TIME DELAY CONTROL OF CABLE-DRIVEN MANIPULATORS WITH ARTIFICIAL BEE COLONY ALGORITHM

<table>
<thead>
<tr>
<th>Journal:</th>
<th>Transactions of the Canadian Society for Mechanical Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manuscript ID</td>
<td>TCSME-2017-0043.R1</td>
</tr>
<tr>
<td>Manuscript Type:</td>
<td>Article</td>
</tr>
<tr>
<td>Date Submitted by the Author:</td>
<td>30-Oct-2017</td>
</tr>
<tr>
<td>Complete List of Authors:</td>
<td>Yan, Fei; Nanjing University of Aeronautics &amp; Astronautics, College of Mechanical and Electrical Engineering</td>
</tr>
<tr>
<td></td>
<td>Wang, Yaoyao; Nanjing University of Aeronautics &amp; Astronautics, College of Mechanical and Electrical Engineering; Zhejiang University, the State Key Laboratory of Fluid Power and Mechatronic Systems</td>
</tr>
<tr>
<td></td>
<td>Xu, Wei; Nanjing University of Aeronautics and Astronautics</td>
</tr>
<tr>
<td></td>
<td>Chen, Bai; Nanjing University of Aeronautics and Astronautics</td>
</tr>
<tr>
<td>Is the invited manuscript for consideration in a Special Issue?</td>
<td>N/A</td>
</tr>
<tr>
<td>Keywords:</td>
<td>cable-driven manipulator, PID, time delay estimation, artificial bee colony algorithm</td>
</tr>
</tbody>
</table>
TIME DELAY CONTROL OF CABLE-DRIVEN MANIPULATORS WITH ARTIFICIAL BEE COLONY ALGORITHM

Fei Yan\textsuperscript{1}, Yaoyao Wang\textsuperscript{1,2}, Wei Xu\textsuperscript{1}, Bai Chen\textsuperscript{1}

\textsuperscript{1}College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing, China.
\textsuperscript{2}the State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou, China

E-mail: chenbye@126.com (Corresponding author: Prof. Chen), yywang_cmee@nuaa.edu.cn

Received Month 0000, Accepted Month 0000
No. 00-CSME-00, E.I.C. Accession Number 0000

\textbf{ABSTRACT}

Cable-driven manipulators (CDM) are widely-used for their unique advantages such as lightweight, low moving mass, high payload-to-weight ratio and large reachable workspace. However, their complex dynamic character and low stiffness with flexible joints make the control designing much more difficult than the traditional robot manipulators. In this paper, time delay control (TDC) which combines PID control method and time delay estimation (TDE) technology will be investigated to build a model-free controller for CDM. PID parameters are reduced dramatically as TDE compensates for a large proportion of unknown dynamics. To handle the problem in tuning parameters of this controller, artificial bee colony (ABC) algorithm is utilized to obtain optimal parameters of PID. Finally, simulations are conducted to verify the effectiveness of the propose controller and the tuning method.

\textbf{Keywords:} cable-driven manipulator; PID; time delay estimation; artificial bee colony algorithm.
1 INTRODUCTION

In recent years, cable-driven manipulators (CDMs) have attracted lots of attention for they can realize a light-weight structure, low moving mass, high payload-to-weight ratio and large reachable workspace (Yeo et al., 2013). Despite CDMs have plenty of advantages; complex dynamic character and low stiffness pose challenges to their industry application. Moreover, complex unknown lumped disturbance greatly impairs the control performance of CDMs.

To solve the above-mentioned problems, plenty of new approaches have been proposed. In 2013, Michele Giorelli et al. applied neural network in learning the inverse kinetics to handle the intrinsic problem of CDM that exact mechanical model and Jacobian matrix are not available (Giorelli et al., 2013). In 2013, Quanzhu Chen et al. were devoted on kinematic calibration of a 7-degree-of-freedom (7-DOF) cable-driven manipulator to make its motion control better. They derived the error model of the serial central linkage mechanism from its forward kinematics (Chen et al., 2013). In 2013, Darwin Lau et al. presented a generalized model to make modeling and analysis simple and convenient. Their model is suitable for multilink cable-driven serial manipulators, which has an arbitrary number of links and arbitrary cable routing (Lau et al., 2013). In 2014, Ryan James Caverly, et al. investigated a planar cable-actuated system about its dynamic model and control method, and created system's model by a lumped-mass method. They modeled a planar cable-actuated system by a lumped-mass method, which takes the change in cable stiffness and winch inertia into consideration (Caverly and Forbes, 2014). In the same year, to handle the problem of actuation redundancy, Wen Bin Lim et al. proposed a tension optimization method to obtain adjustable tension solution (Lim et al., 2014). In 2015, Darwin Lau et al. researched into multilink cable-driven manipulators (MCDMs) considering joint interaction force and moment in the objective function. Besides, they took constraints specific to the inverse dynamics of MCDMs into consideration for the first time (Lau et al., 2015).

