Robotica (2013) volume 31, pp. 643–656. © Cambridge University Press 2012
doi:10.1017/S0263574712000628

Trajectory planning of mobile manipulators using dynamic programming approach

M. H. Korayem*, M. Irani, A. Charesaz, A. H. Korayem and A. Hashemi

Robotic Research Laboratory, Center of Excellence in Experimental Solid Mechanics and Dynamics, School of Mechanical Engineering, Iran University of Science and Technology, P.O. Box 18846, Tehran, Iran

(Accepted September 28, 2012. First published online: November 19, 2012)

SUMMARY

This paper presents a solution for optimal trajectory planning problem of robotic manipulators with complicated dynamic equations. The main goal is to find the optimal path with maximum dynamic load carrying capacity (DLCC). Proposed method can be implemented to problems of both motion along a specified path and point-to-point motion. Dynamic Programming (DP) approach is applied to solve optimization problem and find the positions and velocities that minimize a pre-defined performance index. Unlike previous attempts, proposed method increases the speed of convergence by using the sequential quadratic programming (SQP) formulation. This formulation is used for solving problems with nonlinear constraints. Also, this paper proposes a new algorithm to design optimal trajectory with maximum DLCC for both fixed and mobile base mechanical manipulators. Algorithms for DLCC calculations in previous works were based on indirect optimization method or linear programming approach. The proposed trajectory planning method is applied to a linear tracked Puma and the mobile manipulator named Scout. Application of this algorithm is confirmed and simulation results are compared with experimental results for Scout robot. In experimental test, results are obtained using a new stereo vision system to determine the position of the robot end-effector.

KEYWORDS: DLCC; Dynamic programming; Sequential Quadratic Programming (SQP); Mobile manipulators; Stereo vision.

Nomenclature

F	Nonlinear state equation
x	Vector of state variables
u	Vector of control inputs (torques)
x_0	Initial state
x_f	Final state
$J(x)$	Performance index
$h(x(t))$	Equality constraint
$g(x(t))$	Inequality constraint
q	Generalized coordinate of the system
p	Cartesian displacements of the manipulator end-effector
$T(q)$	Transformation matrix
$j(q)$	Jacobian matrix
$M(q)$	Inertial matrix of manipulator
$C(q, \dot{q})$	Vector of Coriolis and centrifugal forces
$G(q)$	Vector of gravity force
$a(q)$	Lower bound of velocity
$b(q)$	Upper bound of velocity
$v(q, \dot{q})$	Lower bound of torques
$w(q, \dot{q})$	Upper bound of torques
$q_{desired}$	Desired generalized coordinate of the system
Δt	Time of travel between sequential two points
t_f	Final time
U_{i-min}, U_{i-max}	Minimum and maximum allowable torque of ith motor
U_{si}, ω_{si}	Stall torque and no load speed of ith motor
d	Deviation vector
λ	Multipliers vector

* Corresponding author. E-mail: hkorayem@iust.ac.ir

1. Introduction

A major benefit of robotic manipulators is their ability to move on a pre-specified path or between two pre-defined points. Planning a proper trajectory is a fundamental important issue to complete such operations. Trajectory planning has been defined as finding the optimal trajectory from a specific origin to a specified destination (point-to-point motion) or finding history of angular positions and velocities and joint torques for pre-specified end-effector trajectory (pre-specified path). To solve these problems, an optimization problem is defined and then direct or indirect methods are applied to solve this optimization problem. In the optimization problem, optimal angular position and velocity of joints are generated subject to equality, inequality, kinematics, and dynamics constraints.

As mentioned above, two methods can be used to solve an optimization problem: direct and indirect. Direct methods are based on direct search, such as linear and nonlinear programming methods. Indirect methods use optimality conditions to convert the optimization problem into a set of differential equations and try to solve them. Usually, calculus of variations approach is applied to solve the optimization problem. Hamilton-Jacobi-Belman equation

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

and Pontryagin's minimum principle are two well-known indirect approaches.

The dynamic load carrying capacity (DLCC) of a manipulator for a given end-effector trajectory could be defined by various criterions. The maximum load that the manipulator can carry on the optimal trajectory with sufficient accuracy considering the limitation on applied torques is the best. The main novelty of this paper is to consider the problem of finding DLCC and related optimal trajectory as a trajectory-planning problem and implementation of dynamic programming approach to solve it.

Wang and Ravani[1] have formulated the problem of finding the DLCC as an optimization problem using the state space of the dynamic equation of motion. Korayem *et al*.[2] have used iterative linear programming (ILP) method to solve the optimization problem of finding the DLCC for cable driven robots. Korayem *et al*.[3] have converted the problem of finding the DLCC of a mobile manipulator to a trajectory optimization problem. They have applied ILP method to solve this problem. In ref. [4], the DLCC of a manipulator with flexible link that was mounted on a vehicle has been determined using feedback linearization approach. In ref. [5], the maximum load carrying capacity of a mobile robot has been calculated in an environment with obstacles using open loop optimal control approach and considering stability constraints. The stability constraint has been considered by computing the zero moment point (ZMP).

This paper presents a method for determining the optimal trajectory of manipulators considering nonlinear equations of its motion and nonlinear constraints of the joint torques, angular velocities, and positions. A proper approach in the optimal control method is discrete dynamic programming (DDP) that is based on the principle of optimality and sequential quadratic programming (SQP). The purpose of using the dynamic programming approach is to develop a general algorithm. It motivates the user to take it into account for a variety of equality and inequality constraints.

Optimal control of nonlinear systems is one of the most interesting subjects in control theory. Bakker *et al*.[6] have presented an algorithm for hierarchical path planning based on Markov decision processes and dynamic programming. An optimized path planning approach has been presented for flexible dynamic systems.[7] The performance index has included both minimum energy and time terms. Feed forward command profile has been generated by solving a discrete-time optimal control problem via DDP. Gasparreto and Zanotto[8] have presented a new technique for optimal trajectory planning of robot manipulators. In proposed technique, an objective function has been minimized that composed of two terms: the first term is proportional to the total execution time and another one is proportional to the integral of the squared jerk. The proposed technique has taken into account the kinematic constraints expressed as upper bounds on the absolute values of velocity, acceleration, and jerk.

A genetic algorithm with a multi-criteria fitness function has been used for the end-effector motion planning of a redundant manipulator.[9] In ref. [10], two optimal control algorithms have been presented and analyzed consisting of a linear quadratic tracker (LQT) and a dynamic programming (DP) schemes for a differentially driven mobile robot. Stilman *et al*.[11] have used dynamic programming approach in a reduced dimensional subspace and have described an algorithm of time optimal trajectory planning for a 4 DOF biped robot with point feet. The trajectory has been assumed in joint space by a single parameter. By inserting this parameter into the dynamic equations, constraints on input torques are transformed into the second derivative of such a single parameter. In ref. [12], the work space has been discretized into several points. Each point represents one position or configuration of the robot and the optimal path (from start point to goal point) has been searched among these points. Lee and Shih[13] have found an ideal geometric path that consists of several line segments in the Cartesian work space and then a free dynamic motion of minimum time collision has been determined near the ideal geometric path.

In this paper, the problem of finding the optimal trajectory is formulated and then dynamic programming approach is applied to solve this problem for both pre-defined path and point-to-point tasks. This formulation is used in a new algorithm to find the optimal trajectory with maximum dynamic load for mechanical manipulators. Simulations are performed for a linear tracked Puma and wheeled mobile manipulator and finally experimental test is carried out for Scout robot and obtained results are compared with simulations. Actual position of the end-effector is measured using stereo vision system that results in a low-level error (less than 1 mm).

