Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
November-December 2020: 1–17

DOI: 10.1177/1729881420981524
journals.sagepub.com/home/arx
SAGE

Optimal super-twisting sliding mode control design of robot manipulator: Design and comparison study

Ayad Q Al-Dujaili[1], Alaq Falah[2], Amjad J Humaidi[3], Daniel A Pereira[4] and Ibraheem K Ibraheem[5]

Abstract

This article presents a tracking control design for two-link robot manipulators. To achieve robust tracking control performance, a super-twisting sliding mode control (STSMC) is derived. The stability of the system based on the proposed approach is proved based on the Lyapunov theorem. However, one problem with the designed STSMC is to properly set its parameters during the design. Therefore, it is proposed a social spider optimization (SSO) to tune these design parameters to improve the dynamic performance of the robot manipulator controlled considering STSMC. The performance of the STSMC approach based on SSO is compared to that based on particle swarming optimization (PSO) in terms of dynamic performance and robustness characteristics. The effectiveness of the proposed optimal controllers is verified by simulations within the MATLAB software. It is verified that the performance given by SSO-based STSMC outperforms that resulting from PSO-based STSMC. The experimental results are conducted based on LabVIEW 2019 software to validate the numerical simulation.

Keywords

Robotic manipulator, super-twisting SMC, social spider optimization, particle swarm optimization

Date received: 3 January 2020; accepted: 27 November 2020

Topic Area: Robot Manipulation and Control
Topic Editor: Andrey Savkin
Associate Editor: Hailong Huang

Introduction

Robot manipulators are widely used to assist workers in repetitive and insecurity tasks in a myriad of industrial processes, performing them faster and more efficiently. Examples of such activities are transportation, manufacturing, welding, and drilling. However, it is essential that the manipulators track desired trajectories very precisely with a fast convergence.[1]

Although motion control has been extensively studied for robot manipulators, it is still a challenging problem because these systems are highly nonlinear, the model parameters can be time varying and also be uncertain, and the operational conditions are generally subjected to different

[1] Department of Control and Automation Techniques Engineering, Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
[2] Department of Technical Mechatronics, Technical Engineering College, Middle Technical University, Baghdad, Iraq
[3] Control and Systems Engineering Department, University of Technology, Baghdad, Iraq
[4] Department of Automatics, Federal University of Lavras (UFLA), Lavras-MG, Brazil
[5] Electrical Engineering Department, College of Engineering, University of Baghdad, Baghdad, Iraq

Corresponding author:
Ayad Q Al-Dujaili, Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq.
Email: ayad.qasim@mtu.edu.iq

kinds of input disturbances. These difficulties impose the necessity to apply advanced control techniques to solve the motion control problem, such as intelligent control (neural and fuzzy), adaptive control, and sliding mode control (SMC).[2] However, selecting a suitable control technique is a major challenge for many researchers.[3]

Among those controllers, SMC, one of the most significant discoveries in advanced control theory, is an effective and efficient tool to solve the control problems of nonlinear systems.[4,5] It is well known that SMC is robust to model uncertainties, unknown disturbances, and parameter variations.[6,7] Therefore, the SMC has been widely applied in real applications.[8]

SMC, especially on its simple first-order control structure, has many applications to mechanical systems, where the insensitiveness to uncertainties in the model is an important issue.[4,5] But this simple control structure has the limitation of being suitable just when the relative degree of the output is one, besides inducing chattering because of its switching with high frequency.[9] To avoid these restrictions, a natural solution is the application of the SMC with higher-order control structure. The super-twisting sliding mode control (STSMC) is a second-order control structure and characterized by the following advantages[10–13]:

1. The effect of chattering with STSMC is less than that in the case of SMC.
2. The STSMC can make the states of the system reach the equilibrium point in finite time, that is, it is able to zeroing both sliding variable and its derivative in finite time.
3. The STSMC requires only the sliding variable (output variable) and there is no need for its derivative and, hence, it does not require the knowledge of state derivatives as most SMC schemes do. This results in a simple control law and less computation effort.
4. The STSMC avoids the singularity and achieves exact convergence.

The main problem with the design of STSMC design is the presence of many design parameters that have to be properly set to satisfy asymptotical convergence of tracking error. In the same sense, these design parameters play a vital role on the system performance and an improper setting of these design parameters may lead to deterioration of the STSMC performance or even cause instability problems. The try-and-error procedure was the only technique to tune these design parameters in most previous works. However, the problems with this technique are that it is tedious and it could not find the best design parameters for optimal dynamic performance. Instead, modern optimization methods have been introduced to tackle the tuning problem; firstly, to avoid the try-and-error procedure, which is usually tedious, and secondly, to find the optimal design parameters, which give an optimal performance of STSMC. Thus, optimization techniques can be more adequate as a tool to tune the design parameters, providing stability and better performance.

In the field of optimization, several successful swarm algorithms have been developed by a combination of deterministic rules and randomness, mimicking the behavior of insect or animal groups in nature.[14,15] Particle swarm optimization (PSO),[16] genetic algorithm (GA), and artificial bee colony (ABC) are the most popular swarm algorithms for solving complex optimization problems. However, they present serious flaws, such as premature convergence and difficulty to overcome local minima.[15] Swarm algorithm is an optimization strategy that minimizes the behavior of animals in nature, such as insects,[15] and has been applied to complex problems in the field of optimization. GA, PSO, and ABC[16] are good examples of swarm algorithms, but they generally provide suboptimal solutions.[15]

Optimization has been applied in many control problems in the robotics field.[17–23] In the literature,[17] PSO was used to tune the parameters of fuzzy and proportional integral derivative (PID) controllers designed for the trajectory tracking of a robotic system. In this work, changes in parameters, input disturbances, and measurement noises were tested to evaluate the robustness of the algorithms.

In the literature,[18] it was introduced an optimal robust SMC tuned by ingenious multiobjective PSO to address the problem of nonlinear dynamics and tracking systems for a biped robot. In this proposed algorithm, the rate of convergence and variety of solutions are enhanced simultaneously, and innovative methods were proposed to select the global and personal best positions for each particle.

In the literature,[19] PSO was applied to tune PID controllers of aerial robots operating in a hose transportation system that requires robust controllers. The fast convergence of the proposed PSO method was verified experimentally, which also demonstrated that the solution is not unique, and the design is robust.

An efficient and fast method for tuning the controller parameters of a three degrees-of-freedom (3-DOF) robot manipulator in constrained motion was considered in the literature.[20] The optimal design of the controller parameters is tuned offline by a PSO algorithm. For designing the PSO algorithm, different performance indexes are considered in both Cartesian and joint spaces. In this work, the gripper motion was constrained by a circular trajectory to validate the effectiveness of the proposed approach.

In the literature,[21] an active disturbance rejection control was designed to control the position of a robot manipulator, considering a linear and a nonlinear strategy. Parameter uncertainties and disturbance rejection were considered in the design and the robustness of the controllers was compared in terms of the performance deviations due to the uncertainties. An estimator was considered in the control scheme proposed in this work and a PSO algorithm was responsible to tune and enhance the estimation, proving to be useful in the improvement of the system performance.

In the literature,[22] an optimization procedure was developed for the choice of parameters in the sliding surfaces of a fuzzy SMC. The control law is a state feedback implemented by a parallel distributed control scheme, where the feedback gains are calculated using the linear quadratic regular problem with weights tuned by the PSO algorithm.

In the literature,[23] the aim was the presentation of a method for estimation of parameters and modeling of dynamics in a 7-DOF hydraulic manipulator. In this work, it was developed a mechanism to achieve a feasible parameters' set consistent with input, output, and states measurements of the system, considering both noisy and uncertain conditions. A GA was applied for the identification of the system, considering the output error of a joint, resulting in a good accuracy of convergence to the desired solution.

Han and Jia proposed a sliding mode controller to achieve tracking control, anti-interference, and suppression of vibration due to flexibility in 2-DOF two-link rigid–flexible coupling manipulator subjected to bounded disturbance at input torques. The work presented a proof for the existence and uniqueness of the solution for the closed-loop system using sequences of Galerkin approximation.[24]

Yang and Tan designed a sliding mode controller based on adaptive radial basis function (RBF) neural network for joint tracking control and vibration suppression of a single flexible-link manipulator. The flexible link has been treated as a Euler–Bernoulli beam and the partial differential equation has been developed based on Hamilton approach. The adaptive RBF is employed as a neural approximator to enhance the robustness of controller by compensating both internal and external uncertainties.[25]

Rahmani et al. combined both PD control and SMC for tracking control of planar robot manipulator based on the extended gray wolf optimizer (EGWO). The hybrid controller could avoid the drawbacks of PD controller and to have a robust controller against external disturbances and uncertainties. By adding the emphasis coefficients to classical GWO, the EGWO is obtained, which is used to optimize the design parameters of GWO and EGWO-based SMCs. The work conducted a comparison in performance between the proposed control and other controllers via computer simulation.[26]

Jung[27] presented a neurosliding mode control (NSMC) to improve the tracking performance of a three-link manipulator on the basis of nonmodel-based configuration. The neural network is used as a compensator to replace the nonlinear gain in SMC such as to deal with the stability of the controlled system in an intelligent manner. A comparison in performance has been conducted via computer simulation between the proposed controller and the classical SMC to show the improvement obtained by NSMC in both Cartesian and joint spaces.

Feng et al.[28] have proposed a full-order sliding-mode controller for rigid robotic manipulators. The work suggested a time-varying gain to adjust the gain of the signum function. The regulated gain would restrict the upper bounds of uncertainties to be overestimated and has considerably reduced the effort in control signal. The proposed controller led to an attenuation in chattering and an improvement in robustness characteristics against uncertainties and it has resulted in continuous signals, which made it feasible to real-time applications.

Nicolis et al. combined SMC with model predictive control (MPC) to develop robust centralized controller for trajectory tracking and impedance control of redundant manipulators subjected to unmodeled uncertainty and disturbance. The second-order integral SMC law has been suggested to reduce the effect of chattering. The MPC utilized nominal feedback linearized of robot to guarantee motion and to fulfill the actuation constraint. The hybrid model predictive sliding mode controller could cope with the delays acting on the control input torque.[29]

Van et al. proposed adaptive backstepping nonsingular fast terminal SMC (BNFTSMC) for trajectory tracking of robot manipulators. The proposed BNFTSMC combined the merits of nonsingular SMC in terms of finite convergence high robustness and fast transient behavior, also the merit of backstepping control methodology by showing globally asymptotic stability. In addition, an adaptive version of BNFTSMC has been developed to estimate the upper bound of inherited uncertainties.[30]

Lee et al. used adaptive integral SMC based on with time-delay estimator (TDE) for tracking control of robot manipulators subjected to uncertainties. The TDE is used to estimate parametric uncertainty and disturbance. The integral sliding surface is devoted to reduce the reaching phase and to eliminate the chattering noise in control action, while the adaption mechanism can achieve high accuracy. The proposed controller demonstrated high accuracy, chattering-free, and robustness characteristics.[31]

Baek et al. applied a new adaptive sliding mode control (ASMC) based on TDE to control a robot manipulator. Also, the proposed ASMC scheme utilized pole-placement control for achieving good tracking performance with low level of chattering and guaranteed uniformly ultimately bounded of tracking errors.[32]

Ferrara and Incremona presented integral suboptimal second-order sliding mode control for trajectory tracking of robot manipulators. The control algorithm has been developed to reduce the time of reaching phase, to eliminate the chattering level in control action, and to improve the robustness characteristics of controlled system.[33]