Moreover, there are numerous researchers worked on control method. In 2016, Mingxing Yuan et al. elaborated on contouring control in industrial applications (Yuan et al., 2017). These researches have achieved fruitful results, however, precise dynamic model of CDM is hard to obtain, which
makes it impossible to control CDM effectively by model-based controller. In order to get satisfactory control performance and keep controller simple for mass production, traditional PID control method will be combined with time delay estimation (TDE) technique to build a model-free controller. Thanks to intentionally time-delayed information of the closed-loop control system, lumped unknown disturbance can be estimated and compensated effectively. Besides, PID parameters are greatly reduced because TED compensates for a large proportion of the unknown system dynamics.

A PID controller requires a set of input parameters for proper controller service, and as a result, a so-called tuning process is essential to search for the optimal input parameters (Fister et al., 2016). Due to complex dynamic character and low stiffness of CDM, PID turning seems a challenging job. Consequently, Artificial Bee Colony (ABC) Algorithm is utilized to tune PID parameters for best control performance. Compared with other swarm algorithm, ABC Algorithm has unique characters. ABC has three categories of bees that can be categorized as employed bees, onlooker bees, and scout bees. Moreover, ABC includes two colony behaviors, including recruiting onlooker bees to food sources and abandon food sources (Karaboga and Akay, 2009; Akbari et al., 2010). Finally, simulation had been carried out to confirm the effectiveness of our controller and our tuning method.

The remainder is organized as follows. Section II gives the dynamic model of CDM and designs a model-free controller for CDM based on time delay control (TDC). Afterwards, the general procedure of ABC algorithm is introduced, and then the stability analysis of proposed controller is given. In section III, comparative simulations are given to demonstrate the effectiveness of the controller and the feasibility of the tuning method. Finally, Section IV concludes this paper.

2 CONTROLLER DESIGN

2.1 Dynamics Model

Suppose $\theta$ is the angular position vectors of the motors, and $q$ is those of the links. Then the dynamics of a CDM is given as (Jin et al., 2017)
\[ I \ddot{\theta} + D_m \dot{\theta} = \tau_m - \tau_s(q, \dot{q}, \theta, \dot{\theta}) \]  

(1)

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) = \tau_s(q, \dot{q}, \theta, \dot{\theta}) + \tau_d \]

(2)

\[ \tau_s(q, \dot{q}, \theta, \dot{\theta}) = K_s(\theta - q) + D_s(\dot{\theta} - \dot{q}) \]

(3)

where \( \tau_m \) and \( \tau_s(q, \dot{q}, \theta, \dot{\theta}) \) stand for the motor input torque and the joint compliance torque, respectively. \( I \) and \( D_m \) represent inertia and damping matrix of the motor, respectively. \( M(q) \) and \( G(q) \) are mass-inertia matrix of robot links and the gravitational force vector, respectively. \( C(q, \dot{q}) \) represents the Coulomb friction and Coriolis centrifugal force vector of the robot links. \( F(q, \dot{q}) \) stands for other friction force which is a combination of viscous friction and Coulomb friction. Besides, \( \tau_d \) is the unknown but bounded external disturbance. \( K_s \) and \( D_s \) are the matrices of joint stiffness and damping, respectively. The dynamic equation can be rewritten from Eq.(1) and Eq.(2) as

\[ \tau_m = I \ddot{\theta} + D_m \dot{\theta} + M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) - \tau_d \]

(4)

Introducing a constant gain \( \bar{M} \), \( \tau_m \) can be expressed as

\[ \tau_m = \bar{M}\ddot{q} + \tau_e \]

