Robotica (2019) volume 37, pp. 39–61. © Cambridge University Press 2018
doi:10.1017/S0263574718000851

Solving nonlinear optimal path tracking problem using a new closed loop direct–indirect optimization method: application on mechanical manipulators

M. Irani Rahaghi†,* and F. Barat‡

†*Department of Mechanical Engineering, Islamic Azad University, Kashan Branch, Kashan, Iran*
‡*Department of Mechanical Engineering, University of Kashan, Kashan, Iran.*
E-mail: farzaneh_barat@yahoo.com

(Accepted August 8, 2018. First published online: August 31, 2018)

SUMMARY
The purpose of this study is to determine the dynamic load carrying capacity (DLCC) of a manipulator that moves on the specified path using a new closed loop optimal control method. Solution methods for designing nonlinear optimal controllers in a closed-loop form are usually based on indirect methods, but the proposed method is a combination of direct and indirect methods. Optimal control law is given by solving the nonlinear Hamilton–Jacobi–Bellman (HJB) partial differential equation. This equation is complex to solve exactly for complex dynamics, so it is solved numerically using the Galerkin procedure combined with a nonlinear optimization algorithm. To check the performance of the proposed algorithm, the simulation is performed for a fixed manipulator. The results represent the efficiency of the method for tracking the pre-determined path and determining the DLCC. Finally, an experimental test has been done for a two-link manipulator and compare with simulation results.

KEYWORDS: Optimal control, Nonlinear optimization, HJB, Tracking, DLCC of manipulators

1. Introduction

A major benefit of a robotic manipulator is their ability to move on a pre-specified path or between two predefined points. Planning a proper trajectory is a fundamental issue to complete such operations. Trajectory planning has been defined as finding the optimal trajectory from a specific origin to a specified point (point to point motion) or finding the history of angular positions and velocities and joint torques for a pre-specified end-effector trajectory (pre-specified path). In reality, the open-loop method cannot be used for the motion of a robot in a specific path. Open-loop method is extremely sensitive to parameter variations and has no robustness against disturbances and therefor the alternative method is to use a closed-loop controller. At any moment, the closed-loop controller determined the required torque due to the amount of error, actual and desired position, and the speed of the robot.

To increase the amount of load carrying capacity, it is necessary to select the appropriate control method and the required torque for the robot actuators must be reduced. For this reason, a closed loop nonlinear optimal control method is used. For solving a nonlinear optimal control problem in a closed-loop form, the Hamilton–Jacobi–bellman (HJB) equation must be solved. Solving this equation analytically for a nonlinear system is not possible and a numerical method can be used to solve this equation. The numerical method for solving the HJB equation is complex and the process of controller design is very difficult. In this paper, with a combination of direct and indirect methods, a new algorithm was proposed to solve this equation. Initially, optimal conditions are applied according to the indirect optimization method and then the structure of control law is designed and finally, unknown controller parameters were determined by solving the resulting nonlinear optimization problem using MATLAB software.

* Corresponding author. E-mail: irani@kashanu.ac.ir

In ref. [1] Optimal point-to-point motion planning has been investigated based on the Pontryagin's minimum principle for a heavy duty industrial robot to travel between two points in a fixed time interval. In ref. [2], an iterative method for time optimal control of a general type of dynamic systems is proposed. The method uses the indirect solution of the open-loop optimal control problem. The necessary conditions for optimality are derived from Pontryagin's minimum principle and the resulting equations lead to a nonlinear two-point boundary value problem.

In ref. [3], two methods are presented for solving a closed-loop optimal control problem. These control laws are based on the numerical solution to the nonlinear HJB equation. The first approach is the successive approximation for finding the solution of the HJB equation in the closed-loop form, and the second approach is based on solving a state-dependent Riccati equation, which is an extension of the algebraic Riccati equation for nonlinear systems. In ref. [4], a new method is proposed to solve a nonlinear optimal control problem.

Solution methods for designing nonlinear optimal controllers in a closed-loop form are usually based on indirect methods, but the proposed method is a combination of direct and indirect methods. As it mentioned the optimal control law of nonlinear dynamic systems with state feedback form is given by the solution to the nonlinear HJB equation. The Galerkin procedure and a nonlinear optimization algorithm are used to solve this equation numerically. Results show that the proposed method can be used for trajectory planning and designing optimal control with closed form for manipulators.

In ref. [5], optimal point-to-point motion planning of an elastic parallel manipulator is investigated. The dynamic model of a 3R planar parallel manipulator is obtained using Kane equation and the improved curvature based finite element method. Then, an adaptive Gauss Pseudospectral Method is proposed to transcribe the trajectory optimization problem into the Nonlinear Programming (NLP) problem, and the sequential quadratic programming (SQP) approach is used to solve this problem. The proposed method in ref. [6] is compared with two closed-loop nonlinear methods that are usually applied to robotic systems. The proposed method is compared with the feedback linearization, which linearizes the nonlinear dynamic equations using a nonlinear state transformation and a nonlinear feedback law and a robust sliding mode control method. Results show better performance of the proposed nonlinear optimal control approach. Another major subject that is considered in this paper is determining the maximum dynamic load carrying capacity (DLCC) of manipulators. The DLCC, which is an important characteristic of the industrial manipulator, is the maximum allowable load that can be carried by the manipulator in a point to point or predefined trajectory tasks. Finding DLCC of a manipulator and related trajectory is turned out to be an optimization problem that can be solved using various methods.

In ref. [7], a method is proposed for calculating the DLCC for flexible link manipulators with an open-loop form based on Pontryagin minimum principle and indirect optimization. Dynamic equations are based on the method that is detailed in ref. [8]. In ref. [9], a closed loop nonlinear optimal control approach is investigated for flexible joint manipulators and the DLCC of these manipulators is obtained through this approach. DLCC is calculated according to the limits in actuators and tracking accuracy.

Guo *et al.*,[10] proposed a direct optimization method for trajectory planning and obtaining maximum DLCC for a five DOF hybrid manipulator. An approved evolutionary algorithm that was elitist nondominated sorting genetic algorithm (NSGA-II) was used. Wu *et al.*[11] presented a novel method for obtaining the maximum DLCC of a redundant and non-redundant actuated parallel conveyor by optimizing the internal forces. In ref. [12], two-level optimal approach was considered for point-to-point motion planning of a flexible parallel manipulator. It was considered a rigid-flexible coupling dynamic model and actuator dynamics. Then, the multi-interval Legendre–Gauss–Radau (LGR) pseudospectral method was applied to transform the optimal control problem into the NLP problem. Then, the multi-interval LGR Pseudospectral method was applied to transform the optimal control problem into the NLP problem. Fen *et al.*,[13] presented a new improved ant colony algorithm based on the rule of finding a better path in a shorter time that was based on the D–H parameter method for solving the inverse kinematics of a six DOF manipulator.

The load-carrying capacity of the manipulator is often considered to be the same throughout their work space. However, the actual capacity of the manipulator largely depends on their posture, their velocity, their acceleration, and their actuators limits. In ref. [14], a method is proposed to increase the DLCC of the manipulator through trajectory optimization. In ref. [15],

a new two-level method which is a combination of open and closed loop optimal control, is proposed for calculating DLCC of nonholonomic Wheel Mobile Manipulator in the presence of environmental obstacle. This method obtains the maximum load path and the maximum allowable load in a specified time, subject to constraints such as motor torque and jerk limits, accuracy of reaching to the final point, obstacle avoidance and vehicle kinematic and dynamic parameters. The objective of[16] is to identify the trajectory that accomplishes the assigned motion with the maximum DLCC, which is subject to the constraints imposed by the manipulator structure kinematics and dynamics.

In the previous works, the optimal trajectory with maximum DLCC is determined and then a closed-loop controller is designed to realize this trajectory. The proposed method in this paper resolves these deficiencies and designs a nonlinear optimal controller and optimal trajectory with maximum DLCC at the same time. Also, updating and applying this method for a different robot can be implemented without any complicated computations. The structure of paper is as follows; in Section 2, the problem statement is presented and the direct and indirect procedures are explained and the combined method is proposed through an applicable algorithm. Then in Section 3, the DLCC of the manipulator is defined in detail and some of the constraints that deal with the optimization problem are illustrated. Finally, in Section 4, simulations are applied to a rigid planar manipulator and the experimental test is applied on a Puma-type manipulator to show the capability of the proposed method.

2. Problem Statement

Optimal controller design is performed using combination of direct and indirect methods, where the indirect method is based on solving the HJB equation and Galerkin approximation, and the direct method is based on a direct search for an equivalent discrete dummy model. In this section, first the initial optimal control problem is defined and the HJB equation is presented to solve this problem indirectly. Finally the direct-optimization method that is based on the collocation procedure is used to determine the unknown parameters in the resulting optimal control law.

2.1. Optimal control and H–J–B equation (indirect part)

A dynamic system with nonlinear state equation consider as (1). Also, the performance index function and its nonlinear term are defined as (2) and (3).

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x) \qquad x \in R^n; u \in R^r \tag{1}$$

$$J(x) = h(x(t_f), t_f) + \int_{t_0}^{t_f} (l(x) + u^T(x)\bar{R}u(x))\mathrm{d}t \tag{2}$$

$$l(x) = [x(t) - x_d(t)]^T \bar{Q}[x(t) - x_d(t)] \tag{3}$$

where $\bar{Q}$ is $n \times n$ symmetric positive semi-definite and $\bar{R}$ is $r \times r$ symmetric positive definite matrixes, also $x_d(t)$ is vector of the desired state trajectory. This formulation of performance index causes the states to track the desired values described in $x_d(t)$.