2. Formulation of Optimal Trajectory Planning

Optimal trajectory planning problem can be formulated as an optimal control problem. Consider a dynamic system described by state space model as

$$\dot{x} = f(x(t), u(t)). \tag{1}$$

The initial state is given by $x(t_0) = x_0$, the final state is $x(t_f) = x_f$, and the final time is t_f that is either specified or free. The problem is to find the control input vector $u(t)$ and associated state vector $x(t)$ to minimize a performance index that is mentioned by Eq. (2).

$$J(x) = J(x(t_f), t_f) + \int_0^{t_f} L(x(t), u(t))dt \tag{2}$$

In this equation, $L()$ is a positive scalar function and $J(x(t_f), t_f)$ is a constant positive scalar. As it will be shown, the performance index determines the type of optimization problem and leads us to time minimization or energy minimization or mixed time and energy minimization problems. The equality and inequality constraints on the control and state variables can be described by

$$h(x(t), u(t)) = 0 \tag{3}$$

and

$$g(x(t), u(t)) \leq 0. \tag{4}$$

For modeling the kinematic and dynamic equations, q represents the vector of generalized coordinates of the manipulator that is the joint angular position while P represents the position of end-effector.

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

The relation between the end-effector position and the joint coordinates is considered as equality constraint

$$p = T(q),\ P \in R^3, q \in R^n. \tag{5}$$

Also, the kinematic relation between the end-effector velocity and the rate of generalized coordinates (joint angular velocities) is determined as

$$\dot{p} = j(q)\dot{q}, \tag{6}$$

where *j(q)* is the Jacobean matrix. The dynamic motion equation of the manipulator can be stated as

$$U = M(q)\ddot{q} + C(q, \dot{q}) + G(q). \tag{7}$$

This equation can be written in the form of Eq. (1). According to Eqs. (3) and (4), the problem has inequality constraints that describe the specified path and limitations on the joint torques and velocities as

$$a(q) \le \dot{q} \le b(q) \tag{8}$$

and

$$v(q, \dot{q}) \le U \le w(q, \dot{q}), \tag{9}$$

where a and b are the lower and upper bounds of velocity vector while v and w are the lower and upper bounds of the joint torques, respectively. For applying dynamic programming, the dynamic equation of motion (Eq. 7) must be rewritten in the following discrete form:

$$\begin{aligned}\mathrm{M}(q(k))[\dot{q}(k+1) - \dot{q}(k)] - [C(q(k)), \dot{q}(k)) \\ +G(q(k))]\Delta t(k) = u(k)\Delta t(k).\end{aligned} \tag{10}$$

Also, Eq. (2) must be rewritten in discrete form and it should be considered that there are two types of performance index depending on the type of trajectory planning problem: pre-specified path or point-to-point tasks. The discrete performance index of these two problems can be mentioned as:

1. Pre-specified path

$$J = \sum_{j=1}^{N} b\dot{q}_i(j)^2 + cu_i(j)^2, \tag{11}$$

where $\dot{q}_i$ and u_i represent vectors of joint velocities and input joint torques and also b and c are weighting factors. In pre-specified motion, joint variables (*q(t)*) are determined using Eq. (5), and it is not necessary that any term is related to *q(t)* in Eq. (11). The first term in Eq. (11) limits the joint velocities and the second term restricts required joint torques. The smoothness of joint velocity profile can be manipulated via different b and c coefficients in Eq. (11).

2. Point-to-point motion

$$J = \sum_{j=1}^{N} a(q_i(j) - q_i(f))^2 + b(\dot{q}_i(j) - \dot{q}_i(f))^2 + d(u_i(j))^2, \tag{12}$$

where q_f and $\dot{q}_f$ are final values of angular displacement and velocity of the joints, respectively. Final values of joint variables must be considered in performance index to force the optimization algorithm to reach the final point. Finally, the third term in Eq. (12) is added to limit the joint torque in order to increase DLCC of manipulator.

3. Dynamic Programming Algorithm for Pre-specified Trajectory Planning

Let k be the index of the discrete points along the given path ($k = 1, 2, 3, \ldots .N$). The corresponding joint displacement $q(k)$ can be obtained by solving the manipulator kinematic equations. Desired angular displacement of the joints in any discrete point is mentioned as $q_{desired}(k)$. It is required to introduce an inequality constraint described by Eq. (13) to force the end-effector to follow the desired path

$$q(k) - q_{desired}(k) < \delta. \tag{13}$$

It is assumed that the distance of traveling between points k and $k+1$ is small and the acceleration is constant. The acceleration of the joint i at point k is

$$\ddot{q}_i(k) = \frac{\dot{q}_i(k+1) - \dot{q}_i(k)}{\Delta t}. \tag{14}$$

Also Δt, the time of traveling between sequential two points, is given by

$$\Delta t = \frac{t_f - t_0}{N}. \tag{15}$$

According to Bellman's optimality principle, the objective function is defined as

$$\begin{aligned}J^*(q(k), u(k), k) = \min[J(\varphi(\dot{q}(k), u(k), k) \\ +J(u(k), u)(k+1), k)]\end{aligned} \tag{16}$$

$$J(\varphi(\dot{q}(k), u(k), k) = a(\dot{q}(k)^2) + b(u(k)^2) \tag{17}$$

$$J(u(k), u(k+1), k) = c((u(k) - u(k+1))^2), \tag{18}$$

where a, b, and c are weighting factors and second term in the right hand of Eq. (16), which is described by Eq. (18), that decrease the jerk. However, usually the coefficient c is less than a and b; thus, Eq. (17) is more important than the term implies by Eq. (18). The optimization process is performed by starting from the final time and is preceded in a backward manner.

The constraints are divided into two parts: equality nonlinear constraints contain dynamic Eq. (10) and inequality linear constraints contain the limitations of velocities (Eq. 8) and torques (Eq. 9), and specified path (Eq. 13). Also, the velocities and torques at point $k+1$ are used to calculate the objective function at point k and this procedure is repeated from N to *1*.

Sequential quadratic programming algorithm is used to determine the optimal velocities and torques at point k so a new optimization problem is defined at each step:

Minimize $J(q, \dot{q}, u)$
Subject to $h_s(q, \dot{q}, u) = 0, s = 1, 2, \ldots, S$
$g_1(q, \dot{q}, u) \le 0, l = 1, 2, \ldots, L,$

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

where S and L are the numbers of constrained equations and limitations on velocities and torques, respectively. The Lagrangian function can be defined as

$$L((q,\dot{q},u),\lambda) = J((q,\dot{q},u)) + \sum_{s=1}^{S}\lambda_s h_s(q,\dot{q},u) = J((q,\dot{q},u)) + \lambda^T h(q,\dot{q},u), \quad (19)$$

where λ is the vector of Lagrange multipliers. If deviations of $q, \dot{q}, u$ are shown by vector d, then the value of Lagrangian at the next step will be

$$L((q_{in},\dot{q}_{in},u_{in})+d,\lambda_{in}) = L((q_{in},\dot{q}_{in},u_{in}),\lambda_{in}) + [\nabla_{(q,\dot{q},u)}L((q_{in},\dot{q}_{in},u_{in}))]^T d + \frac{1}{2}d^T\nabla^2 L((q_{in},\dot{q}_{in},u_{in}))d \quad (20)$$

where $q_{in}, \dot{q}_{in}$, and u_{in} are initial values of states and torques and also λ_{in} is a multiplier vector (a suitable choice is $\lambda_{in} = 0$). $L((q_{in},\dot{q}_{in},u_{in})+d,\lambda_{in})$ in Eq. (20) can be computed by Eq. (21):

$$L((q_{in},\dot{q}_{in},u_{in})+d,\lambda_{in}) = [J(q_{in},\dot{q}_{in},u_{in}) + \lambda_{in}^T h(q_{in},\dot{q}_{in},u_{in})] + \left[\nabla_{(q,\dot{q},u)}J(q_{in},\dot{q}_{in},u_{in}) + \lambda_{in}^T\nabla_{(q,\dot{q},u)}h(q_{in},\dot{q}_{in},u_{in})\right]^T d + \frac{1}{2}d^T\left[\nabla^2_{(q,\dot{q},u)}(q_{in},\dot{q}_{in},u_{in}) + \sum_{s=1}^{S}\lambda_{in}\nabla^2_{(q,\dot{q},u)}h_s(q_{in},\dot{q}_{in},u_{in})\right]d \quad (21)$$

