlayersSize* is the number of players in the league, *ProblemSize* is the dimension of the problem and S stands for skill. Each team has a franchise player and the league's MVP is the best player of the league. An example of two players with their corresponding level of skills for each one is shown in Fig. 1.

Figure (1): Example of two players with their skills.

The following explains in detail the main phases of the MVPA [16]:

a) *Initialization*; a population of the player size; players are randomly generated in the search space.

b) *Team formation*; the teams are named as nT_1' and 'nT_2' are first team and second team respectively. Also, the players are named such as nP_1' and 'nP_2' are the players of the first and second team respectively. These variables are calculated as follow [16]:

$$nP_1 = ceil\left(\frac{PlayersSize}{TeamsSize}\right) \dots (3)$$

$$nP_2 = nP_1 + 1 \qquad \dots (4)$$

$$nT_1 = PlayersSize \\ - nP_2 \\ \times TeamsSize \dots (5)$$

$$nT_2 = TeamsSize - nT_1 \qquad \dots (6)$$

c) *Team competition*; players are debating each other individually in order to find which one is the best player who has the best skills. This competition is calculated using the following expressions [16]:

$$TEAM_i \\ = TEAM_i \\ + rand(FranchisePlayer_j - TEAM_i) \\ + 2rand(MVP - TEAM_i) \qquad \dots (7)$$

If $TEAM_i$ is chose to play against $TEAM_j$ and $TEAM_i$ wins the player's performance of $TEAM_i$ are expressed as follow [16]:

$$TEAM_i + rand(TEAM_i \\ - FranchisePlayer_j) . (8)$$

Otherwise, they are expressed as follow [16]:

$$TEAM_i + rand(FranchisePlayer_j \\ - TEAM_i) \qquad \dots (9)$$

d) *Application of greediness;* a new solution is selected after the comparison of the population is done. Each selection is made based on a better objective function value.

e) *Application of elitism*; the best (elite) players are selected and the other players are replaced with the best ones.

f) *Remove duplicates*; if the best players have been selecting twice. Then, one of them is dropped.

g) *Termination criterion;* in the MVPA, this criterion is option implemented by the user himself or the number of the iterations will be the termination criterion.

Aforementioned phases illustrate the MVPA calculations to find the optimal parameters. The following explains how this method is working on the system in this paper. First, assign the number of teams, players and the problem dimensions which are the parameters of the controller that need to be optimized. These parameters represent the skills of a player in MVPA. Next, assign the objective function

which represents the cost function used in this paper (Integral Square Error). Then, the players gained skills (parameters) throughout the phases that mentioned previously which are embodied in Matlab code (m-files). After that, the pre-optimized parameters are calculated and applied simultaneously into the controlled system to calculate the measured error. Consequently, the measured error used to find the cost function and compared to the previous cost at each iteration in order to obtain best cost and then the optimal parameters. Finally, this process is repeated until the optimal parameters are obtained after certain number of iterations. It can be seen in Fig 2, the steps of the MVPA in detail with respect to the system that can be used later [16].
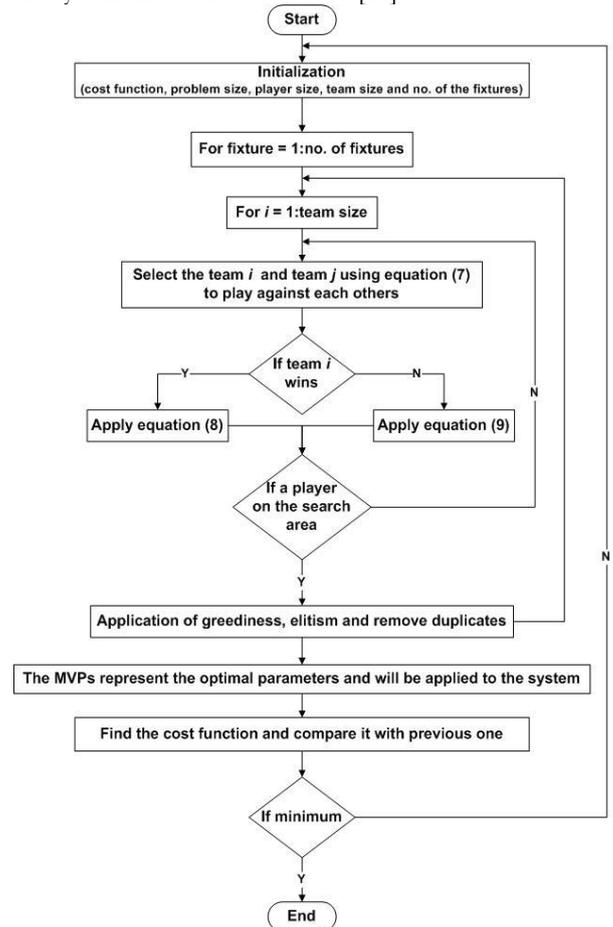
Figure (2): The flowchart of MVPA with the system.