The optimal value of $J(x(t), t)$ is subject to input control vector *u(t)* denoted by $J^*(x(t), t)$.

$$J^*(x, t) = \min_{\substack{u(t) \\ 0<t<t_f}} \left\{ \int_0^{t_f} (l(x(t), t) + u(x, t)^T \bar{R}u(x, t))\mathrm{d}t \right\} \tag{4}$$

The Hamiltonian is defined as follows:

$$H(x(t), u(t), p(t), t) = (l(x(t), t) + u(x, t)^T \bar{R}u(x, t)) + p^T(t)[f(x(t)) + g(x)u(x, t)] \tag{5}$$

where *p(t)* is the vector of co-states. According to calculus of variations, the necessary conditions for *u*(t)* and *x*(t)*, to be optimal control and optimal state, respectively, are as follows:

$$\begin{aligned}
\dot{x}^*(t) &= \frac{\partial H}{\partial p}(x^*(t), u^*(t), p^*(t), t) \\
\dot{p}^*(t) &= -\frac{\partial H}{\partial x}(x^*(t), u^*(t), p^*(t), t) \\
H(x^*(t), u^*(t), p^*(t), t) &\leq H(x^*(t), u(t), p^*(t), t)
\end{aligned} \tag{6}$$

The boundary conditions for two sets of ordinary differential equations are as follows:

$$x^*(t_0) = x_0; \quad p^*(t_f) = \frac{\partial h}{\partial x}(x^*(t_f), t_f) = 0 \tag{7}$$

As it shown in ref. [17], $p^*(t) = \frac{\partial J^*}{\partial x}(x^*(t))$ and Eq. (8) must be satisfied for $J^*(x^*(t))$ in order to be the minimum value of $J(x)$.

$$\frac{\partial J^*}{\partial t} + \left(l(x^*, t) + u^*(x^*, t)^T \bar{R} u^*(x^*, t)\right) + \left[\frac{\partial J^*}{\partial x}\right]^T \left[f(x^*(t)) + g(x^*)u^*(x^*, t)\right] = 0 \tag{8}$$

This is HJB partial differential equation and it is not possible to obtain its analytical solution for nonlinear systems. The solution of this equation for linear time invariant systems leads to a well-known algebraic Riccati equation.[17] A new approach for solving this equation for nonlinear systems is presented using a combination of Galerkin approximation and dynamic optimization.

After solving H–J–B equation, the solution to the main optimal control problem is given by the state feedback law.

$$u^*(x, t) = \frac{1}{2}\bar{R}^{-1} g^T(x) \frac{\partial J^*}{\partial x}\left(x^*(t)\right) \tag{9}$$

That $J^*(x^*(t))$ is solution to the HJB equation.

2.2. Galerkin approximation and nonlinear optimization (direct part)

According to the numerical Galerkin method,[18] it is considered that one can find a complete set of basic functions $\{\phi_j(x)\}_{j=1}^{\infty}$ with boundary conditions $\{\phi_j(0) = 0\}_{j=1}^{\infty}$ and the solution of the HJB equation can be written as follows:

$$J^*(x(t), t) = \sum_{j=1}^{\infty} c_j(t)\phi_j(x) \tag{10}$$

It is assumed that this summation converges in a set of state variables such as Ω. An approximate solution to the HJB equation is obtained by truncating this summation.

$$J^*{}_N(x(t), t) = \sum_{j=1}^{N} c_j(t)\phi_j(x) \tag{11}$$

Using this procedure, the optimal control law can be written as follow:

$$u_N(x, t) = \frac{1}{2}\bar{R}^{-1} g^T(x) \frac{\partial J^*{}_N}{\partial x} = \frac{1}{2}\bar{R}^{-1} g^T(x) \left[\sum_{j=1}^{N} c_j(t) \frac{\partial \phi_j(x)}{\partial x}\right] \tag{12}$$

Nonlinear programming method is used to find time varying coefficients $c_j(t)$. The controller $u_N(x, t)$ has a nonlinear form and has the same robustness as a linear quadratic controller.[19] Important

issue of this procedure is a proper selection of basic functions and Ω. Some controls should be taken in the selection of basic functions, since their partials with respect to the states ($\frac{\partial \phi_j(x)}{\partial x}$) will ultimately appear in the control law. In this paper, second-order polynomial functions of states that will result linear terms with respect to the states in control law are selected. It is shown in ref. [20] that selection higher order terms to basic functions have no advantages. The optimal control problem explained by Eq. (12) can be mentioned as a nonlinear optimization problem as follows to find unknown parameters $c_j(t)$.

Algorithm 1.

$$\begin{aligned}
&\min_{c_j(t)} \left\{ h(x(t_f), t_f) + \int_0^{t_f} (l(x(t), t) + u_N(x, t)^T \bar{R} u_N(x, t)) dt \right\} \\
&Subject\ to: \\
&\dot{x} = f(x(t)) + g(x) u_N(x, t) \\
&x(t) \in \Omega \\
&x(0) = x_0; x(t_f) = x_f \\
&u_N(x, t) = \tfrac{1}{2} \bar{R}^{-1} g^T(x) \left[\sum_{j=1}^{N} c_j(t) \frac{\partial \phi_j(x)}{\partial x} \right] \\
&u_{min} \le u_N(x, t) \le u_{max}
\end{aligned}$$

This optimization problem, which has a differential constraint, can be solved using the collocation method and the "fmincon" command from MATLAB software.[20] It should be noted that MATLAB software can only solve nonlinear optimization. It includes algebraic constraints and cannot solve dynamic optimization problem. The collection of functions increases the capability of the MATLAB optimization toolbox, especially of the constrained nonlinear minimization of the routine fmincon.

By definition of an additional state variable x_{n+1} and differential Eq. (13), the performance index can be defined as the standard Mayer form (14).

$$\dot{x}_{n+1} = l(x(t), t) + u_N(x, t)^T \bar{R} u_N(x, t); \quad x_{n+1}(0) = 0 \tag{13}$$

$$J(u(t)) = h(x(t_f), t_f); \quad x \in R^{n+1} \tag{14}$$

Using Eqs. (13) and (14), the optimization problem defined by Algorithm 1 is converted to the following Mayer form optimization.

Algorithm 2.

$$\begin{aligned}
&\min_{C(t)} \{h(x(t_f), t_f)\} \\
&\text{such that:} \\
&\dot{x} = f(x(t), C(t), t) \qquad x(0) = x_0 \\
&Eq(x(t), C(t), t) = u_N(x, t) - \tfrac{1}{2} \bar{R}^{-1} g^{\mathrm{T}}(x) \left[\sum_{j=1}^{N} c_j(t) \frac{\partial \phi_j(x)}{\partial x} \right] = 0 \\
&g_1(x(t), C(t), t) = u_N(x, t) \le u_{\max} \\
&g_2(x(t), C(t), t) = -u_N(x, t) \le -u_{\min} \\
&x^{LB} \le x(t) \le x^{UB}; \qquad \Omega = \left[x^{LB}, x^{UB}\right]
\end{aligned}$$

where

$$C(t) = [c_1(t), c_2(t), \ldots, c_N(t)]^T \tag{15}$$

$Eq, g = \{g_1, g_2\}^{\mathrm{T}}, x^{LB}; x^{UB}, u_{\min}; u_{\max}$ are equality, inequality constraint vectors, state and input profile bounds.

As it mentioned, the collocation finite element is used to convert the differential constraint equation into an algebraic one. Based on the collocation method, the residual equations that are evaluated at

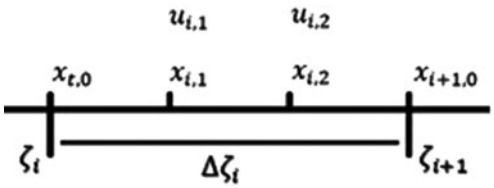

Fig. 1. Time interval and collocation points ($K = 2$).

any shifted roots of Lagrange polynomials shown in Eq. (16), form a set of algebraic equations.[20]

$$x_i(t) = \sum_{j=0}^{K} x_{i,j} \left[\prod_{\substack{k=0 \\ k\neq j}}^{K} \frac{(t - t_{i,k})}{(t_{i,j} - t_{i,k})} \right] \qquad i = 1, ..., N \tag{16}$$

$$c_i(t) = \sum_{j=0}^{K} c_{i,j} \left[\prod_{\substack{k=0 \\ k\neq j}}^{K} \frac{(t - t_{i,k})}{(t_{i,j} - t_{i,k})} \right] \qquad i = 1, ..., N \tag{17}$$

Differential constraint equation, equality and inequality must be discretized into N time intervals, also as shown in Fig. 1, each time interval ($t \in [\zeta_i, \zeta_{i+1}]$) is divided into K collocation points. State and input variables on each point are approximated by Lagrange polynomials with $x_{i,j}$ and $c_{i,j}$ unknown coefficients. In addition, the continuity conditions must be satisfied at any endpoints ($x_{i,0}(\zeta_i) = x_{i-1,K}(\zeta_i)$). Using $K = K_x = K_u$ point represent orthogonal collocation on finite elements. This procedure leads to the following algebraic constraint optimization over its time interval:

Algorithm 3.

$$\min_{x_{ij}, c_{ij}} \left[h(x_f, t_f) \right]$$
subject to:
$$x_{1,0} - x_0 = 0$$
$$x_{i,0}(\zeta_i) - x_{i-1,K}(\zeta_i) = 0 i = 2, ..., N j = 1, ..., K$$
$$x_f - x^N_{N+1,0}(\zeta_{N+1}) = 0$$
$$x^{LB} \leq x_{i,j} \leq x^{UB} i = 1, ..., N j = 1, ..., K$$
$$Eq(t_{i,j}, x_{i,j}, u_{i,j}) = 0$$
$$g(t_{i,j}, x_{i,j}, u_{i,j}) \leq 0$$

Now, MATLAB Optimization Toolbox can be chosen to minimize objective function with respect to nonlinear algebraic equality and inequality constraints. Output and inputs of nonlinear optimization are shown in Fig. 2. In this Figure, Φ vector is second-order polynomial base functions that consider as follows:

$$\Phi(x) = \{\phi_1(x), \phi_2(x), \cdots, \phi_{10}(x)\}^T = \left\{x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4, x_4^2\right\}^T \tag{18}$$

The output of this optimization is used in the nonlinear feedback controller as shown in Fig. 3.