Constraint functions h_s and g_l can be rewritten as:

$$h_s(q,\dot{q},u) + [\nabla_{(q,\dot{q},u)}h_s(q,\dot{q},u)]^T d = 0, s = 1,2,\ldots,S \quad (22)$$

$$g_1(q,\dot{q},u) + [\nabla_{(q,\dot{q},u)}g_1(q,\dot{q},u)]^T d \le 0, l = 1,2,\ldots,L \quad (23)$$

New optimization problem must be solved to find optimum values of d and λ, which can be shown by d^* and λ^*. It should be considered that dynamic Eq. (10) is defined as *h(x)* while limitations of velocities and torques are presented as *g(x)*.

4. The Dynamic Programming Algorithm for Point-to-Point Trajectory Planning

There is no specified path for point-to-point motion and the objective function and constraints have been changed. Unlike pre-specified trajectory planning, in a point-to-point motion, the optimization process is performed by starting from an initial state and the objective function is defined as:

$$J^*(q(k),u(k),k) = \min[J(\varphi(\dot{q}(k),u(k),k) + J(u(k),u)(k+1),k)] \quad (24)$$

$$J(\varphi(q(k),\dot{q}(k),u(k),k)) = a(q(k)-q(f))^2 + b(\dot{q}(k)^2) + c(u(k)^2) \quad (25)$$

$$J(u(k),u(k+1),k) = c((u(k)-u(k-1))^2) \quad (26)$$

Equation (25) has one term more than Eq. (17) and this term is added to force the joint displacements to be converged to final states.

5. Algorithm for DLCC Calculation

As it was stated in the introduction, the DLCC of a manipulator is used to compare the performance of the proposed method with previous algorithms. A new approach that is shown in Fig. 1 is proposed to calculate the DLCC and its related optimal angular position and velocity of the joints and also the related end-effector path. As it is shown in Fig. 1, the allowable load is increased by an iteration procedure. The main factor that limits the DLCC of a mechanical manipulator is the limitation on the applied torques in the manipulator joints. Joints' torque is generated by DC motors and the maximum and minimum torques of the motor are

$$U_{i-\max}(k) = U_{si} - \frac{U_{si}}{\omega_{si}}\dot{q}_i(k) \quad (27)$$

and

$$U_{i-\min}(k) = -U_{si} - \frac{U_{si}}{\omega_{si}}\dot{q}_i(k), \quad (28)$$

where U_{si} is the stall torque, ω_{si} is the no load speed, and $\dot{q}_i(k)$ is the angular velocity of the ith motor. Another constraint that limits the DLCC is Eq. (8) that implies bounds on angular velocity of the joints. Selection of appropriate weighting factors for performance index is done using trial-and-error procedure so that the optimization algorithm leads the system to have a smooth trajectory and minimum chattering in resultant joint velocities. Also, the number of steps, N, must be selected carefully because large N leads to low speed convergence and small N causes divergence of optimization algorithm. m_p in Fig. 1 is the load that is carried by manipulator in any iteration, m_{pmin} is the initial value considered for m_p that is increased in each iteration, and Δm_p is the increment of m_p.

6. Simulation for Linear Tracked Puma (Pre-Specified Path)

Schematic view of a Puma manipulator mounted on linear mobile base is shown in Fig. 2 and the manipulator characteristics and D-H parameters according to[14] are given in Table I.

A pre-specified path that has been designed in ref. [14] between initial coordinates $E_0 = (x_e = 0.5\,\text{m}, y_e = 0, z_e = -0.1\,\text{m})$ and final coordinates $E_f = (x_e = 0, y_e = 1.2\,\text{m}, z_e = 1.04\,\text{m})$ is considered that must be performed by 24 sec. The generalized coordinates can be considered as $q = [x_f \quad \theta_1 \quad \theta_2 \quad \theta_3]$. Also, the point mass mounted on the end-effector has three degrees of freedom but the manipulator has four degrees of freedom. Manipulator

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

Table I. Parameters and inertia properties of Puma manipulator.

NO	MASS (k)	Moment of inertia (kgm^2)	Center of mass (m)	D-H parameters θ_i	α_i	a_i (m)	d_i(m)
1	12	$\begin{bmatrix}0.2 & 0 & 0\\ 0 & 0.2 & 0\\ 0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}-0.25\\ 0\\ 0\end{bmatrix}$	θ_1	90	0	0.4
2	10	$\begin{bmatrix}0 & 0 & 0\\ 0 & 0.2 & 0\\ 0 & 0 & 0.2\end{bmatrix}$	$\begin{bmatrix}-0.25\\ 0\\ 0\end{bmatrix}$	θ_2	0	0.5	0
3	5	$\begin{bmatrix}0 & 0 & 0\\ 0 & 0.1 & 0\\ 0 & 0 & 0.1\end{bmatrix}$	$\begin{bmatrix}-0.25\\ 0\\ 0\end{bmatrix}$	θ_3	0	0.5	0

has one degree of redundancy and needs one additional kinematical constraint for redundancy resolution, so a pre-specified fifth-order polynomial from $x_f(0) = 0$ to $x_f(t_f) = 0.347$ m is considered for the base motion.[14]

Stall torque of motors, U_{s1}, U_{s2}, and U_{s3} are considered to be [10, 30, 6.67] N.m and no load speeds [ω_{s1}, ω_{s2}, ω_{s3}] are [5.71, 6.41, 4.6] rad/sec. Proposed algorithm is applied and the maximum payload is found to be 8.4 kg. Configuration of manipulator and pre-specified and actual optimal paths of the end-effector are shown in Fig. 3. Angular position and velocity of the joints are illustrated in Figs. 4–9. Also, the joints' torques are displayed in Figs. 10–12. Finally,