3. Nonlinear Control Algorithm Design

In this section, the problem formulation of the Single Link Manipulator (SLM), the Variable Transformation Technique (VTT), the model reference design and the Lyapunov Stability Analysis (LSA) are presented in order to obtain a proper controller that can deal with nonlinearity and improve the performance of the system.

3.1 Mathematical Model Description

The following is of Single Link Manipulator (SLM) system [13]:

$$\dot{x} = f(x) + g(x)u$$
$$I\ddot{q}_1 + Mgl\sin(q_1) + k(q_1 - q_2) = 0$$

$$J\ddot{q}_2 + k(q_2 - q_1) = \tau$$

Or $\quad f(x) = \begin{pmatrix} x_2 \\ -asin(x_1) - b(x_1 - x_3) \\ x_4 \\ c(x_1 - x_3) \end{pmatrix},$

$$g(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ d \end{pmatrix} \quad ....(10)$$

with $a = \frac{MgL}{I}, b = \frac{K}{I}, c = \frac{K}{J}$ and $d = \frac{1}{J}$.

where x_1, x_2, x_3 and x_4 are the states of the SLM system, τ is the input torque, K is the spring constant, M shows the total mass of payloads, L represents the flexible link length, I and J are the depict moments of equivalent inertia. Table (1) shows the values of the system parameters as follow:

Table (1): Parameters of the SLM system [13].

Parameters	a	b	c	d
Values	20	1	0.5	0.4

Figures 3 and 4 represent the open loop and the closed loop responses of the state trajectories (rotate angle, deflection angle, rotate velocity and deflection velocity); before applying the controller. Moreover, it shows the bad performance of the nonlinear Single Link Manipulator (MLS) system with the applied input $r = sin(0.5t) - cos(t)$.
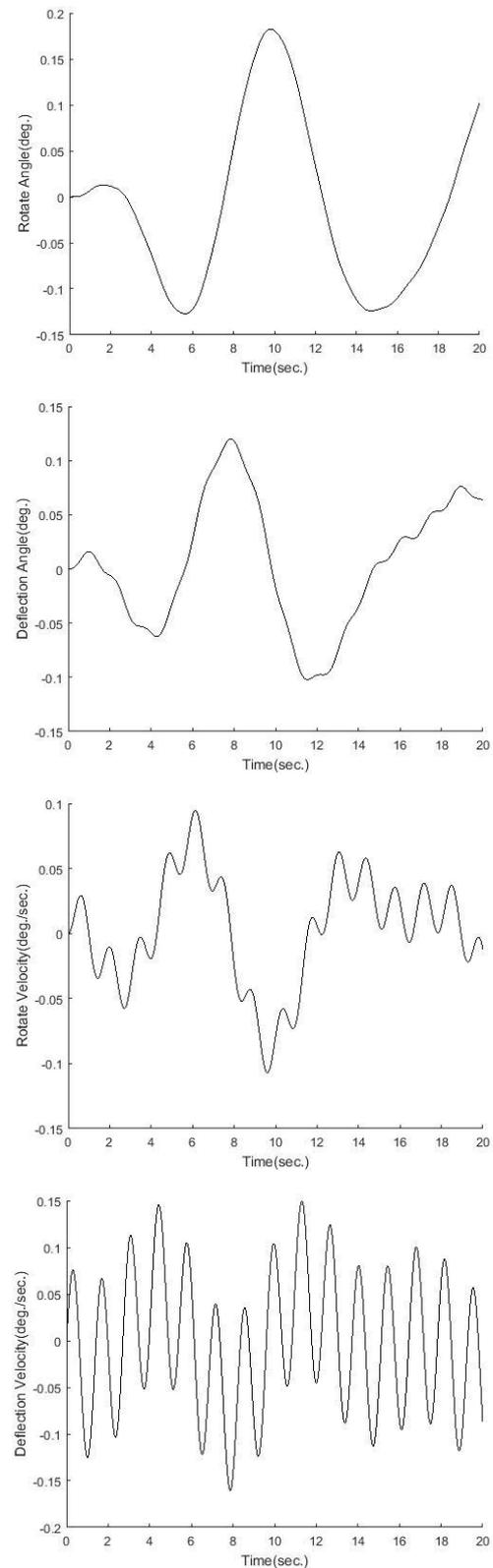
Figure (3): Open loop system responses.
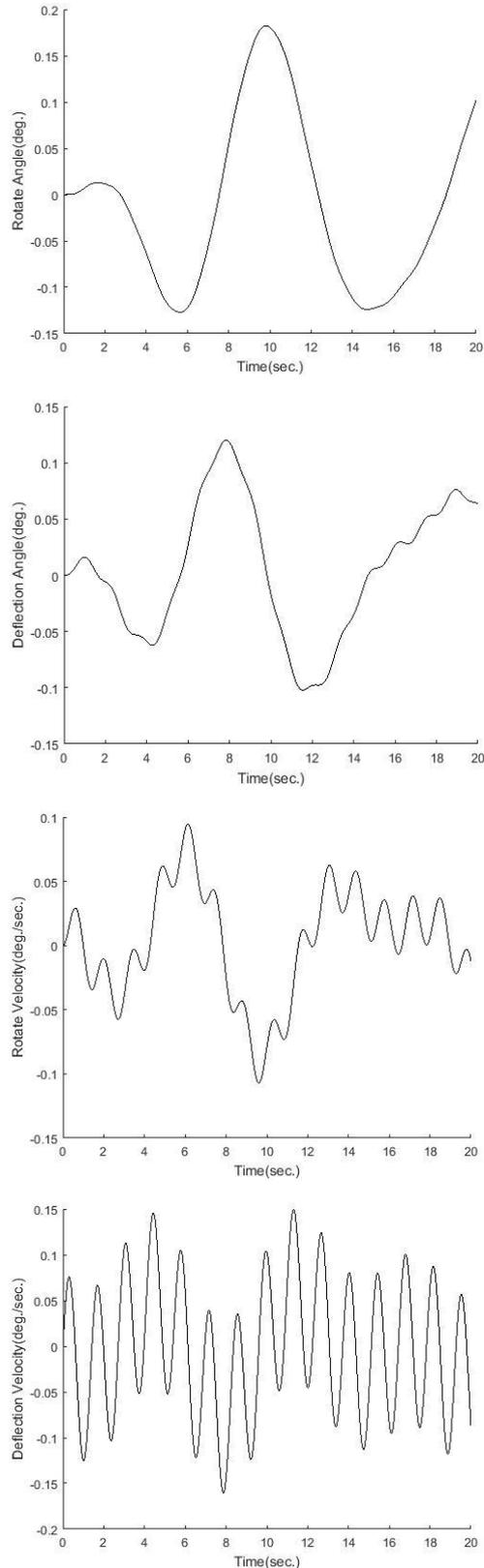
Figure (4): Closed loop system responses before applying the controller.

