ITC 1/47
Journal of Information Technology and Control
Vol. 47 / No. 1 / 2018
pp. 26-44
DOI 10.5755/j01.itc.47.1.15031

Terminal Sliding Mode Control with Evolutionary Algorithms for Finite-Time Robust Tracking of Nonholonomic Systems

Received 2017/05/11 | Accepted after revision 2018/01/22

http://dx.doi.org/10.5755/j01.itc.47.1.15031

Terminal Sliding Mode Control with Evolutionary Algorithms for Finite-Time Robust Tracking of Nonholonomic Systems

Hossein Ghasemi, Behrooz Rezaie, Zahra Rahmani
Babol Noshirvani University of Technology, Department of Electrical and Computer Engineering,
Shariati Ave., Babol, Mazandaran, Iran

Corresponding author: brezaie@nit.ac.ir

This paper deals with utilizing a recursive fast terminal sliding mode control method for finite-time robust tracking in a class of nonholonomic systems described by an extended chained form of differential equations. To enhance the performance of the proposed method, the constrained parameters of the controller are exactly tuned using evolutionary algorithms such that the tracking error reaches zero in a short time while chattering is significantly reduced. A comparative study is also presented among the applied evolutionary algorithms, namely, differential evolution, bat optimization, cuckoo optimization and bacterial foraging optimization. Applying the proposed design method leads to a considerable reduction in convergence time of the states as well as the chattering phenomenon. It is shown that the method is robust against disturbance in the input of the system. Numerical simulations for a well-known nonholonomic system, i.e., wheeled mobile robot demonstrate effective improvement in the results compared with conventional terminal sliding mode control method.

KEYWORDS: Nonholonomic systems, Terminal sliding mode control, Finite-time tracking, Evolutionary algorithms, Chained form.

1. Introduction

An important class of general nonlinear systems is nonholonomic (NH) systems which have been studied as classical mechanical systems for many years. These systems have many applications in the fields of mechanics, mobile robotics, electro-magnetics and electromechanics. In such systems, there exists a special type of conditions which restricts their motions. Some examples of the NH systems are sledges that slide on a plane, wheels and spheres that roll without slipping on a plane, mobile cars and car-like vehicles, wheeled mobile robots (WMRs), knife-edge, under-actuated satellites, surface vessel and space robots [1, 3, 10, 14, 16].

In recent years, issues related to the control of the NH systems have received much attention. Due to the fact that Brockett's necessary condition cannot be satisfied for the NH systems [4], such systems cannot be stabilized by smooth or even continuous stationary feedback control laws while they are controllable. In addition, such systems are highly nonlinear and they are also generally not invertible, and thereby designing controller for stabilizing or tracking is not an easy task. Therefore, different control methods based on nonlinear control strategies such as discontinuous control methods, time varying control methods, the methods based on system conversion, sliding mode control methods, hybrid control methods and so on, have been devoted to the problem of stabilizing or tracking in the NH systems [12, 16, 17, 20, 33, 37, 38].

One of the important methods for controlling the NH system is based on system conversion into subsystems. For instance, Ploeg et al. in [29] have proposed a position control method based on feedback linearization, in which multi-cycle robots have been converted to a set of identical unicycles. Whenever each unicycle is controlled, as a consequence, the whole NH system position is controlled. Even though their method has some advantages, uncertainty in the dynamics has not been considered.

NH systems can also be described in a chained form, and many control strategies have been proposed based on this form [16, 20, 21, 34]. WMR is a well-known example of the NH systems that can be stated in chained form with an appropriate coordinate transformation. Such transformations are useful tools for making the system suitable for applying control strategies in order to force the system to move along a desired trajectory.

Sliding mode control (SMC) method, especially, terminal sliding mode control (TSMC) is a discontinuous control technique that has attracted the attention of many researchers in recent years due to its superior features like robustness, easy implementation, finite-time convergence and high precision performance [18, 19, 35, 41]. Recently, new types of TSMC with different sliding surfaces and also some improvements by using adaptive or intelligent methods have been proposed in the literature [13, 22, 23]. Such mechanisms are useful for improving the finite-time convergence of the control systems. In addition, nonsingular versions of the TSMC method have also been studied in [7, 11] to tackle the problem of singularity in TSMC.

Utilizing SMC for controlling the NH systems have been reported in several papers [2, 6, 9, 21, 24, 34, 36]. In these works, backstepping-based second-order SMC [9], backstepping-based adaptive SMC [6] have been used to control the NH systems in presence of uncertainties. Moreover, in [2, 21, 24, 34, 36], TSMC has been proposed for stabilizing the NH systems with chained form of state equations. In addition, the intelligent control strategies such as fuzzy and neural controllers and their combinations with SMC-based methods have also been widely proposed for trajectory tracking problems [5, 8, 15, 25-27, 32, 42].

Despite the advantages of the SMC-based control methods developed for the NH systems, most of them possess low speed of convergence and tracking error takes a long period of time to reach zero. Even in some cases, tracking error does not reach zero, especially in presence of uncertainty and disturbance. Therefore, the problem of robust finite-time tracking of the NH systems with chained form in presence of disturbances and uncertainties is still an open challenge in the literature. Moreover, making improvements in the SMC-based control methods in the NH systems, by using some supervisory mechanisms such as intelligent control algorithms, can be an open challenge which motivates the current research. To address these problems, this paper uses a TSMC method, in which the main targets are to minimize the tracking error and to maximize the convergence speed.

In this paper, we utilize a recursive TSMC for controlling the chained-form NH systems in order to minimize the tracking error in these systems. Despite the effectiveness of the recursive TSMC, there are some constrained parameters in the structure of recursive terminal sliding surface that should be determined accurately in order to reach such control objectives as tracking and the convergence of the states. Exact adjustment of such parameters leads to better tracking results and increases the convergence speed. The idea studied in this paper is to determine such parameters by using the intelligent algorithms. For this purpose, an intelligent TSMC is proposed, in which the TSMC is combined with evolutionary algorithms. By using such intelligent optimization methods, the recursive terminal sliding structure is improved by reducing the convergence time of tracking error for the NH systems described in an extended chained form. Four intelligent algorithms, namely, differential evolution (DE), bat optimization (BO), cuckoo optimization (CO) and bacterial foraging optimization (BFO) are studied and compared in this paper. Such strategies are separately used to adjust the TSMC in order to minimize the tracking error of the NH system. Input disturbance is also considered in this paper, and it is shown that the convergence is guaranteed in presence of the disturbance. The simulation results show the effectiveness of the proposed method and it will be shown that by using the proposed method, a better tracking error can be achieved compared with the conventional TSMC.

The paper is organized as follows. In Section 2, the dynamic model and problem formulation are described. The main proposed controller is introduced in Section 3, in which a recursive TSMC is applied to the NH system in order to achieve finite-time tracking control of the desired output in presence of external disturbances. The intelligent optimization algorithms are also utilized in this section to adjust the coefficients of the TSMC in order to improve the performance of the closed loop system. Section 4 illustrates the application of the proposed method to a nonholonomic WMR as a well-known benchmark example. Simulation results are illustrated in Section 5 to show the applicability and effectiveness of the proposed method. Finally, Section 6 gives some conclusion remarks.

2. The Problem Statement

Consider NH systems in generalized extended chained form described by the following equations [34]:

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \\ &\vdots \\ \dot{x}_n &= x_{n-1} u_1, \end{aligned} \tag{1}$$

where u_1 and u_2 are control inputs of the system and $x = [x_1, \ldots, x_n]^T$ is the state vector of the system.

The desired dynamics, which is to be tracked by the NH system, is considered as:

$$\begin{aligned} \dot{x}_{1d} &= u_{1d} \\ \dot{x}_{2d} &= u_{2d} \\ \dot{x}_{3d} &= x_{2d} u_{1d} \\ &\vdots \\ \dot{x}_{nd} &= x_{(n-1)d} u_{1d}, \end{aligned} \tag{2}$$

where u_{1d} and u_{2d} are reference control inputs. Now, consider the following dynamic extension of system (1) in order to solve the tracking control problem:

$$\begin{aligned} \dot{u}_1 &= f_1 \\ \dot{u}_2 &= f_2, \end{aligned} \tag{3}$$

where $\dot{u}_1$ and $\dot{u}_2$ are dynamic extension of the system and f_1, f_2 are adjustable inputs of the dynamical model. Let $x_e = x - x_d$ be the tracking error. The tracking error dynamics can be obtained as:

$$\begin{aligned} \dot{x}_{1e} &= u_1 - u_{1d} \\ \dot{x}_{2e} &= u_2 - u_{2d} \\ \dot{x}_{3e} &= x_{2e} u_{1d} + x_2 (u_1 - u_{1d}) \\ &\vdots \\ \dot{x}_{ne} &= x_{(n-1)e} u_{1d} + x_{n-1} (u_1 - u_{1d}) \\ \dot{u}_1 &= f_1 \\ \dot{u}_2 &= f_2. \end{aligned} \tag{4}$$

The objective is to design control inputs, f_1 and f_2, such that the states of the system track the desired dynamics of the reference system (2) and control strategy makes the tracking error to reach zero in a short period of time.

3. The Main Results

3.1. Recursive Terminal Sliding Mode Control

In this section, the TSMC method is considered for controlling a NH system such that the tracking error converges to zero in finite time. For this purpose, (4) is considered as two subsystems. The first subsystem can be described by:

$$\begin{aligned} \dot{x}_{1e} &= u_1 - u_{1d} \\ \dot{u}_1 &= f_1. \end{aligned} \tag{5}$$

The first subsystem (5) has been described in the canonical form of nonlinear SISO system. To design the fast terminal sliding control law for this subsystem, according to the conventional procedure of TSMC studied in the existing references such as [2, 21, 34-36], recursive sliding surfaces should be defined as:

$$\begin{aligned} s_0 &= x_{1e} \\ s_1 &= \dot{s}_0 + \beta s_0^{q/p}, \end{aligned} \tag{6}$$

where p and q are odd positive integers satisfying $q < p$, and β is a positive constant. For the first subsystem, the recursive sliding surfaces are selected as (6) and the control law is taken as:

$$f_1 = \dot{u}_{1d} - \beta \frac{q}{p} x_{1e}^{\frac{q}{p}-1} (u_1 - u_{1d}) - k_1 (u_1 - u_{1d} + \beta x_{1e}^{\frac{q}{p}})^{\frac{q'}{p'}}, \tag{7}$$

where $k_1 > 0$ and p' and q' are odd positive integers such that $q' < p'$. Therefore:

$$\begin{aligned} \dot{s}_1 &= \ddot{s}_0 + \frac{d}{dt}(\beta s_0^{q/p}) \\ &= \ddot{s}_0 + \beta \frac{q}{p} s_0^{q/p-1} \dot{s}_0 \\ &= \ddot{x}_{1e} + \beta \frac{q}{p} s_0^{q/p-1} \dot{x}_{1e} \\ &= \frac{d}{dt}(u_1 - u_{1d}) + \beta \frac{q}{p} s_0^{q/p-1} (u_1 - u_{1d}) \\ &= f_1 - \dot{u}_{1d} + \beta \frac{q}{p} s_0^{q/p-1} (u_1 - u_{1d}). \end{aligned} \tag{8}$$

Thus, using the control input in the form of (7), we have:

$$\dot{s}_1 + k_1 s_1^{q'/p'} = 0. \tag{9}$$

It can be proven that the solution s_1 of this equation will reach zero in finite-time, i.e., the equilibrium $s_1 = 0$ is a fast terminal attractor. Based on the recursive structure of (6), $s_0 = 0$ will be reached in finite time and thereby the state errors, x_{1e} will reach zero in finite time. It means that the system state trajectory starting from any initial state will reach the sliding mode $s_1 = 0$ in finite-time and $u_1 = u_{1d}$. Therefore, the convergence of tracking error for the subsystem (5) will be guaranteed.