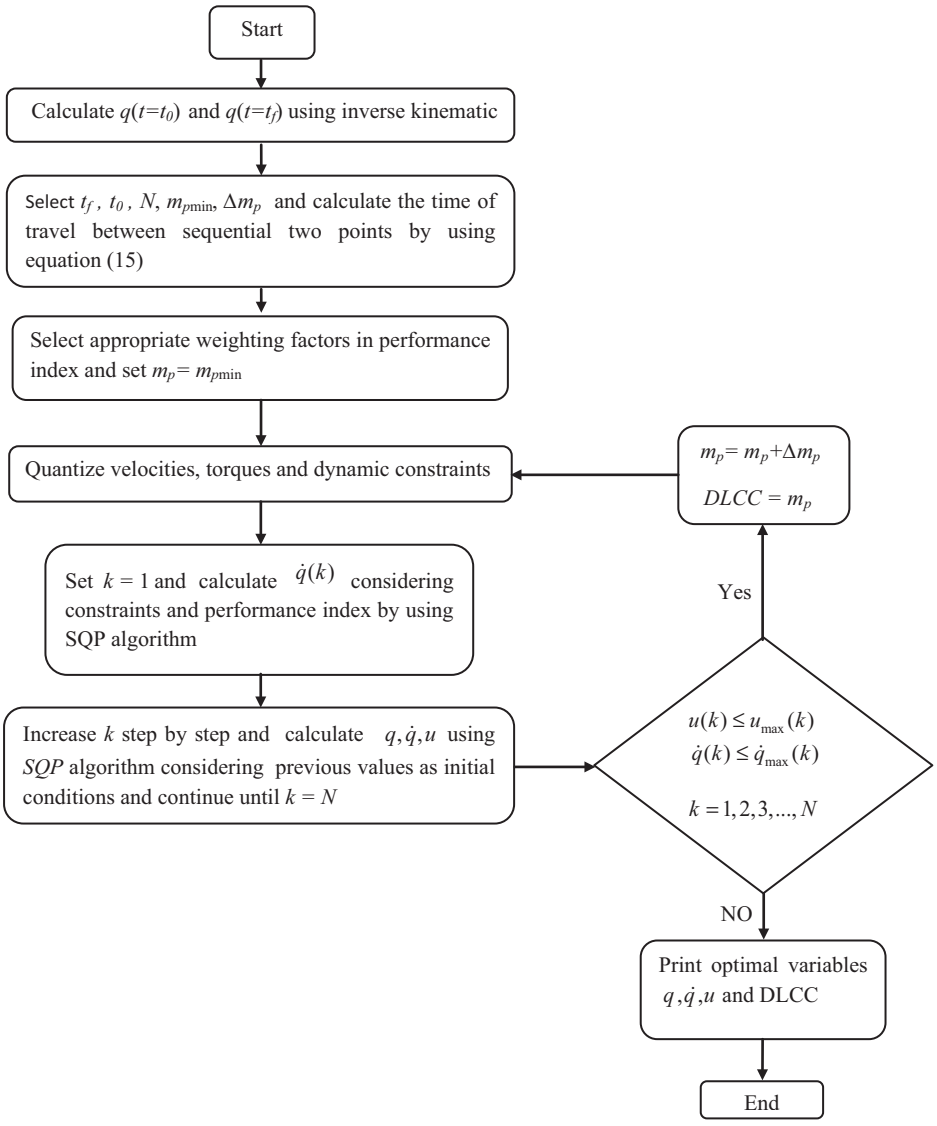


Fig. 1. DLCC calculation algorithm.

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

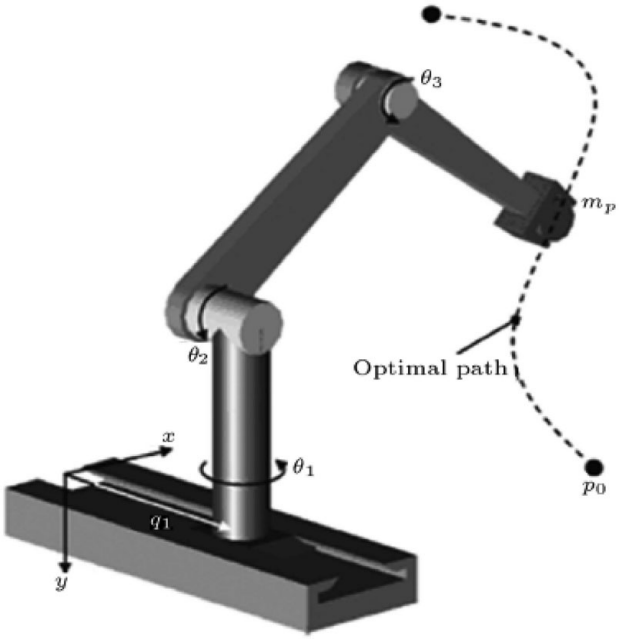


Fig. 2. Puma mobile manipulator with linear tracked base.

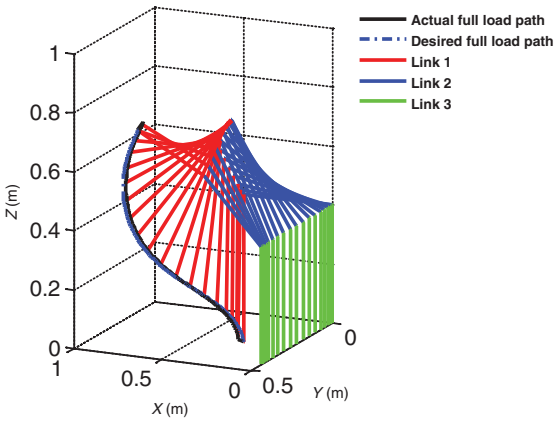


Fig. 3. (Colour online) Actual and desired path of end-effector.

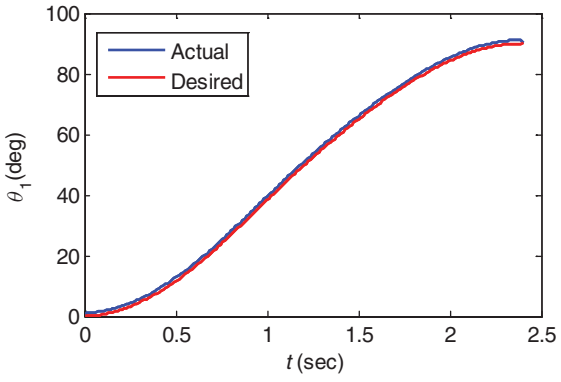


Fig. 4. (Colour online) Angular position of joint 1.

Fig. 13 shows the position error between the actual and desired positions. It should be considered that the allowable accuracy for tracking is selected to be $\delta = 0.05$ m but the maximum error that is occurred is less than 0.035 m. Figure 11 mentions that during the motion, the torque of joint 2 is equal to maximum allowable value and if smaller value of δ is considered in the dynamic programming algorithm then the joint torques will be increased and the torque constraints will be failed. So the saturation of the second joint torque causes an error between the actual and desired paths.

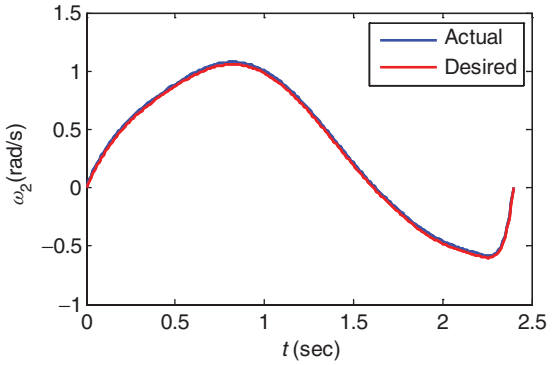


Fig. 5. (Colour online) Angular velocity of joint 2.

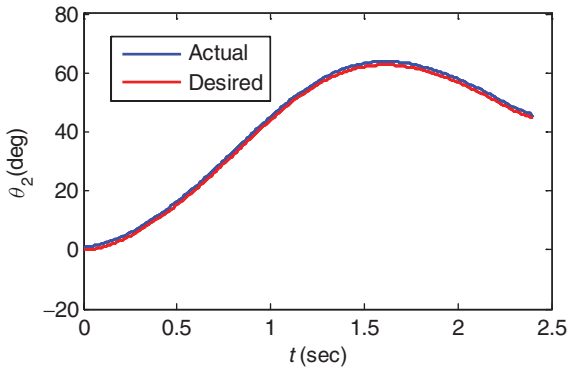


Fig. 6. (Colour online) Angular position of joint 2.