3. Dynamic Load Carrying Capacity

DLCC of a manipulator is used as a criterion to measure the performance of a control system. It is defined as the maximum allowable load that the manipulator can carry on the specified path from the initial to the final point in the work space with a predefined acceptable accuracy. Other constraints that limit the DLCC are the upper and lower limits for output torque in joints, which can be calculated

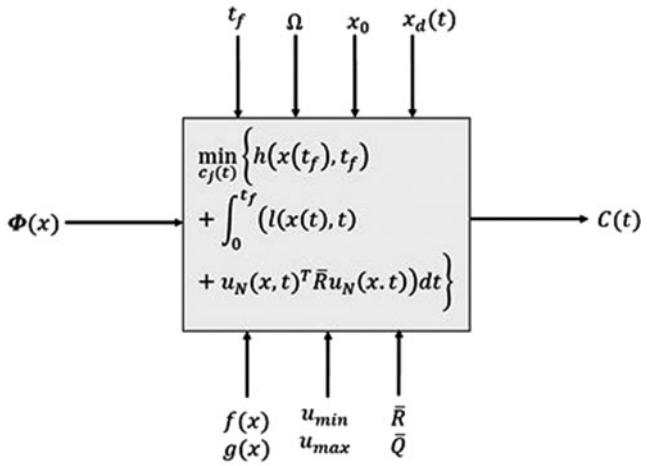

Fig. 2. Nonlinear optimization problem.

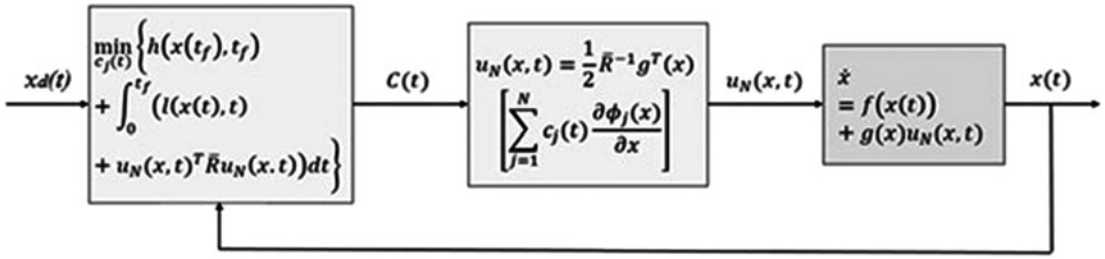

Fig. 3. Nonlinear feedback control system.

from the following equation:[21]

$$u_{max} = K_1 - K_2\omega(t) \tag{19}$$

$$u_{min} = -K_1 - K_2\omega(t) \tag{20}$$

where

$$K_1 = \left[\tau_{s_1}, \tau_{s_2}, ..., \tau_{s_n}\right]^T \tag{21}$$

$$K_2 = \left[\frac{\tau_{s_1}}{\omega_{s_1}}, \frac{\tau_{s_2}}{\omega_{s_2}}, ..., \frac{\tau_{s_n}}{\omega_{s_n}}\right]^T \tag{22}$$

τ_s is stall torque and ω_s is maximum no-load speed of the motor and $\omega(t)$ is vector of angular velocity of motor.

4. Simulation Results

As a first case study, a simple rigid manipulator with two links is shown in Fig. 4 is considered and the new proposed algorithm is applied to design control law and find the DLCC of the manipulator during the specified path. Also, characteristics of this manipulator are listed in Table I.

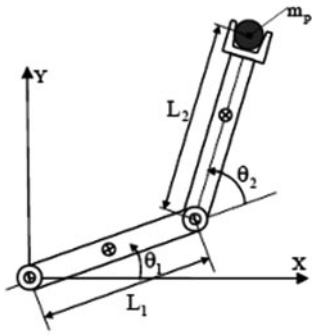

Fig. 4. Simple rigid manipulator.

Table I. Characteristics of manipulator.

Parameter	Value	Unit
Length of links	$L_1 = L_2 = 1$	m
Center of mass	$L_{c1} = L_{c2} = 0.5$	m
Mass	$m_1 = m_2 = 2$	kg
Moment of inertia	$I_1 = I_2 = 0.166$	kg m^2
Max.no load speed	$\omega_{s1} = \omega_{s2} = 5.6$	Rad/sec
Actuator stall torque	$\tau_{s1} = \tau_{s2} = 50$	N m

The state variables are selected to be $x_1 = \theta_1$; $x_2 = \theta_2$; $x_3 = \dot{\theta}_1$; $x_4 = \dot{\theta}_2$ so the state space model of manipulator is as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ D^{-1}\left[\tau - C - G\right] \end{bmatrix} \tag{23}$$

Matrix and vectors that appeared in Eq. (23) are as follows:

$$\begin{aligned} D &= \begin{bmatrix} a_1 + a_2 + 2a_3 cos(x_2) & a_2 + a_3 cos(x_2) \\ a_2 + a_3 cos(x_2) & a_2 \end{bmatrix} \\ G &= \begin{bmatrix} g(a_4 cos(x_1) + a_5 cos(x_1 + x_2)) \\ g a_5 cos(x_1 + x_2) \end{bmatrix} \\ C &= \begin{bmatrix} -a_3 sin(x_2)(2x_3 x_4 + x_4^2) \\ a_3 sin(x_2) x_3^2 \end{bmatrix} \end{aligned} \tag{24}$$

where constant coefficients are as follows:

$$\begin{aligned} a_1 &= I_1 + m_1 L_{c1}^2 + (m_2 + m_p) L_1^2 \\ a_2 &= I_2 + m_2 L_{c2}^2 + m_p L_2^2 \\ a_3 &= L_1 (m_2 L_{c2} + m_p L_2) \\ a_4 &= m_1 L_{c1} + L_1 (m_2 + m_p) \\ a_5 &= m_2 L_{c2} + m_p L_2 \end{aligned} \tag{25}$$

As it mentioned, second-order polynomial basic functions are selected as follows:

$$\{\phi_j\}_{j=1}^{10} = \{x_1^2, x_1 x_2, x_2^2, x_1 x_3, x_2 x_3, x_3^2, x_1 x_4, x_2 x_4, x_3 x_4, x_4^2\} \tag{26}$$

Final time t_f is selected to be 1 second and the proposed algorithm is applied to this manipulator to move on the path with the maximum available load carrying capacity. During simulation, the path with maximum load carrying capacity is examined with two criteria. The first criterion is the error

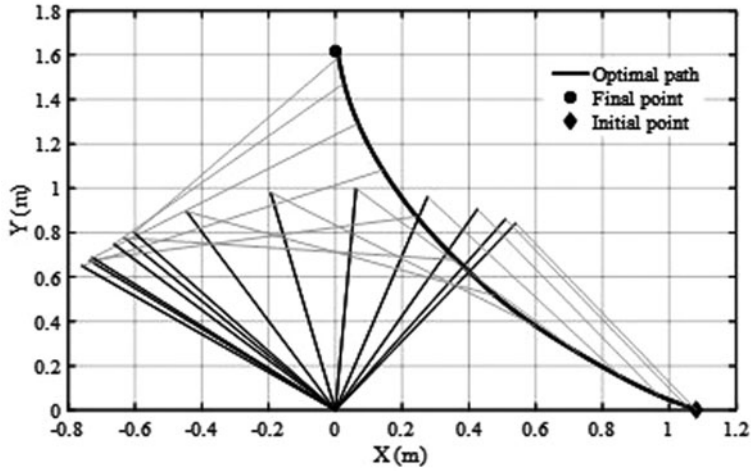

Fig. 5. Optimal path with maximum DLCC for point to point motion.

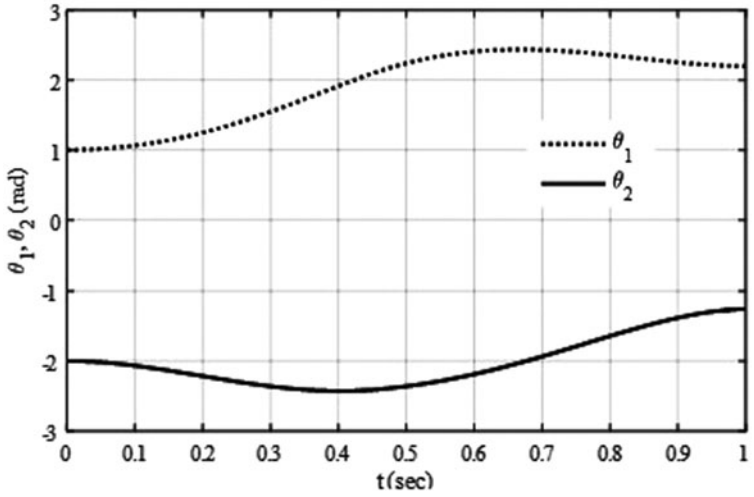

Fig. 6. Optimal angles with maximum DLCC.

between the actual and the desired angular displacement for the traveled path and the second criterion is the maximum amount of torque that should be between the lower and upper allowable torque. Also a point to point motion simulation is performed to compare validity and benefit of existing methodology and literature.

4.1. Point to point motion

A point to point motion is simulated from initial point $P_0(1.1, 0)$ to final point $P_f(0, 1.6)$ and results are compared with ref. [4]. In ref. [4], the DLCC was computed as 5.7 kg while in present work and using proposed methodology the DLCC is computed as 9 kg. Resulting end effector path and histories of angular position and velocity of links also joint torques are shown in Figs. 5–9.

4.2. Polynomial trajectory

For this specified path, the parameters that must be selected in the algorithm and the angular displacement of each link that represents the specified path of the robot end-effector motion are shown in Table II. The simulation is performed and the DLCC of this robot is computed as 5.8 kg and with this load carrying capacity the second criterion, that is, the maximum torque value checked. Figure 10 shows that the first criterion, which is the error between the actual and the desired angular displacement for the travelled path, is less than 0.1 rad. The specified path and the path travelled by the end effector are shown in Fig. 11 with the obtained maximum DLCC mp = 5.8 kg. Angular velocity

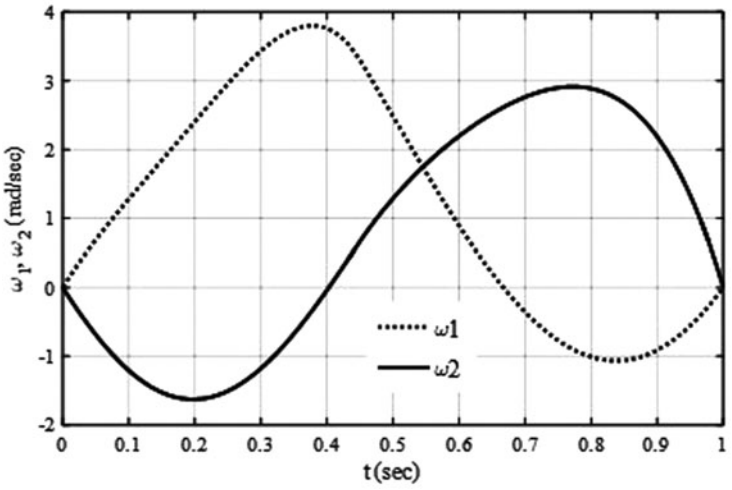

Fig. 7. Angular velocity of links.