In order to overcome the singularity problem in this recursive method, the initial value should be defined carefully to avoid trajectory from reaching $s_i = 0$ $(i = 1,\ldots, j-1)$ before $s_j = 0$ is reached. Hence, if the initial value $x_{1e}(0)$ and $u_1(0)$ is defined such that $s_0 > 0$ and $s_1 > 0$, then the switching manifolds s_1 and s_0 reach zero sequentially and the singularity problem will not occur.

The second subsystem can be described as follows:

$$\begin{aligned} \dot{x}_{2e} &= u_2 - u_{2d} \\ \dot{x}_{3e} &= x_{2e} u_{1d} + x_2 (u_1 - u_{1d}) \\ &\vdots \\ \dot{x}_{ne} &= x_{(n-1)e} u_{1d} + x_{n-1} (u_1 - u_{1d}) \\ \dot{u}_2 &= f_2. \end{aligned} \tag{10}$$

By changing variables of the second subsystem, the system can be written as classical form so that the TSMC design can be applied.

Considering $y_1 = x_{ne}$, $y_2 = x_{(n-1)e}$, ... , $y_{n-1} = x_{2e}$, $y_n = u_2 - u_{2d}$, and assuming $u_1 = u_{1d}$, the subsystem (10) can be transformed into the following form:

$$\begin{aligned} \dot{y}_1 &= u_{1d} y_2 \\ \dot{y}_2 &= u_{1d} y_3 \\ &\vdots \\ \dot{y}_{n-1} &= y_n \\ \dot{y}_n &= f_2. \end{aligned} \tag{11}$$

For this system, the following recursive fast terminal sliding surface structure is used to design the control law f_2 [35]:

$$\begin{aligned} s_0 &= y_1 \\ s_1 &= \dot{s}_0 + \beta_1 s_0^{q_1/p_1} \\ &\vdots \\ s_{n-1} &= \dot{s}_{n-2} + \beta_{n-1} s_{n-2}^{q_{n-1}/p_{n-1}}, \end{aligned} \tag{12}$$

where p_i and q_i are odd positive integers satisfying $q_i < p_i$ and β_i are positive constants $(i = 1,\ldots, n-1)$. Equation (12) can be rewritten as:

$$\begin{aligned} &s_0 = y_1 \\ &s_1 = \dot{s}_0 + \beta_1 s_0^{q_1/p_1} \\ &s_2 = \ddot{s}_0 + \beta_1 \frac{d}{dt}(s_0^{q_1/p_1}) + \beta_2 s_1^{q_2/p_2} \\ &\vdots \\ &s_{n-1} = s_0^{(n-1)} + \sum_{i=1}^{n-1} \beta_i \frac{d^{n-i-1}}{dt^{n-i-1}} s_{i-1}^{q_i/p_i}. \end{aligned} \tag{13}$$

Taking derivative of s_{n-1}, we have:

$$\dot{s}_{n-1} = s_0^{(n)} + \sum_{i=1}^{n-1} \beta_i \frac{d^{n-i}}{dt^{n-i}}(s_{i-1}^{q_i/p_i}). \tag{14}$$

On the other hand, we have:

$$y_n = \dot{y}_{n-1} = \frac{\ddot{y}_{n-2}}{u_{1d}} = \cdots = \frac{y_1^{(n-1)}}{u_{1d}^{n-2}} = \frac{s_0^{(n-1)}}{u_{1d}^{n-2}}. \tag{15}$$

Therefore, $s_0^{(n)} = u_{1d}^{n-2}\dot{y}_n$, and (14) can be rewritten as:

$$\dot{s}_{n-1} = u_{1d}^{n-2}\dot{y}_n + \sum_{i=1}^{n-1} \beta_i \frac{d^{n-i}}{dt^{n-i}}(s_{i-1})^{q_i/p_i}, \tag{16}$$

where $\dot{y}_n = f_2$. Let

$$f_2 = -\frac{1}{u_{1d}^{n-2}}\left(\sum_{i=1}^{n-1} \beta_i \frac{d^{n-i}}{dt^{n-i}}(s_{i-1})^{q_i/p_i} + k_2 s_{n-1}^{q_n/p_n}\right), \tag{17}$$

where p_n and q_n are also odd positive integers such that $q_n < p_n$ and k_2 is a positive constant. Substituting this control law into (16), $\dot{s}_{n-1}$ becomes:

$$\dot{s}_{n-1} = -k_2 s_{n-1}^{q_n/p_n}. \tag{18}$$

Equation (18) is a fast terminal attractor and by solving (18), it can be shown that after a finite time, s_{n-1} and $\dot{s}_{n-1}$ will reach zero. Therefore, based on the recursive structure of (13), it can be proven that y_i, for $i = 1,\dots,n$ and thereby the state errors, x_{ie}, for $i = 2,\dots,n$ will reach zero in finite time and keep zero afterward as well as $u_2 = u_{2d}$.

The above analysis has been summarized as the following theorem.

Theorem 1. *For the system described by (4), for which the control inputs are considered as (7) and (17), the state errors, x_{ie}, for $i = 1,\dots,n$ will reach zero in finite time, i.e., $u_i = u_{id}$ for $i = 1, 2$ and the sliding manifolds are according to terminal attractors in (9) and (18).*

3.2. Robustness Analysis

Considering disturbance in the input f_2 of the second subsystem (4), the tracking error dynamics can be re-written as:

$$\begin{aligned} &\dot{x}_{1e} = u_1 - u_{1d} \\ &\dot{x}_{2e} = u_2 - u_{2d} \\ &\dot{x}_{3e} = x_{2e}u_{1d} + x_2(u_1 - u_{1d}) \\ &\vdots \\ &\dot{x}_{ne} = x_{(n-1)e}u_{1d} + x_{n-1}(u_1 - u_{1d}) \\ &\dot{u}_1 = f_1 + \Delta f_1 \\ &\dot{u}_2 = f_2 + \Delta f_2, \end{aligned} \tag{19}$$

where Δf_1 and Δf_2 are additional terms which may be due to the disturbance in the input of the system. We assume that $|\Delta f_1| \le M_1$ and $|\Delta f_2| \le M_2$ where M_1 and M_2 are positive bounds on disturbances.

The first subsystem can be described by:

$$\begin{aligned} &\dot{x}_{1e} = u_1 - u_{1d} \\ &\dot{u}_1 = f_1 + \Delta f_1. \end{aligned} \tag{20}$$

Moreover, by changing variables for the second subsystem, we have:

$$\begin{aligned} &\dot{y}_1 = u_{1d} y_2 \\ &\dot{y}_2 = u_{1d} y_3 \\ &\vdots \\ &\dot{y}_{n-1} = y_n \\ &\dot{y}_n = f_2 + \Delta f_2. \end{aligned} \tag{21}$$

Therefore, we have the following theorem.

Theorem 2. *For the system described by (19) with bounded disturbance $|\Delta f_1| \le M_1$ and $|\Delta f_2| \le M_2$, the control inputs considered as (7) and (17), the state error x_{1e} and also the state errors x_{ie}, for $i = 2,\dots,n$ will reach the neighborhood Ω_1 and Ω_2 of zero in finite time, respectively, and also the sliding manifolds for the subsystems are according to terminal attractors $\dot{s}_1 = -\gamma_1 s_1^{q'/p'}$ and $\dot{s}_{n-1} = -\gamma_2 s_{n-1}^{q_n/p_n}$, respectively, where:*

$$\gamma_1 = k_1 - \frac{(\Delta f_1)}{s_{n-1}^{q'/p'}}$$

$$\gamma_2 = k_2 - \frac{(\Delta f_2)u_{1d}^{n-1}}{s_{n-1}^{q_n/p_n}}$$

$$k_1 = \frac{M_1}{|s_1|^{q'/p'}} + \eta_1, \quad \eta_1 > 0$$

$$k_2 = \frac{M_2 |u_{1d}|^{n-1}}{|s_{n-1}|^{q_n/p_n}} + \eta_2, \quad \eta_2 > 0 \tag{22}$$

$$\Omega_1 = \left\{ y : |s_{n-1}| \le \left(\frac{M_1}{k_1} \right)^{p'/q'} \right\}$$

$$\Omega_2 = \left\{ y : |s_{n-1}| \le \left(\frac{M_2 |u_{1d}|^{n-1}}{k_2} \right)^{p_n/q_n} \right\}.$$

Proof. For the first subsystem described by (20), taking derivate of s_1 defined in (6), similar to (8), we can obtain:

$$\begin{aligned} \dot{s}_1 &= \ddot{s}_0 + \frac{d}{dt}(\beta s_0^{\frac{q}{p}}) \\ &= \frac{d}{dt}(u_1 - u_{1d}) + \beta \frac{q}{p} s_0^{\frac{q}{p}-1}(u_1 - u_{1d}) \\ &= f_1 - \dot{u}_{1d} + \beta \frac{q}{p} s_0^{\frac{q}{p}-1}(u_1 - u_{1d}) + \Delta f_1. \end{aligned} \tag{23}$$

Considering f_1 as (7), it can be obtained that

$$\dot{s}_1 = -k_1 s_1^{q'/p'} + \Delta f_1. \tag{24}$$

Thus:

$$\dot{s}_1 = -\gamma_1 s_1^{q'/p'}. \tag{25}$$

To prove the fast terminal convergence, we must prove that $\gamma_1 > 0$. According to the condition (22), we have:

$$\begin{aligned} \gamma_1 &= \eta_1 + \frac{M_1}{|s_1|^{q'/p'}} - \frac{\Delta f_1}{s_1^{q'/p'}} \\ &\ge \eta_1 + \frac{M_1}{|s_1|^{q'/p'}} - \frac{|\Delta f_1|}{|s_1|^{q'/p'}} \ge \eta_1 > 0. \end{aligned} \tag{26}$$

Therefore, the conditions in (22) guarantee that $\gamma_1 \ge \eta_1 > 0$. Thus, the error will reach the region Ω_1 constrained by $|s_1| \le (M/k)^{q'/p'}$ in finite time.