(5)

\( \tau_e \) is the lumped unknown dynamics of CDM which is defined as

\[ \tau_e = \underbrace{I \ddot{\theta} + D_m \dot{\theta}}_{\text{Motor dynamics element}} + \underbrace{(M - \bar{M})\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q})}_{\text{Link dynamics element}} - \tau_d \]

(6)

### 2.2 Controller Design

The proposed control scheme is given as

\[ \tau_m = \bar{M}(\ddot{q}_d + K_p e + K_i \int e + K_d \dot{e}) + \dot{\tau}_e \]

(7)

where \( e \) is the trajectory tracking error and \( q_d \) is the desired angular position vector of the joints. The tracking error \( e \) is defined by

\[ e = q_d - q \]

(8)
Time delay estimation (TDE) technique is utilized to obtain \( \tau_e \), whose estimation is denoted by \( \hat{\tau}_e \).

Thus, \( \hat{\tau}_e \) can be expressed as

\[
\hat{\tau}_e = \tau_{e(t-\eta)} - \bar{\eta}_{(t-\eta)}
\]  

where \( t \) and \( \eta \) stands for current time and delay time which is often selected as several sampling period. \( \tau_{(t-\eta)} \) denotes the control output at time \( t-\eta \). Combining Eq.(7) and Eq.(9), the proposed controller for CDM is given as

\[
\tau_m = \bar{M}\left(\hat{q}_d + K_p e + K_I \int e + K_D \dot{e}\right) + \tau_{(t-\eta)} - \bar{\eta}_{(t-\eta)}
\]  

It is obvious that the controller is model-free. \( K_p, K_I, K_D \) will be tuned by ABC Algorithm introduced in the following section.

2.3 ABC Algorithm

ABC is an approximate model of nature honeybees that are categorized as employed bees, onlooker bees and scout bees (Abachizadeh et al., 2010). The value of food source stand for feasible solution to the target problem, and every employed bee corresponds to a single food source (Geng et al., 2014). The whole procedure of ABC Algorithm can be divided into four phases, i.e. initializing phase, employed bee phase, onlooker bee phase, and scout bee phase.

1) Initializing phase: Possible solutions are generated and the quality of these solutions is evaluated. Supposing the target question has \( n \) dimension, one of its solution called a food source, can be denoted as \( x_m = (x_{m1}, x_{m2}, \ldots, x_{mn}) \). Thus, possible solutions can be obtained by Eq.(11)

\[
x_{mi} = l_i + \vartheta \times (u_i - l_i)
\]

where \( m = 1, 2, \ldots, FD \) is the index of possible solutions and \( FD \) denotes the number of food source. \( x_{mi} \) is the \( i-th \) component of \( x_m \). \( \vartheta \) is a random number in the range of \([0,1]\). \( l_i \) and \( u_i \) are the lower bound and the upper bound of \( x_{mi} \), respectively.

2) Employed bee phase: employed bees search for nearby food source according to Eq.(12)

\[
v_{mi} = x_{mi} + \varphi_{mi} \cdot (x_{mi} - x_{ki})
\]
where $\nu_m$ denotes new food source and $\phi_m$ is a random number in the search range around the food source. $x_i$ is a food source selected at random. Whenever a food source is searched out, its fitness value which is the criterion of choosing better food source, will be calculated by the following equation.

$$fit(x_m) = 1/(1 + obj(x_m))$$

(13)

in which $obj(x_m)$ is the objective function of $x_m$, and $fit(x_m)$ is the fitness value of $x_m$.

3) Onlooker bees phase: Onlooker bees will randomly choose food source based on certain probability. The probability of an onlooker bee to choose a specific food source is calculated by the formula given as follows

$$P(x_m) = fit(x_m)/\sum_{m=1}^{\text{PD}} fit(x_m)$$

(14)

4) Scout bees phase: If a food source remain unimproved after trials of given time, the employed bee to that food source will abandon it and becomes a scout bee. A scout bee will search for a new food source according to Eq.(11).

Except Initializing phase, other phases are executed repeatedly for several cycles to find out the food source with the biggest fitness value.