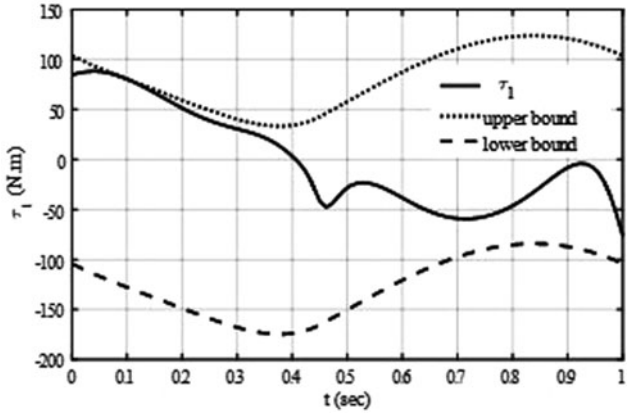

Fig. 8. Torque applied on joint 1.

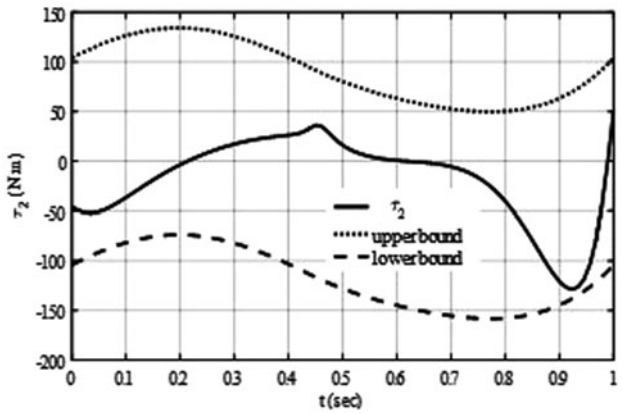

Fig. 9. Torque applied on joint 2.

of the links and angular position error are shown in Figs. 12 and 13. Torques need to be applied to the joints are shown in Figs. 14 and 15. These figures illustrate high positive torque at the beginning of motion and high negative torque to achieve zero velocity at the end of motion. The coefficients of the closed-loop controller, which are outputs of the nonlinear optimization problem, are illustrated in Figs. 16–18.

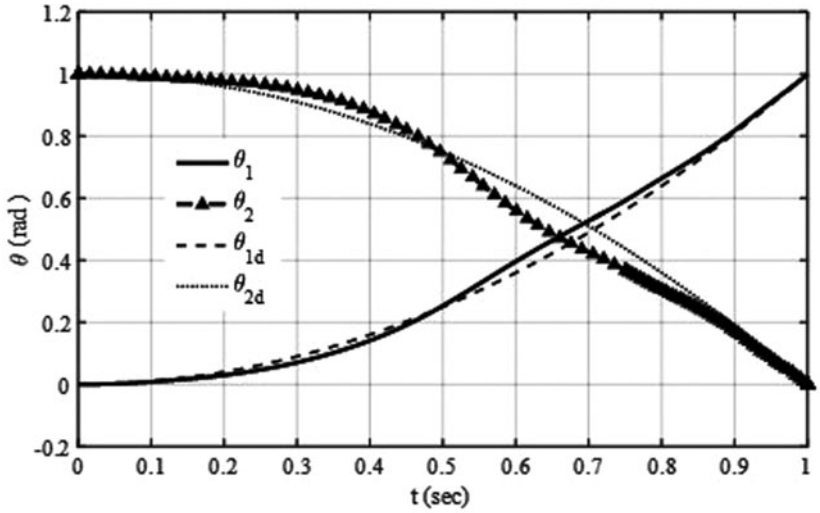

Fig. 10. Optimal angles with maximum DLCC.

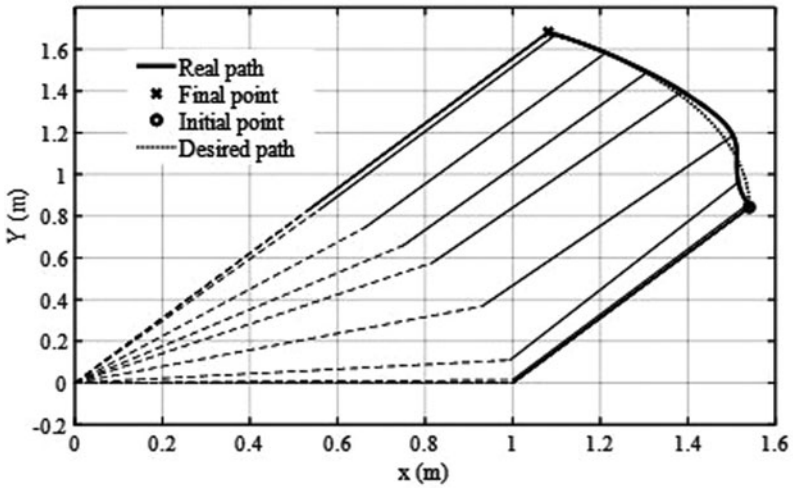

Fig. 11. Optimal path and configuration of links.

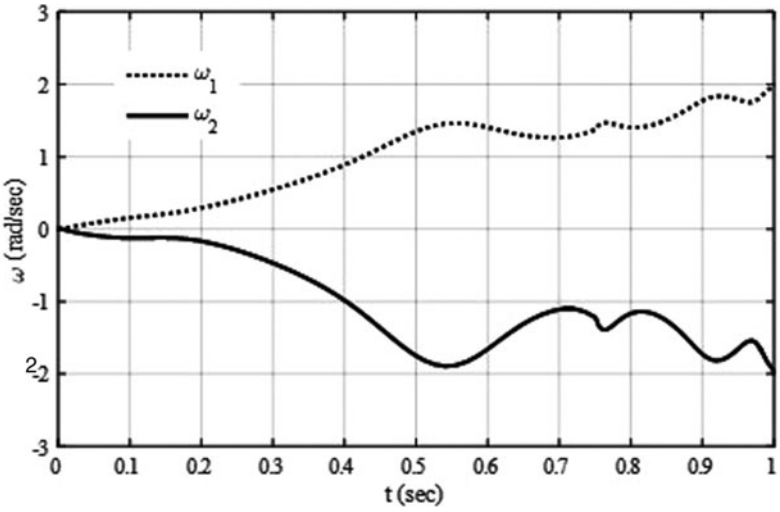

Fig. 12. Angular velocity of links.

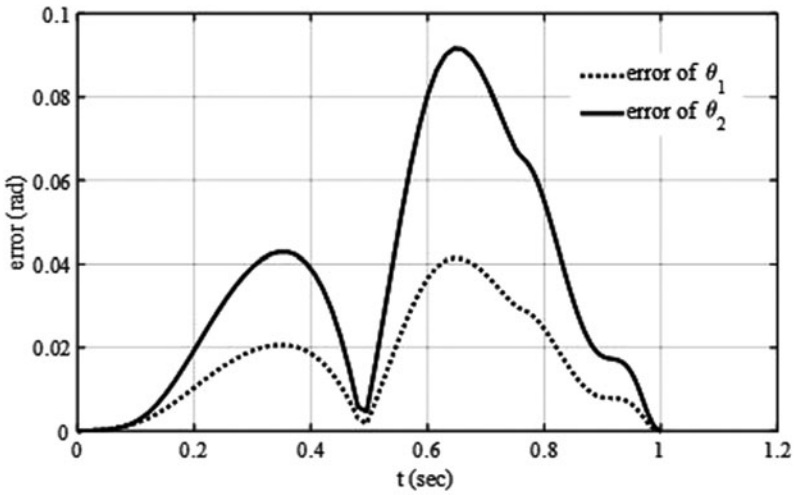

Fig. 13. Trajectory tracking error.

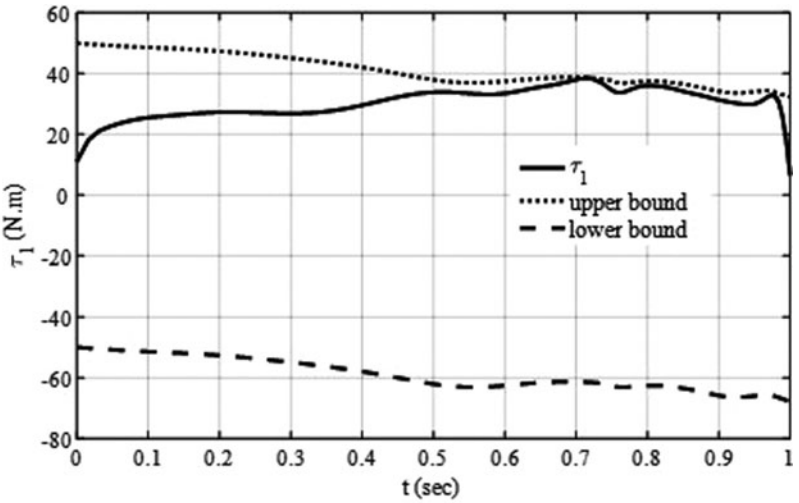

Fig. 14. Torque applied on joint 1.