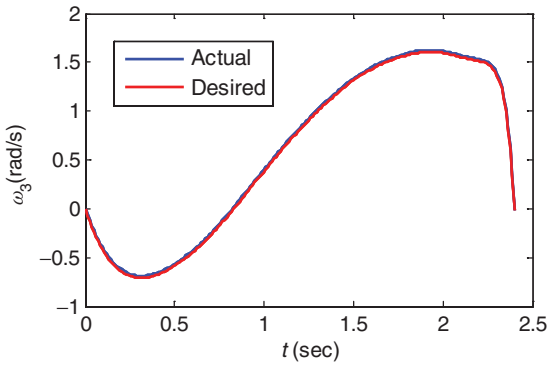


Fig. 7. (Colour online) Angular velocity of joint 3.

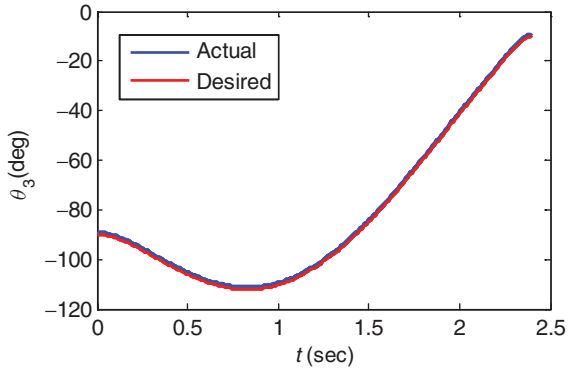


Fig. 8. (Colour online) Angular position of joint 3.

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

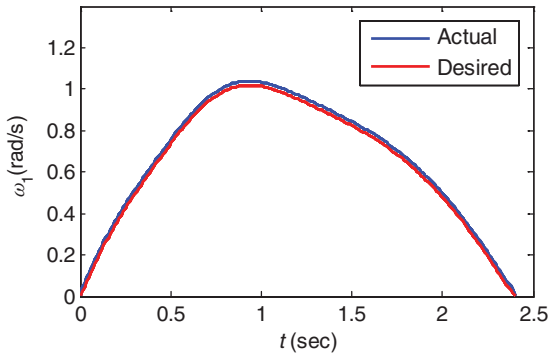


Fig. 9. (Colour online) Angular velocity of joint 3.

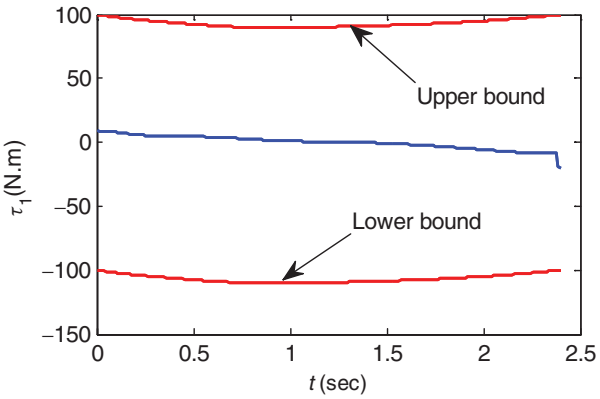


Fig. 10. (Colour online) Torques of joint 1.

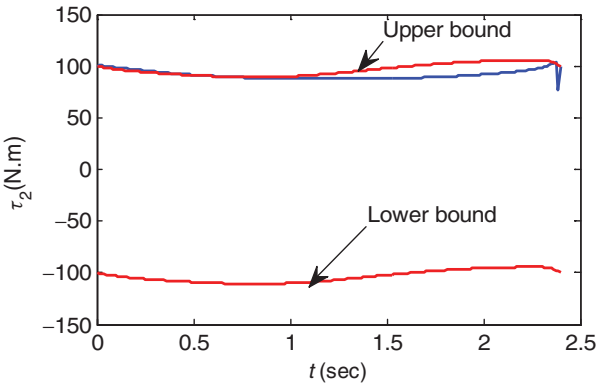


Fig. 11. (Colour online) Torques of joint 2.

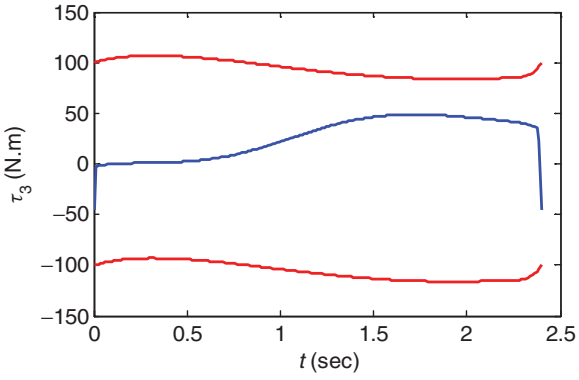


Fig. 12. (Colour online) Torques of joint 3.

7. Simulation for Wheeled Mobile Robot (Point-to-Point Motion)

The proposed algorithm is applied on Scout robot that is a wheeled mobile robot and is shown in Fig. 14. A schematic

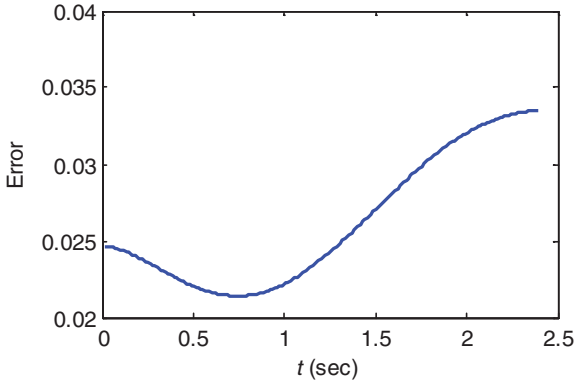


Fig. 13. (Colour online) Position error.

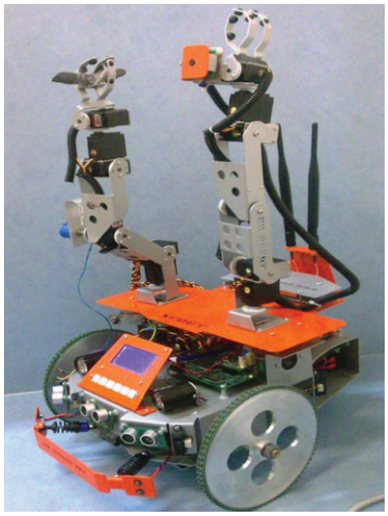


Fig. 14. (Colour online) Scout mobile robot.

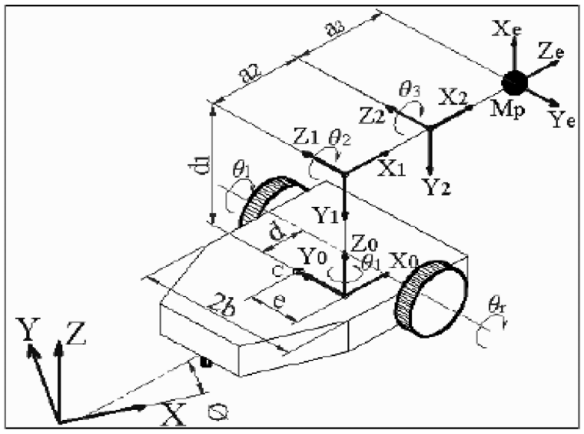


Fig. 15. Schematic of Scout robot with three links.

view of this mobile robot with three active links of the left arm is shown in Fig. 15 and the parameters of robot are presented in Table II.