Figure 5 shows the tracking properties of the system. It is clearly shown the gap between the reference and the output signals.
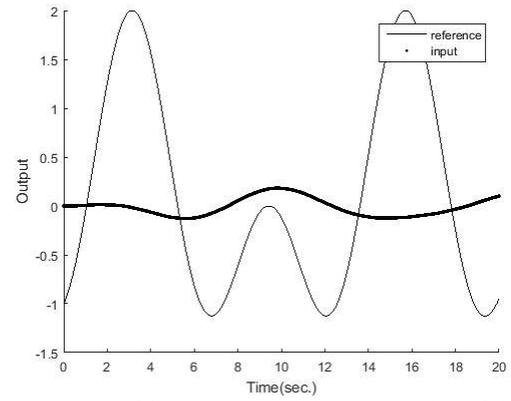
Figure (5): The output tracking properties of the system.

3.2 Variable Transformation Technique

In this subsection, the new technique is introduced to overcome the complexity of the structures in the SLM system. This technique is used to simplifying the complicated structure of the systems. VTT is a technique that transforms a variable by a function of that variable, or it is a replacement of the shape of the nonlinear system equations (variable distributions and relationships) [17], [18]. The technique is similar to backstepping technique in the back-substituting way, but it has been worked for non-adaptive cases so far. It can be derived by using the recursive derivative of the first state in order to find other states to be transformed based on pre-assumptions as follow [19]:

Let $\quad z_1 = x_1, \dot{z}_1 = \dot{x}_1 = x_2 = z_2$

and $\quad \dot{z}_2 = \ddot{x}_1 = \dot{x}_2 = -a sin(x_1) - b(x_1 - x_3) = -a sin(z_1) - b(z_1 - x_3) = z_3$

$$z_3 + a sin(z_1) = -b(z_1 - x_3)$$
$$z_3 + a sin(z_1) + b z_1 = b x_3$$

Thereby $\quad x_3 = \frac{z_3 + a sin(z_1) + b z_1}{b} \quad …(11)$

$\dot{z}_3 = \dddot{x}_1 = \ddot{x}_2 = -a\dot{x}_1 cos(x_1) - b(\dot{x}_1 - \dot{x}_3)$
$$= z_4$$
$$z_4 = -a z_2 cos(z_1) - b(z_2 - x_4)$$

Then $\quad x_4 = \frac{z_4 + a z_2 cos(z_1) + b z_2}{b} \quad …(12)$

$\dot{z}_4 = \frac{\partial \dddot{x}_1}{\partial t} = \ddddot{x}_2 = -a[\dot{x}_1 \dot{x}_1(-sin(x_1) + \ddot{x}_1 cos(x_1)] - b(\dddot{x}_1 - \dddot{x}_3)$
$= a z_2^2 sin(z_1) - a z_3 cos(z_1) - b z_3 + b \dot{x}_4$
$= a z_2^2 sin(z_1) - a z_3 cos(z_1) - b z_3 + bc(x_1 - x_3) + bdu \quad ..(13)$

Substituting equation (11) in equation (13) gives:
$\dot{z}_4 = a z_2^2 sin(z_1) - a z_3 cos(z_1) - b z_3 + bc\left(z_1 - \frac{z_3 - a sin(z_1) - b z_1}{b}\right) + bdu$
$= a z_2^2 sin(z_1) - a z_3 cos(z_1) - b z_3 + \cancel{bcz_1} - cz_3 - acsin(z_1) - \cancel{bcz_1} + bdu$
$= -(b + c)z_3 + bdu + a z_2^2 sin(z_1) - acsin(z_1) - a z_3 cos(z_1) \quad ..(14)$

The resulting new system is:
$$\dot{z} = f(z, u, t) =$$

34

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \\ \dot{z_3} \\ \dot{z_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -(b+c) & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ bd \end{bmatrix} u$$

$$+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \delta \quad ...(15)$$

where $\delta = az_2^2 sin(z_1) - acsin(z_1) - az_3cos(z_1)$.