For the second subsystem described by (21), defining differentiation of s_{n-1} along its variables, we have:

$$\dot{s}_{n-1} = s_0^{(n)} + \sum_{i=1}^{n-1} \beta_i \frac{d^{n-i}}{dt^{n-i}}(s_{i-1}^{q_i/p_i}). \tag{27}$$

Considering f_2 as (17), it can be obtained that:

$$\dot{s}_{n-1} = -k_2 s_{n-1}^{q_n/p_n} + u_{1d}^{n-1}(\Delta f_2). \tag{28}$$

Thus,

$$\dot{s}_{n-1} = -\gamma_2 s_{n-1}^{q_n/p_n}. \tag{29}$$

To be able to have the fast terminal convergence, we have to prove that $\gamma_2 > 0$. For this, according to the condition (22), we have:

$$\begin{aligned} \gamma_2 &= \eta_2 + \frac{M_2 |u_{1d}|^{n-1}}{|s_{n-1}|^{q_n/p_n}} - \frac{(\Delta f_2)u_{1d}^{n-1}}{s_{n-1}^{q_n/p_n}} \\ &\ge \eta_2 + \frac{M_2 |u_{1d}|^{n-1}}{|s_{n-1}|^{q_n/p_n}} - \frac{|\Delta f_2||u_{1d}^{n-1}|}{|s_{n-1}|^{q_n/p_n}} \ge \eta_2 > 0. \end{aligned} \tag{30}$$

It means that the conditions in (22) guarantee that $\gamma_2 \ge \eta_2 > 0$. Thus, it can be seen that the error will reach the region Ω_2 constrained by $|s_{n-1}| \le (M_2 |u_{1d}|^{n-1} / k_2)^{q_n/p_n}$ in finite time. Hence the proof of theorem has been completed.

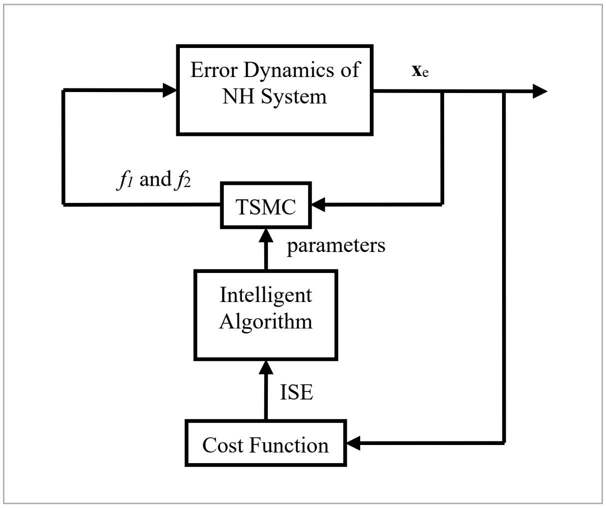
In the proposed controllers in (7) and (17), instead of sign function, k_1 and k_2 have to be selected as (22) in Theorem 2 in order to guarantee the robustness. Similar to switching control inputs like sign functions, it has been proven in Theorem 2 that the proposed method can compensate the effects of disturbances and the state errors will reach the neighborhood of zero in finite time. In fact, by replacing k_1 and k_2 in (22), into (7) and (17), it can be seen that these controllers act similar to sign function and can suppress the disturbances. In this method, the existence of η_1 and η_2 plays an important role to reduce the chattering. On the other hand, increasing the parameters k_1 and k_2 leads to increasing the gains γ_1 and γ_2 which causes faster convergence of terminal attractors $\dot{s}_1 = -\gamma_1 s_1^{q'/p'}$ and $\dot{s}_{n-1} = -\gamma_2 s_{n-1}^{q_n/p_n}$.

3.3. Intelligent Tuning of Parameters

The error dynamics of the NH system reach zero in a finite period of time if the parameters are designed properly. Therefore, the design parameters should be chosen carefully, otherwise, it may take very long time for tracking error to reach zero or tracking error may fluctuate with the amplitude around zero without reaching it, as it will be shown in simulation results presented in Section 5. We separately utilize four intelligent optimization algorithms, namely, DE, BO, BFO and CO to design these parameters such that an enhanced tracking error can be achieved. Figure 1 shows the block diagram of the main problem.

Figure 1

A block diagram of the proposed method.



In this paper, our goal is to design the parameters p_i, q_i $(i=1,...,n)$, β_i $(i=1,...,n-1)$ and k_2 considering the constraints mentioned above for each parameter by using evolutionary algorithms such that the tracking errors reach zero in finite time. In the following subsections, it will be illustrated how such algorithms work.

3.3.1. Differential Evolution (DE) Algorithm

DE algorithm has been proposed by Storn and Price in 1995 [31]. They have used this algorithm to solve continuous optimization problems, and recently it has been used for integer, discrete and other type of engineering problems. This algorithm has many similarities with other intelligent algorithms, such as genetic algorithm (GA). However, the main difference is its exclusive way to produce new population.

Three main operators in this algorithm are crossover, mutation and selection. DE algorithm changes the ordering of operators used in this algorithm. In addition, the way of using mutation is exclusive. The algorithm is explained in the following order:

Parameter Selection. The problem formulation includes the objective function, the unknown variables or parameters and decision parameters which have to be defined at the first stage. The ranges of unknown parameters have to be also determined. In addition, the parameters of the algorithm such as the population size and scaling as well as crossover factors have to be chosen.

Initialization. The initial population with N_p vectors is produced randomly in the acceptable range defined for parameters:

$$\begin{aligned}&\mathbf{x}_{j,g}=\mathbf{b}_{j,L}+rand_j(0,1)\times(\mathbf{b}_{j,U}-\mathbf{b}_{j,L})\\&\text{for } j=0,1,...,N_p-1\end{aligned} \tag{31}$$

where $\mathbf{b}_{j,L}$ and $\mathbf{b}_{j,U}$ are lower and upper limit vectors for unknown parameters, respectively. $\mathbf{x}_{j,g}$ is the base vector, g denotes the generation and $rand_j(0,1)$ produces a random number with normal distribution in the interval of (0,1].

Mutation. Differential mutation is a difference vector that is being sampled randomly from the initial population. If $\mathbf{v}_i$ is the target vector, the mutation vector is produced with the following equation:

$$\mathbf{v}_{i,g}=\mathbf{x}_{r_0,g}+F\times(\mathbf{x}_{r_1,g}-\mathbf{x}_{r_2,g}), \tag{32}$$

where $\mathbf{x}_{r_0,g}$ (base vector) and $\mathbf{x}_{r_1,g}-\mathbf{x}_{r_2,g}$ (difference vector) are selected randomly from initial population considering $r_0\neq r_1\neq r_2\neq i$. Moreover, F is a scaling factor which is a strictly positive real number that varies in the interval (0 , 1].

Crossover. Each current vector combines with mutation vector and produces a new temporary response. In this paper, we use uniform crossover with the following equation:

$$\mathbf{u}_{j,g}=\begin{cases}\mathbf{v}_{j,g} & \text{if } rand_j(0,1)\leq C_r \text{ or } j=j_{\text{rand}}\\ \mathbf{x}_{j,g} & \text{O.W.}\end{cases}, \tag{33}$$

where $C_r \in [0, 1]$ is user defined crossover probability and $\mathbf{u}_{j,g}$ is called trial vector. In fact, (33) implies that trial vector is constructed at the randomly chosen parameter index, j_{rand}, which implies that the trial vectors are inherited from the mutant vector until $rand_j\,(0,1) \le C_r$. The first time that $rand_j\,(0,1) \le C_r$, all the remaining parameters are obtained from the target vector.

Selection. We select the next generation with the following equation. It means that if the value of the objective function, evaluated with the trial vector, was better than the one evaluated with the target vector, in the next generation, the target vector is replaced with the trial vector, otherwise we keep the previous one:

$$\mathbf{x}_{j,g+1} = \begin{cases} \mathbf{u}_{j,g} & \text{if } f(\mathbf{u}_{j,g}) \le f(\mathbf{x}_{j,g}) \\ \mathbf{x}_{j,g} & \text{O.W.} \end{cases}, \tag{34}$$

where f denotes the objective function.

Termination Criteria. Many conditions can be used for stopping the algorithm. In this paper, we used a stopping criterion based on maximum iteration (g_{max}).

3.3.2. Bat Optimization (BO) Algorithm

Another subcategory of swarm intelligence algorithms is BO algorithm that has been presented by Yang in 2010 [39]. This algorithm has been proposed based on the sound reflection of special groups of bats called micro-bats that have forearm length between 2.2-11 centimeters. These bats can find their prey even in environments with complete darkness. They send out a very loud sound pulse and listen the echo returning from the surrounding objects. Usually, it takes about 300-400 microseconds to integrate the received signals. They follow louder sounds when they are looking for prey. The complete steps of the BO algorithm in simulation are described as follows:

Step1. Initialize the number of bats (n), decision variable (d), initial velocity vector (v_i), define admissible interval for frequency [$f_{\min}$, $f_{\max}$], allocate the initial loudness, A_0, in (1, 2) and initial emission rate, r_0, in (0, 1), for each bat and also produce initial bat population vector x_i in search space randomly as:

$$\mathbf{x}_i = \mathbf{x}_{\text{low}} + \text{rand}(\mathbf{x}_{\text{high}} - \mathbf{x}_{\text{low}}). \tag{35}$$

Step2. Evaluate objective function and determine the global best, $\mathbf{x}_{\text{best}}$.

Step3. Update the frequency, velocity and position:

$$\begin{aligned} f_i &= f_{\min} + \beta(f_{\max} - f_{\min}) \\ \mathbf{v}_i &= \mathbf{v}_i + f_i(\mathbf{x}_i - \mathbf{x}_{\text{best}}) \\ \mathbf{x}_i &= \mathbf{x}_i + \mathbf{v}_i, \end{aligned} \tag{36}$$

where $\beta \in [0, 1]$ is a random number.

Step4. Produce a random number ($rand$) with uniform distribution between 0 and 1. If $rand > r_i$, then do local search, i.e.:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \varepsilon \mathbf{A}, \tag{37}$$

where $\varepsilon \in$ [-1, 1] and $\mathbf{A}$ is the average loudness of all the bats at this time step.

Step5. Evaluate the objective function.

Step6. Produce a random number ($rand$) with uniform distribution between 0 and 1. If $rand < A_j$ and $f(\mathbf{x}_{\text{new}}) < f(\mathbf{x}_{\text{best}})$, then accept the new solution and update A_i, r_i as:

$$\begin{aligned} A_i &= \alpha A_i \\ r_i &= r_i[1 - \exp(-\gamma t)], \end{aligned} \tag{38}$$

where $0 < \alpha < 1$ and $\gamma > 0$ are constant values.

If the above condition is not satisfied, rank the bats and find the best one.

Step7. Check the termination criteria. If it is not satisfied, go to step 3.

Step8. End.

3.3.3. Bacterial Foraging Optimization (BFO) Algorithm

BFO is an optimization algorithm based on swarm intelligence that has been presented by Passino in 2001 to solve continuous optimization problems [28]. The main idea of this algorithm is based on natural selection which actually tends to eliminate animals with poor "foraging strategies" and also spread those genes of animals that have successful foraging strategies. After many generations, poor foraging strategies were either eliminated or got better.