The dynamic motion equation of the Scout mobile robot is[15]

$$M(q)\ddot{q} + V(q, \dot{q}) = E(q)u - A^T(q)\lambda. \tag{29}$$

Equation (29) can be rewritten as Eq. (30) to cancel the last term in the right hand of Eq. (29)[14].

$$u = S^T M\dot{S} + S^T MS\dot{v} + S^T V \tag{30}$$

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

Table II. Parameters of Scout robot.

Parameter	Value	Unit
Length of links	$d_1 = 0.06, a_2 = 0.1, a_3 = 0.21$	m
Mass of links	$M_1 = 0.028, M_2 = 0.1, M_3 = 0.231$	kg
Moment of inertia of link 1	$I_{1x} = 0, I_{1y} = 0, I_{1z} = 0.00005$	kg.m^2
Moment of inertia of link 2	$I_{2x} = 0.00008, I_{2y} = 0.00091, I_{2z} = 0.00092$	kg.m^2
Moment of inertia of link 3	$I_{3x} = 0.00008, I_{3y} = 0.00091, I_{3z} = 0.00092$	kg
Mass of base	6	kg
Mass of wheels	0.32	kg
Moment of inertia of base about Z axis	$I_{zzb} = 0.06363$	kg.m^2
Moment of inertia of wheels about rotation axis	0.0008	kg.m^2
b	0.145	m
r	0.08	m
d	0.065	m
e	0.08	m

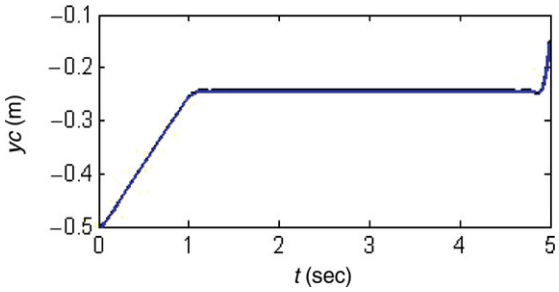


Fig. 16. (Colour online) Base position along the x-axis.

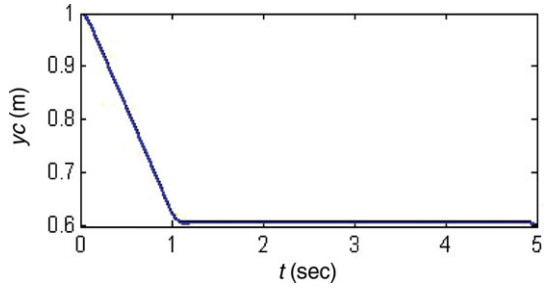


Fig. 17. (Colour online) Base position along the y-axis.

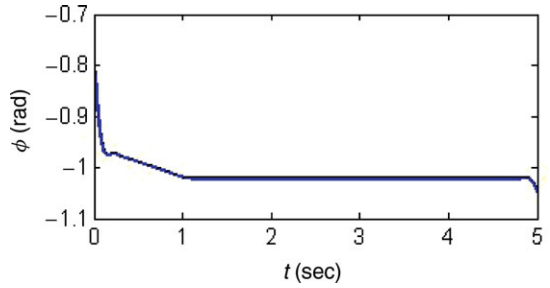


Fig. 18. (Colour online) Rotation angle of base.

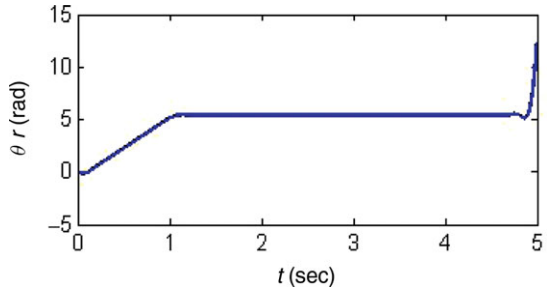


Fig. 19. (Colour online) Angular position of right wheel.

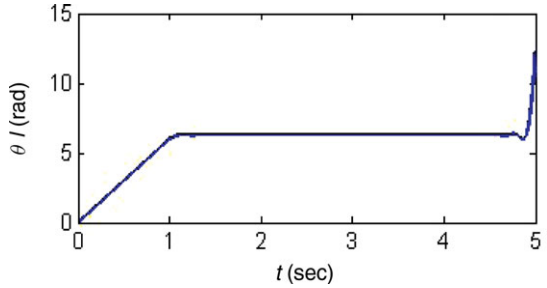


Fig. 20. (Colour online) Angular position of left wheel.

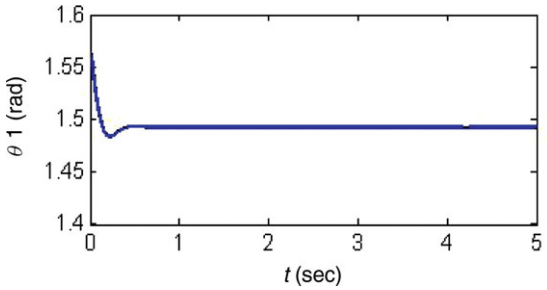


Fig. 21. (Colour online) Angular position of joint 1.

The vector of state variables for the Scout mobile robot is determined as:

$$[X]_{1\times 12} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{31}$$

$$q = [x_c \quad y_c \quad \phi \quad \theta_1 \quad \theta_2 \quad \theta_3]^T \tag{32}$$

$$\dot{q} = [\dot{x}_c \quad \dot{y}_c \quad \dot{\phi} \quad \dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T \tag{33}$$

Also, vector of joint torques is considered as control vector

$$u = [\,\tau_r \quad \tau_l \quad \tau_1 \quad \tau_2 \quad \tau_3\,]^T. \tag{34}$$

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

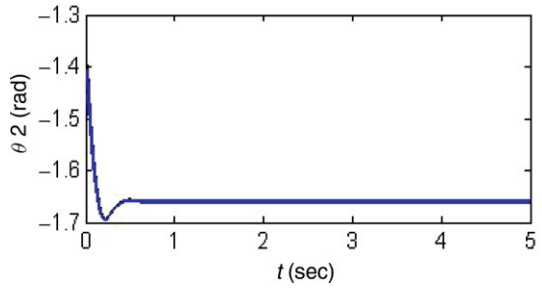


Fig. 22. (Colour online) Angular position of joint 2.

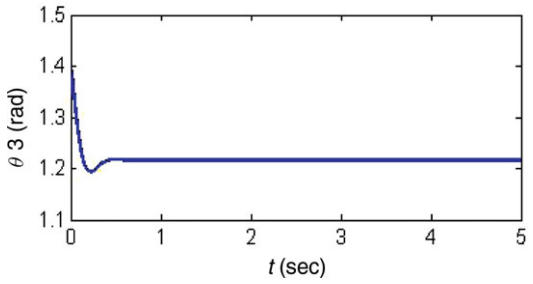


Fig. 23. (Colour online) Angular position of joint 3.