Islam and Liu proposed an SMC technique to 2-DOF robot manipulator based on multiple model-control approach to reduce the effect of parametric uncertainty and also to reduce the gains of controller and observer. The uniformly compact set of unknown parameters is firstly divided into smaller compact subsets and a candidate SMC is designed for each subset. The candidate model, used for approximating the plant instantaneously, can be identified based on the time derivative of the Lyapunov function.[34]

Zhang et al. presented fixed-time SMC for the global fixed-time trajectory tracking of robot manipulators under bounded external disturbances and uncertain dynamics. The fixed-time sliding surface with singularity-free sliding mode controller is synthesized and the global convergence of tracking errors and sliding variable to the origin has been proven. The proposed control scheme showed fixed-time tracking, fast response, high tracking precision, and time-independent settling time with the presence of uncertainties.[35]

Ma and Sun proposed a dual terminal SMC scheme for tracking control of robot manipulator. The work has addressed the problem of underactuated issue and input limitation. The proposed controller could achieve finite-time convergence of tracking errors and it has been combined with the adaptive approach to compensate the adverse effect of input limitation. The proposed methods are characterized by chattering suppression, nonsingularity, effectiveness, and high efficiency.[36]

Wang et al. proposed a robust adaptive fuzzy terminal sliding mode controller (FTSMC) based on low-pass filter (LPF) for trajectory tracking of robot manipulator in the presence of parametric uncertainty and external disturbance. The FTSMC could satisfy desirable tracking precision and fast convergence, while the LPF keeps smooth position profile. The controller could successfully reject the effect of uncertainties without prior knowledge of their upper bound.[37]

Su and Zheng presented a new terminal sliding mode control (TSMC) for 2-DOF robot manipulators under uncertainty. A nonsingular TSMC is developed by proposing a novel integral sliding surface based on finite-time stability theory. The tracking errors and sliding surface has been proved to achieve a global finite-time convergence.[38]

Gao et al. have presented an improved robust super-twisting algorithm in the presence of unbounded uncertainties for hybrid robot system based on adaptive switching gains. The fast-adaptive law has been designed to effectively reduce the chattering effect and a global robust sliding surface has been designed to eliminate the reaching phase such that the global robustness is ensured.[39]

Ahmed et al. proposed a model-free adaptive fractional STSMC for trajectory tracking of robotic manipulators subjected to external disturbances and uncertainties. The proposed controller integrates a fractional order control, which is responsible for fast finite-time convergence, chatter-free control inputs, and better tracking performance and robustness, with an adaptive STSMC, which is designed to cope with the unknown upper-bounded uncertainty. The state estimation is performed utilizing robust exact differentiator.[40]

Louise et al. proposed STSMC based on adaptive gains for trajectory tracking of underwater swimming manipulator. The state estimation is performed by proposing a higher-order sliding mode observer. The ultimate boundedness of the tracking errors has been proven. As compared to linear PD controller, the proposed control law showed better performance characteristics.[41]

Shokoohinia and Fateh presented a robust SMC for SCARA robot based on Fourier series expansion for estimating the uncertainties. An adaptive law based on Fourier series expansion is developed to avoid try-and-error procedure in tuning the fundamental frequency.[42]

In the literature, other interesting robotic applications, such as mobile robot, underwater robot, and autonomous farm vehicles, have applied sliding control either for tracking or stabilizing control purposes.

Yang and Kim have proposed a novel sliding mode controller for a robust tracking control of nonholonomic wheeled mobile robots subjected to bounded external disturbances. The computed torque method is utilized for feedback linearization of robot dynamic system. It is shown that the tracking errors are asymptotically stabilized and the proposed scheme gives better performance in terms of accuracy and robustness.[43]

Matveev et al. have developed guidance laws for automatic path tracking of farm vehicles subjected to both constrained steering angle and bounded uncertainties. The first guidance law was synthesized based on SMC, while the other combined a smooth nonlinear control to reduce the chattering effect in control signal. The global convergence and robust stability of the proposed guidance laws have been rigorously proved and its performance has been numerically verified and experimentally tested and applied.[44]

Sarfraz et al. presented adaptive integral SMC for robust stabilizing of nonholonomic underwater systems subjected to uncertainties in both model and control inputs. The system was firstly transformed into chain form with matched uncertainties and secondly into a form with a nominal part and unknown terms. A nominal controller is devoted for nominal part and an adaptive controller for uncertain part compensation. The proposed control scheme has been developed based on Lyapunov analysis and its effectiveness has been verified via computer simulation.[45]

To address the problem of design parameters emerging throughout the design of STSMC, which have direct impact on the dynamic performance of controlled robotic system, modern optimization techniques have been introduced for optimal tuning of these parameters. In the present work, two optimization algorithms have been developed to tune the design parameters of STSMC in such a way to improve the performance of STSMC-based robotic system. One technique is based on social spider optimization (SSO) and the other is based on PSO.

The PSO technique was firstly proposed by Kennedy and Eberhart in 1995 and it was inspired by the behavior of organisms.[16] On the other hand, the SSO algorithm, proposed by Cuevas et al. in 2013,[15] is based on the simulation of the cooperative behavior of social spiders. This algorithm realistically emulates the cooperative behavior of the swarms and incorporates computational mechanisms to

avoid critical flaws that are popular in other algorithms.[46] These characteristics have motivated the use of the SSO algorithm to solve many engineering applications in different areas.[47] Other modern and generalized optimization techniques can be employed either to improve the optimization process or to make a comparison in performance among each other.[48–53]

Therefore, this study has addressed two main contributions: one concerning the solutions of chattering problem in control signal and the tracking control problem of Pelican-type manipulator in the presence of uncertainty by designing a robust controller based on STSMC. The other contribution focuses on the development of two optimization algorithms, based on SSO and PSO methods, to tune the design parameters that emerge in the design of STSMC, such to reach the optimal performance of STSMC-based robotic system. Also, a comparative study has been conducted between the performances of STSMC based on the proposed optimization techniques. Thus, the contribution of the present work can be embodied by pursuing the following steps of objectives.

Development of the control law for the STSMC-based 2-DOF manipulator (Pelican-type). This includes the proof of asymptotic stability for the controlled system based on Lyapunov theorem.

- Development of optimization algorithm for SSO and PSO techniques.
- Conducting a comparative study between SSO and PSO algorithms in terms of dynamic performance and robustness characteristics.
- Experimental verification of simulated results based on LABView software.

The article is organized as follows: In the second section, the dynamic model of the robot manipulators system is presented. The third section is dedicated to the design of the STSMC. The fourth section is dedicated to utilize the SSO algorithm to find the optimal design parameters of the proposed controller. In the fifth section, a simulation study is given to verify the effectiveness of the proposed scheme. The experimental results have been conducted in the sixth section to validate the results of numerical simulation. In the seventh section, the conclusions drawn based on numerical and experimental observations are given.

Modeling of pelican robot manipulator

As shown in Figure 1, the robot manipulator under consideration is a 2-DOF planar arm composed of a revolute joint and two links. Each link is actuated by an electrical motor, so one is in the base and the other is in the elbow. The motor axes and the links are directly coupled.

The arm of the manipulator has planar movement in the Cartesian directions $x - y$ and is composed of two links,

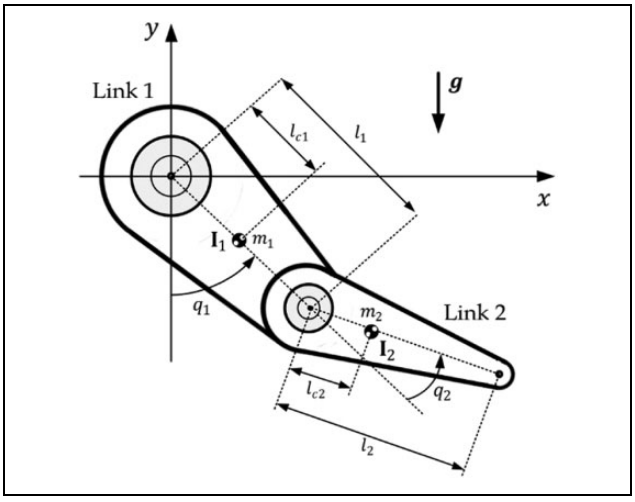

Figure 1. Schematic diagram of Pelican robot.

considered to be rigid bodies, with lengths I_1 and I_2, and masses m_1 and m_2. These parameters of the manipulator can be seen in Figure 1. The moments of inertia are denoted by l_{c1} and l_{c2}, each one calculated with respect to the axes parallel to the direction x that cross the center of mass of the respective link. The parameters l_{c1} and l_{c2} represent, respectively, the distance between the rotating axe and the center of mass of each link. The angles q_1 and q_2, both positive counterclockwise, denote the DOF of the manipulator.[54]

The positions of the joints are represented by the vector $q = [q_1 \quad q_2]^{\mathrm{T}}$. Aiming a representation suitable for the application of control methods, the following compact form of the dynamic model can be written as follows

$$\begin{aligned} &\boldsymbol{M}(\boldsymbol{q})\,\ddot{q} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \boldsymbol{G}\,(\boldsymbol{q}) + \boldsymbol{d} = \boldsymbol{\tau} \\ &\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix}\ddot{q} + \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix}\dot{q} \\ &\quad + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \end{aligned} \tag{1}$$

where τ_1 and τ_2 are the external torques delivered by the actuators at joints 1 and 2 and $\boldsymbol{d}$ represents the uncertainty vector in system parameters. Also, the elements of matrices $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{G}$ are given by

$$\begin{aligned} M_{11}(q) &= m_1 l_{c1}{}^2 + m_2 l_1{}^2 + m_2 l_{c2}{}^2 + 2m_2 l_1 l_{c2}\cos(q_2) + I_1 + I_2 \\ M_{12}(q) &= M_{21}(q) = m_2 l_{c2}{}^2 + m_2 l_1 l_{c2}\cos(q_2) + I_2 \\ M_{22}(q) &= m_2 l_{c2}{}^2 + I_2 \\ C_{11}(q,\dot{q}) &= -m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2 \\ C_{12}(q,\dot{q}) &= -m_2 l_1 l_{c2}\sin(q_2)[\dot{q}_1 + \dot{q}_2] \\ C_{21}(q,\dot{q}) &= m_2 l_1 l_{c2}\sin(q_2)\dot{q}_1 \\ C_{22}(q,\dot{q}) &= 0 \\ g_1(q) &= [m_1 l_{c1} + m_2 l_1]g\sin(q_1) + m_2 g l_{c2}\sin(q_1 + q_2) \\ g_2(q) &= m_2\, g l_{c2}\sin(q_1 + q_2) \end{aligned}$$

Table 1. Physical parameters of pelican robot manipulator.

Symbol	Description	Value
l_1	Length of link 1 (m)	0.26
l_2	Length of link 2 (m)	0.26
l_{c1}	Distance to the center of mass (link 1) (m)	0.0983
l_{c2}	Distance to the center of mass (link 2) (m)	0.0229
m_1	Mass of link 1 (kg)	6.5225
m_2	Mass of link 2 (kg)	2.0458
I_1	Inertia relative to the center of mass (link 1) (kg m^2)	0.1213
I_2	Inertia relative to the center of mass (link 2) (kg m^2)	0.0116
g	Gravity acceleration (m/s^2)	9.81

Equation (1) can be rewritten as follows

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}(\boldsymbol{q})^{-1}[\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\ \dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{d}] \tag{2}$$

It is important to notice that q_1 and q_2, the angular positions, and $\dot{q}_1$ and $\dot{q}_2$, their derivatives, are the state variables. So, the dynamic model should be rewritten considering these state variables

$$\frac{d}{dt}\begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \boldsymbol{M}(\boldsymbol{q})^{-1}[\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\ \dot{\boldsymbol{q}} - \boldsymbol{G}\ (\boldsymbol{q}) - \boldsymbol{d}] \end{bmatrix} \tag{3}$$

In Table 1, it can be found the values adopted for each parameter of the manipulator.