The algorithm is based on the foraging behavior of Escherichia coli (E. coli) bacteria which are present in human intestine. The E. coli bacteria that are present in human intestine have a foraging strategy governed by four processes, namely, chemotaxis, swarming, reproduction and elimination-and-dispersal. All the steps of this algorithm are explained as follows:

Initialization: In this step, the parameters of the algorithm, i.e., $s, p, N_s, N_c, N_{re}, N_{ed}, p_{ed}$ that are respectively the number of bacteria, problem dimensions, number of swims, number of chemotactic steps, reproduction steps, elimination-dispersal steps, elimination-dispersal probability are initialized. Moreover, $C(i)$, for i=1,2,...,s, is defined as chemotactic step size of each bacteria and θ_i is considered as initial position of all bacteria.

Main Loop: The main loop consists of the following steps:

Step1. Elimination-dispersal loop: $l = l + 1$,

Step2. Reproduction loop: $k = k + 1$,

Step3. Chemotactic loop: $j = j + 1$,

Step4. For each bacterium, $i = 1, 2, \ldots, s$ take the chemotactic step as follows:

- Calculate the evaluation function and add the attractant effect to the nutrient concentration:

$$\begin{aligned} J(i,j,k,l) &= J(i,j,k,l) \\ &\quad + J_{cc}\left(\theta_i(j,k,l), P(j,k,l)\right) \\ J_{\text{last}} &= J(i,j,k,l), \end{aligned} \tag{39}$$

where J is the objective function that has to be minimized, and J_{cc} is the cell-to-cell attractant effect to the nutrient concentration. In addition, P represents the position of each member in the population of s bacteria at the j-th chemotactic step, k-th reproduction step, and l-th elimination-dispersal event.

- Tumble: Generate a random vector, $\Delta(i)$, on [-1, 1].
- Move: Obtain new position by:

$$\theta_i(j+1,k,l) = \theta_i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}. \tag{40}$$

- For this position, evaluate the objective function $J(i, j+1, k, l)$.
- Swim:

Let $m = 0$ (counter for swim length).

While $m < N_s$:

Let $m = m + 1$.

If moving condition is satisfied, i.e.,

$J(i, j+1, k, l) < J_{\text{last}}$, then move and let

$J_{\text{last}} = J(i,j+1,k,l)$.

Otherwise and also if $m = N_s$, swimming loop is done.

- If $i \neq s$, go to the next bacterium and repeat the above steps.
- If $j < N_s$, go to step 3; otherwise moving loop ends.

Step5. Reproduction:

a For the given k, l and for each bacterium, sort bacteria based on the objective evaluation function.

b A half of the bacteria with worst value dies and the other half with better value is divided into two parts in the same position.

c If $k < N_{re}$, go to step 2.

d Otherwise, end the reproduction loop.

Step6. Elimination-dispersal loop: $l = l + 1$.

a If condition of elimination and dispersal is satisfied, the bacterium is eliminated and replaced with another bacterium that is produced randomly.

b If $l < N_{ed}$, go to step 1.

c Otherwise, end the elimination-dispersal loop.

3.3.4. Cuckoo Optimization (CO) Algorithm

In 2009, research about cuckoo search has been developed by Yang and Deb [40]. Since then, CO algorithm has been presented by Rajabioun in 2011 [30]. The main reason that motivates researches to develop this optimization algorithm is that cuckoo has a different life style in comparison with other birds. CO algorithm starts with random initial population similar to the other optimization algorithms. Each cuckoo lays a random number of eggs. Hence, the population is divided into two types, cuckoos and eggs. Some kind of effort occurs between this population and the better cuckoos survive in this competition. These better cuckoos immigrate to a better environment, start reproduction and laying eggs. This cycle continues until the stopping criteria are satisfied. Steps of this algorithm are as follows:

Step1. Generate the initial habitats for cuckoos randomly in the search space.

Step2. Dedicate a random number of eggs to each of the initial cuckoo habits (in nature it is between 5 and 20).

Step3. Define the egg laying radius (ELR) for each cuckoo with the following equation:

$$ELR = \alpha \times \frac{\text{Number of current cuckoo's eggs}}{\text{Total number of eggs}} \times (\text{var}_{\text{hi}} - \text{var}_{\text{low}}), \quad (41)$$

where var_{hi} and var_{low} are, respectively, the upper and lower limits of decision variables and α is an integer used to handle the maximum value of ELR.

Step4. Each cuckoo lays a random number of eggs in other host birds' nest inside its corresponding ELR.

Step5. Those eggs that are verified because of lesser similarity to the host birds' eggs will be killed by the host birds.

Step6. Let eggs hatch and grow. The first-born cuckoo ruins other eggs because of its three times larger body; it eats more food and other chicks die from starvation. Hence, we have only one cuckoo remaining in each nest in the end.

Step7. Evaluate the habitat of each newly grown cuckoo.

Step8. Determine the maximum number of cuckoos that can live in each environment and kill those that live in the worst habitats.

Step9. Cluster cuckoos, find the best group and select the best of it as a target habitat.

Step10. Immigrate toward the target habitat:

$$\mathbf{x}_{\text{next}} = \mathbf{x}_{\text{current}} + F(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{current}}), \quad (42)$$

where F is a parameter defined between 0 and 1.

Step11. If stopping condition is not satisfied, go to Step 2.

Step12. End.

3.3.5. Parameters for Adjustment

To determine the parameters of terminal sliding surface with intelligent optimization algorithms, we firstly put all the parameters needed to be designed in the following vector:

$$Param = [\beta_1, \beta_2, ..., \beta_{n-1}, k_2, q_1, p_1, ..., q_n, p_n]. \quad (43)$$

The algorithms firstly randomly produce the initial population. Each member of the population contains the vector above. Parameters in the vector must satisfy the constraints, in which p_i and q_i (i = 1, 2, ... , n) are odd positive integers such that $q_i < p_i$ and β_i (i = 1, 2, ... , n-1) and k_2 are certain positive constants. For each set of parameters shown in the form of the vector above, we can define the control law f_2 as in (17). By applying the control law to the system, error tracking dynamics (x_{ie}, for i = 2, 3, ... , n) can be obtained. Having these errors, we can define the integral square error (ISE) as an objective function with the following equation:

$$\text{ISE} = \int_{t_0}^{t} (\sum_{i=2}^{n} x_{ie}^2(\tau)) d\tau. \quad (44)$$

In order to choose the parameters that are determined by the intelligent optimization algorithm, we make a change in this objective function. For tracking purposes, occurring peaks at the beginning are prevalent, while peaks at the end are not desirable. In order to avoid the integration of these two peaks in the objective function, they should be separated. Therefore, we consider the integration of the square error from a short time after the beginning. In fact, for simulation, we take t_0 = 0.4 and t = 10 in (44).

The value of ISE obtained from (44) is allocated to each parameter vector. It is obvious that the vector with less value of the allocated ISE is more successful in the algorithm. We continue the algorithm until the algorithm gets us the parameter vector with the best objective function value.

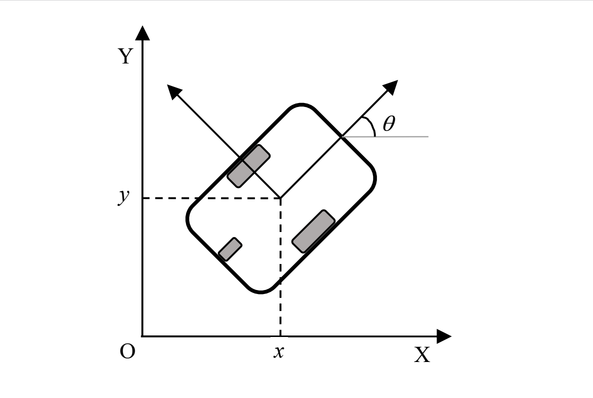
4. Mathematical Model of Wheeled Mobile Robots (WMRs)

WMRs are one of the applications of the NH systems which are especially being used in environments in which a motion on smooth surfaces such as shopping centers, hospitals and industrial areas is needed. Furthermore, it plays an important role in security, transportation inspecting, painting, soccer playing robots and other aspects. In fact, the main control problem in such systems is to force systems to move along a desired trajectory. In this paper, a nonholonomic WMR with two driving wheels and one passive wheel in its behind is considered, as shown in Figure 2.

The nonholonomic constraint for WMR is due to the perfect rolling constraint in which driving wheels must purely roll without any longitudinal or lateral

Figure 2

A nonholonomic WMR with two driving wheels and a passive wheel



slipping. In other words, the WMR can only move in a direction which is perpendicular to the axis of the driving wheels. Therefore, we have:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0, \quad (45)$$

where $[x\ y\ \theta]^T$ represents the position of the WMR in the Cartesian space.

It has been well known that, under the assumption of pure rolling, state space equations of WMR with two driving wheels and a passive wheel in the behind is described by:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}-\sin\theta & 0\\ \cos\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}$$
$$\begin{bmatrix}\dot{v}\\ \dot{\omega}\end{bmatrix} = M^{-1}\begin{bmatrix}T_1\\ T_2\end{bmatrix}, \quad (46)$$

where v is the linear velocity of the wheel and ω is the angular velocity around the vertical axis. M is a symmetric positive matrix. T_1 is the pushing force in the direction of heading angle and T_2 is the torque about the vertical axis for steering. It is assumed that v and ω are the available control inputs, which can be easily calculated from torques of the motor driving wheels.

In order to use the proposed TSMC design method, we convert the equations of the system into extended chained form of the NH systems considering the following states and control transformation:

$$\begin{aligned} x_1 &= \theta\\ x_2 &= -x\sin\theta + y\cos\theta\\ x_3 &= x\cos\theta + y\sin\theta\\ u_1 &= \omega\\ u_2 &= v - x_3\omega, \end{aligned} \quad (47)$$

where (x_2, x_3) denotes the coordinates of the center of mass, x_2 is the axis aligned with the vehicle orientation. Moreover, u_1 and u_2 are the control inputs which depend on the pushing force T_1, steering torque T_2 and linear and angular velocity as defined in (48):

$$\begin{bmatrix}\dot{u}_1\\ \dot{u}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1 & -x_3\end{bmatrix} M^{-1}\begin{bmatrix}T_1\\ T_2\end{bmatrix} + \begin{bmatrix}0\\ -x_2\end{bmatrix}\omega^2. \quad (48)$$

Using above definition, state space equations can be described as:

$$\begin{aligned} \dot{x}_1 &= u_1\\ \dot{x}_2 &= u_2\\ \dot{x}_3 &= x_2 u_1\\ \dot{u}_1 &= f_1\\ \dot{u}_2 &= f_2, \end{aligned} \quad (49)$$

where f_1 and f_2 are the adjustable control inputs of the dynamical model. These third order equations illustrate a NH system in the extended chained form same as equation (1) with n=3. Now assume that the desired trajectory for the third-order NH system, $x_d = [x_{1d}, x_{2d}, x_{3d}]^T$, is generated by:

$$\begin{aligned} \dot{x}_{1d} &= u_{1d}\\ \dot{x}_{2d} &= u_{2d}\\ \dot{x}_{3d} &= x_{2d} u_{1d}, \end{aligned} \quad (50)$$

where u_{1d} and u_{2d} are the reference controls. Furthermore, we denote the tracking error as $x_e = x - x_d$ where x is the state vector. Tracking error for the third-order NH system, considering the dynamic extension, satisfies the following equations:

$$\begin{aligned} \dot{x}_{1e} &= u_1 - u_{1d}\\ \dot{x}_{2e} &= u_2 - u_{2d}\\ \dot{x}_{3e} &= x_{2e} u_{1d} + x_2(u_1 - u_{1d})\\ \dot{u}_1 &= f_1 + \Delta f_1\\ \dot{u}_2 &= f_2 + \Delta f_2. \end{aligned} \quad (51)$$