3.3 Model Reference Design

In this subsection, the way of the selection/design of the model reference is explained. In many works a second order model reference was used for single-input single-output systems. A few of them used Van Der Pol oscillator model for multi-input multi-output (MIMO) systems [20]. In this work, the model reference is selected/designed based on the system properties. It is designed and optimized using MVPA in order to figure out the optimal model reference. Since, the number of states in the actual systems is $n = 4$. Then, a fourth order model reference is designed to fit the design procedure as follow:

$$\begin{bmatrix} \dot{z}_{1d} \\ \dot{z}_{2d} \\ \dot{z}_{3d} \\ \dot{z}_{4d} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 & -a_4 \end{bmatrix} \begin{bmatrix} z_{1d} \\ z_{2d} \\ z_{3d} \\ z_{4d} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_1 \end{bmatrix} r$$

$$\dot{z}_d = Az_d + Bv$$
$$y = z_1$$

T.F $\quad \dfrac{Y(s)}{R(s)} = \dfrac{S+b}{S^4+a_4S^3+a_3S^2+a_2S+a_1} \quad ...(16)$

Consequently, the optimal parameters of the selected model reference which are optimized using the MVPA are following: $a_1 = 1.00005e + 03$, $a_2 = 1.21217e + 02$, $a_3 = 88.63046$, $a_4 = 2.10716$ and $b_1 = 9.226367e + 02$. In addition, the model reference is stable with negative eigenvalues after using Matlab syntax eig(A) such as $\lambda_1 = -0.2903 + 8.6138i$, $\lambda_2 = -0.2903 - 8.6138i$,

$\lambda_3 = -0.7633 + 3.5889i$ and $\lambda_4 = -0.7633 - 3.5889i$.

3.4 Lyapunov Stability Analysis

In this subsection, the Lyapunov Stability Analysis (LSA) is presented in order to obtain a proper controller that can confirm the stability and improve the performance of the SLM system. The error between the system and the reference model can be defined as follow:

$$e = z_d - z \quad ...(17)$$
$$\dot{e} = \dot{z}_d - \dot{z} \quad ...(18)$$

Then substituting Equations (16), (17) and (18) in Equation (19) gives:

$$\dot{e} = Az_d + Bv - f(z, u, t)$$
$$\dot{e} = Az_d + Bv - f(z, u, t) + Az - Az$$
$$\dot{e} = A(z_d - z) + Az + Bv - f(z, u, t)$$
$$\dot{e} = Ae + Az + Bv - f(z, u, t) \quad ...(19)$$

Lyapunov quadratic function (LQF) is used as follow:
$$V(e) = e^T P e \quad ...(20)$$

The time-derivative of Lyapunov quadratic function (LQF) $\dot{V}(e)$ yields:

$$\dot{V}(e) = \dot{e}^T P e + e^T P \dot{e} \quad ...(21)$$

Substituting Equation (19) in Equation (21) yields:

$$\dot{V}(e) = [Ae + Az + Bv - f(z, u, t)]^T P e + e^T P [Ae + Az + Bv - f(z, u, t)]$$

$$\dot{V}(e) = e^T(A^T P + PA)e + 2e^T P[Az - f(z, u, t) + Bv]$$

$$\dot{V}(e) = -e^T Qe + 2M \quad ...(22)$$

with $\quad M = e^T P[Az + Bv - f(z, u, t)] \quad ...(23)$
where Q and P are positive, symmetric matrix.

By substituting equations (15) and (16) in equation (23) as follow:

$$M = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{12} & p_{22} & p_{23} & p_{24} \\ p_{13} & p_{23} & p_{33} & p_{34} \\ p_{14} & p_{24} & p_{34} & p_{44} \end{bmatrix}$$

$$\times \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 & -a_4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_1 \end{bmatrix} r - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -(b+c) & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ bd \end{bmatrix} u - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \delta \right\}$$

$$M = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{12} & p_{22} & p_{23} & p_{24} \\ p_{13} & p_{23} & p_{33} & p_{34} \\ p_{14} & p_{24} & p_{34} & p_{44} \end{bmatrix}$$

$$\times \left\{ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -a_1 & -a_2 & -(a_3+b+c) & -a_4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_1 r - bdu - \delta \end{bmatrix} \right\}$$