Design of super-twisting sliding mode control for two degrees of freedom manipulator

The sliding surface vector $\boldsymbol{s} = [s_1\ s_2]^T$ is given by

$$\boldsymbol{s} = \dot{\boldsymbol{e}} + c\ \boldsymbol{e} \tag{4}$$

where $c = [c_1\ c_2]^{\mathrm{T}}$ is a positive design parameter and $\boldsymbol{e} = [e_1\ e_2]^{\mathrm{T}}$ is the error value between the desired and actual angular positions ($\boldsymbol{q}_d$) and ($\boldsymbol{q}$), respectively

$$\boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q} \tag{5}$$

where $\boldsymbol{q}_d = [q_{d1}\ q_{d2}]^{\mathrm{T}}$ represents the desired joint positions. The time derivative of equation (4) gives

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{e}} + c\ \dot{\boldsymbol{e}} = \ddot{\boldsymbol{q}}_d - \ddot{\boldsymbol{q}} + c\ \dot{\boldsymbol{e}} \tag{6}$$

Substituting equation (18) into equation (6), we have

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}}_d - \boldsymbol{M}(\boldsymbol{q})^{-1}[\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\ \dot{\boldsymbol{q}} - \boldsymbol{G}\ (\boldsymbol{q}) - \boldsymbol{d}] + c\dot{\boldsymbol{e}} \tag{7}$$

The control signal is composed of two terms, which are the equivalent (u_{eq}) and the switching (u_{sw}) control laws, represented by

$$\boldsymbol{\tau} = \boldsymbol{M}(q)\ \left(\boldsymbol{u}_{\mathrm{eq}} + \boldsymbol{u}_{\mathrm{sw}}\right) \tag{8}$$

where $\boldsymbol{u}_{\mathrm{sw}} = [u_{\mathrm{sw1}}\ u_{\mathrm{sw2}}]^{\mathrm{T}}$. The equivalent law is given by

$$\boldsymbol{u}_{\mathrm{eq}} = \ddot{\boldsymbol{q}}_d + c\dot{\boldsymbol{e}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\ \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\ \dot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{G}(\boldsymbol{q}) \tag{9}$$

Also, the design of the switching law is derived from the super-twisting algorithm and the control law is written as

$$\boldsymbol{u}_{\mathrm{swi}} = k_i\sqrt{|s_i|}\ \mathrm{sgn}(s_i) + w_i\int \mathrm{sgn}\ (s_i)\ \mathrm{d}t \tag{10}$$

where $i = 1, 2$, k_i and w_i are positive design parameters. The equivalent law is given by

Substituting equations (9) and (10) into equation (8), we have

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ \left(\ddot{q}_d + c\ \dot{\boldsymbol{e}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{u}_{\mathrm{swi}}\right) \tag{11}$$

Then, substituting equation (11) into equation (7), we get

$$\begin{aligned}\dot{\boldsymbol{s}} = \Bigg[& -k_1\sqrt{|s_1|}\ \mathrm{sgn}(s_1) - w_1\int \mathrm{sgn}(s_1)\mathrm{d}t \\ & -k_2\ \sqrt{|s_2|}\ \mathrm{sgn}(s_2) - w_2\int \mathrm{sgn}(s_2)\mathrm{d}t\Bigg]^{\mathrm{T}} + \boldsymbol{b}\end{aligned} \tag{12}$$

where $\mathbf{b} = [b_1\ b_2]^{\mathrm{T}} = \left(\boldsymbol{M}(q)^{-1}\ \boldsymbol{d}\right)^{\mathrm{T}}$. The Lyapunov method can be applied for the stability analysis of the system with the STSMC. The quadratic Lyapunov candidate function can be written in terms of the sliding surfaces

$$V(\boldsymbol{s}) = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{s}$$

The Lyapunov function derivative is calculated as

$$\dot{V}(\boldsymbol{s}) = s_1\dot{s}_1 + s_2\dot{s}_2 \tag{13}$$

Using equations (12) and (13) becomes

$$\begin{aligned}\dot{V}(s) = {} & s_1\left(-k\sqrt{|s_1|}\ \mathrm{sgn}(s_1) - w_1\int \mathrm{sgn}\ (s_1)\ \mathrm{d}t + b_1\right) \\ & + s_2\left(-k_2\sqrt{|s_2|}\ \mathrm{sgn}(s_2) - \ w_2\int \mathrm{sgn}\ (s_2)\ \mathrm{d}t + b_2\right)\end{aligned}$$

or

$$\begin{aligned}\dot{V} \leq {} & -k_1\sqrt{|s_1|}\ |s_1| - |s_1|\int w_1\ \mathrm{d}t + |s_1|b_1 - k_2\sqrt{|s_2|}\ |s_2| \\ & - |s_2|\int w_2\ \mathrm{d}t + |s_2|b_2\end{aligned} \tag{14}$$

Writing b_1 and b_2 in integral forms, equation (14) becomes

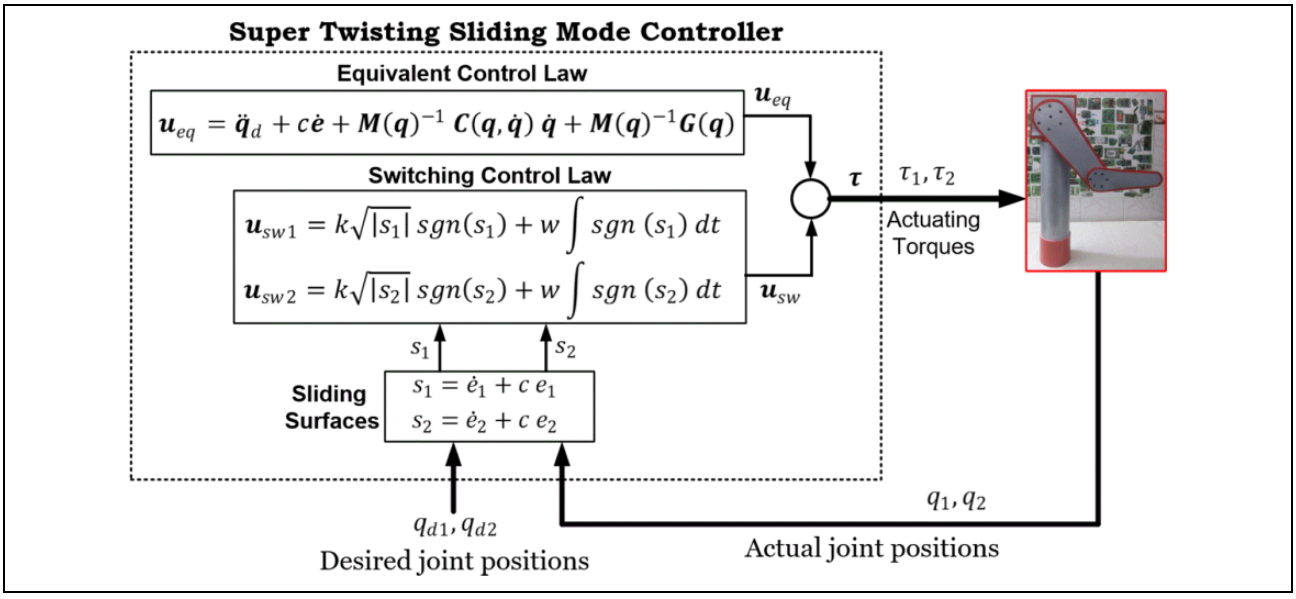

Figure 2. Scheme of 2-DOF manipulator controlled by STSMC. STSMC: super-twisting sliding mode control; 2-DOF: two degrees of freedom.

$$\begin{aligned}\dot{V} &\le -k_1\sqrt{|s_1|}\,|s_1| - |s_1|\int w_1\,\mathrm{d}t + |s_1|\int \dot{b}_1\,\mathrm{d}t \\ &\quad - k_2\sqrt{|s_2|}\,|s_2| - |s_2|\int w_2\,\mathrm{d}t + |s_2|\int \dot{b}_2\,\mathrm{d}t \\ &\le -k_1\sqrt{|s_1|}\,|s_1| - |s_1|\int w_1\,\mathrm{d}t + |s_1|\int \delta_1\,\mathrm{d}t \\ &\quad - k_2\sqrt{|s_2|}\,|s_2| - |s_2|\int w_2\,\mathrm{d}t + |s_2|\int \delta_2\,\mathrm{d}t\end{aligned}$$

or

$$\begin{aligned}\dot{V} \le -k_1\sqrt{|s_1|}\,|s_1| - |s_1|\left(\int (w_1 - \delta_1)\mathrm{d}t\right) \\ - k_2\sqrt{|s_2|}\,|s_2| - |s_2|\left(\int (w_2 - \delta_2)\mathrm{d}t\right) \le 0\end{aligned} \tag{15}$$

The STSMC design is schematically represented in Figure 2.

***Remark 1*:** The manipulator will have guaranteed stability if w_1 and w_2 have values satisfying $w_1 > \delta_1 > |\dot{b}_1|$ and $w_2 > \delta_2 > |\dot{b}_2|$. Under this condition, V will be positive definite and $\dot{V}$ will be negative definite for all $t \to 0$. Also, sliding variables s_1 and s_2 will converge to zero as $t \to \infty$.

Optimized super-twisting sliding mode control for two degrees of freedom manipulator

During the development of STSMC algorithm for robot manipulator system, three design parameters have appeared, denoted by c, k, and w. These design parameters are essential for stability and performance of the closed loop system. The trial and error procedure for the adjustment of these parameters is not really practical and, in fact, makes it impossible to find an optimal set of values. In the present work, two optimization techniques, represented by SSO and PSO algorithms, are applied for the finding of optimal design parameters in terms of best dynamic performance of a 2-DOF manipulator controlled by the STSMC.

Social spider optimization-based optimization of super-twisting sliding mode control

In this part, the performance of the robot manipulator is optimized by the SSO algorithm, which is formulated to find an optimal value for the design parameters of the STSMC.