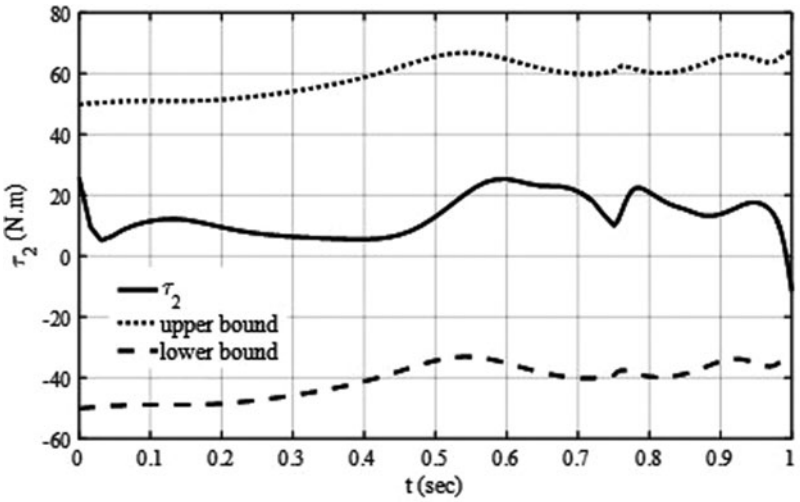

Fig. 15. Torque applied on joint 2.

Table II. Parameters of algorithm for finding DLCC.

Parameter	Value
The first angular displacement	$\theta_1 = t^2$
The second angular displacement	$\theta_2 = 1 - t^2$
Final time	$t_f = 1$ sec
Allowable tracking error	$\varepsilon = 0.1$ *rad*
Weight of states in J	$\bar{Q} = \text{diag}(0.1)$
Weight of control in J	$\bar{R} = \text{diag}(1)$

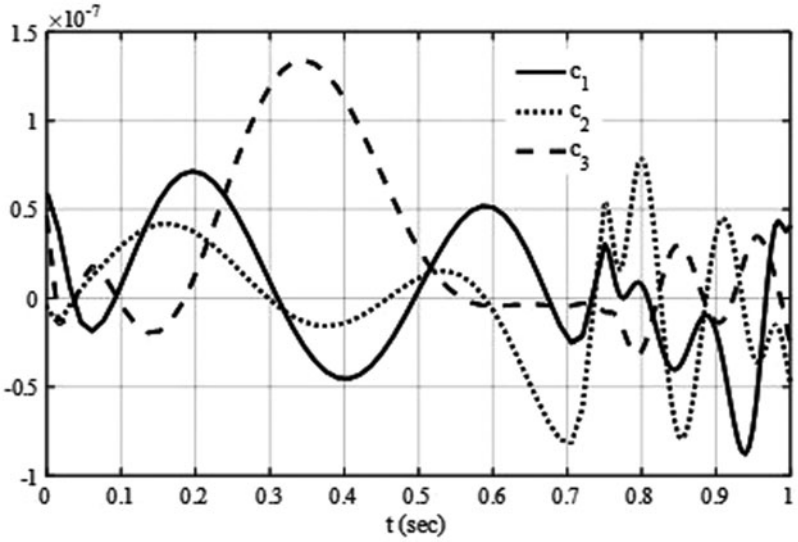

Fig. 16. Coefficients c_1, c_2, c_3.

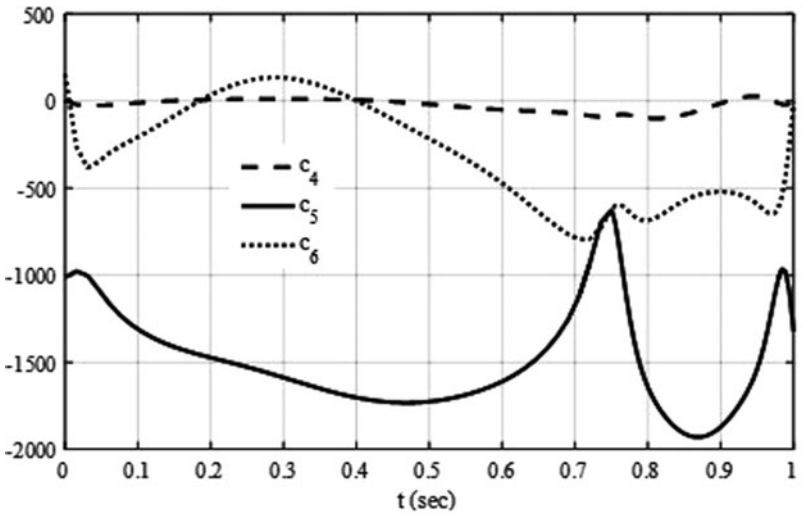

Fig. 17. Coefficients c_4, c_5, c_6.

4.3. Harmonic trajectory

In this simulation, harmonic functions are selected for the desired angle of links. Table III mentions the path parameters and the desired angular displacement of each link that represents the specified path of robot end-effector motion. DLCC of this robot is 5 kg and with this load carrying capacity, the first criterion is tested, error between actual and desired angular displacement for the travelled path. The actual and specified angles with maximum DLCC ($m_p = 5$ kg) are shown in Fig. 19. The optimal

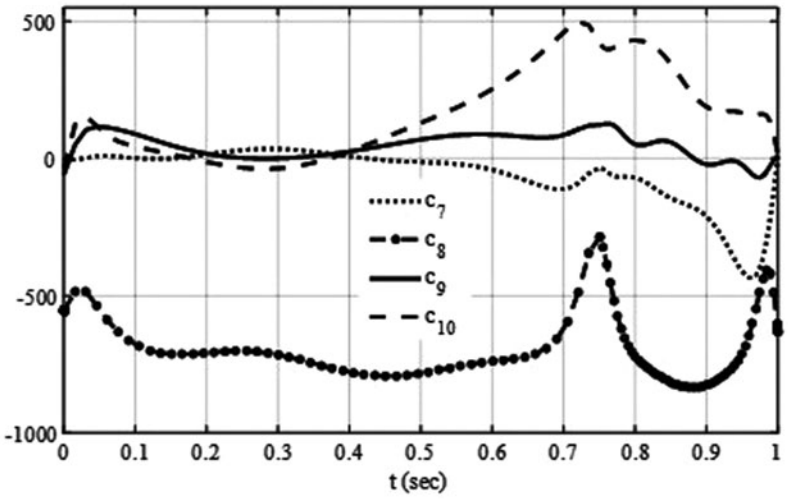

Fig. 18. Coefficients c_7, c_8, c_9, c_{10}.

Table III. Parameters of algorithm for finding DLCC.

Parameter	Value
The first angular displacement	$\theta_1 = \sin(\omega t)$
The second angular displacement	$\theta_2 = \cos(\omega t)$
Final time	$t_f = 1$ sec
Allowable tracking error	$\varepsilon = 0.15$rad
Weight of states in J	$\bar{Q} = \text{diag}(0.01)$
Weight of control in J	$\bar{R} = \text{diag}(1)$

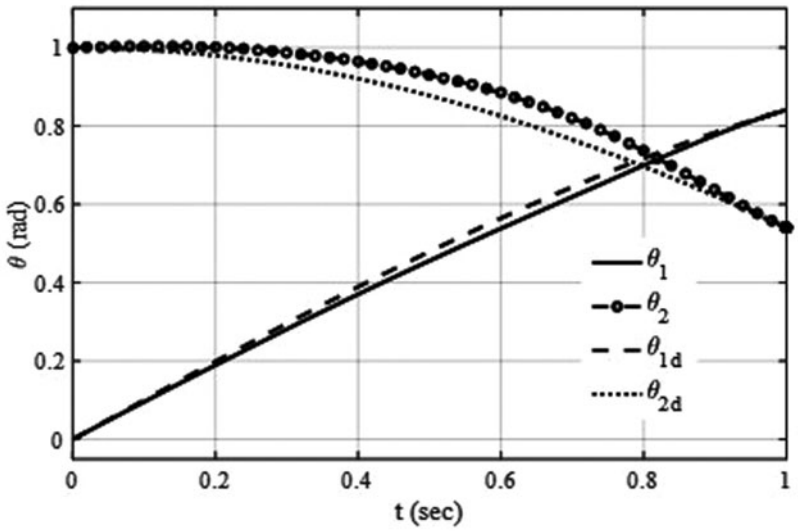

Fig. 19. Optimal angles with maximum DLCC.

end-effector position and the angular velocity of the links are shown in Figs. 20 and 21. Torques need to be applied to the joints are shown in Figs. 22 and 23. As in previous simulation, joint torque histories show a high positive torque at the beginning of motion and a high negative torque to achieve zero velocity at the end of motion. Closed-loop controller coefficients are mentioned in Figs. 24–26.

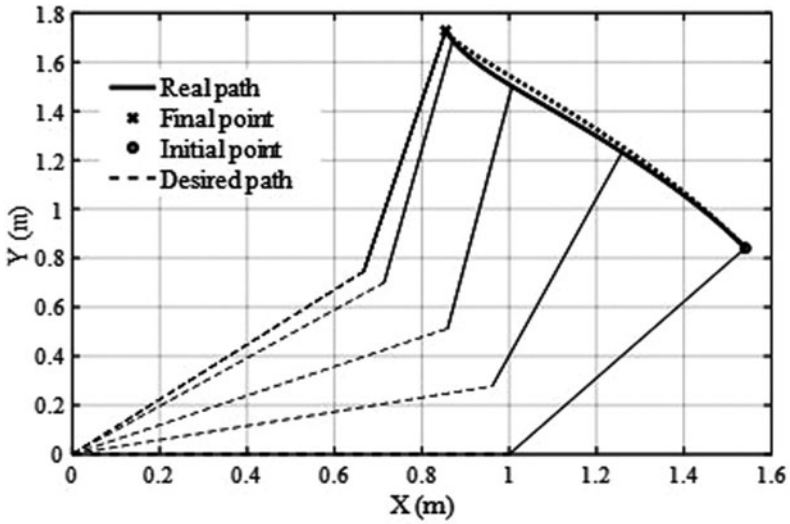

Fig. 20. Optimal end-effector position

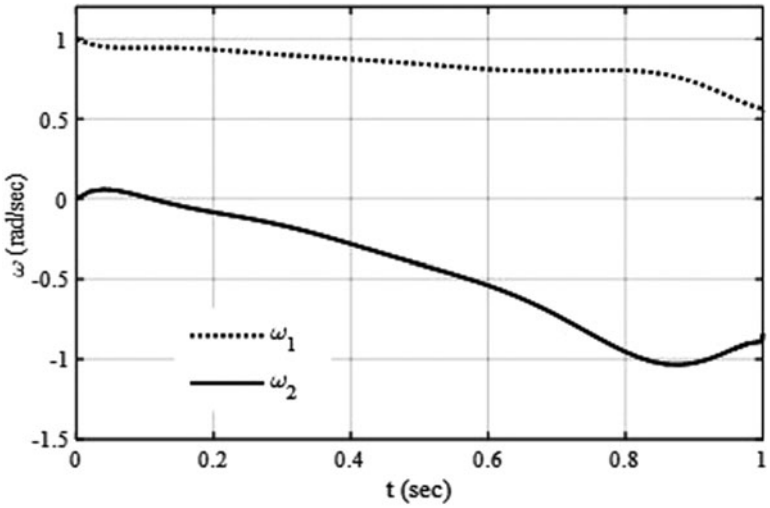

Fig. 21. Angular velocity of links.