The first subsystem is controlled by f_1 as defined in (7). In the proposed method, the control task is to design f_2, as (17), using the intelligent TSMC method such that tracking error becomes zero in finite time. By using contexts stated in Section 3 for controlling second subsystem of generalized NH systems in extended chained form (51), system assuming $y_1 = x_{3e}$, $y_2 = x_{2e}$ and $y_3 = u_2 - u_{2d}$, we should convert the equations of third-order NH into the following equations:

$$\begin{aligned} \dot{y}_1 &= u_{1d} y_2 \\ \dot{y}_2 &= y_3 \\ \dot{y}_3 &= f_2 + \Delta f_2 . \end{aligned} \tag{52}$$

To solve these equations, we consider terminal sliding surface as:

$$\begin{aligned} s_0 &= y_1 \\ s_1 &= \dot{s}_0 + \beta_1 s_0^{\frac{q_1}{p_1}} \\ s_2 &= \dot{s}_1 + \beta_2 s_2^{\frac{q_2}{p_2}} \end{aligned} \tag{53}$$

and the control input as:

$$f_2 = -\frac{1}{u_{1d}}[\beta_1 \frac{d^2}{dt^2}(s_0^{q_1/p_1}) + \beta_2 \frac{d}{dt}(s_1^{q_2/p_2}) + k_2(s_2^{q_3/p_3})]. \tag{54}$$

The simulation results in the next section show that if we design parameters using intelligent optimization algorithms, then the tracking errors will reach and remain on zero in a short period of time.

5. Simulation Results

In this section, we show the results of computer simulation performed in MATLAB/SIMULINK. The desired trajectory is described as follows:

$$\begin{aligned} x_d &= t(\sin t + \cos t) \\ y_d &= t(\sin t - \cos t) \\ \theta_d &= t. \end{aligned} \tag{55}$$

For the first subsystem described by (20), by considering disturbance as $\Delta f_1 = 0.1\sin(t)$, the control law of the form (7) with parameters $q = q' = 3$, $p = p' = 5$, and $\beta = k_1 = 3$ has been applied to the system and the results have been shown in Figure 3. It can be seen that, using the defined control law in (7), the tracking error x_{1e} reaches zero in a few seconds and the input u_1 reaches the desired value in a finite time as well.

In Figure 4, the trajectory errors x_{2e} and x_{3e}, the input u_2 and control law f_2 for the second subsystem (52) have been shown when the parameters of terminal sliding surface are selected randomly and without using the intelligent algorithms. The parameters assumed in this simulation are q_1=45, p_1=117, q_2=23, p_2=123, q_3=3, p_3=5, β_1=26, β_2=54, and k_2=44. It can be observed that for the case that the parameters of terminal sliding surface are determined properly, the tracking error reaches zero in a finite time and it may fluctuate around zero with high amplitude.

Figure 3

a) Tracking error of x_{1e}, input u_1 of first subsystem (5), b) control law f_1 defined in (7)

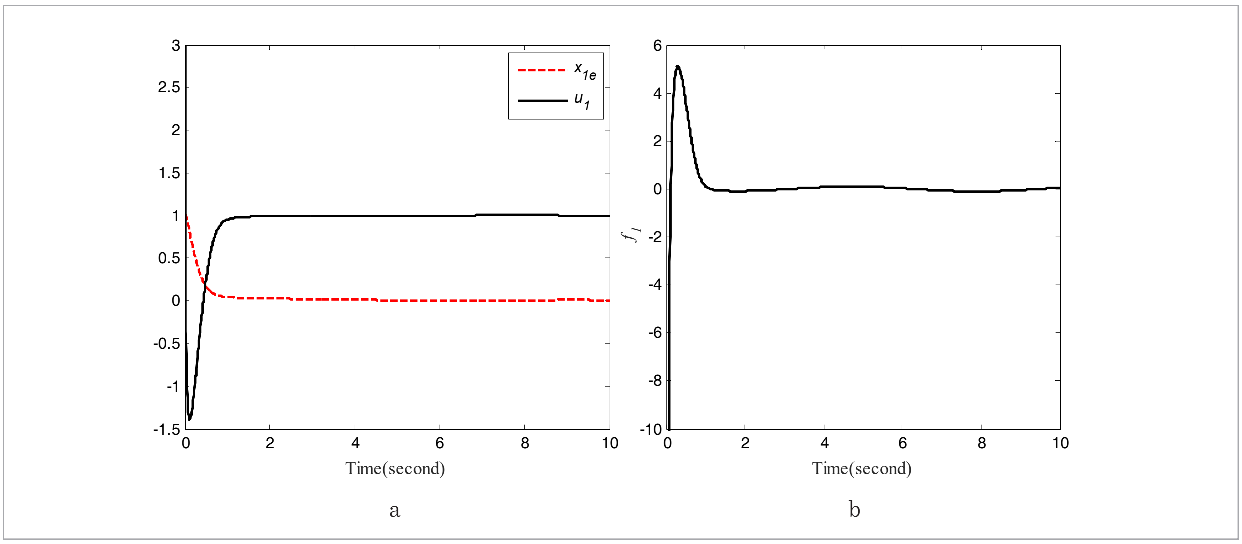
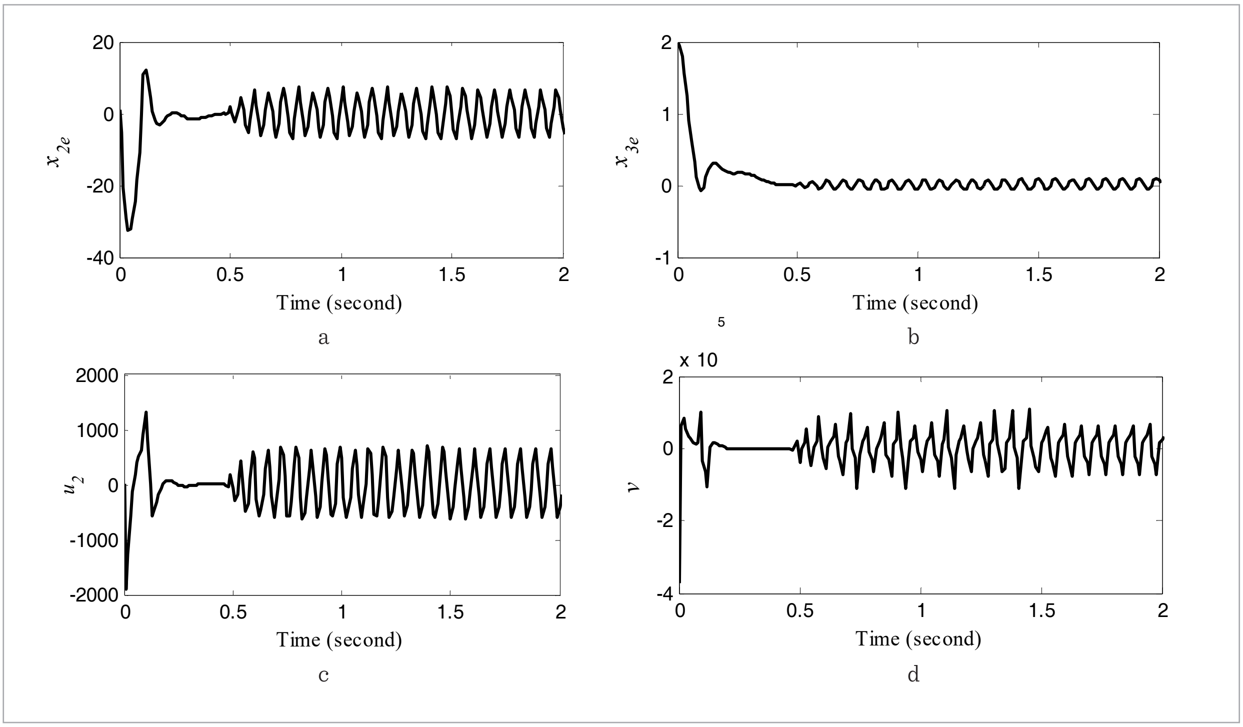


Figure 4

a) Tracking error of x_{2e}, b) tracking error of x_{3e}, c) dynamic control law u_2 and d) WMR linear velocity v when the parameters of terminal sliding surface structure are selected randomly



Now, we apply each algorithm presented in Section 3 to find the parameters of the terminal sliding mode surface structure proposed to control the second subsystem of WMR. For each algorithm, in order to compare the performance of the algorithms, the same objective function is used which is integral square error, ISE, defined in (44). Each algorithm needs its own preliminary parameters to be defined. These parameters are taken into consideration as discussed below:

DE: For simulation use, the population size is assumed as 50, and the scaling factor is considered randomly in the range of $0.2 < F < 0.8$ with uniform distribution. We additionally take $C_r = 0.2$ as the crossover probability.

BFO: For this algorithm, we assume that $s = 50$, $N_c = 10$, $N_s = 4$, $N_{re} = 4$, $N_{ed} = 2$ and $P_{ed} = 0.25$.

BO: We use 30 bats with $f_{\min} = 0$, $f_{\max} = 2$. Other parameters are selected randomly in the range stated in Section 3.

CO: For simulating this algorithm, a number of 20 cuckoos is used. There are between two to five eggs. For clustering, we use k-nearest-neighborhood (KNN) clustering method with three clusters. In addition, we take $\alpha = 5$, $F = 9$.

Considering disturbance for the second subsystem defined in (51) as $\Delta f_2 = \sin(t)$, simulation results of controlling second subsystem with TSMC tuned using these four algorithms and by considering the input disturbance have been depicted in Figures 5 and 6. In addition, Table 1 compares the results of applying the intelligent algorithms to the system by considering the number of function evaluations (NFE), consumed time for running the algorithms and the best cost of ISE.