Objective value is used to quantify control performance. As for manipulators tracking, we hope that the manipulator can track specific trajectory precisely with low energy cost. In other words, we hope the tracking error $e(t)$ and the control output of motors $u(t)$ as small as possible. However, we can hardly get minimum $e(t)$ with lowest energy consumption. The cost of minimizing $e(t)$ usually turns out to be increasing $u(t)$. Therefore, a suitable criterion is necessary to evaluate control performance.

Integral absolute error (IAE) is one of the most commonly used objective functions in Eq.(13) (Elkhateeb and Badr, 2013). In addition, integral squared control output (ISCO) is used to restrict energy consumption (Pan et al., 2011, Kumar et al., 2017). In this paper, IAE will be combined with ISCO as the objective function to evaluate control performance.
\[ \text{obj}(x_m) = \omega_1 \int_{0}^{\infty} |e(t)| dt + \omega_2 \int_{0}^{\infty} u(t)^2 dt \]  \hspace{1cm} (15)

where \( \omega_1 \), and \( \omega_2 \) are used to balance the impact of these objective functions. If there are more than one motors and joints, the objective value of each motor and joint will be calculated separately, then the sum of these objective values will be used as objective value.

Afterwards, controller parameters can be optimized by ABC algorithm by proper computer program. Details of implementation are shown in Fig.1.

2.4 Stability Analysis

The stability of the proposed controller can be proven based on the method given by (Jin et al., 2009). From Eq.(5) and Eq.(7) , we can get

\[ \ddot{e} + K_p e + K_I \int e + K_D \dot{e} = \varepsilon(t) \]  \hspace{1cm} (16)

in which \( \varepsilon = -\mathcal{M}^{-1}(\ddot{\tau}_e - \tau_e) \) is defined as the TDE error. Equation (16) shows a typical linear system which is stable on condition that \( K_p \), \( K_I \), and \( K_D \) are properly chosen and \( \varepsilon(t) \) is bounded. If \( \text{obj}(x_m) \) in Eq.(13) is properly chosen, preferable \( K_p \), \( K_I \), and \( K_D \) can be searched out by ABC Algorithm. Then we will prove the boundedness of \( \varepsilon(t) \) base on the method given in (Jin et al., 2009).

Denote

\[ u = \ddot{q}_a + K_p e + K_I \int e + K_D \dot{e} \]  \hspace{1cm} (17)

then Eq.(7) can be rewritten as

\[ \tau_m = \mathcal{M}u + \ddot{\tau}_e \]  \hspace{1cm} (18)

and \( \varepsilon(t) \) can be expressed as

\[ \varepsilon(t) = u - \dot{q} \]  \hspace{1cm} (19)

from Eq.(6), and Eq.(9), we can get

\[ \ddot{\tau}_e = I \ddot{\theta}_{(v-q)} + D \dddot{\theta}_{(v-q)} + (M - \mathcal{M}) \ddot{q}_{(v-q)} + C\dddot{q}_{(v-q)} + G_{(v-q)} + F_{(v-q)} - \tau_d_{(v-q)} \]  \hspace{1cm} (20)
A combination of Eq.(1), Eq.(2), Eq.(18) and Eq.(19) gives

\[ M \ddot{e}(t) = Mu - \bar{M}u - \dot{\tau}_e + I \ddot{\theta} + D_m \dot{\theta} - \tau_d + C \dot{q} + G + F \]  
\[ \text{(21)} \]

Substituting Eq.(20) into Eq.(21), we have

\[ M \ddot{e}(t) = \left( M - \bar{M} \right)u - \left( M_{(i-q)} - \bar{M} \right)\ddot{q}_{(i-q)} + \Delta \]
\[ \text{(22)} \]

where

\[ \Delta = I \ddot{\theta} + D_m \dot{\theta} + C \dot{q} + G + F - \tau_d 
- I \ddot{\theta}_{(i-q)} - D_m \dot{\theta}_{(i-q)} - C \dot{q}_{(i-q)} - G_{(i-q)} - F_{(i-q)} + \tau_d_{(i-q)} \]
\[ \text{(23)} \]