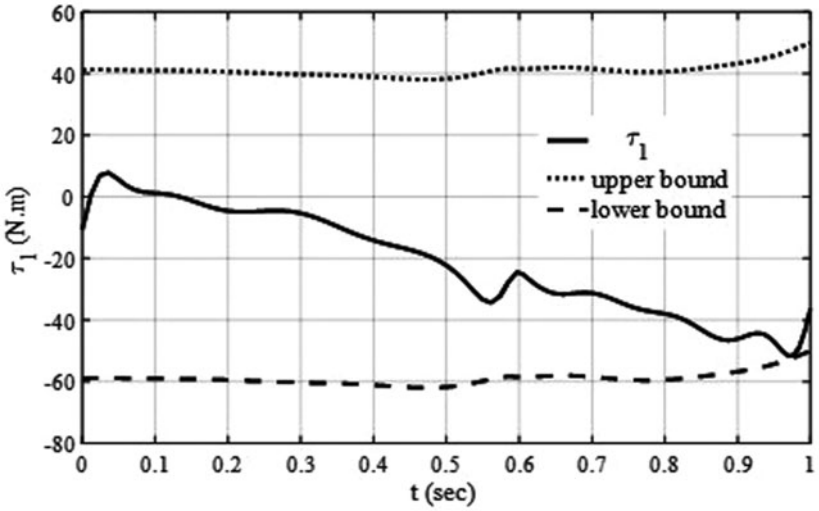

Fig. 22. Torque applied on joint 1.

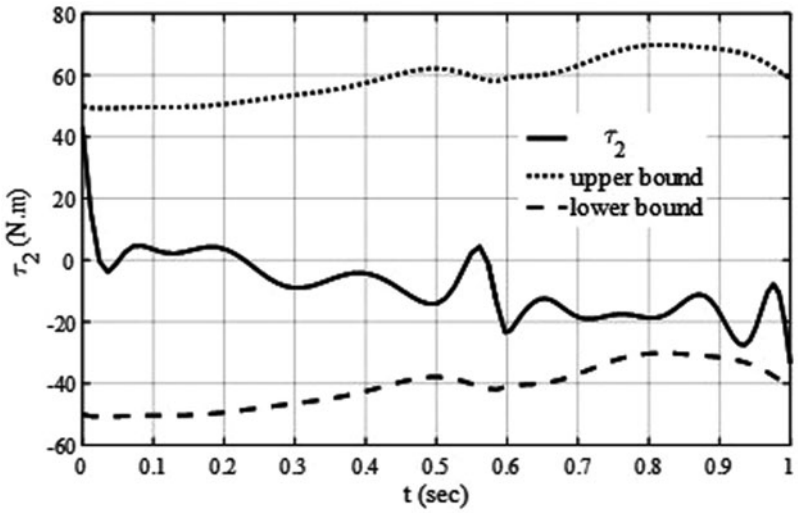

Fig. 23. Torque applied on joint 2.

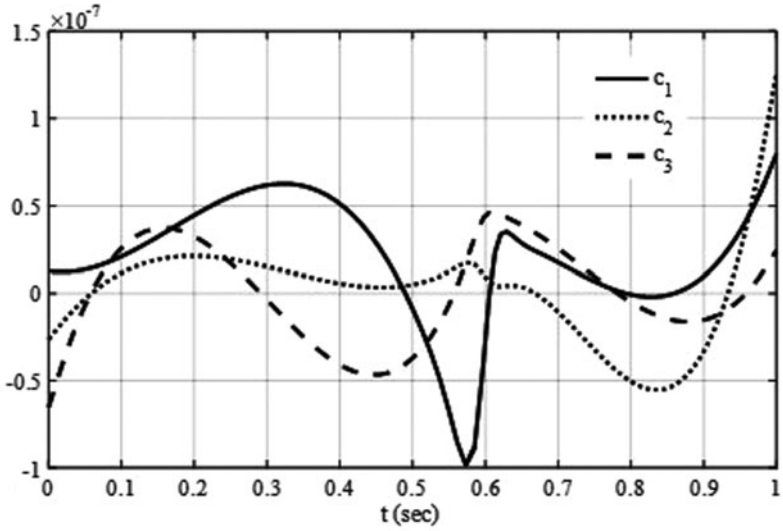

Fig. 24. Coefficients c_1, c_2, c_3.

As can be seen, the DLCC using polynomial trajectory is more than DLCC using harmonic trajectory and the error accrued is less than harmonic trajectory so in this case, polynomial trajectory is better than harmonic trajectory.

5. Experimental Test

To show the capability of the proposed tracking controller design methodology, a simulation for a three link manipulator with fixed base (Fig. 27) is applied and compared with experimental results. Also, according to the existing limitations, only two links are examined and one of the links is fixed. In this mechanical arm, the end effector of the robot is first placed at the point where it is intended. Reverse kinematics is used to position the end effector of the robot in the desired position. In fact, with the robot's end point and the direction that the robot should approach this point, the robot's coordinates are positioned in such a way that the robot is in the desired position. Characteristics of this robot can be seen in Table III. Specification of the end-effector path and other parameters are shown in Tables IV and V, respectively.

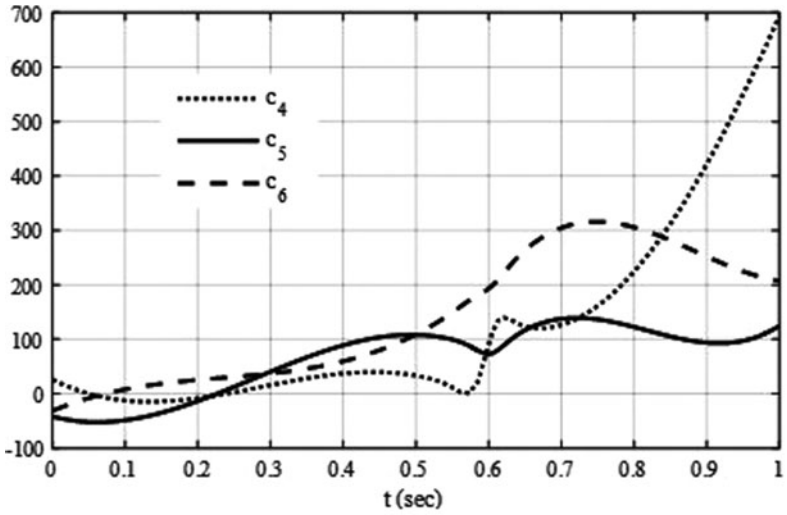

Fig. 25. Coefficients c_4, c_5.

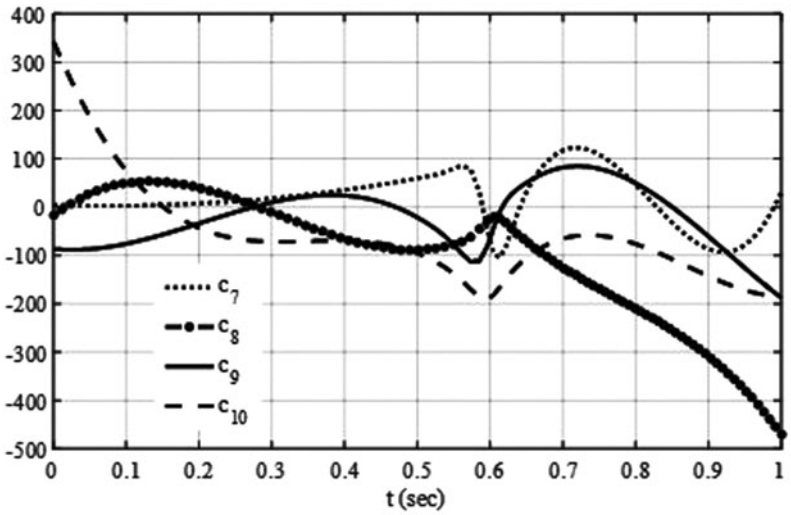

Fig. 26. Coefficients c_6, c_7, c_8, c_9, c_{10}.

Fig. 27. Experimental set.

Table IV. Characteristics of manipulator.

Parameter	Value	Unit
Length of links	$L_1 = 0.21$ $L_2 = 0.18$	m
Center of mass	$L_{c1} = 0.105$ $L_{c2} = 0.09$	m
Mass	$m_1 = 0.42$ $m_2 = 0.36$	kg
Moment of inertia	$I_1 = I_2 = 0.166$	kg m^2
Max.no load speed	$\omega_{s1} = \omega_{s2} = 5.6$	Rad/sec
Actuator stall torque	$\tau_{s1} = \tau_{s2} = 50$	N m

Table V. Parameters of algorithm for finding DLCC of manipulator.