It can be seen that by applying the TSMC in which the parameters of terminal sliding surface structure are chosen using the intelligent algorithms (Figures 5 and 6), the results are considerably improved compared with the case in which the parameters have been selected randomly (Figure 4). Simulation results show that the intelligent TSMC is robust against disturbance, i.e., applying disturbance to the system does not affect the tracking error dynamics of the system. By using the intelligent algorithm, the tracking errors

Figure 5

a) Tracking error of x_{2e}, b) tracking error of x_{3e}, c) dynamic control law u_2 and d) WMR linear velocity v when the parameters of the proposed method are selected using intelligent algorithms

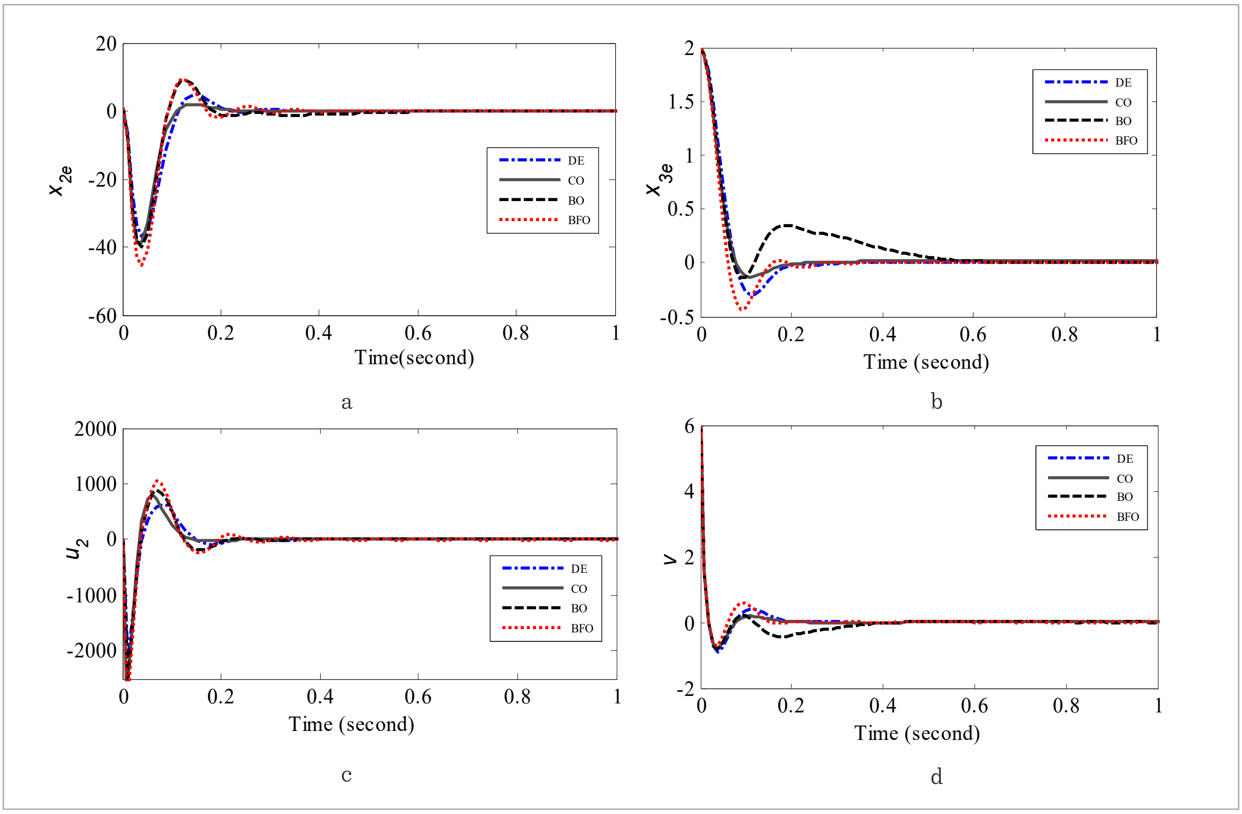


Figure 6

Control input f_2 when the parameters of TSMC are selected using intelligent algorithms

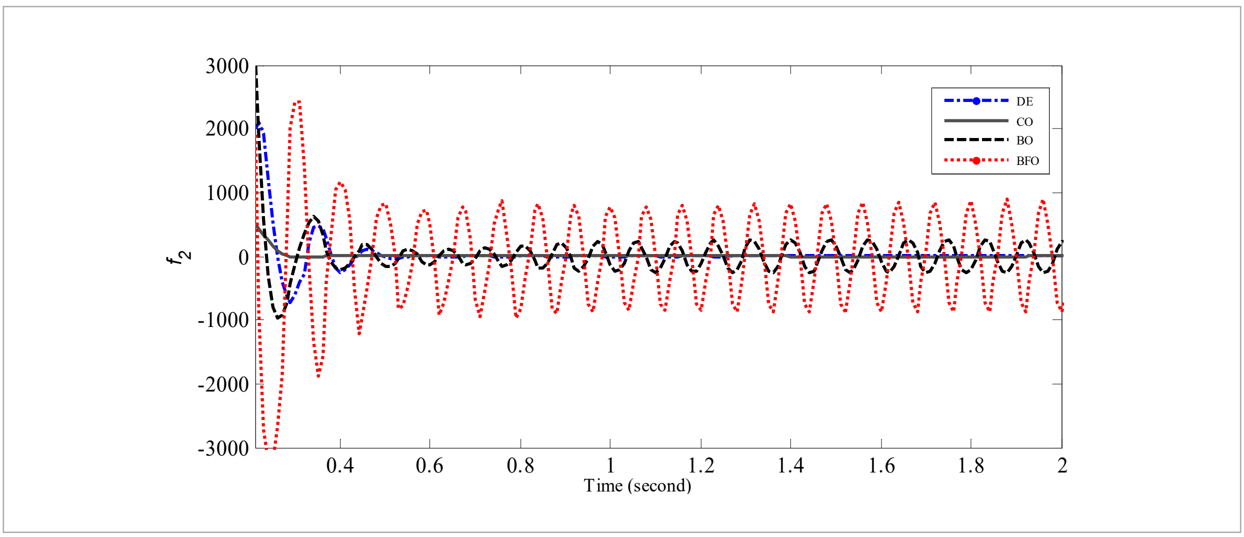


Table 1

The results of using intelligent algorithms with or without disturbance for WMR. Parameters are a vector with the form of $[\beta_1, \beta_2, k_2, q_1, p_1, q_2, p_2, q_3, p_3]$ that shows the simulation outputs of the algorithms

Algorithm	NFE	Time	Best Cost	Best Parameters
CO	3516	52 min	0.01138	[21.7, 10.6, 1.6, 49, 41, 65, 97,9, 39]
DE	2050	40 min	0.01139	[11.7, 20.8, 1, 177, 217, 169, 197, 5, 271]
BO	1230	24 min	0.0119	[14.3, 15, 54.1, 275, 299, 235, 267, 113, 177]
BFO	8000	3 h 54 min	0.04	[22, 16, 148, 219, 261, 219, 261,5, 131]

converge to zero very quickly in less than a second. In addition, the control effort has been reduced by using the intelligent control scheme.

In order to see how tuning parameters affects the trajectory tracking, Figures 7 to 10 show the effectiveness of each algorithm in tracking the desired trajectory. It