The friction term in Eq.(23) can be expressed as \( F = F_v + F_c \), where viscous friction \( F_v \) is continuous, while Coulomb friction \( F_c \) is bounded but discontinuous at velocity reversal. \( \Delta \) can be expressed as

\[ \Delta = \Delta_c + \Delta_d \]
\[ \text{(24)} \]
in which \( \Delta_c \) and \( \Delta_d \) represents the continuous part and discontinuous part of \( \Delta \), respectively. \( \Delta_c \) and \( \Delta_d \) are given as follows

\[ \Delta_c = I \ddot{\theta} + D_m \dot{\theta} + C \dot{q} + G + F_v - \tau_d 
- I \ddot{\theta}_{(i-q)} - D_m \dot{\theta}_{(i-q)} - C \dot{q}_{(i-q)} - G_{(i-q)} - F_{v{(i-q)}} + \tau_d_{(i-q)} \]
\[ \text{(25)} \]

\[ \Delta_d = F_c - F_{c{(i-q)}} \]
\[ \text{(26)} \]

Denote

\[ \Gamma = I \ddot{\theta} + D_m \dot{\theta} + C \dot{q} + G + F_v - \tau_d \]
\[ \text{(27)} \]

If boundedness and continuous condition of \( \Gamma \) can be satisfied, there exists

\[ \Gamma = O(\eta^2) \]
\[ \text{(28)} \]
in which \( O \) is used to describe the error term in an approximation to a mathematical function (Su et al., 2000). Reducing the delay time \( \eta \) can make the approximation error smaller. It is obvious that \( \Delta_d \) satisfies the following inequality

\[ \Delta_d \leq \begin{cases} b, & \text{when velocity reverses} \\ 0, & \text{other conditions} \end{cases} \]
\[ \text{(29)} \]
in which \(\dot{b}\) becomes a constant vector when \(\eta\) is sufficient small. Overall, the bound of \(\Delta\) is given by

\[
\Delta \leq b + \Gamma
\]  
(30)

Substituting \(\dot{q}(t_{\eta}) = u(t_{\eta}) - e(t)\) from Eq.(19) into Eq.(22) yields

\[
M\dot{e}(t) = (M - \bar{M})u - (M - \bar{M})(u(t_{\eta}) - e(t)) + (M - M(t_{\eta}))\dot{q}(t_{\eta}) + \Delta
\]  
(31)

\[
M\dot{e}(t) = (M - \bar{M})e(t) + (M - \bar{M})(u(u(t_{\eta})) + (M - M(t_{\eta}))\dot{q}(t_{\eta}) + \Delta
\]  
(32)

Therefore, \(e(t)\) can be expressed as

\[
e(t) = Ee(t-\eta) + E\alpha + \beta
\]  
(33)

in which

\[
E = I - M^{-1}\bar{M}
\]  
(34)

\[
\alpha = u - u(t_{\eta})
\]  
(35)

\[
\beta = M^{-1}\left[(M - M(t_{\eta}))\dot{q}(t_{\eta}) + \Delta\right]
\]  
(36)

since \(\Delta\) is bounded, for sufficiently small \(\eta\), \(\alpha\) and \(\beta\) are ought to be bounded. Note that \(\|E\|<1\) can be easily satisfied by suitable choice of \(\bar{M}\) (Spong and Vidyasagar, 1987).

In the discrete-time domain, (33) can be expressed as

\[
e(k) = E(k)e(k-1) + E(k)\alpha(k) + \beta(k)
\]  
(37)

\(\|E\|<1\) implies that roots of \(E(k)\) reside inside a unit circle. Consequently, the first-order difference Eq. (37) is asymptotically bounded with bounded forcing function \(\alpha\) and \(\beta\).

Suppose the desired trajectory and the initial error are bounded, \(\|E\|<1\) guarantees the boundedness of \(e(t)\) or \(e(k)\), which assure the boundedness of the error, and then \(\Gamma\) is bounded. As an initial condition of the problem, the error is bounded, and as a result, the error and the TDE error \(e(t)\) or \(e(k)\) will always keep bounded which guarantees the stability of the system.
3 SIMULATION STUDIES

3.1 Simulation Setup

We can get from Eq.(1)-(3) that

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) = \tau_m + \tau_d - I\dot{\theta} - D\dot{\theta} \]  \hspace{1cm} (38)

Considering \( \zeta = -\tau_d + I\dot{\theta} + D\dot{\theta} \) as the disturbance of manipulator, we can get

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q, \dot{q}) + \zeta = \tau_w \]  \hspace{1cm} (39)