The SSO algorithm considers both male and female spiders, being their numbers defined as members inside the search space when the algorithm is initialized. The number related to the females is N_f and it is chosen within the range between 65% and 90%, randomly, of the whole society N_s, in selected former times. Consequently, N_f is given by[15]

$$N_f = \text{floor}\,[(0.9 - 0.25 \times \text{rand}) \cdot Ns] \tag{16}$$

where floor (.) maps a real number to an integer number, while rand is a random number lies between.$[0,\ 1]$ The number of male spiders N_m is calculated based on N_f and N_s as follows[55]

$$N_m = Ns - N_f \tag{17}$$

The complete population S, composed by N_s elements, are divided in two subgroups F and M. The group f assembles the set of female individuals ($f = \{f_1, f_2, \ldots, f_{Nf}\}$) while M

groups the male members ($M = \{m_1, m_2, \ldots, m_{Nf}\}$), where $S = f \cup M$ ($S = \{s_1, s_2, \ldots, s_{Nf}\}$), such that

$$S = \{s_1 = f_1, s_2 = f_2, s_{Nf} = f_{Nf}, s_{Nf+1} = m_1, s_{Nf+2} = m_2, \ldots, s_N = m_{Nm}\}.$$

Assignation of fitness. Each spider in the population S is quantified based on its fitness value and weight. The weight w_i assigned to each spider qualifies the spider i of the population. The weight of each spider is assessed based on the following equation[15]

$$w_i = \frac{J(s_i) - (\text{worst})_s}{(\text{best})_s - (\text{worst})_s} \tag{18}$$

being $J(s_i)$ the fitness at the spider position s_i, calculated by the objective function $J(.)$. Also, $(\text{worst})_s$ and $(\text{best})_s$ values are given by

$$\text{best}_s = \max_{k \in (1,2,\ldots N)} J_{(sk)} \tag{19}$$

$$\text{worst}_s = \min_{k \in (1,2,\ldots N)} J_{(sk)} \tag{20}$$

Modeling of the vibrations. In the colony, the information among members are encoded based on small vibrations. Such vibrations are employed to initiate the spider and they are related to spider weight and the distance between spiders. The model of vibration process can be described by

$$Vb_{i,j} = w_j \, e^{-d_{i,j}^2} \tag{21}$$

where i and j denote the indices of spider individuals, $d_{i,j}$ is the distance from spider j and i defined in terms of the Euclidean norm as[55]

$$d_{i,j} = s_i - s_j \tag{22}$$

Initialization of population. As it is already obvious, the SSO algorithm is iterative and requires a randomic initialization of the population of males and females. The iterative procedure is started with definition of the N spider locations and the set S. The location of the female spider f_i (or male m_i) contains the parameters values that must be optimized. These values are comprised inside a range of predefined upper and lower bounds, p_j^{high} and p_j^{low}, respectively, respecting a uniform and random distribution within this range. These relations are described by[15]

$$f_{i,j}^0 = p_j^{\text{low}} + \text{rand}\left(p_j^{\text{high}} - p_j^{\text{low}}\right) \tag{23}$$

$$m_{i,j}^0 = p_j^{\text{low}} + \text{rand}\left(p_j^{\text{high}} - p_j^{\text{low}}\right) \tag{24}$$

where $i = 1, 2, \ldots, N_f$, $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, N_m$, and $f_{i,j}$ ($m_{i,j}$) represents the j'th parameter of the i'th female (male) spider position.

Cooperative operator

Female cooperative operator. Female spiders are attracted or not by other spiders. This is defined by the consistency of the vibrations emitted by the spider over the communal net. The attraction movement is activated if r_m, a random number uniformly distributed in the range [0, 1], is smaller than a threshold PF. If this is satisfied, the attraction movement is produced based on the following formula[46]

$$f_i^{k+1} = f_i^k + \alpha.\, V_i\, bc_i\, .\, \left(s_c - f_i^k\right) + \beta.\, V_i bb_i.\, \left(s_b - f_i^k\right) + \delta.(\text{rand} - 0.5) \tag{25}$$

Otherwise, a repulsion movement is generated as

$$f_i^{k+1} = f_i^k + \alpha.\, V_i\, bc_i\, .\, \left(s_c - f_i^k\right) - \beta.\, V_i bb_i.\, \left(s_b - f_i^k\right) + \delta.(\text{rand} - 0.5) \tag{26}$$

where β, α, δ, and rand(.) are created randomly within [0, 1], while t represents the iteration number. The individuals s_b and s_c represent the nearest member to i that holds a higher weight and the best individual of the entire population S, respectively.

Male cooperative operator. The biological features of the social cooperation determine the classification of a male population as dominant (D) or nondominant (ND), taking as reference their position relatively to the median individual. In terms of size and weight, the D spiders present better fitness than the ND spiders. Additionally, the D males are attracted by the nearest female spiders inside the communal web. The male individual is called a dominant (D) member when its weight value is bigger than the median value for the males, while the nondominant (ND) male is that member, which has weight less than the median value. The position change of male spider can be modeled using the following formula[46]

$$m_i^{k+1} = \begin{cases} m_i^k + \alpha.\, Vbf_i.\, \left(s_f - m_i^k\right) + \delta.\, (\text{rand} - 0.5) \text{ if } w_{Nf+i} > w_{Nf+m} \\ m_i^k + \alpha.\left(\dfrac{\sum_{h=1}^{Nm} m_h^k.\, w_{Nf+h}}{\sum_{h=1}^{Nm} w_{Nf+h}} - m_i^k\right) \text{ if } w_{Nf+i} \leq w_{Nf+m} \end{cases} \tag{27}$$

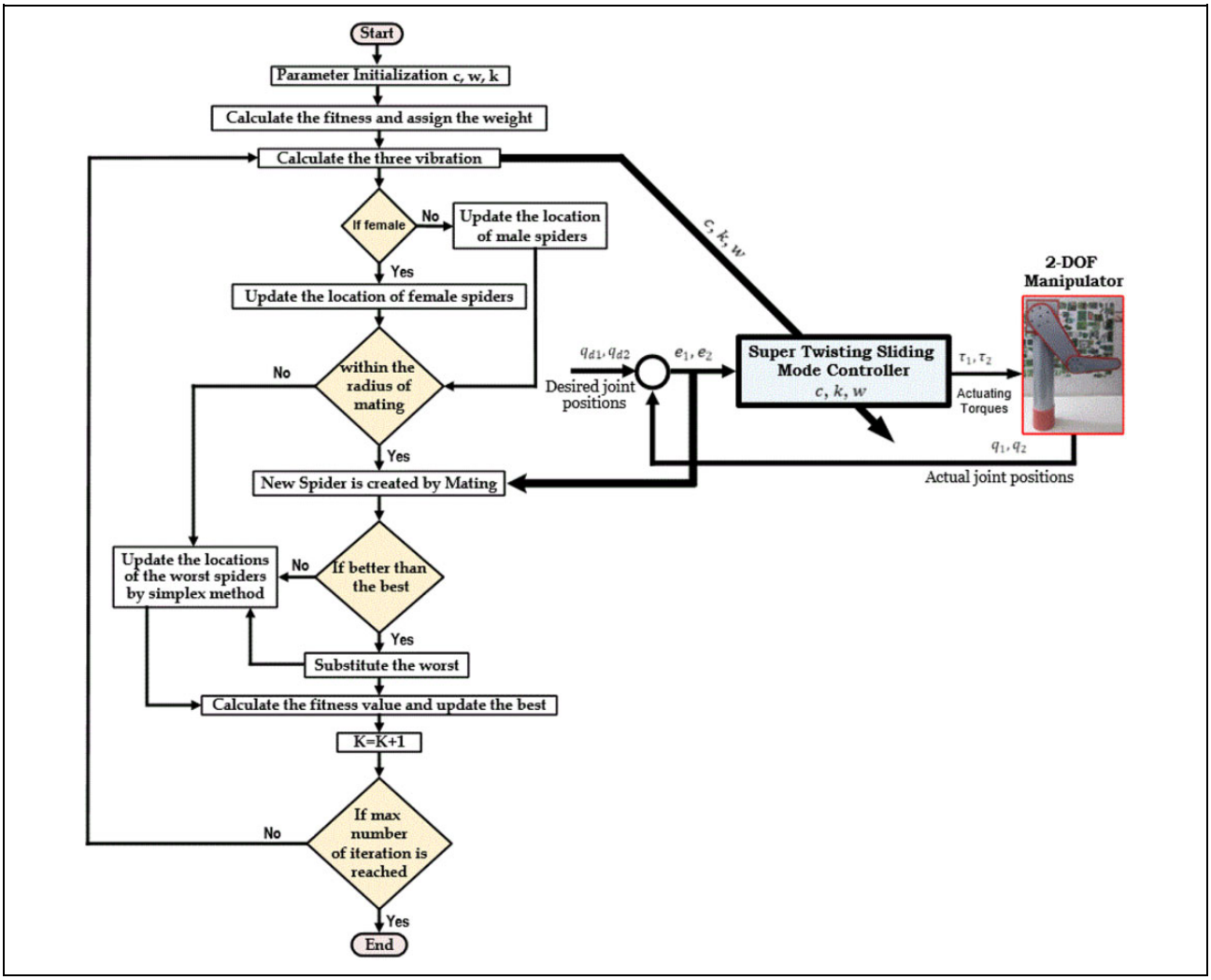

Figure 3. Flowchart of the SSO algorithm for robot manipulator system controlled by STSMC. STSMC: super-twisting sliding mode control; SSO: social spider optimization.

where w_{Nf+m} denotes the median male member, s_f is the closest female individual to the male individual i, and $\frac{\sum_{h=1}^{Nm} m_h^k \cdot w_{Nf+h}}{\sum_{h=1}^{Nm} w_{Nf+h}} - m_i^k$ represents the mean weight in the male population M.

Mating operator. The mating phenomenon in the colony of social spiders is performed by female male dominant individuals. If the dominant male m_g spider $(g \in D)$ identifies a set E^g of female individuals inside a range r_e, it mates, constituting a new brood s_{new} that is created taking into account the components of the set T^g, that is, this new generation is given by the union $E^g \cup m_g$. The radius r_e gives a region that is related to the search area being calculated by[15]

$$r_e = \frac{\sum_{j=1}^{n} \left(P_j^{\text{high}} - P_j^{\text{low}} \right)}{2\, n} \tag{28}$$

The probability of a spider, in the mating technique, having an impact on each member of the new brood is defined by the weight of the related spider. Bigger weights increase the probability to affect the new product. The Roulette method can be used to measure the influence probability P_{si} of the individuals as[46,47]

$$P_{si} = \frac{w_i}{\sum_{j \in T^k} w_j} \tag{29}$$

where $i \in T^g$. After shaping the new spider, the new candidate s_{wo} should be compared to the worst spider s_{new}, in terms of their weights (where $w_{\text{wo}} = \min_{l \in (1,2,\ldots N)}(w_l)$. In case of higher value, the new spider will become the worst spider. If not, the colony will stay unchanged. When the worst spider is substituted by the new one, the new spider assumes the index and the gender of the replaced one. So, the whole population S continues with the original rate between male and female individuals. A flowchart describing the SSO algorithm can be seen in Figure 3.