$$M = [e_1 \quad e_2 \quad e_3 \quad e_4] \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{12} & p_{22} & p_{23} & p_{24} \\ p_{13} & p_{23} & p_{33} & p_{34} \\ p_{14} & p_{24} & p_{34} & p_{44} \end{bmatrix}$$

$$\times \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ -a_1 z_1 - a_2 z_2 - (a_3 + b + c)z_3 - a_4 z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b_1 r - bdu - \delta \end{bmatrix} \right\}$$

$$M = [e_1 p_{1j} \quad e_2 p_{2j} \quad e_3 p_{3j} \quad e_4 p_{4j}] \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ -a_1 z_1 - a_2 z_2 - (a_3 + b + c)z_3 - a_4 z_4 + b_1 r - bdu - \delta \end{bmatrix}$$

$$M = \begin{pmatrix} e_1 p_{14} + e_2 p_{24} \\ + e_3 p_{34} + e_4 p_{44} \end{pmatrix} \times [-a_1 z_1 - a_2 z_2 - \begin{pmatrix} a_3 + \\ b + c \end{pmatrix} z_3 - a_4 z_4 + b_1 r - bdu - \delta]$$

Let $u = \frac{1}{bd}(-a_1 z_1 - a_2 z_2 - (a_3 + b + c)z_3 - a_4 z_4 + b_1 r - \delta - c_1 z_2^2 + c_2 z_2^2 sign(e_1 p_{14} + e_2 p_{24} + e_3 p_{34} + e_4 p_{44}))$ ...(24)

Then $M = (e_1 p_{14} + e_2 p_{24} + e_3 p_{34} + e_4 p_{44})[(c_1 - c_2 sign(e_1 p_{14} + e_2 p_{24} + e_3 p_{34} + e_4 p_{44}))z_2^2]$ ...(25)

Substituting equation (25) in equation (22) gives:
$\dot{V}(e) = -eQe^T + 2(e_1 p_{14} + e_2 p_{24} + e_3 p_{34} + e_4 p_{44})[(c_1 - c_2 sign(e_1 p_{14} + e_2 p_{24} + e_3 p_{34} + e_4 p_{44}))z_2^2]$ ...(26)

The abovementioned subsections have explained the proposed control design in order to obtain the controller (as in equation (24)) that can compensate the SLM system. In the next section, the simulation results prove the efficiency of the proposed nonlinear control. Moreover, to find the optimal parameters of the nonlinear control algorithm, the integral square error performance index (ISE) is used. It is expressed by [21]:

$$J = \int_0^t e^2(t)\, dt \qquad ... (27)$$

where $e(t)$ is the difference value between the model reference output and the system output.

4. Simulation Results

In this section, the simulation results of the SLM system is presented after applying the proposed control strategy. The MVPA is implemented to obtain the optimal parameters of the proposed controller. Table (2) contains the MVPA settings and the optimal parameters of the proposed controller.

Table (2): MVPA settings and optimal parameters

MVPA Settings	Values	Optimal parameters	Values
LowerBound	0.00051	P_{14}	50
UpperBound	50	P_{24}	50
ProblemSize	6	P_{34}	4.51681
PlayerSize	10	P_{44}	0.5
TeamSize	5	c_1	0.55774
Iterations	40	c_2	50

Figure 6 shows the stability properties of the system state trajectories (rotate angle, deflection angle, rotate velocity and deflection velocity). In addition, it shows that the proposed controller has stabilized the Single-Link Manipulator (SLM) system effectively.
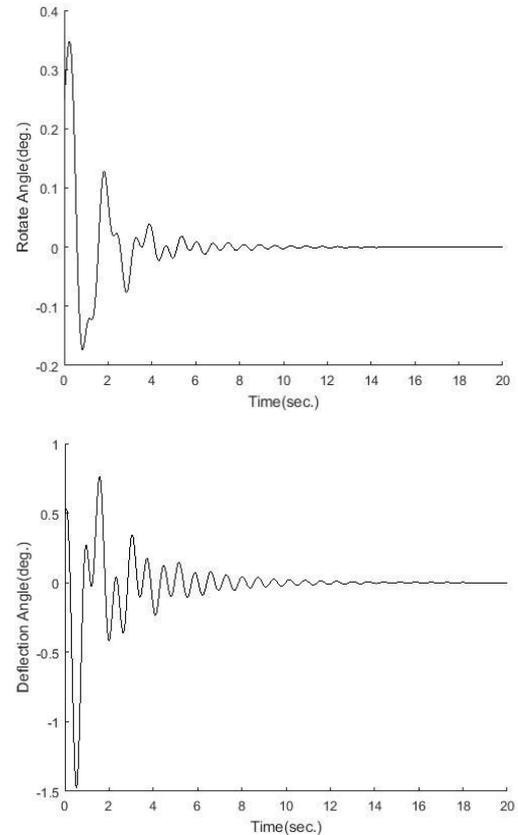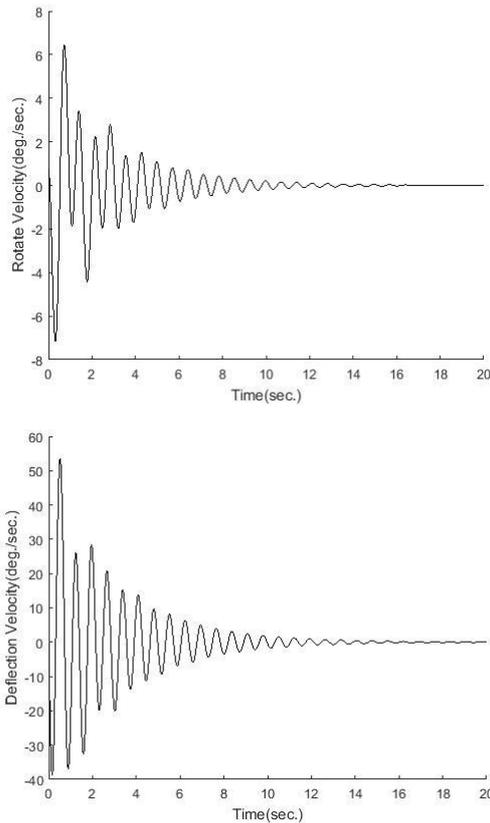
Figure (6): Stability properties of the system state trajectories.