Parameter	Value
The first angular displacement	$\theta_1 = t^2$
The second angular displacement	$\theta_2 = 1 - t^2$
Final time	$t_f = 1$ sec
Allowable tracking error	$\varepsilon = 0.08$ rad
Weight of states in J	$\bar{Q} = \text{diag}(0.01)$
Weight of control in J	$\bar{R} = \text{diag}(1)$

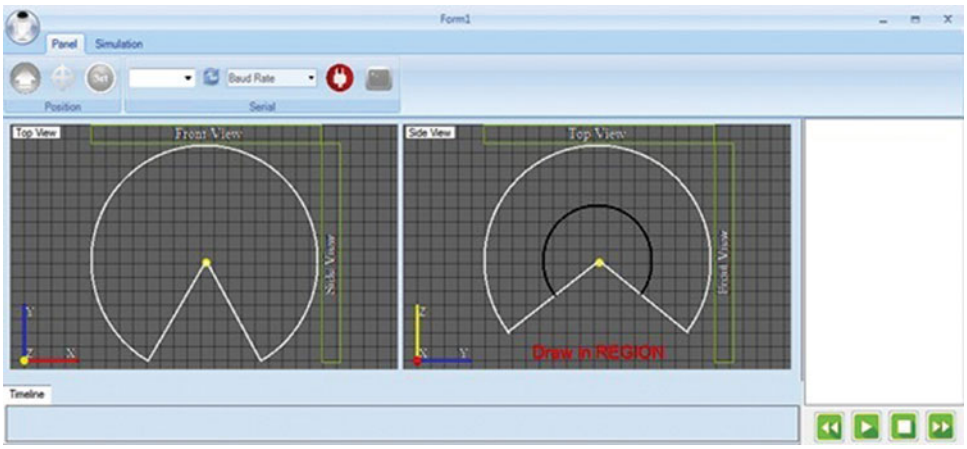

Fig. 28. GUI of robot software.

A view of robot software that is programed using Visual C#, is shown in Fig. 28. Desired trajectory is the input of software and controller is implemented in LPC2368 microcontroller. LPC2368 produces required PWM to run AX-12A DC servo motors.

The algorithm is applied to the robot and the optimal path is obtained when the robot carries the maximum load. Maximum load carrying capacity of the robot during the specified path with an error less than 0.08 rad is equal to 5kg. The specified path and path traveled by the manipulator by considering the maximum DLCC have been shown in Fig. 29. The actual and desired angular position and speed of links are shown in Figs. 30 and 31. Error history of the path has been shown in Fig. 32, also torques needed to be applied on joints are presented in Figs. 33 and 34.

These figures illustrate high positive torque at the beginning of motion and high negative torque to achieve zero velocity at the end of motion. The coefficients of the closed-loop controller, which are outputs of the nonlinear optimization problem, are shown in Figs. 35–37. Finally, an experimental

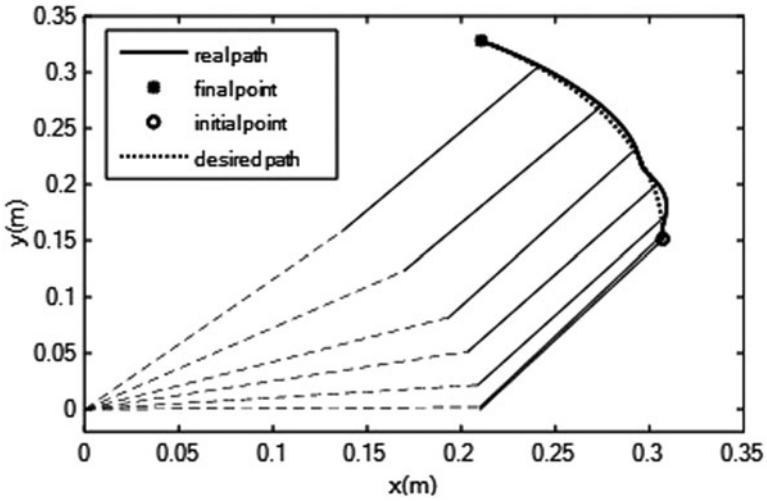

Fig. 29. Optimal and desired path with maximum DLCC.

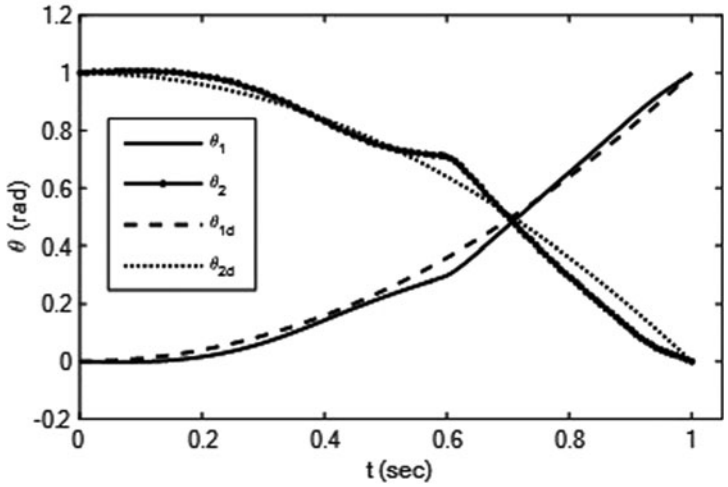

Fig. 30. Optimal angular position of links.

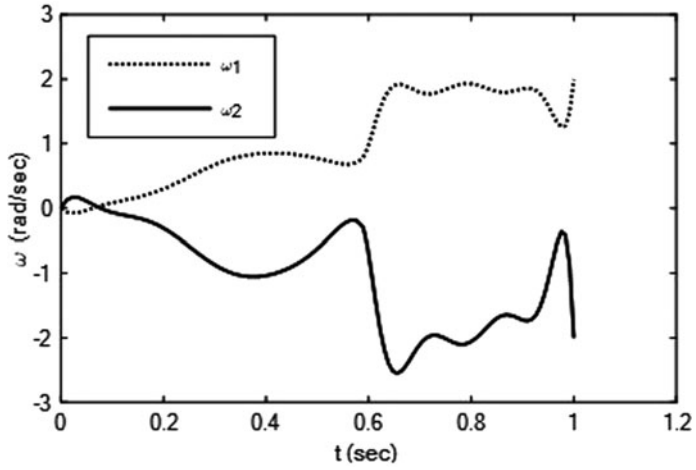

Fig. 31. Angular velocity of links.

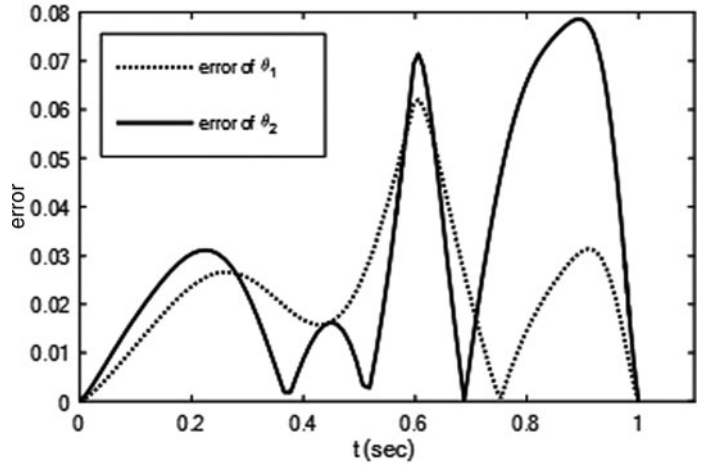

Fig. 32. Tracking error.

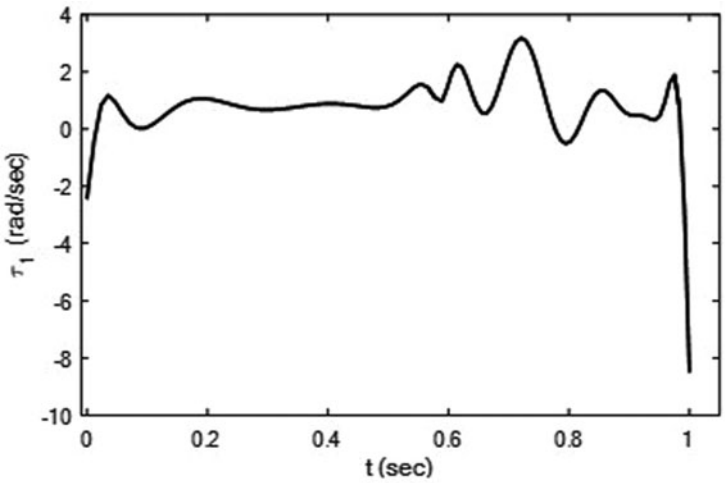

Fig. 33. Torque applied on joint 1.

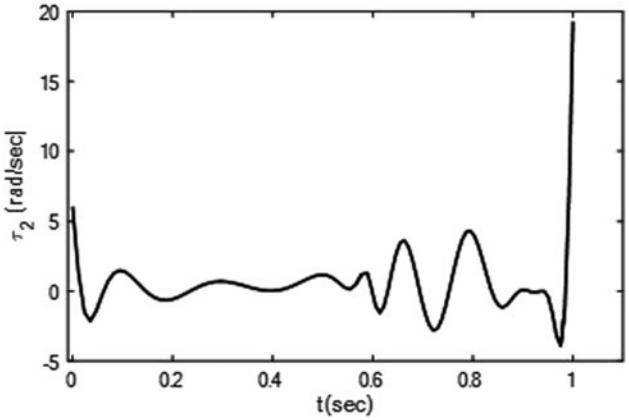

Fig. 34. Torque applied on joint 2.

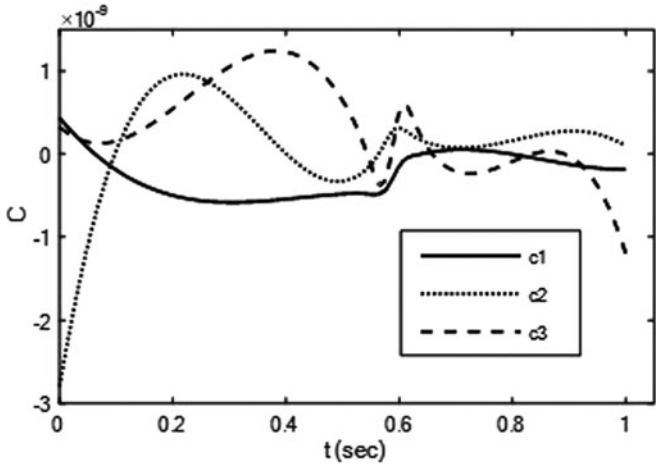

Fig. 35. Coefficients c_1, c_2, c_3.