Figure 7

Tracking the desired trajectory of x by four algorithms

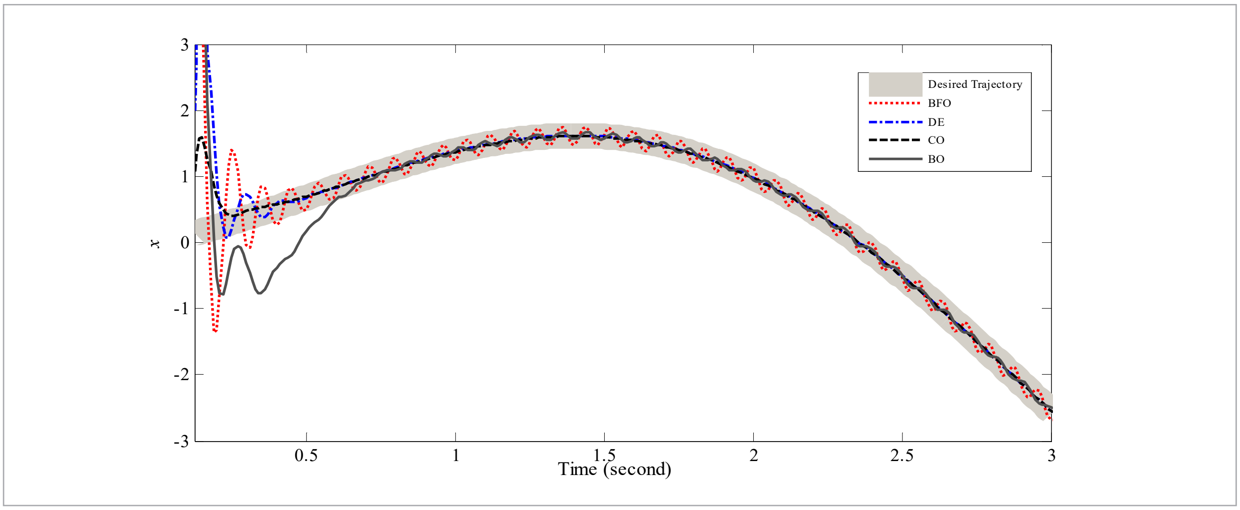


Figure 8

Tracking the desired trajectory of y by four algorithms

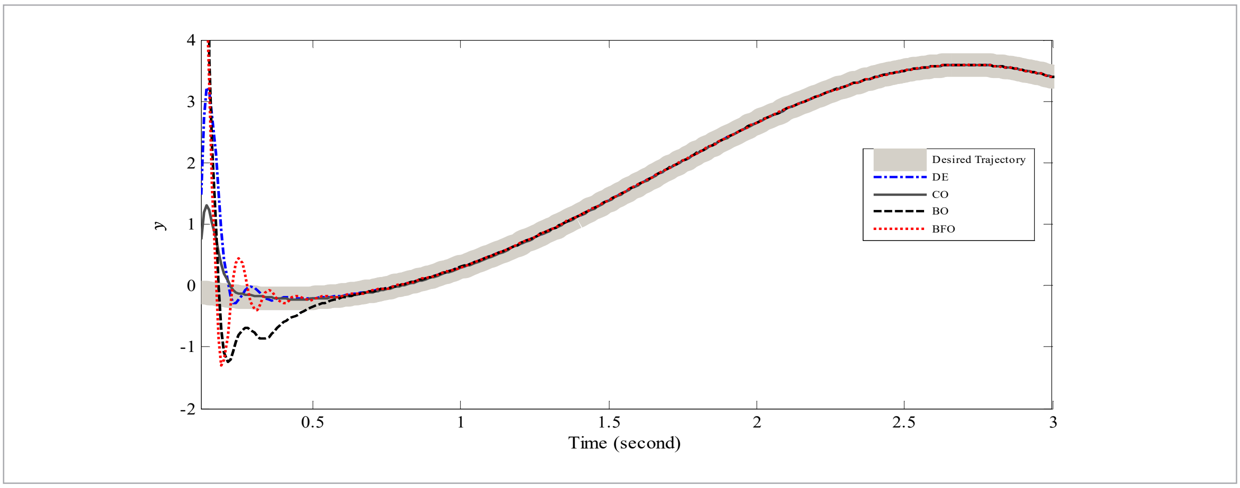


Figure 9

Tracking the desired trajectory of θ by four algorithms

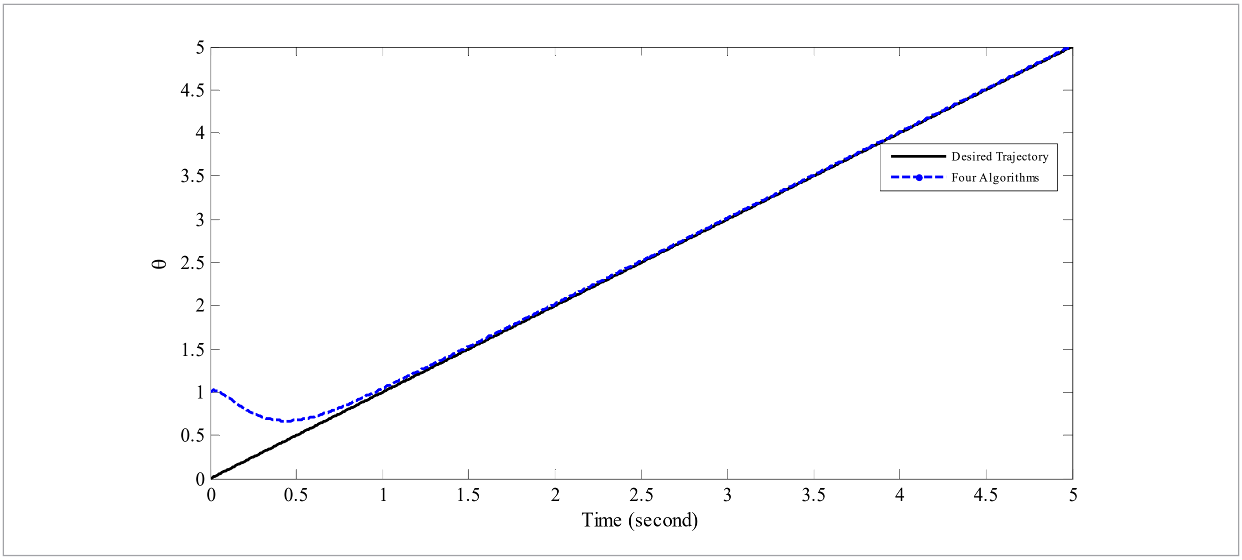
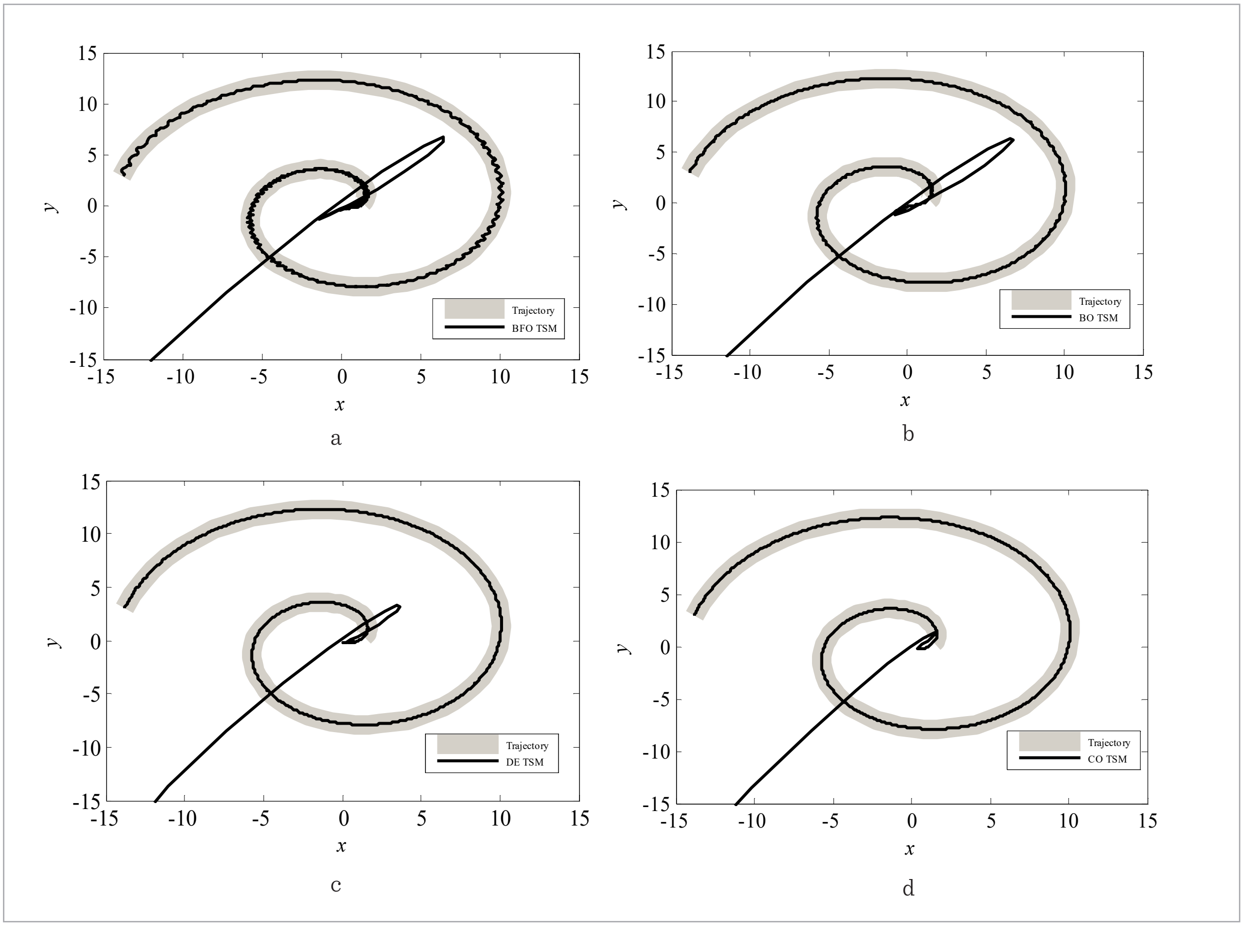


Figure 10

Trajectory tracking of WMR in Cartesian coordinate using four intelligent algorithms a) BFO, b) BO, c) DE, d) CO

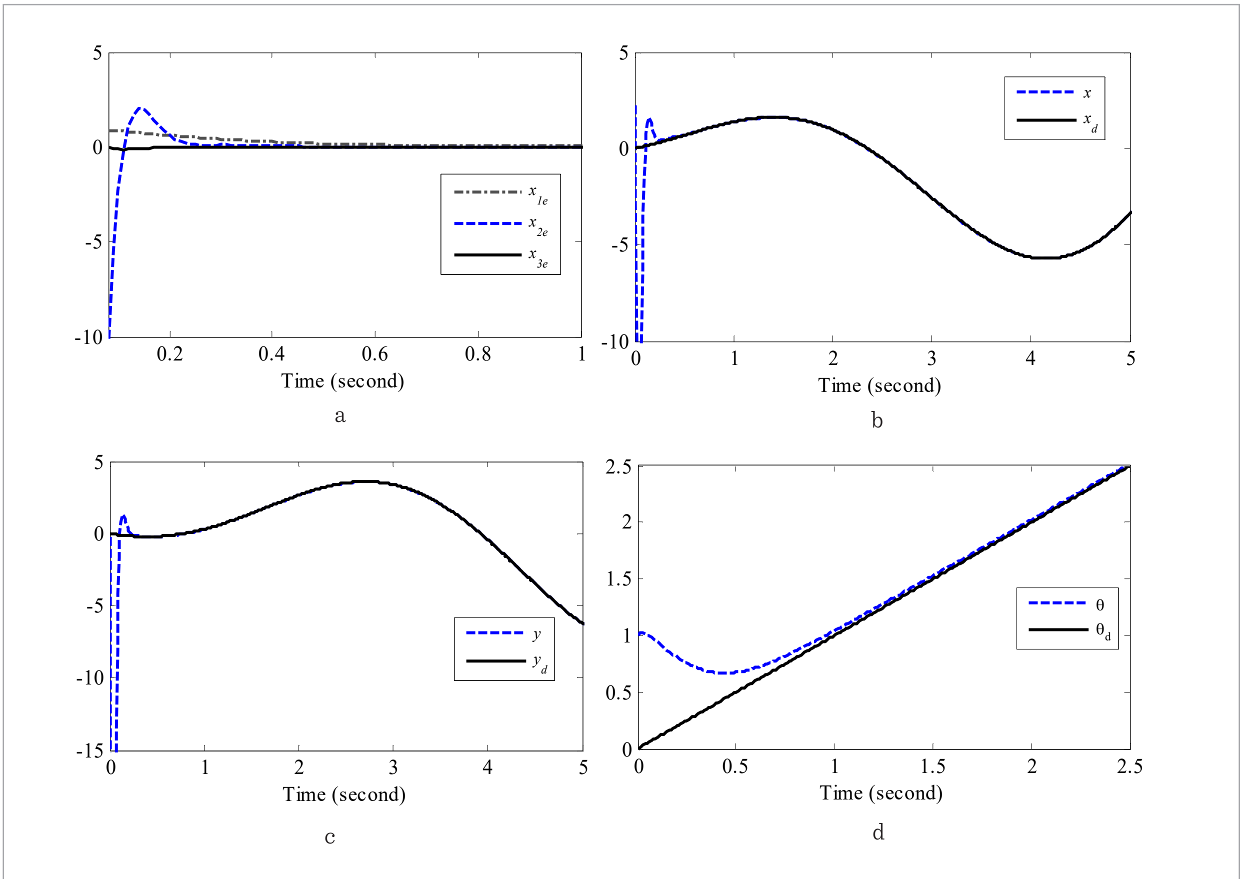


can be seen from these figures that using the parameters of TSMC obtained from BFO and BO algorithms, the desired trajectory is tracked with a little fluctuation. However, using DE and CO algorithms, the parameters are properly tuned and therefore the trajectory is accurately tracked in a short period of time.

Simulation results show that among these four algorithms, by using CO algorithm, the best parameter values can be obtained for trajectory tracking of WMR. Table 1 also reveals that BFO leads to a greater objective function and it takes much time compared with the others. Therefore, it could not find the best parameter values. However, CO algorithm tunes the parameters of terminal sliding surface such that leads to the least ISE as objective function. Thus, it can be concluded that CO algorithm is the best algorithm for tuning these parameters. The tracking error and the state variables, when CO algorithm is used for selecting the parameters, have been shown in Figure 11.

Figure 11

a) Error of transformed system x_{1e}, x_{2e}, x_{3e}, b) tracking of desired trajectory x, c) tracking of desired trajectory y, d) tracking of desired trajectory θ when the parameters of terminal sliding surface are selected using CO algorithm



6. Conclusion

In this paper, an intelligent TSMC has been proposed for controlling the NH systems in extended chained form. In this method, intelligent algorithms have been used to determine the parameters of the recursive terminal sliding surface structure designed for error dynamic tracking. Disturbance has also been considered in tracking dynamic error of the NH system. BFO, BO, DE and CO are the optimization algorithms, used for designing the parameters. It has been shown that, using these intelligent algorithms, appropriate parameters can be found such that the system can track the desired trajectory very well. Simulation results performed on WMR as a benchmark of the NH systems have shown the effectiveness of the proposed design for error dynamic tracking even in the presence of the input disturbance. In addition, simulation results show that CO algorithm has provided the most effective tuning of parameters among other algorithms such that the tracking error can reach zero in a very short period of time with a small amount of chattering.