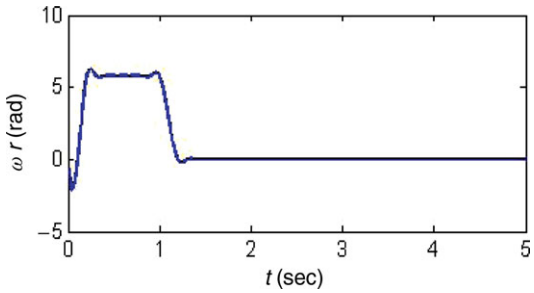


Fig. 24. (Colour online) Angular velocity of right wheel.

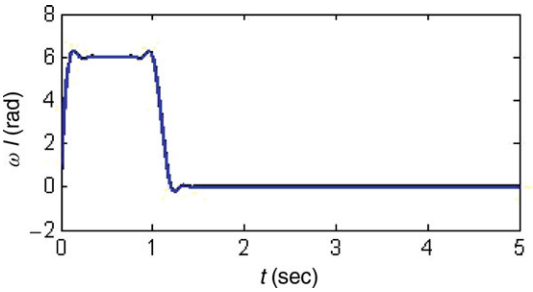


Fig. 25. (Colour online) Angular velocity of left wheel.

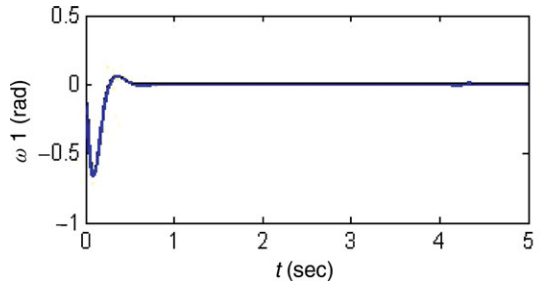


Fig. 26. (Colour online) Angular velocity of joint 1.

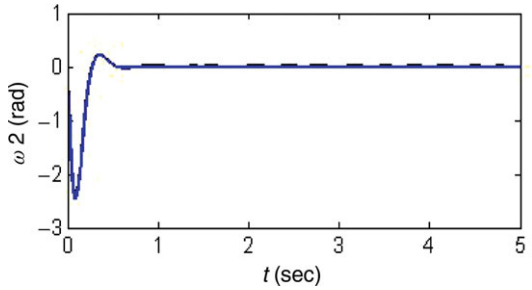


Fig. 27. (Colour online) Angular velocity of joint 2.

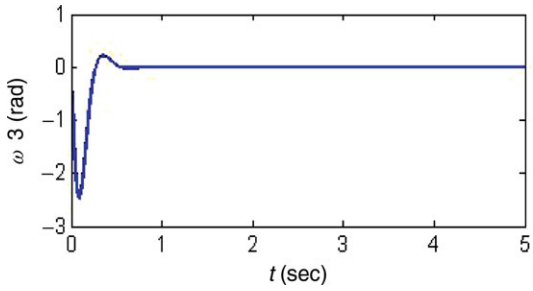


Fig. 28. (Colour online) Angular velocity of joint 3.

Matrix S and vector v in Eq. (30) have the following forms:

$$S = \begin{bmatrix} r/2^* \cos(\phi) & r/2^* \cos(\phi) & 0 & 0 & 0 \\ r/2^* \sin(\phi) & r/2^* \sin(\phi) & 0 & 0 & 0 \\ r/2b & -r/2b & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{35}$$

$$v = [\dot{\theta}_r \quad \dot{\theta}_1 \quad \dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T \tag{36}$$

The dynamic programming algorithm is used to design point-to-point optimal path for Scout mobile robot and the objective function is defined as

$$J = a[(q(k) - q(0))^2 + (\dot{q}(k) - \dot{q}(0))^2 + b[u(k)^2] + c[(u(k) - u(k-1))^2]. \tag{37}$$

Since all of the calculations are done in a forward way, the acceleration of the joints at point k must be computed by Eq. (38):

$$\ddot{q}(k) = \frac{\dot{q}(k) - \dot{q}(k-1)}{\Delta t} \tag{38}$$

Equality constraint described by Eq. (3) can be written using dynamic Eq. (30):

$$u(k) - (S^T(k)M(k)\dot{S}(k)) - (S^T(k)M(k)S(k)\dot{v}) - S^T V = 0 \tag{39}$$

The optimal trajectory and related control vector is computed through the dynamic programming and SQP method based on the proposed method. In this simulation, point-to-point

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

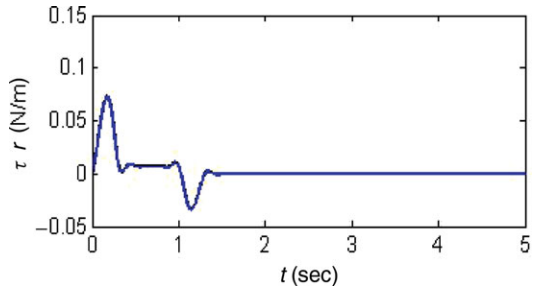


Fig. 29. (Colour online) Right wheel torque.

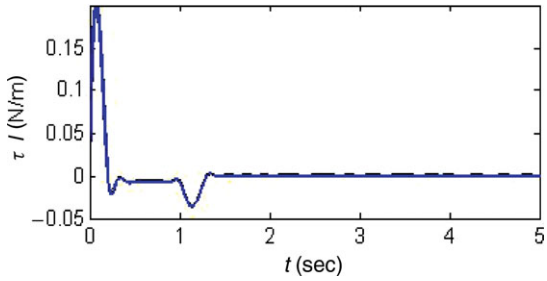


Fig. 30. (Colour online) Left wheel torque.

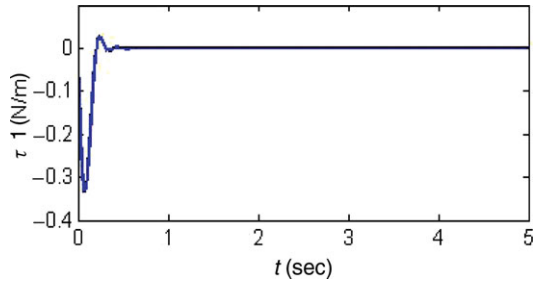


Fig. 31. (Colour online) Torque of joint 1.

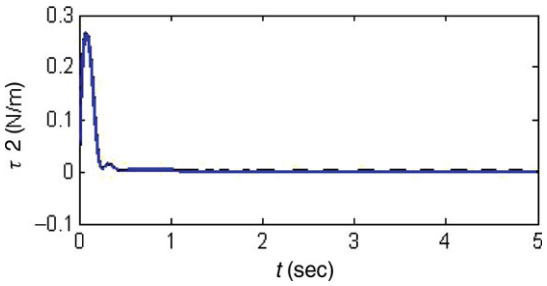


Fig. 32. (Colour online) Torque of joint 2.