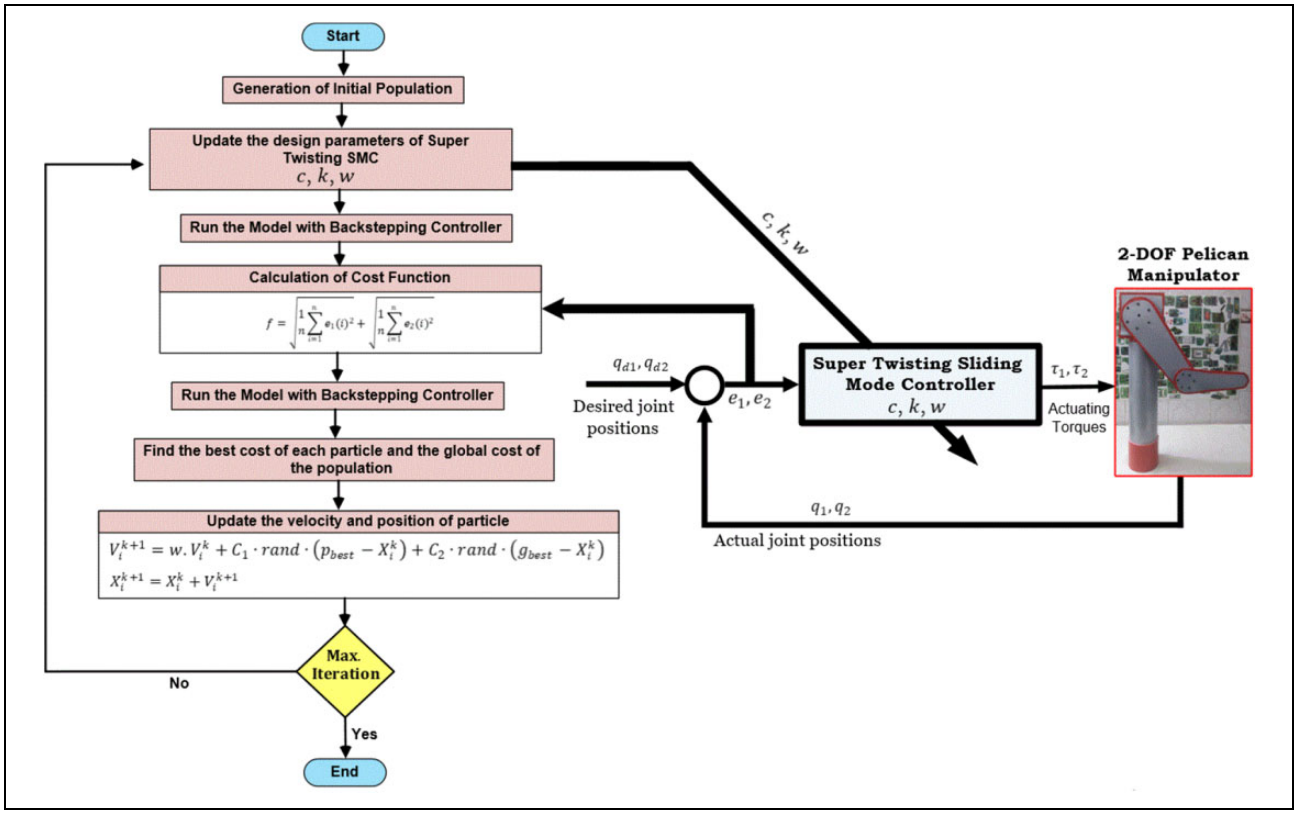

Figure 4. Flowchart of PSO algorithm for tuning the design parameters of STSMC. STSMC: super-twisting sliding mode control; PSO: particle swarm optimization.

Particle swarm optimization-based optimization super-twisting sliding mode control

The PSO was widely method category of swarm intelligence for solving optimization problems and find the local and global solution first come in by Kennedy and Eberhart in 1995,[16] the basic PSO algorithm depends on three steps (generating particles positions and velocities, velocity update, and position update). This algorithm is taken from the conduct of some animals, such as birds and fish.[11,14]

The velocity of the particle in each iteration is computed by[56,57]

$$V_i^{k+1} = w.\ V_i^k + C_1 \cdot \text{rand} \cdot \left(p_{\text{best}} - X_i^k\right) + C_2 \cdot \text{rand} \cdot \left(g_{\text{best}} - X_i^k\right) \tag{30}$$

where the weights w, C_1, and C_2 are, respectively, the inertia, the self-confidence, and the swarm confidence. The suitable value range (C_1 and C_2) is between (1–2), but the setting of value 2 is the most suitable in many problems. The function "rand" generates random numbers with zero mean and the inertia weight can be given by

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right) k/k_{\max}$$

where k is the current number of iterations, $k_{\max}$ is the maximum number of iteration, $w_{\max}$ and $w_{\min}$ are the maximum and minimum weights, respectively. The appropriate value of $w_{\min}$ is 0.4, while that for $w_{\max}$ is 0.9. The position updated by[56,57]

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{31}$$

where X_i^k and X_i^{k+1} represent the current and updated values, respectively, containing the STSMC design parameters c, k, and w, which are required to be tuned. The PSO-based STSMC scheme for the 2-DOF robot manipulator system is depicted in Figure 4.

The PSO algorithm tries to run the manipulator system controlled by STSMC many times until finding the optimal value of the cost function and finishing the algorithm. Then, this set of optimal design parameters is addressed to the proposed controller. The objective function f considered for the evaluation of every particle during the search of the better value is based on the minimization of the following criterion

$$f = \text{ITAE} = \int_0^T t \cdot |e|\ \mathrm{d}t \tag{32}$$

Simulation results

The computer results for the STSMC are presented for the tracking control of the robot manipulator system based on

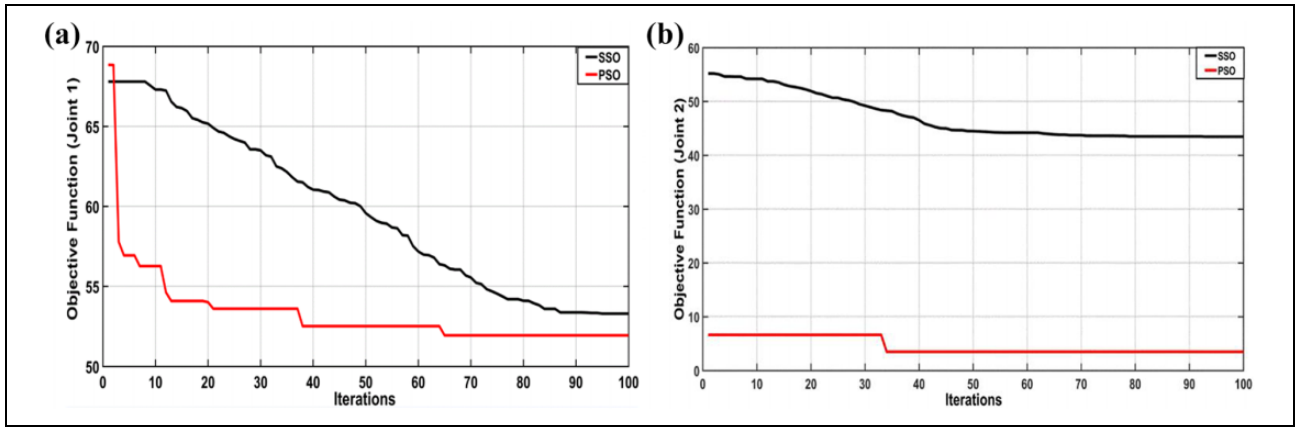

Figure 5. Behaviors of objective function based on SSO and PSO of (a) q_1, and (b) q_2. PSO: particle swarm optimization; SSO: social spider optimization.

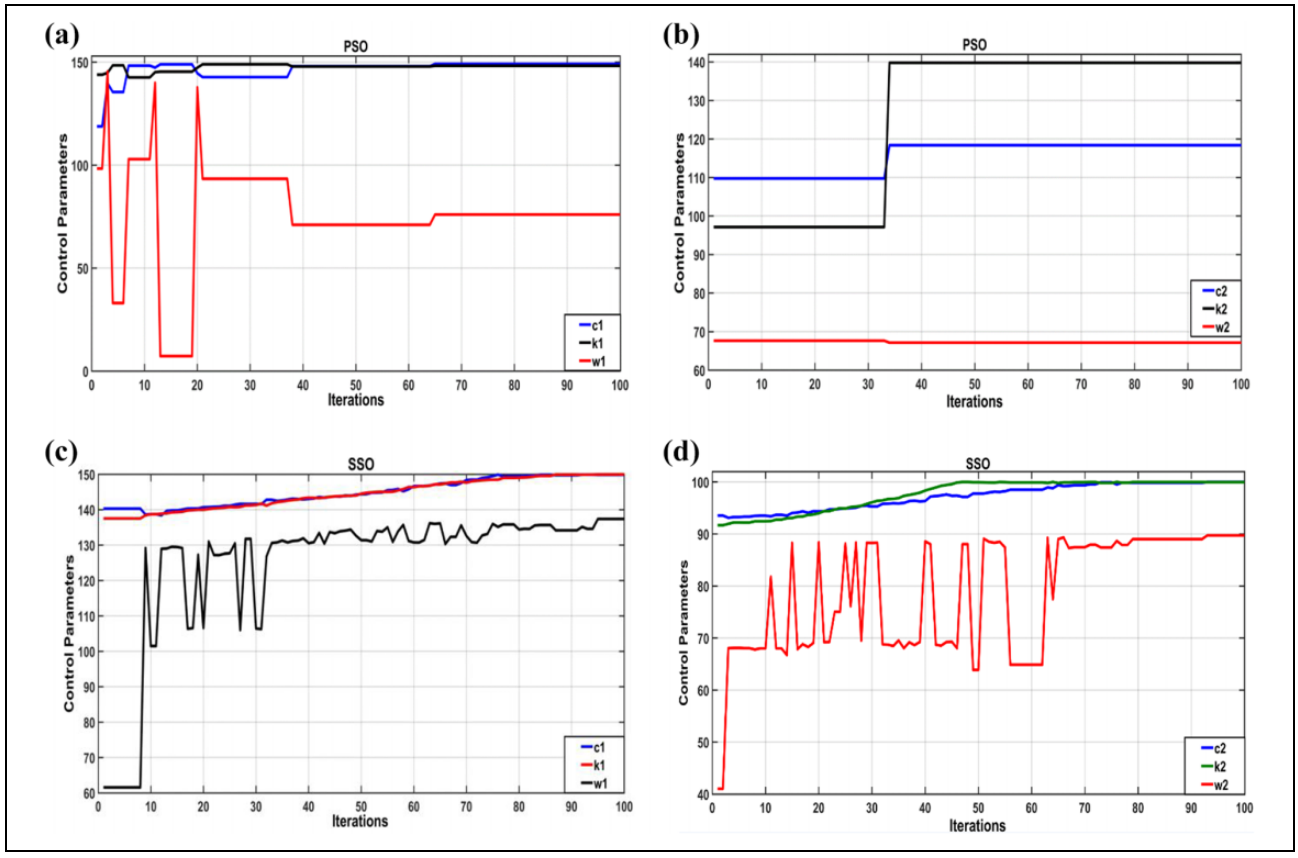

Figure 6. Parameter evolution along iterations: (a) c_1, w_1, and k_1 parameters based on PSO; (b) c_2, w_2, and k_2 parameters based on PSO; (c) c_1, w_1, and k_1 parameters based on SSO; and (d) c_2, w_2, and k_2 parameters based on SSO. PSO: particle swarm optimization; SSO: social spider optimization.

an SSO and PSO algorithms. The effectiveness of STSMC and the performance of optimization techniques are verified using computer simulation within MATLAB environment.

Figure 5 shows the trace of fitness function over the iteration of both PSO and SSO techniques for each manipulator joint. It is clear from the figure that the PSO technique gives better optimization performance than the SSO technique for manipulator controlled by STSMC. Figure 6 shows the behavior of the best design parameters c_j, w_j, and k_j with respect to iteration in the case of SSO and PSO algorithms, where $j = 1, 2$. The optimal values of design parameters at the end of each optimization process are listed in Table 2. These optimal values are set in the proposed controller and, hereafter, the controller can be termed as "optimal super twisting SMC."

Figure 7 shows the angular position and velocity of joint 1 (a, b), and angular velocity and, the angular position and velocity of joint 2 (c, d) and its reference trajectory based on optimal STSMC. Figure 8 shows the control signal of joint 1 (a), joint 2 (b) and sliding surface signal in joint 1 (c), joint 2 (d).

Next, the performance of STSMC based on PSO and SSO algorithms under uncertainty in parameters has been investigated. In this scenario, a change by 8% ratio of nominal value in moment of inertias (I_1 and I_2) has been allowed. Figures 9 and 10 show the angular positions (and velocities) of manipulator joints and the actuating torques of both joints, respectively.

Table 2. Optimal design parameters of STSMC.

Design constants for the STSMC	PSO	SSO
c_1	149.1205	149.8627
k_1	148.2037	149.9814
w_1	76.0179	137.3982
c_2	118.3952	99.9934
k_2	139.7738	99.9583
w_2	67.1226	89.7139

PSO: particle swarm optimization; SSO: social spider optimization; STSMC: super-twisting sliding mode control.

The performances of STSMC based on SSO and PSO techniques are numerically reported in Table 3. This table indicates that the PSO-based STSMC is better than that based on SSO algorithm in terms of transient and robustness characteristics.