Figure 7 shows that the proposed controller can effectively compensate the system states (rotate angle, deflection angle, rotate velocity and deflection velocity) under the nonlinearity effects.
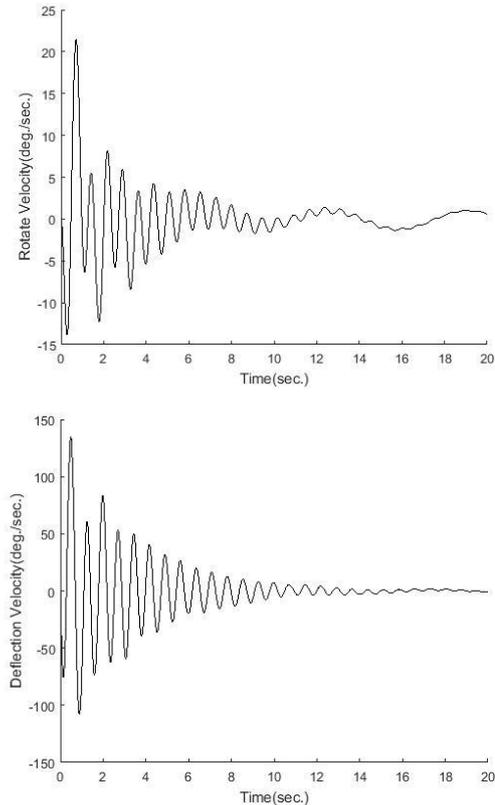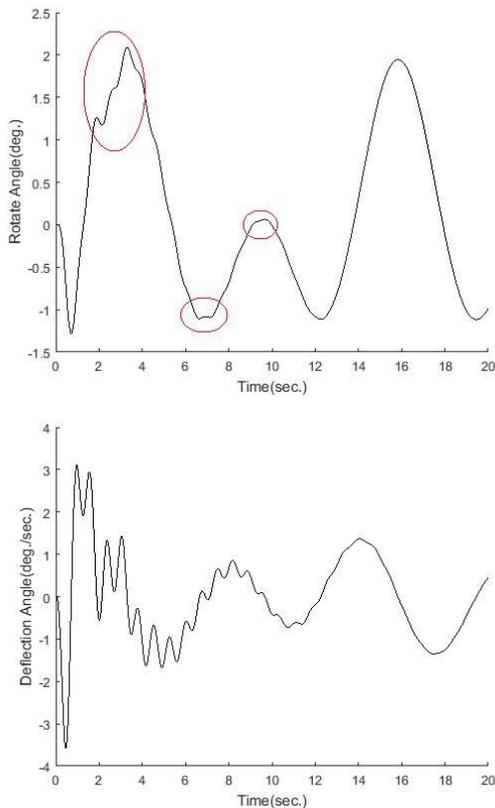
Figure (7): State trajectories of the SLM system.

Figure 8 shows that the tracking properties have been achieved between the system output and the reference input after applying the proposed controller. Since, there is no model reference (with $n = 4$) for the SLM system that fits Lyapunov stability analysis to produce a controller, a new model reference is designed and optimized (using MVPA) to be used in this case thereby the red spots in the tracking and rotate angle trajectory in Fig 7 are caused by the coupling in the model reference and nothing to do with SLM system.
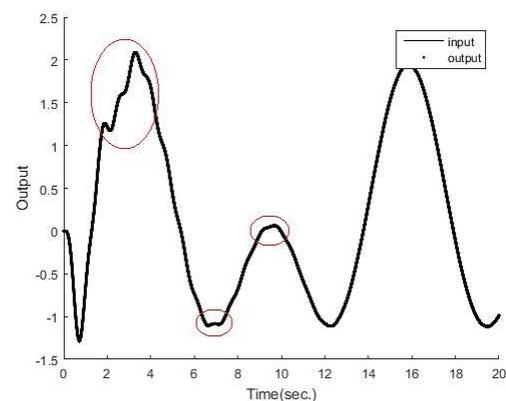
Figure (8): Rotate angle tracking of the system.