Acknowledgments

We would like to present our thanks to anonymous reviewers for their helpful suggestions.

References

1. Astolfi, A. Discontinuous Control of Nonholonomic Systems. Systems and Control Letters, 1996, 27(1), 37-45. https://doi.org/10.1016/0167-6911(95)00041-0
2. Binazadeh, T., Shafiei, M. H. Nonsingular Terminal Sliding-Mode Control of a Tractor–Trailer System. Systems Science and Control Engineering, 2014, 2(1), 168-174. https://doi.org/10.1080/21642583.2014.890552
3. Bloch, A. M. Nonholonomic Mechanics and Control. Springer-Verlag, New York, 2003. https://doi.org/10.1007/b97376
4. Brockett, R. W. Asymptotic Stability and Feedback Stabilization. In: Brockett R. W., Millman R. S., Sussmann H. J. (Eds.), Differential Geometric Control Theory, Birkhauser, Boston, 181-191, 1983.
5. Chen, C. Y., Li, T. H. S., Yeh, Y. C. EP-Based Kinematic Control and Adaptive Fuzzy Sliding-Mode Dynamic Control for Wheeled Mobile Robots. Information Sciences, 2009, 179(1-2), 180-195. https://doi.org/10.1016/j.ins.2008.09.012
6. Chen, N., Song, F., Li, G., Sun, X., Ai, C. S. An Adaptive Sliding Mode Backstepping Control for the Mobile Manipulator with Nonholonomic Constraints. Communications in Nonlinear Science and Numerical Simulation, 2013, 18(10), 2885-2899. https://doi.org/10.1016/j.cnsns.2013.02.002
7. Chen, S. Y., Lin, F. J. Robust Nonsingular Terminal Sliding-Mode Control for Nonlinear Magnetic Bearing System. IEEE Transactions on Control Systems Technology, 2011, 19(3), 636-643. https://doi.org/10.1109/TCST.2010.2050484
8. Fang, Y., Yuan, Z., Fei, J. Adaptive Fuzzy Backstepping Control of MEMS Gyroscope Using Dynamic Sliding Mode Approach. Information Technology and Control, 2015, 44(4), 380-386. https://doi.org/10.5755/j01.itc.44.4.9110
9. Ferrara, A., Giacomini, L., Vecchio, C. Control of Nonholonomic Systems with Uncertainties via Second-Order Sliding Modes. International Journal of Robust and Nonlinear Control, 2008, 18(4-5), 515-528. https://doi.org/10.1002/rnc.1202
10. Jiang, Z. P. Robust Exponential Regulation of Nonholonomic Systems with Uncertainties. Automatica, 2000, 36(2), 189-209. https://doi.org/10.1016/S0005-1098(99)00115-6
11. Jin, M., Lee, J., Chang, P. H., Choi, C. Practical Nonsingular Terminal Sliding-Mode Control of Robot Manipulators for High-Accuracy Tracking Control. IEEE Transactions on Industrial Electronics, 2009, 56(9), 3593-3601. https://doi.org/10.1109/TIE.2009.2024097
12. Khooban, M. H. Design an Intelligent Proportional-Derivative (PD) Feedback Linearization Control for Nonholonomic-Wheeled Mobile Robot. Journal of Intelligent and Fuzzy Systems, 2014, 26(4), 1833-1843.
13. Kocamaz, U. E., Göksu, A., Taşkın, H., Uyaroğlu, Y. Synchronization of Chaos in Nonlinear Finance System by Means of Sliding Mode and Passive Control Methods: A Comparative Study. Information Technology and Control, 2015, 44(2), 172-181. https://doi.org/10.5755/j01.itc.44.2.7732
14. Kolmanovsky, I., McClamroch, N. H. Developments in Nonholonomic Control Problems. IEEE Control Systems, 1995, 15(6), 20-36. https://doi.org/10.1109/37.476384
15. Lee, T. H., Lam, H. K., Leung, F. H. F., Tam, P. K. S. A Practical Fuzzy Logic Controller for the Path Tracking of Wheeled Mobile Robots. IEEE Control Systems, 2003, 23(2), 60-65. https://doi.org/10.1109/MCS.2003.1188772
16. Liang, Z. Y., Wang, C. L. Robust Stabilization of Nonholonomic Chained form Systems with Uncertainties. Acta Automatica Sinica, 2011, 37(2), 129-142.
17. Liu, Y. L., Wu, Y. Q. Output Feedback Control for Stochastic Nonholonomic Systems with Growth Rate Restriction. Asian Journal of Control, 2011, 13(1), 177-185. https://doi.org/10.1002/asjc.230
18. Man, Z., Paplinski, A. P., Wu, H. R. A Robust MIMO Terminal Sliding Mode Control Scheme for Rigid Robotic Manipulators. IEEE Transactions on Automatic Control, 1994, 39(12), 2464-2469. https://doi.org/10.1109/9.362847

19. Man, Z., Yu, X. H. Terminal Sliding Mode Control of MIMO Linear Systems. IEEE Transactions on Circuits and Systems I: Fundamental, Theory and Applications, 1997, 44(11), 1065-1070. https://doi.org/10.1109/81.641769
20. Marchand, N., Alamir, M. Discontinuous Exponential Stabilization of Chained form Systems. Automatica, 2003, 39(2), 343-348. https://doi.org/10.1016/S0005-1098(02)00229-7
21. Mobayen, S. Finite-Time Tracking Control of Chained-Form Nonholonomic Systems with External Disturbances Based on Recursive Terminal Sliding Mode Method. Nonlinear Dynamics, 2015, 80, 669-683. https://doi.org/10.1007/s11071-015-1897-4
22. Mon, Y. J., Lin, C. M. ANFIS-Based Integral Terminal Sliding Mode Control for Disturbed Chaotic System. Journal of Intelligent and Fuzzy Systems, 2014, 27(1), 443-450.
23. Mon, Y. J., Lin, C. M. Double Inverted Pendulum Decoupling Control by Adaptive Terminal Sliding-Mode Recurrent Fuzzy Neural Network. Journal of Intelligent and Fuzzy Systems, 2014, 26(4), 1723-1729.
24. Ou, M., Du, H., Li, S. Finite-Time Tracking Control of Multiple Nonholonomic Mobile Robots. Journal of Franklin Institute, 2012, 349(9), 2834-2860. https://doi.org/10.1016/j.jfranklin.2012.08.009
25. Pan, Y., Liu, Y., Xu, B., Yu, H. Hybrid Feedback Feedforward: An Efficient Design of Adaptive Neural Network Control. Neural Networks, 2016, 76(6), 122-134. https://doi.org/10.1016/j.neunet.2015.12.009
26. Pan, Y., Yu, H. Biomimetic Hybrid Feedback Feedforward Neural-Network Learning Control. IEEE Transactions on Neural Networks and Learning Systems, 2016, 28(6), 1481-1487. https://doi.org/10.1109/TNNLS.2016.2527501
27. Pan, Y., Yu, H., Er, M. J. Adaptive Neural PD Control with Semiglobal Asymptotic Stabilization Guarantee. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(12), 2264-2274. https://doi.org/10.1109/TNNLS.2014.2308571
28. Passino, K. M. Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems, 2002, 22(3), 52-67. https://doi.org/10.1109/MCS.2002.1004010
29. Ploeg, J., Schouten, H. E., Nijmeijer, H. Position Control of a Wheeled Mobile Robot Including Tire Behavior. IEEE Transactions on Intelligent Transportation Systems, 2009, 10(3), 523-533. https://doi.org/10.1109/TITS.2009.2026316
30. Rajabioun R. Cuckoo Optimization Algorithm. Applied Soft Computing, 2011, 11(8), 5508-5518. https://doi.org/10.1016/j.asoc.2011.05.008
31. Storn, R., Price, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 1997, 11, 341-359. https://doi.org/10.1023/A:1008202821328
32. Sun, T., Pei, H., Pan, Y., Zhou, H., Zhang, C. Neural Network-Based Sliding Mode Adaptive Control for Robot Manipulators. Neurocomputing, 2011, 74(14-15), 2377-2384. https://doi.org/10.1016/j.neucom.2011.03.015
33. Tian, Y. P., Li, H. Exponential Stabilization of Nonholonomic Dynamic Systems by Smooth Time-Varying Control. Automatica, 2002, 38(7), 1139-1146. https://doi.org/10.1016/S0005-1098(01)00303-X
34. Wu, Y., Wang, B., Zong, G. D. Finite-Time Tracking Controller Design for Nonholonomic Systems with Extended Chained Form. IEEE Transactions on Circuits and Systems II: Express Briefs, 2005, 52(11), 798–802. https://doi.org/10.1109/TCSII.2005.852528
35. Wu, Y., Yu, X., Man, Z. Terminal Sliding Mode Control Design for Uncertain Dynamic Systems. Systems and Control Letters, 1998, 34(5), 281-287. https://doi.org/10.1016/S0167-6911(98)00036-X
36. Wu, Y. Q., Zhu, C. L., Zhang, Z. C. Finite-Time Stabilization of a General Class of Nonholonomic Dynamic Systems via Terminal Sliding Mode. International Journal of Automation and Computing, 2016, 13(6), 585-595. https://doi.org/10.1007/s11633-015-0931-9
37. Xi, Z. R., Feng, G., Jiang, Z. P., Cheng, D. Output Feedback Exponential Stabilization of Uncertain Chained Systems. Journal of Franklin Institute, 2007, 344(1), 36-57. https://doi.org/10.1016/j.jfranklin.2005.10.002
38. Xu, W. L., Huo, W. Variable Structure Exponential Stabilization of Chained Systems Based on the Extended Nonholonomic Integrator. Systems and Control Letters, 2000, 41(4), 225-235. https://doi.org/10.1016/S0167-6911(00)00057-8
39. Yang, X. S. A New Metaheuristic Bat-Inspired Algorithm. In: Gonzalez, J. R., Pelta, D. A., Cruz C., Terrazas, G., Krasnogor, N. (Eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, Berlin, 65-74, 2010. https://doi.org/10.1007/978-3-642-12538-6_6
40. Yang, X. S., Deb, S. Engineering Optimization by Cuckoo Search. International Journal of Mathematical Modeling and Numerical Optimization, 2010, 1(4), 330-343. https://doi.org/10.1504/IJMMNO.2010.035430
41. Yu, X., Man, Z. Model Reference Adaptive Control Systems with Terminal Sliding Modes. International Journal of Control, 1996, 64(6), 1165-1176. https://doi.org/10.1080/00207179608921680
42. Zhang, C., Sun, T., Pan, Y. Neural Network Observer-Based Finite-Time Formation Control of Mobile Robots. Mathematical Problems in Engineering, 2014, 267307. https://doi.org/10.1155/2014/267307