Real-time implementation of optimal super-twisting sliding mode control for pelican manipulator

Figure 11 shows the hardware setup for practical implementation of proposed optimal versions of STSMC. The overall hardware setup consists of the following elements: Pelican-type robot manipulator, NI-6212 interfacing device, the personal computer (PC), and drive circuits.

The experiment setup has been carried out on manufactured Pelican-type manipulator. The manipulator is a 2-DOF and it consists of two arms moving on a vertical plane. The total length of arm links, when they are aligned in straight line, is 0.52 m long and these rigid links are tailored from 6061 aluminum sheet. The joints of rigid arms are directly connected to drive motors via revolute joints.

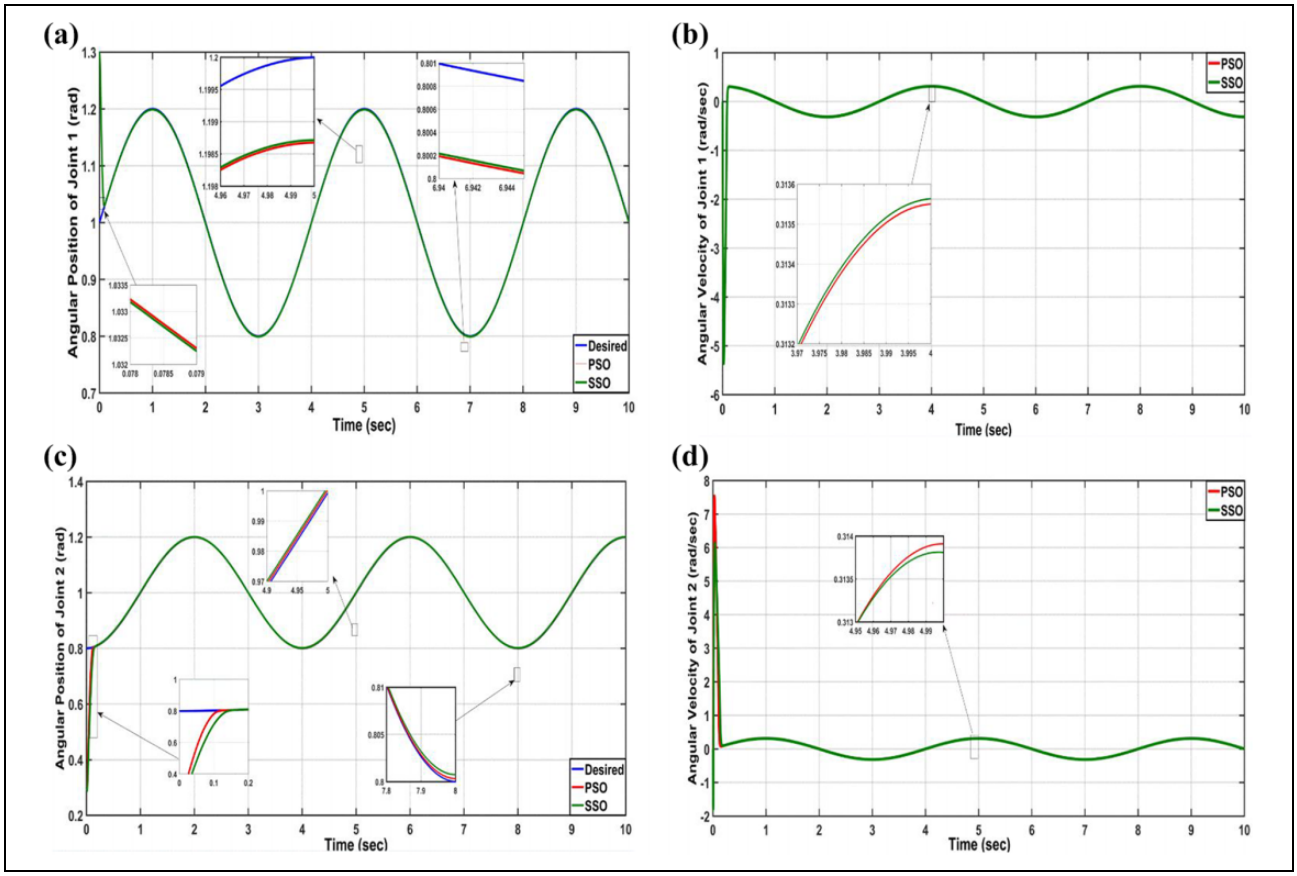

Figure 7. (a, b) The angular position and velocity of joint 1 and (c, d) the angular position and velocity of joint 2 and its reference trajectory.

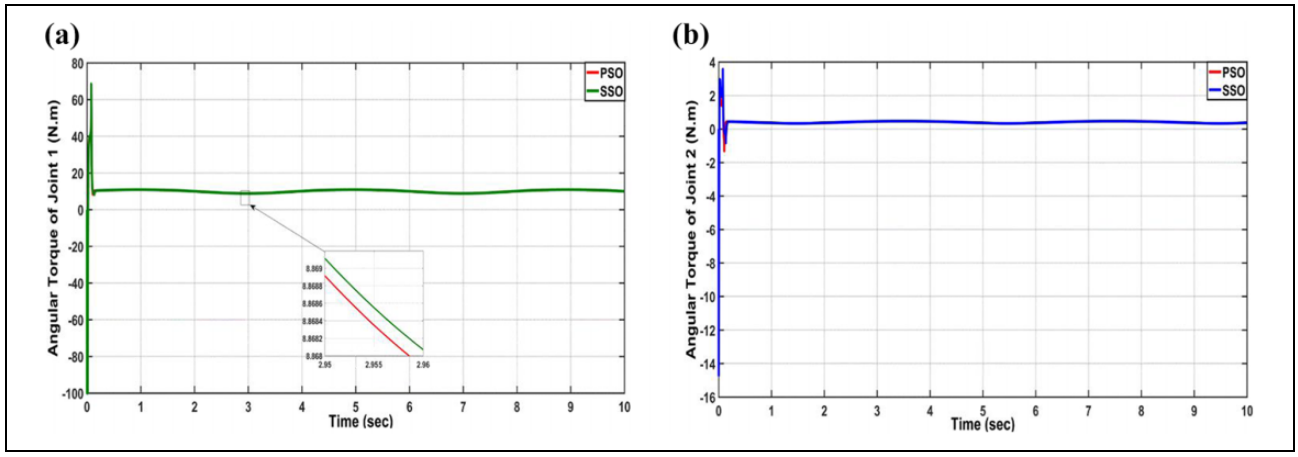

Figure 8. The control signal of (a) joint 1 and (b) joint 2 based on PSO and SSO optimization methods. PSO: particle swarm optimization; SSO: social spider optimization.

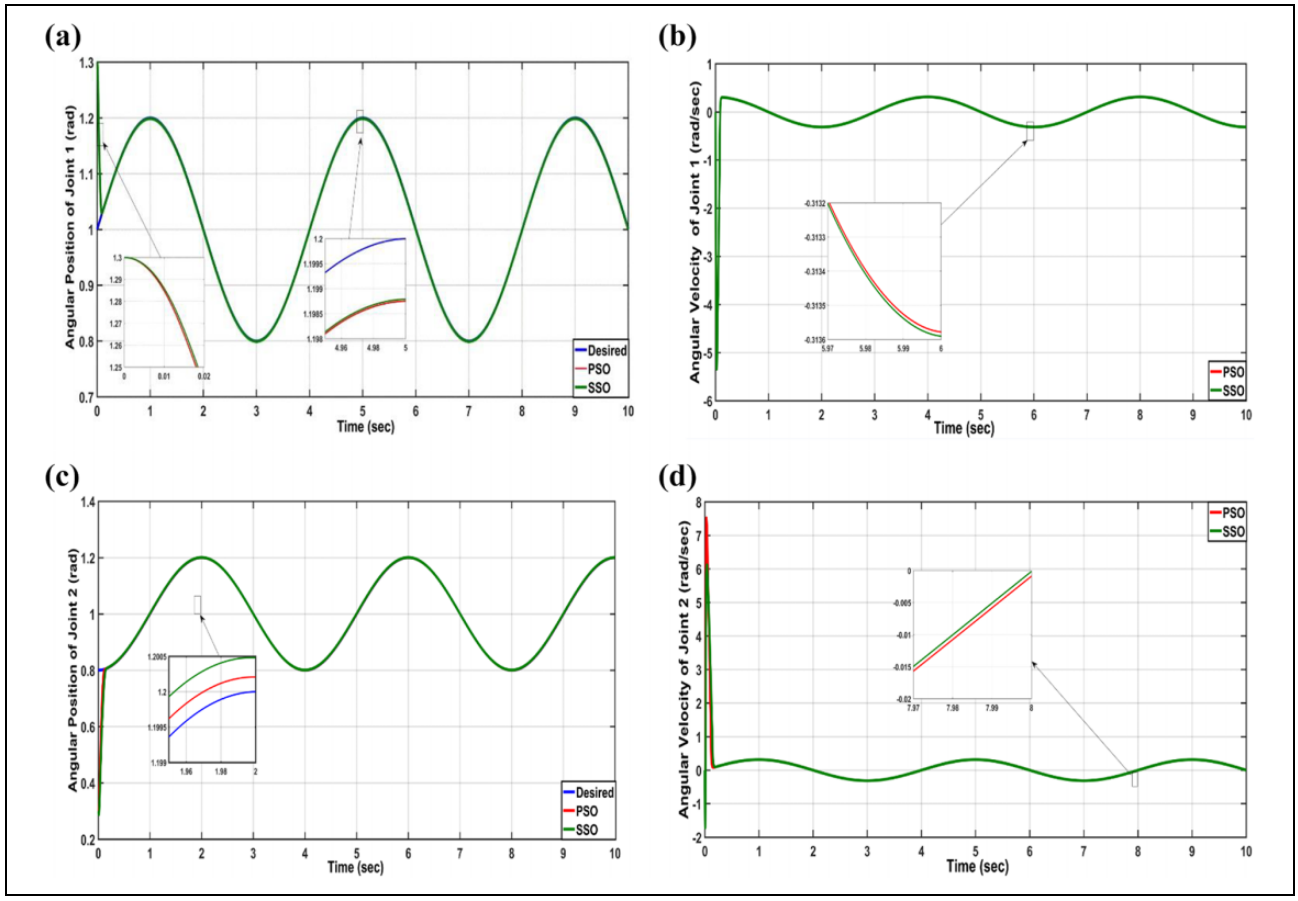

Figure 9. (a, b) The angular position and velocity of joint 1 and (c, d) the angular position and velocity of joint 2 and its reference trajectory with 8% uncertainty in I_1 and I_2.