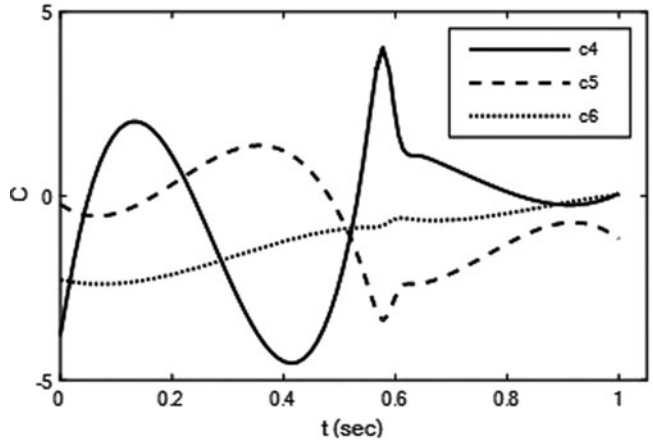

Fig. 36. Coefficients c_4, c_5.

test was performed to compare simulation and actual results. Figures 38 and 39 show the simulated and actual angular position of the links and illustrate the appropriate tracking accuracy.

6. Conclusion

The purpose of this paper is to use a combined direct–indirect methods for designing a tracking controller, which realizes a pre-determined path for a manipulator with the lowest possible error and maximum load carrying capacity. Main contribution from this work is listed below:

- Unlike previous studies, that the linear optimization method and the open loop optimal control method have been used, the proposed method has not only a closed-loop form and therefore is applicable, but also is based on nonlinear optimization technique.
- The method for solving a nonlinear optimal controller in a closed-loop form is based on the solution of the HJB equation (indirect part). To reduce the required computation in solving this equation, the coefficients appeared in the solution of the HJB equation are determined using a new direct nonlinear optimization method.
- The simulation was performed to confirm the results for two different angular trajectories. The results show that the proposed method can be used as well for tracking specified trajectory.
- It is shown that using polynomial trajectory the DLCC is more than one using harmonic trajectory. and the error accrued is less than harmonic trajectory

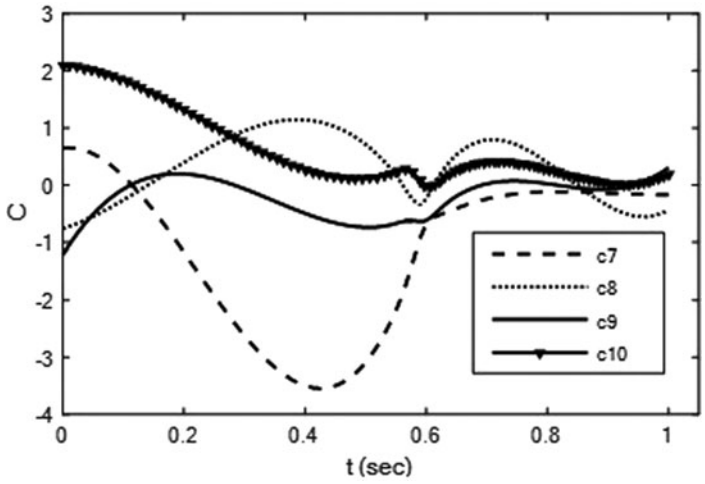

Fig. 37. Coefficients c_6, c_7, c_8, c_9, c_{10}.

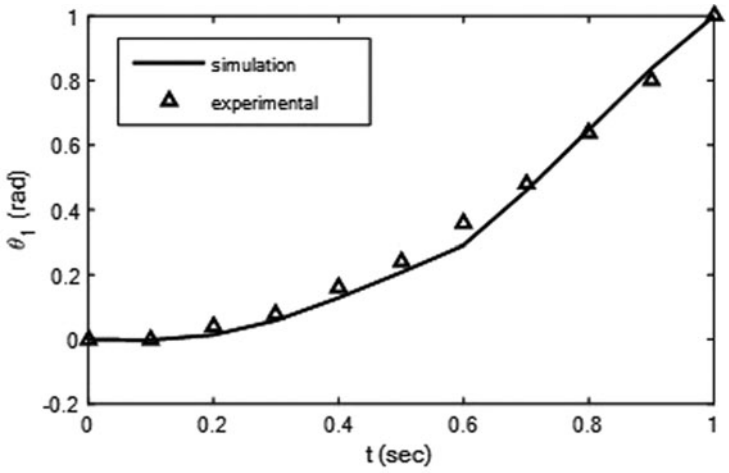

Fig. 38. Angular position of the first joint.

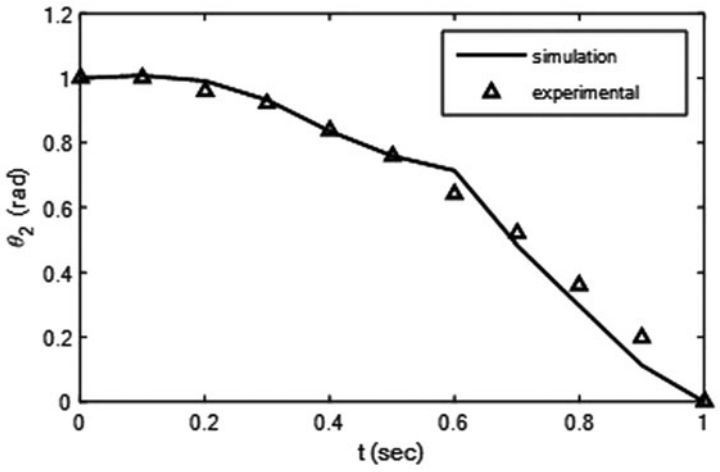

Fig. 39. Angular position of the second joint.

- Experimental test express the applicability of the proposed method which is based on the fact that the method has a closed-loop optimal manner.

References

1. M. Kong, Z. Chen, C. Ji and W. You, "Optimal Point-to-point Motion Planning of Heavy-Duty Industry Robot with Indirect Method," *International Conference on Robotics and Biomimetics* (2013) pp. 768–773.
2. N. Kashiri, M. H. Ghasemi and M. Dardel, "An iterative method for time optimal control of dynamic systems," *Arch. Control Sci.* **21**(1), 5–23 (2011).
3. M. H. Korayem, M. Irani and S. Rafee Nekoo, "Motion control and dynamic load carrying capacity of mobile robot via nonlinear optimal feedback," *Int. J. Manuf. Mater. Sci.* **2**(1), 16–21 (2012).
4. M. H. Korayem and M. Irani, "New optimization method to solve motion planning of dynamic systems: Application on mechanical manipulators," *Multibody Syst. Dyn.* **31**(2), 169–189 (2014).
5. M. Kong, Z. Chen, C. Ji and M. Liu, "Optimal Point-to-point Motion Planning of Flexible Parallel Manipulator with Adaptive Gauss Pseudo-Spectral Method," *International Conference on Advanced Intelligent Mechatronics (AIM)* (2014) pp. 852–858.
6. A. H. Korayem, M. Irani Rahagi, H. Babaee and M. H. Korayem, "Maximum load of flexible joint manipulators using nonlinear controllers," *Robotica* — **35**(1), pp. 119-142 (2017).
7. A. M. Shafei and M. H. Korayem, "Theoretical and experimental study of DLCC for flexible robotic arms in point-to-point motion," *Optim. Control Appl. Methods* **38**(6), 963–972 (2017).
8. M. H. Korayem, A. M. Shafei and H. R. Shafei, "Dynamic modeling of nonholonomic wheeled mobile manipulators with elastic joints using recursive Gibbs-Appell formulation," *Sci. Iran.* **19**(4), 1092–1104 (2012).
9. M. H. Korayem, M. Irani Rahagi, and S. Rafee Nekoo, "Load maximization of flexible joint mechanical manipulator using nonlinear optimal controller," *Acta Astronaut* **69**(7–8), 458–469 (2011).
10. W. Guo, R. Li, C. Cao, X. Tong and Y. Gao, "A new methodology for solving trajectory planning and dynamic load-carrying capacity of a robot manipulator," *Math. Probl. Eng*. (2016) 28 pages.
11. J. Wu, X. Chen, L. Wang and X. Liu, "Dynamic load-carrying capacity of a novel redundantly actuated parallel conveyor," *Nonlinear Dyn.* **78**(1), 241–250 (2014).
12. M. Kong, Z. Chen and C. Ji, "Optimal Point-to-point Motion Planning of Flexible Parallel Manipulator with Multi-Interval Radau Pseudospectral Method," *MATEC Web of Conferences, 42, EDP Science* (2016).
13. L. Fen, Z. Jiang-hai, S. Xiao-bo, Z. Pei-ying, F. Shi-hui and L. Zhong-jie, "Path Planning of 6-DOF Humanoid Manipulator Based on Improved Ant Colony Algorithm," *Control and Decision Conference (ccdc)* (2012) pp. 4158–4161.
14. A. Gallant and C. Gosselin, "Parametric trajectory optimisation for increase payload," *Trans. Canadian Soc. Mech. Eng.* **40**(2), 125–137 (2016).
15. M. H. Korayem, V. Azimirad and M. Irani Rahagi, "Maximum allowable load of mobile manipulator in the presence of obstacle using non-linear open and closed loop optimal control," *Arab. J. Sci. Eng.* **39**(5), 4103–4117 (2014).
16. C. T. Chen and T. T. Liao, "Trajectory planning of parallel kinematic manipulators for the maximum dynamic load-carrying capacity," *Meccanica* **51**(8), 1653–1674 (2016).
17. D. E. Kirk, *Optimal Control Theory*. (Prentice-Hall Inc, Englewood Cliffs, New Jersey, USA, 1970).
18. C. Fletcher, "Computational Galerkin Methods," **In:** *Springer Series in Computational Physics* (Springer Verlag, New York, 1984).
19. S. C. Beeler, H. T. Tran and H. T. Banks, "Feedback control methodologies for nonlinear systems," *J. Optim. Theory Appl.* **107**(1), 1–33 (2000).
20. M. Cizniar, M. Fikar and M. A. Latifi, "Matlab Dynamic Optimisation Code," Institute of Information Engineering, Automation, and Mathematics, Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Radlinsk´eho 9, 812 37 Bratislava, Slovak Rebublic (2009).
21. M. H. Korayem and A. Nikoobin, "Maximum payload for flexible joint manipulators in point-to point task using optimal control approach," *Int. J. Adv. Manuf. Technol.* **38**, 1045–1060 (2008).