Eq.(39) has the same structure as a typical 2-DOF manipulator model shown in Fig. 2 which will be used in simulation. \( M(q), C(q, \dot{q}), G(q), F(q, \dot{q}) \) in that equation are as follows.

\[
\begin{align*}
M(q)_{11} &= I_{11}^2m_2 + 2l_1l_2m_2c_2 + l_1^2(m_1 + m_2) \\
M(q)_{12} &= M(q)_{21} = l_2^2m_2 + l_1l_2m_2c_2 \\
M(q)_{22} &= l_2^2m_2 \\
C(q, \dot{q})\dot{q} &= \begin{bmatrix} -l_1l_2m_2s_2\dot{q}_2^2 - 2l_1l_2m_2s_2\dot{q}_1\dot{q}_2 \\ l_1l_2m_2s_2\dot{q}_2^2 \end{bmatrix} \\
G(q) &= \begin{bmatrix} l_2m_2gc_{12} + l_1gc_1(m_1 + m_2) \\ l_2m_2gc_{12} \end{bmatrix} \\
F(q, \dot{q}) &= \begin{bmatrix} F_{v1}\dot{q}_1 + F_{c1}\text{sgn}(\dot{q}_1) \\ F_{v2}\dot{q}_2 + F_{c2}\text{sgn}(\dot{q}_2) \end{bmatrix}
\end{align*}
\]  \hspace{1cm} (40-45)

where \( s_i = \sin(q_i), c_i = \cos(q_i), \) and \( c_{ij} = \cos(q_i + q_j). \) The parameters of the 2-DOF manipulator are shown in Table 1. Each joint’s objective function is calculated separately by Eq.(15) and their sum is used as the objective function in Eq.(13). The integral algorithm is set as ‘ode4’ in Simulink version 8.8 with the sample time 0.001s. In this paper, \( \omega_1 \) is set to be eight for each joint, and \( \omega_2 \) is set as 0.0002 and 0.001 for joint 1 and joint 2, respectively. The unit for \( e \) and \( u \) is degree and \( N \cdot m \), respectively. Other parameters of ABC algorithm are shown in Table 2. \( \bar{M} \) in eq.(5) is given by
3.2 Simulation Results

Three cases are studied for comparison:

1) **Case 1**: controllers with TDE, and the range of all PID parameters are [0, 20].

2) **Case 2**: controllers without TDE, and the range of all PID parameters are [0, 200].

3) **Case 3**: controllers without TDE, and the range of all PID parameters are [0, 2000].

Tuning results are shown in Table 3. The development of fitness value during tuning process is shown in Fig. 3. Triangular wave, shown in Fig. 4, is used as desired position in tuning process. We can see that after 50 cycles the fitness values roughly remain stable. After 200 cycles of optimization, case 1 achieves the highest fitness value of 0.2578; following by case 3, whose fitness value finally come to 0.2479. In contrast, case 2 achieves a much lower fitness value of 0.1545. PID parameters in case 1 are much smaller than those in case 3, but the final fitness value in case 1 is slightly higher than that in case 3. This result comes down to the application of TDE method which greatly compensates for lumped unknown dynamics. Therefore, TDE is an effective tool in controller design considering tracking error and control output.

In Fig. 4, we can see that in all cases, joints can precisely tracking the desired position except for small errors near turning point. All of three cases generate smooth output except for some chattering in case 3. However, when it comes to position error, case 2 has poor performance due to slow convergence. Case 3 may performance better due to large PID parameters. However, large PID parameters may lead to instability when disturbance occurs. We can conclude that case 1 achieved satisfactory performance with small PID parameters.

In order to test whether the tuning results shown in Table 3 suitable for other trajectories, another simulation is conducted. In that simulation, the desired trajectory is given by

\[
q_d = \begin{cases} 
15\degree \times \sin(90\degree \times t - 90\degree) + 15\degree, & 0 \leq t \leq 2 \\
30\degree \times \sin(45\degree \times t), & 2 \leq t \leq 8
\end{cases}
\]  

(47)
where \( q_d \) stands for desired trajectory in degree, and \( t \) is simulation time in second. Simulation results are shown in [Error! Reference source not found.].