Figure 9 shows the LQF with indefinite time-derivative function (equation 26) which indicates that the SLM controlled system becomes asymptotically

stable.
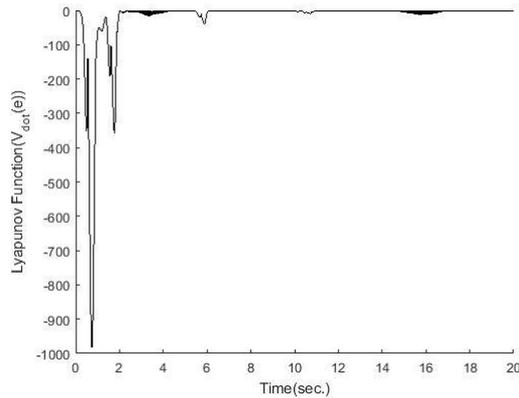
Figure (9): Time derivative of the LQF.

Figure 10 explains the resulting control signal with optimal parameters which is applied to the system. However, the shadow parts are due to implying the signum function in the controller (equation 24). In addition, the high torque values are due to the highly nonlinear effects in the SLM system.
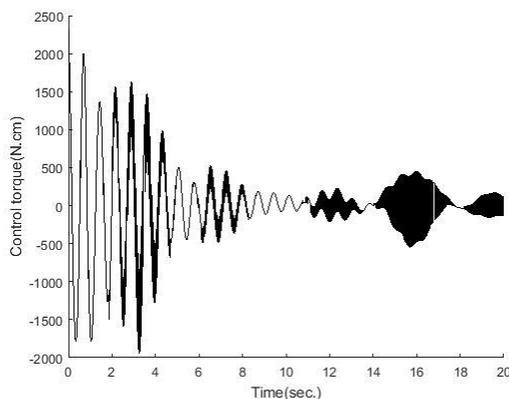
Figure (10): The resulting control torque.

Figure 11 shows that the phase-plane trajectory reaches to zero after applying the controller which improves the stability of the SLM system.
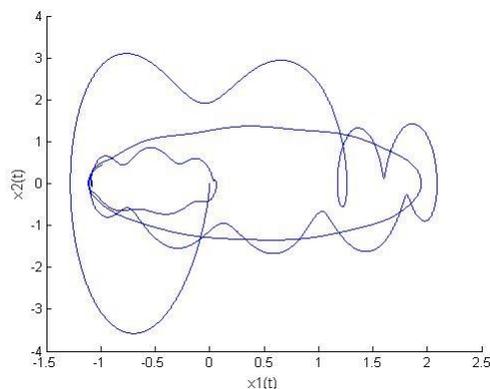
Figure (11): Phase-plane trajectory.

Furthermore, many simulation tests have been done therefore; it was noticed that the settings of MVPA were adequate for this system by minimize the error to the value of 0.5. In addition, the number of iterations is equal to 12 indicates that MVPA

achieves the best minimization of the cost function. Eventually, these results are compared to a fixed time super-twisting sliding mode controller that applied to the same system [13]. In reference [13], the control torque was swing between the values $33000\ N.m$ and $-33000 N.m$ for the first $4\ sec.$; however, the control torque of the proposed controller is swing between the values $2000\ N.m$ and $-1750\ N.m$ for the first $4\ sec$ as shown in Figure 10. In reference [13], the leader-follower was used in the Lyapunov stability analysis to find out the stability of the system and determines the performance. Consequently, a fourth order model reference is designed and optimized to be used in the Lyapunov stability analysis to find out the stability of the system and determines the performance.

5. Conclusion

In this paper, a new nonlinear control algorithm is proposed for a single link manipulator system the proposed controller achieved the stability and a desirable performance. Variable Transformation Technique (VTT) has been used to simplify the nonlinearity in SLM system in order to compensate all those bad terms by the proposed controller. A new model reference has been designed and optimized to fit the analysis procedure for Lyapunov stability analysis. Moreover, a proper controller has been successively obtained using Lyapunov Stability Analysis (LSA) which make the system stable. Finally the simulation results have proved that the stability and the performance of the SLM system. In addition, the simulation results have shown the effectiveness of the proposed nonlinear control algorithm by compensating the bad terms in the SLM system.

6. References

[1] H. Ali, *Swarm Intelligence to Robust Control Design,* United Scholars Publications, USA, 2018.

[2] M. Karkoub, G. Balas, K. Tamma and M. Donath, "Robust control of flexible manipulators via μ-synthesis", *Control Engineering Practice*, vol. 8, no. 7, pp. 725-734, 2000. Available: 10.1016/s0967-0661(00)00006-x.

[3] D. Kim and C. Park, "Adaptive robust control design and experimental demonstration for flexible joint manipulators", *Journal of Mechanical Science and Technology*, vol. 21, no. 1, pp. 57-73, 2007. Available: 10.1007/bf03161712.