The control operation is achieved through software, which is called "Laboratory Virtual Instrument Engineering Workbench" or LabVIEW. The LabVIEW has a facility to interface with MATLAB, where the interface document within LabVIEW environment permits the user to define calls to a MATLAB file. The algorithm of STSMC has been coded inside MATLAB function, which is called by the LabVIEW programing software via interfacing nodes. The sampling period for executing the control algorithm has been set at 2.5 ms since this sampling period is fast enough to approximate continuous control signals produced by executed continuous-time strategy. The control

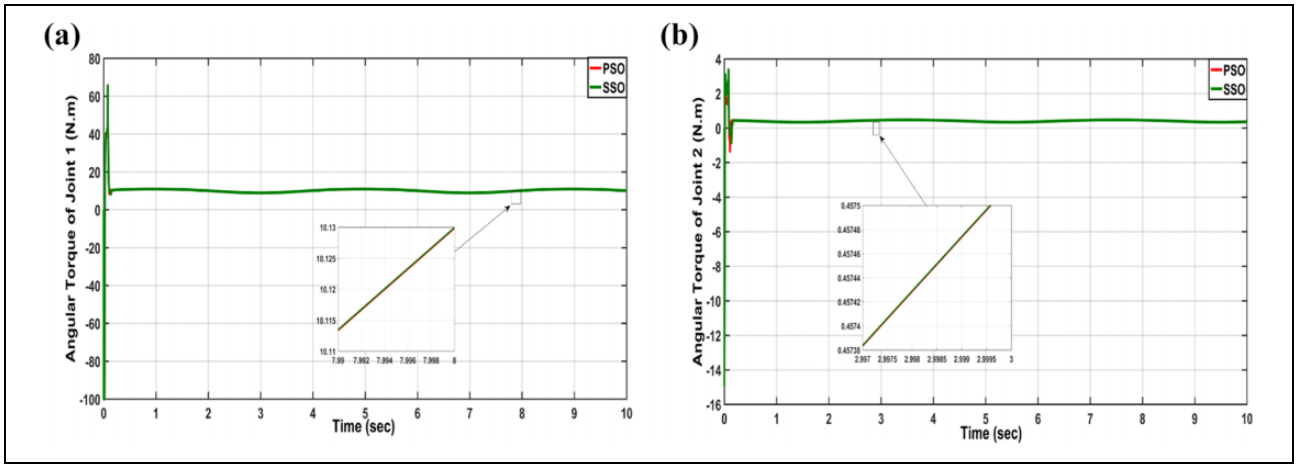

Figure 10. The control signal of (a) joint 1 and (b) joint 2 based on PSO and SSO methods with 8% uncertainty in I_1 and I_2. PSO: particle swarm optimization; SSO: social spider optimization.

Table 3. Report of the dynamic performances of STSMC.

Optimization method	State variable	RMSE	Setting time (s)
Normal case			
SSO	q_1	0.0156	0.1
	q_2	0.0352	0.183
PSO	q_1	0.0155	0.1
	q_2	0.0303	0.14
Uncertainty case (+8% in I_1 and I_2)			
SSO	q_1	0.0156	0.1
	q_2	0.0353	0.183
PSO	q_1	0.0155	0.1
	q_2	0.0303	0.14

PSO: particle swarm optimization; SSO: social spider optimization; STSMC: super-twisting sliding mode control; RMSE: root mean squared error.

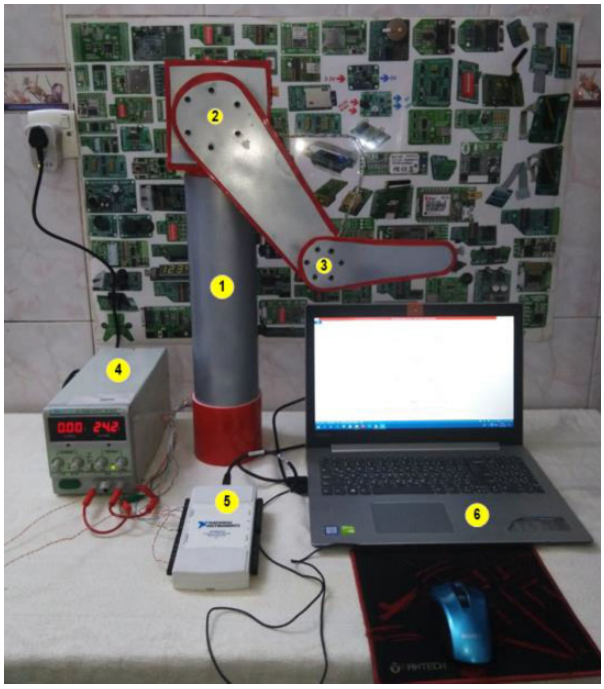

Figure 11. The hardware set-up of Pelican robot manipulator controlled by PSO-based and SSO-based STSMC controllers. (1) Pelican manipulator, (2) shoulder joint motor, (3) elbow joint motor, (4) power supply, (5) NI-6212 interfacing device and (6) PC. PSO: particle swarm optimization; SSO: social spider optimization; STSMC: super-twisting sliding mode control; PC: personal computer.

signals generated by LabVIEW-based super-twisting sliding mode controller is exported to the environment via the national instrument device NI-USB-6212. The control signals are transferred to motor drives via I/O analogue ports of NI-USB-6212 data acquisition card (DAC). The information of feedback signals are acquired by the PC (LabVIEW package) via I/O digital and analogue port of DAC.

The actuating motors used in the real-time implementation are from Parker Compumotor of type DM1200-A and DM1015-B for the shoulder joint and elbow joint, respectively. These actuating motors are high torque without gear reduction and they are operated in "torque mode"; that is, they work as torque sources and they are commanded with analog voltages for requested torques. The motor at shoulder joint (DM1200-A) can develop a maximum torque of 150 Nm, while the elbow motor (DM1015-B) delivers maximum torque up to 15 Nm. Based on data sheet of applied motors, their resolutions are 1,024,000 steps/rev, that is, the accuracy of both motors reaches 0.0069.

The data information of position for both joints (shoulder and elbow joints) based on mounted encoders are acquired within the environment of LabVIEW programming format and they are exported as to MATLAB environment for the purpose of plotting. Figures 12 and 13 show the experimental position behavior of q_1 and q_2 subjected to sinusoidal reference input. However, this scenario has been implemented based on normal condition. To evaluate the performances of both optimal controllers, the root

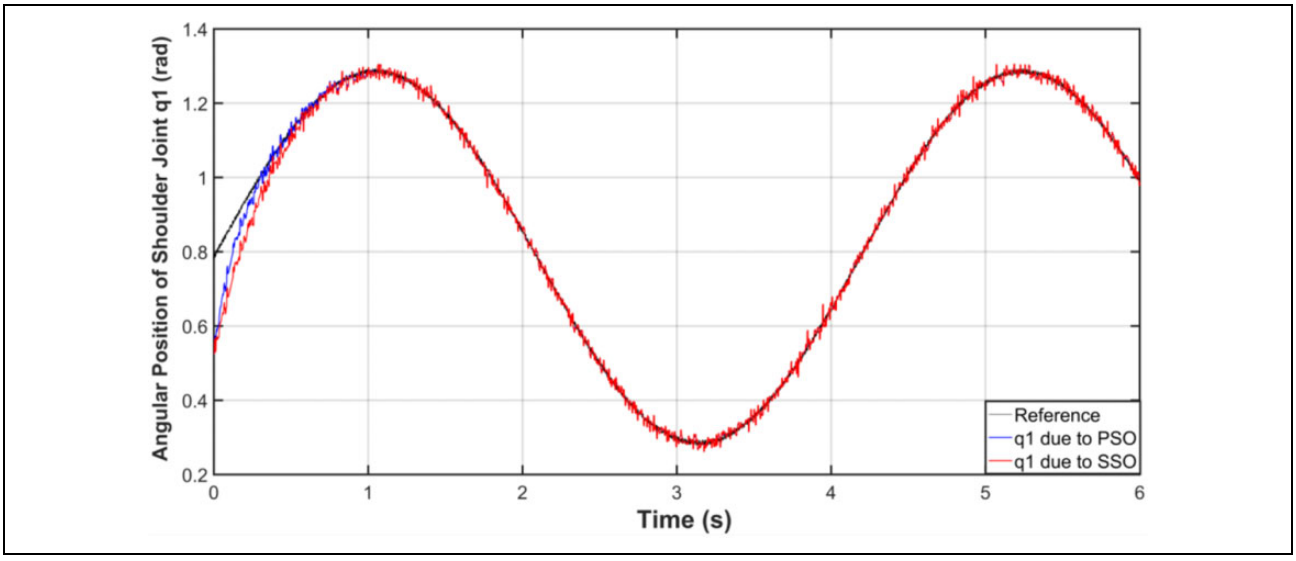

Figure 12. The real behaviours of shoulder joint based on PSO and SSO algorithms. PSO: particle swarm optimization; SSO: social spider optimization.

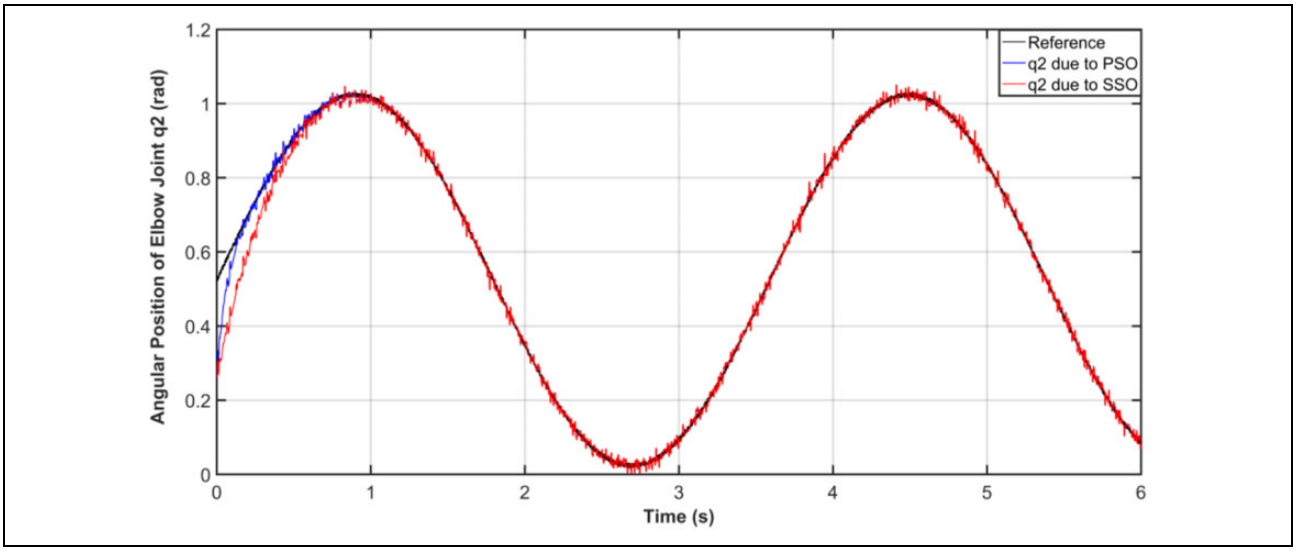

Figure 13. The real behaviours of elbow joint based on PSO and SSO algorithms. PSO: particle swarm optimization; SSO: social spider optimization.

mean squared error (RMSE) is used as a measuring index for evaluation. Based on shoulder joint information (Figure 11), the RMSE for PSO-based controller is calculated to be 0.012 rad, while that for SSO-based controller is given by 0.019 rad. On the other hand, the same calculation of RMSE is repeated for elbow joint information based on Figure 12. The RMSE value of q_2 for PSO-based controller is found to be 0.015, while the RMSE value is equal to 0.021 based on SSO-based STSMC. This indicates that better improvement of performance can be obtained by PSO technique than that given by SSO technique. Moreover, the experimental results showed that the response of q_1 and q_2 is faster in case of PSO-based controller than those based on SSO-based controller.

Conclusion

In this article, an STSMC has been developed for tracking control of robot manipulator. The design procedure of STSMC is developed and the asymptotic stability is proved and discussed in detail. Instead of relying on the trial and error method to determine the optimal design parameters of the STSMC algorithm, a modern optimization technique based on SSO and PSO tuneris to find the optimal design parameters of the proposed controller. A comparative study between SSO and PSO techniques is made toward better dynamic performance and the computer simulation showed that dynamic performance of STSMC based on PSO is better than that based on SSO. Also, PSO-based STSMC

of manipulator has better robustness characteristics than that based on SSO technique. The conclusions based on numerical simulation have been validated by conducting experimental results, which showed the superior of using PSO technique to the SSO algorithm in terms of control performance.