From the results, we can see that in all of three cases joints can tracking desired trajectory well, and there is no major difference in control output among three cases. However, when it comes to position error, case 2 has much bigger position error than the other cases. The position error of joint 1 in case 3 is the smallest between 0.30s and 1.30s, however, it is larger than that in case 1 during other period, especially at the beginning of simulation. The position error of joint 2 in case 3 outperforms the other cases, as a result of big PID parameters. Nevertheless, the absolute position error of joint 2 is rather small compared with that of joint 1. We would say that the position error of joint 2 in case 1 is also acceptable. It is worth to mention that too big PID parameters may make the controller vulnerable of disturbance. Above all, we can conclude that case 1 achieves the best performance.

4 CONCLUSIONS

A controller for CDM is proposed and its tuning method is investigated. Traditional PID control method and TDE technique is combined to build a model-free controller, while small PID gains can be obtained to guarantee good robustness. PID is utilized to regulate the dynamic performance, and TDE is adopted to compensate CDM’s the lumped unknown dynamics. To cope with tuning problems brought by complex dynamic character and low stiffness of CDM, ABC algorithm is utilized to find optimal PID parameters. PID parameters are greatly reduced thanks to TDE that compensates for a large proportion of unknown dynamics. Finally, simulations are carried out to verify the feasibility of ABC algorithm in PID tuning, and the effectiveness of the optimized PID parameter is tested.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 51705243 and 51575256), Natural Science Foundation of Jiangsu Province (SBK20170789) and the Open Foundation of the State Key Laboratory of Fluid Power and Mechatronic Systems (GZKF-201606).
REFERENCES


### NOMENCLATURE (OPTIONAL SECTION)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>angular position vectors of the motors</td>
</tr>
<tr>
<td>$q$</td>
<td>angular position vectors of the links</td>
</tr>
<tr>
<td>$\tau_m$</td>
<td>motor input torque</td>
</tr>
<tr>
<td>$\tau_s$</td>
<td>joint compliance torque</td>
</tr>
<tr>
<td>$I$</td>
<td>inertia matrix of the motor</td>
</tr>
<tr>
<td>$D_m$</td>
<td>damping matrix of the motor</td>
</tr>
<tr>
<td>$M$</td>
<td>mass-inertia matrix of robot links</td>
</tr>
<tr>
<td>$D_L$</td>
<td>viscous friction matrix</td>
</tr>
<tr>
<td>$C$</td>
<td>Coulomb friction and Coriolis centrifugal force vector of the robot links</td>
</tr>
<tr>
<td>$F$</td>
<td>friction force vector</td>
</tr>
<tr>
<td>$G$</td>
<td>gravitational force vector</td>
</tr>
<tr>
<td>$\tau_d$</td>
<td>unknown but bounded external disturbance</td>
</tr>
<tr>
<td>$K_s$</td>
<td>matrices of joint stiffness</td>
</tr>
<tr>
<td>$D_s$</td>
<td>matrices of joint damping</td>
</tr>
<tr>
<td>$\bar{M}$</td>
<td>a constant gain</td>
</tr>
<tr>
<td>$\tau_e$</td>
<td>lumped unknown dynamics of cable-driven manipulators</td>
</tr>
<tr>
<td>$e$</td>
<td>trajectory tracking error</td>
</tr>
<tr>
<td>$q_d$</td>
<td>desired angular position vector of the joints</td>
</tr>
<tr>
<td>$\hat{\tau}_e$</td>
<td>the estimation of lumped unknown dynamics of cable-driven manipulators</td>
</tr>
<tr>
<td>$t$</td>
<td>current time</td>
</tr>
<tr>
<td>$\eta$</td>
<td>delay time</td>
</tr>
<tr>
<td>$x_m$</td>
<td>a solution of the target question</td>
</tr>
<tr>
<td>$m$</td>
<td>index of possible solutions</td>
</tr>
</tbody>
</table>
\( x_{mi} \) the \( i-th \) component of \( x_m \)

\( FD \) number of food source

\( \vartheta \) a random number in the range of [0,1]

\( l_i \) lower bound of \( x_{mi} \)

\( u_i \) upper bound of \( x_{mi} \)