[4] W. Cheng, "Neural-fuzzy control of a flexible dynamic tracking and adjusting manipulator", *Transactions of the Institute of Measurement and Control*, vol. 37, no. 7, pp. 883-891, 2014. Available: 10.1177/0142331213510549.

[5] M. Fateh and M. Souzanchikashani, "Indirect adaptive fuzzy control for flexible-joint robot manipulators using voltage control strategy", *Journal of Intelligent & Fuzzy Systems*, vol. 28, no. 3, pp. 1451-1459, 2015. Available: 10.3233/ifs-141430.

[6] M. Soltanpour, M. Khooban and P. Otadolajam, "Robust control strategy for electrically driven

robot manipulators: adaptive fuzzy sliding mode", *IET Science, Measurement & Technology*, vol. 9, no. 3, pp. 322-334, 2015. Available: 10.1049/iet-smt.2013.0265.

[7] H. Yang, Y. Yu, Y. Yuan and X. Fan, "Back-stepping control of two-link flexible manipulator based on an extended state observer", *Advances in Space Research*, vol. 56, no. 10, pp. 2312-2322, 2015. Available: 10.1016/j.asr.2015.07.036.

[8] A. Izadbakhsh, "Robust control design for rigid-link flexible-joint electrically driven robot subjected to constraint: theory and experimental verification", *Nonlinear Dynamics*, vol. 85, no. 2, pp. 751-765, 2016. Available: 10.1007/s11071-016-2720-6.

[9] B. Niu, C. Ahn, H. Li and M. Liu, "Adaptive Control for Stochastic Switched Nonlower Triangular Nonlinear Systems and Its Application to a One-Link Manipulator", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 10, pp. 1701-1714, 2018. Available: 10.1109/tsmc.2017.2685638.

[10] F. Dong, X. Zhao, J. Han and Y. Chen, "Optimal fuzzy adaptive control for uncertain flexible joint manipulator based on D-operation", *IET Control Theory & Applications*, vol. 12, no. 9, pp. 1286-1298, 2018. Available: 10.1049/iet-cta.2017.0835.

[11] Z. Ma and G. Sun, "Dual terminal sliding mode control design for rigid robotic manipulator", *Journal of the Franklin Institute*, vol. 355, no. 18, pp. 9127-9149, 2018. Available: 10.1016/j.jfranklin.2017.01.034.

[12] D. Nam, P. Loc, N. Huong and D. Tan, "A Finite-Time Sliding Mode Controller Design for Flexible Joint Manipulator Systems Based on Disturbance Observer", *International Journal of Mechanical Engineering and Robotics Research*, pp. 619-625, 2019. Available: 10.18178/ijmerr.8.4.619-625.

[13] S. Binazadeh and R. Ghasemi, "Distributed fixed time super-twisting sliding mode leader–follower tracking protocol design for nonlinear networked flexible joint robots", *International Journal of Dynamics and Control*, vol. 8, no. 3, pp. 908-916, 2020. Available: 10.1007/s40435-019-00603-z.

[14] H. Ali, H. Jassim and A. Hasan, "Optimal Nonlinear Model Reference Controller Design for Ball and Plate System", *Arabian Journal for Science and Engineering*, vol. 44, no. 8, pp. 6757-6768, 2018. Available: 10.1007/s13369-018-3616-1.

[15] Y. Chen, C. Lee, S. Tseng and W. Hu, "Nonlinear Optimal Control Law of Autonomous Unmanned Surface Vessels", *Applied Sciences*, vol. 10, no. 5, p. 1686, 2020. Available: 10.3390/app10051686.

[16] H. Bouchekara, "Most Valuable Player Algorithm: a novel optimization algorithm inspired from sport", *Operational Research*, vol. 20, no. 1, pp. 139-195, 2017. Available: 10.1007/s12351-017-0320-y.

[17] G. Thomas, *Thomas' calculus*. Boston, MA: Pearson, 2017.

[18] J. Brown and R. Churchill, *Complex Variables And Applications*. New York: McGraw-Hill Education, 2013.

[19] C. Xu, H. Lei, J. Li, J. Ye and D. Zhang, "Adaptive Neural Control for Nonaffine Pure-Feedback System Based on Extreme Learning Machine", *Mathematical Problems in Engineering*, vol. 2019, pp. 1-13, 2019. Available: 10.1155/2019/5613212.

[20] C. Xu, H. Lei, J. Li, J. Ye and D. Zhang, "Adaptive Neural Control for Nonaffine Pure-Feedback System Based on Extreme Learning Machine", *Mathematical Problems in Engineering*, vol. 2019, pp. 1-13, 2019. Available: 10.1155/2019/5613212.

[21] W. Kanso, "Model reference adaptive control algorithm on a second order dynamic model", *Proceedings of the 33rd Midwest Symposium on Circuits and Systems*. Available: 10.1109/mwscas.1990.140937.

[22] M. Rahimian and M. Tavazoei, "Improving integral square error performance with implementable fractional-order PI controllers", *Optimal Control Applications and Methods*, vol. 35, no. 3, pp. 303-323, 2013. Available: 10.1002/oca.2069.