The work can be extended to include the observer or adaptation mechanism to account for unmeasured states or uncertainty. The other suggestion is to conduct the proposed scheme in this study for other robotic applications, such as wheeled mobile robot and underwater and flying robots.[45,58,59]

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Ayad Q Al-Dujaili https://orcid.org/0000-0002-1126-3290
Amjad J Humaidi https://orcid.org/0000-0002-9071-1329
Ibraheem K Ibraheem https://orcid.org/0000-0002-4486-0710

References

1. Baek J and Kwon W. Practical Adaptive Sliding-Mode Control Approach for Precise Tracking of Robot Manipulators, *Appl. Sci*. 2020; 10(8): 1–16.
2. Fatma M, Dorsaf E, and Tarak D. Robust Control for a Two DOF Robot Manipulator. *J Elec and Comput Eng* 2019; 1–13.
3. Shahnaz T-H, Farzin P, and Jong-Myon K. Robust composite high-order super-twisting sliding mode control of robot manipulators. *Robotics* 2018; 7(1): 13.
4. Maolin J, Jinoh L, Pyung HC, et al. Practical non-singular terminal sliding-mode control of robot manipulators for high-accuracy tracking control. *IEEE Trans Industr Electr* 2009; 56(9): 3593–3601.
5. Young KD, Utkin VI, and Ozguner U. A control engineer's guide to sliding mode control. *IEEE Trans Control Syst Technol* 1999; 7(3): 328–342.
6. Utkin VI. *Sliding modes in control and optimization*. Berlin, Germany: Springer Science & Business Media, 2013.
7. Edwards C and Spurgeon S. *Sliding mode control: theory and applications*. Boca Raton, Florida: CRC Press, 1998.
8. Kamal S, Moreno JA, Chalanga A, et al. Continuous terminal sliding mode controller. *Automatica* 2016; 69: 308–314.
9. Jeong C-S, Kim J-S, and Han S-I. Tracking error constrained super-twisting sliding mode control for robotic systems. *Int J Control Auto Syst* 2018; 16(2): 804–814.
10. Humaidi AJ and Hameed AH. PMLSM position control based on continuous projection adaptive sliding mode controller. *Syst Sci Control Eng* 2018; 6(3): 242–252.
11. Humaidi AJ and Hasan AF. Particle swarm optimization–based adaptive super-twisting sliding mode control design for 2-degree-of-freedom helicopter. *Meas Control J* 2019; 52(9–10): 1403–1419.
12. Feng Z and Fei J. Design and analysis of adaptive super-twisting sliding mode control for a micro-gyroscope. *PLoS One* 2018; 13(1): e0189457.
13. Yaoyao W, Fei Y, Jiawang C, et al. Optimal non-singular terminal sliding mode control of cable-driven manipulators using super-twisting algorithm and time-delay estimation. *IEEE Access* 2018; 6: 61039–61049.
14. Humaidi AJ and Hameed M. Development of a new adaptive backstepping control design for a non-strict and under-actuated system based on a PSO tuner. *Information* 2019; 10(2): 38: 1–17.
15. Cuevas E, Cienfuegos M, Zaldívar D, et al. A swarm optimization algorithm inspired in the behavior of the social-spider. *Exp Syst Appl* 2013; 40(16): 6374–6384.
16. Kennedy J and Eberhart R. Particle swarm optimization. *IEEE Int Conf Neural Network* 1995; 4: 1942–1948.
17. Zafer B and Oguzhan K. A fuzzy logic controller tuned with PSO for 2DOF robot trajectory control. *Exp Syst Appl* 2011; 38(1): 1017–1031.
18. Mahmoodabadia MJ, Taherkhorsandi M, and Bagherib A. Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO. *Neurocomputing* 2014; 124: 194–209.
19. Estevez J and Graña M. Robust control tuning by PSO of aerial robots hose transportation. In: Ferrández Vicente J, Álvarez-Sánchez J, de la Paz López F, et al.. (eds) *Bio-inspired computation in artificial systems lecture notes in computer science*. Berlin: Springer, Vol. 9108: 291–300, 2015.
20. Haifa M and Olfa B. Impedance controller tuned by particle swarm optimization for robotic arms. *Int J Adv Robot Syst* 2011; 8(5): 93–103.
21. Humaidi AJ and Badr HM. Linear and nonlinear active disturbance rejection controllers for single-link flexible joint robot manipulator based on PSO tuner. *J Eng Sci Technol Rev* 2018; 11(3): 133–138. doi:10.25103/jestr.113.18
22. Chaouech L, Soltani M, Dhahri S, et al. A new optimal sliding mode controller design using scalar sign function. *Iranian J Fuzzy Syst* 2017; 14(4): 67–85.
23. West C, Montazeri A, Monk S, et al. A genetic algorithm approach for parameter optimization of a 7-DOF robotic manipulator. *IFAC-PapersOnLine* 2016; 49(12): 1261–1266.
24. Han F and Jia Y. Sliding mode boundary control for a planar two-link rigid-flexible manipulator with input disturbances. *Int J Control AutoSyst* 2020; 18: 351–362.
25. Yang H-J and Tan M. Sliding mode control for flexible-link manipulators based on adaptive neural networks. *Int J Auto Comput* 2018; 15: 239–248.
26. Rahmani M, Komijani H, and Rahman M. New sliding mode control of 2-DOF robot manipulator based on extended grey wolf optimizer. *Int J Control Auto Syst* 2020; 18(X): 1–9.

27. Jung S. Improvement of tracking control of a sliding mode controller for robot manipulators by a neural network. *Int J Control Auto Syst* 2018; 16(2): 937–943.
28. Feng Y, Zhou M, Yu X, et al. Full-order sliding-mode control of rigid robotic manipulators. *Asian J Control* 2019; 21(4): 1–9.
29. Nicolis D, Allevi F, and Rocco P. Operational space model predictive sliding mode control for redundant manipulators. *IEEE Trans Robot* 2020; 36(4): 1348–1355.
30. Van M, Mavrovouniotis M, and Ge S. An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *IEEE Trans Syst Man Cybernet Syst* 2019; 49(7): 1448–1458.
31. Lee J, Hun P, and Jin M. Adaptive integral sliding mode control with time-delay estimation for robot manipulators. *IEEE Trans Industr Electr* 2017; 64(8): 6796–6804.
32. Baek J, Jin M, and Han S. A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans Industr Electr* 2016; 63(6): 3628–3637.
33. Ferrara A and Incremona G. Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators. *IEEE Trans Control Syst Technol* 2015; 23(6): 2316–2325.
34. Islam S and Liu X. Robust sliding mode control for robot manipulators. *IEEE Trans Industr Electr* 2011; 58(6): 2444–2453.
35. Zhang L, Wang Y, Hou Y, et al. Fixed-time sliding mode control for uncertain robot manipulators. *IEEE Access* 2019; 7: 149750–149763.
36. Ma Z and Sun G. Dual terminal sliding mode control design for rigid robotic manipulator. *J Franklin Inst* 2018; 355(18): 9127–9149.
37. Wang P, Zhang D, and Lu B. Trajectory tracking control for chain-series robot manipulator: robust adaptive fuzzy terminal sliding mode control with low-pass filter. *Int J Adv Robot Syst* 2020; 17: 1–12.
38. Su Y and Zheng C. A new nonsingular integral terminal sliding mode control for robot manipulators. *Int J Syst Sci* 2020; 51(8): 1418–1428.
39. Gao G, Zhang S, and Ye M. Global robust super-twisting algorithm with adaptive switching gains for a hybrid robot. *Int J Adv Robot Syst* 2020; 17: 1–12.
40. Ahmed S, Ahmed A, Mansoor I, et al. Output feedback adaptive fractional-order super-twisting sliding mode control of robotic manipulator. *Iranian J Sci Technol Trans Electr Eng* 2020. DOI: 10.1007/s40998-020-00364-y.
41. Louise I, Borlaug G, Gravdahl JT, et al. Trajectory tracking for underwater swimming manipulators using a super twisting algorithm. *Asian J Control* 2019; 21(1): 208–223.
42. Shokoohinia MR and Fateh MM. Robust dynamic sliding mode control of robot manipulators using the Fourier series expansion. *Trans Inst Measure Control* 2018; 41: 2488–2495.
43. Yang JM and Kim JH. Sliding mode control for trajectory tracking of non-holonomic wheeled mobile robots. *IEEE Trans Robot Auto* 1999; 15(3): 578–587.
44. Matveev AS, Hoy H, Katupitiya J, et al. Nonlinear sliding mode control of an unmanned agricultural tractor in the presence of sliding and control saturation. *Robot Autonom Syst* 2013; 61(9): 973–987.
45. Sarfraz M, Rehman F, and Shah I.Robust stabilizing control of nonholonomic systems with uncertainties via adaptive integral sliding mode: an underwater vehicle example. *Int J Adv Robot Syst* 2017; 14: 1–11.
46. Zhou Z, Yongquan M, Zhou Y, et al. A simple method-based social spider optimization algorithm for clustering analysis. *Eng Appl Artificial Intell* 2017; 64: 67–82.
47. Alberto L-C, Erik C, Fernando F, et al. Social spider optimization algorithm: modifications, applications, and perspectives. *Math Probl Eng* 2018; 2018: 1–29.
48. Moezi SA, Zakeri E, and Zare A. A generally modified cuckoo optimization algorithm for crack detection in cantilever Euler-Bernoulli beams. *Precision Eng* 2018; 52: 227–241.
49. Allawi ZT, Ibraheem KI, and Humaidi AJ. Fine-tuning meta-heuristic algorithm for global optimization. *MDPI Process J* 2019; 7: 657.
50. Gao ZM and Juan Z. An improved grey wolf optimization algorithm with variable weights. *Comput Intell Neurosci* 2019; 2019: 1–13.
51. Humaidi AJ, Kadhim SK, and Gataa AS. Development of a novel optimal backstepping control algorithm of magnetic impeller-bearing system for artificial heart ventricle pump. *Cyber Syst An Int J* 2020; 51(4): 521–541.
52. Jhila N and Farzin MK. A whale optimization algorithm (WOA) approach for clustering. *Cogent Math Stat* 2018; 5: 1.
53. Seyedali M. A sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 2016; 96: 120–133.
54. Kelly R, Santibáñez D, Victor LP, et al. Control of robot manipulators in joint space. In: Michael JG and Lorenzo M (eds) *Advanced textbooks in control and signal processing*. London: Springer-Verlag London, 2005.
55. Zhao Z, Ruxin A, Luo Q, et al. Elite opposition-based social spider optimization algorithm for global function optimization. *Algorithms* 2017; 10(1): 9.
56. Humaidi AJ, Ibraheem IK, Azar AT, et al. A new adaptive synergetic control design for single link robot arm actuated by pneumatic muscles. *MDPI Entropy* 2020; 22: 723.
57. Humaidi AJ, Oglah AA, Abbas SJ, et al. Optimal augmented linear and nonlinear PD control design for parallel robot based on PSO tuner. *Int Rev Model Simulat* 2019; 12(5): 281–291.
58. Hailong H and Savkin AV. Towards the internet of flying robots: a survey. *MDPI Sensors* 2018; 18(11): 4038.
59. Pavol B, Akkad MAA, Peter B, et al. Navigation control and stability investigation of a mobile robot based on a hexacopter equipped with an integrated manipulator. *Int J Adv Robot Syst* 2017; 14(6): 1–13.