\( v_{mi} \) a new food source

\( \varphi_{mi} \) a random number in the search range around the food source

\( x_{kl} \) a food source selected at random

\( \varepsilon \) TDE error

\( F_y \) viscous friction

\( F_c \) Coulomb friction

\( q_d \) desired trajectory
### TABLES

#### Table 1. Parameters of The 2-DOF Manipulator

<table>
<thead>
<tr>
<th>parameters</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of link 1 ((l_1))</td>
<td>1.0 m</td>
</tr>
<tr>
<td>Length of link 2 ((l_2))</td>
<td>1.0 m</td>
</tr>
<tr>
<td>Equivalent mass at joint 1 ((m_1))</td>
<td>1.0 kg</td>
</tr>
<tr>
<td>Equivalent mass at joint 2 ((m_2))</td>
<td>1.0 kg</td>
</tr>
<tr>
<td>Viscous friction coefficient of link 1 ((F_v^1))</td>
<td>5.0 N \cdot m \cdot s / rad</td>
</tr>
<tr>
<td>Viscous friction coefficient of link 2 ((F_v^2))</td>
<td>5.0 N \cdot m \cdot s / rad</td>
</tr>
<tr>
<td>Coulomb friction coefficient of link 1 ((F_c^1))</td>
<td>5.0 N \cdot m</td>
</tr>
<tr>
<td>Coulomb friction coefficient of link 2 ((F_c^2))</td>
<td>5.0 N \cdot m</td>
</tr>
<tr>
<td>Gravity acceleration</td>
<td>9.8 m / s²</td>
</tr>
</tbody>
</table>
Table 2. Parameters of ABC Algorithm

<table>
<thead>
<tr>
<th>parameters</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Colony Size</td>
<td>20</td>
</tr>
<tr>
<td>Number of Food Sources</td>
<td>10</td>
</tr>
<tr>
<td>Maximum Number of Cycles</td>
<td>200</td>
</tr>
<tr>
<td>Number of Optimized Parameters</td>
<td>6</td>
</tr>
</tbody>
</table>
### Table 3. Tuning Results

<table>
<thead>
<tr>
<th>joint</th>
<th>parameters</th>
<th>case 1</th>
<th>case 2</th>
<th>case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$K_r$</td>
<td>20.0</td>
<td>200.0</td>
<td>250.7</td>
</tr>
<tr>
<td>1</td>
<td>$K_i$</td>
<td>0.01386</td>
<td>200.0</td>
<td>2000.0</td>
</tr>
<tr>
<td>1</td>
<td>$K_d$</td>
<td>0.7187</td>
<td>4.284</td>
<td>5.521</td>
</tr>
<tr>
<td>2</td>
<td>$K_r$</td>
<td>20.0</td>
<td>200.0</td>
<td>2000.0</td>
</tr>
<tr>
<td>2</td>
<td>$K_i$</td>
<td>0.04721</td>
<td>200.0</td>
<td>2000.0</td>
</tr>
<tr>
<td>2</td>
<td>$K_d$</td>
<td>1.907</td>
<td>0.5076</td>
<td>0.2901</td>
</tr>
</tbody>
</table>
FIGURES CAPTIONS

Fig. 1. Details of implementation in applying ABC algorithm

Fig. 2. A 2-DOF manipulator

Fig. 3. Optimizing process

Fig. 4. Results of experiment 1. a) Tracking performance of joint 1; b) Tracking performance of joint 2; c) Position error of joint 1; d) Position error of joint 2; e) Control output of joint 1; f) Control output of joint 2;

Fig. 5. Results of experiment 2. a) Tracking performance of joint 1; b) Tracking performance of joint 2; c) Position error of joint 1; d) Position error of joint 2; e) Control output of joint 1; f) Control output of joint 2;
Fig. 1. Details of implementation in applying ABC algorithm
Fig. 2. A 2-DOF manipulator

190x178mm (300 x 300 DPI)
Fig. 3. Optimizing process

82x73mm (300 x 300 DPI)
Fig. 4. Results of experiment 1. a) Tracking performance of joint 1; b) Tracking performance of joint 2; c) Position error of joint 1; d) Position error of joint 2; e) Control output of joint 1; f) Control output of joint 2;
Fig. 5. Results of experiment 2. a) Tracking performance of joint 1; b) Tracking performance of joint 2; c) Position error of joint 1; d) Position error of joint 2; e) Control output of joint 1; f) Control output of joint 2;