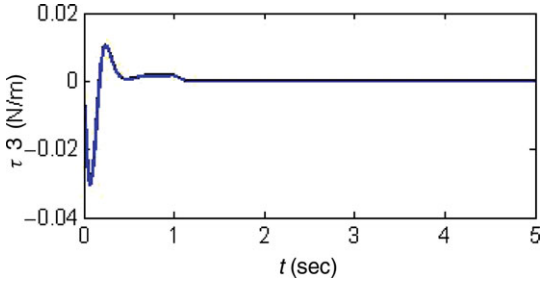


Fig. 33. (Colour online) Torque of joint 3.

motion is considered between initial point of $P_0 = (-0.4, 1.1, 0.4)$ and final point of $P_f = (-0.05, 0.6, 0.5)$. Time duration is considered $t_f = 5$ sec. Also, 0.5 kg DLCC is obtained and the optimal variables of mobile base such as position and angle of the base and angular position of the wheels are shown in Figs. 16–20. Optimal angular positions of the links are implied in Figs. 21–23. Moreover, the angular velocity of the wheels and links are displayed in Figs. 24–28. Figures 29–33 show the applied torques of the wheels and

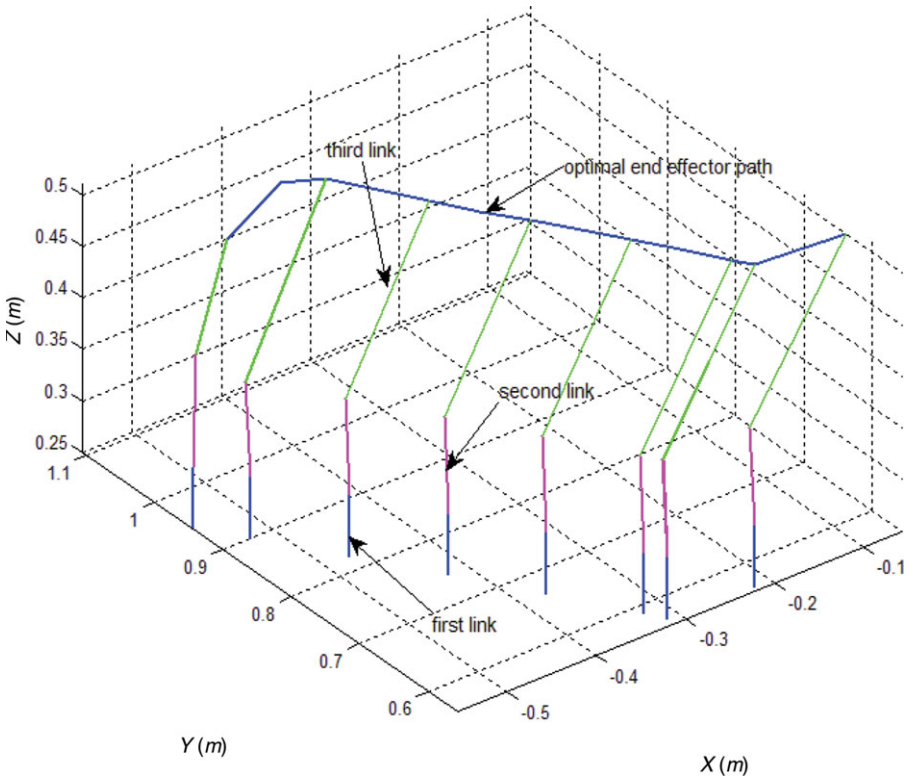


Fig. 34. (Colour online) Configuration of Scout robot (first simulation).

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

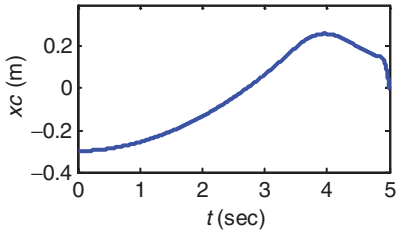


Fig. 35. (Colour online) Base position along x-axis.

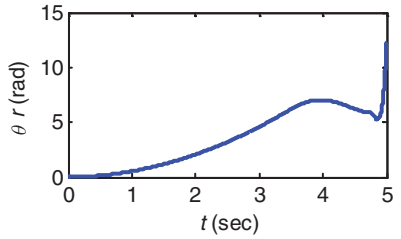


Fig. 36. (Colour online) Angular position of right wheel.

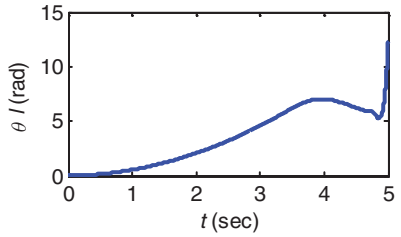


Fig. 37. (Colour online) Angular position of left wheel.

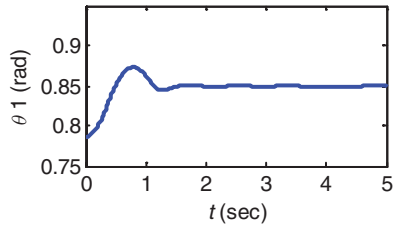


Fig. 38. (Colour online) Angular position of joint 1.

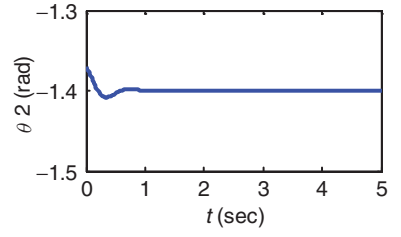


Fig. 39. (Colour online) Angular position of joint 2.

joints. Finally, the optimal end-effector path and its related configuration of the links are displayed in Fig. (34).

Another simulation is done to consider the ability of the proposed algorithm and in this new study, even further restriction is imposed on angular velocities. Initial position of the end-effector is $P_0 = (0.4, 1, 0.38)$ and its final position is $P_f = (-0.1, 1, 0.4)$. In this simulation, the path of the base is considered to be a straight line along x coordinate. Resultant base position, angular positions, velocities, torques, and also the optimal trajectory of the end-effector are displayed in Figs. 35–46, respectively.

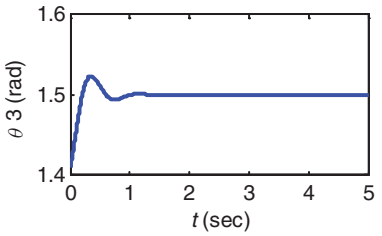


Fig. 40. (Colour online) Angular position of joint 3.

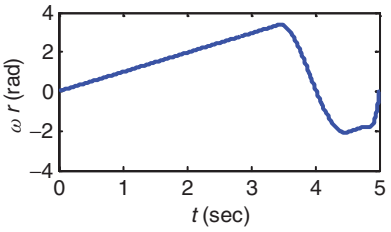


Fig. 41. (Colour online) Angular velocity of right wheel.

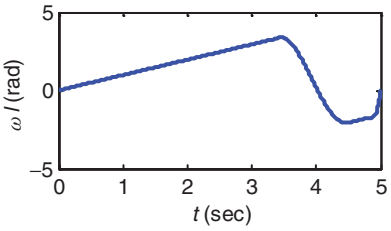


Fig. 42. (Colour online) Angular velocity of left wheel.

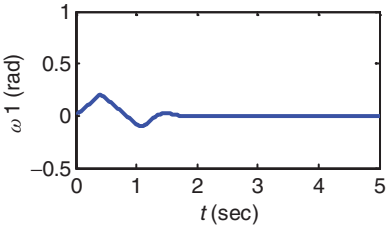


Fig. 43. (Colour online) Angular velocity of joint 1.

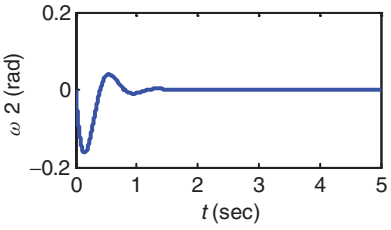


Fig. 44. (Colour online) Angular velocity of joint 2.

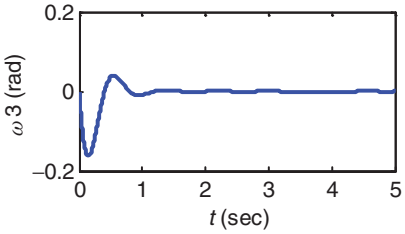


Fig. 45. (Colour online) Angular velocity of joint 3.

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

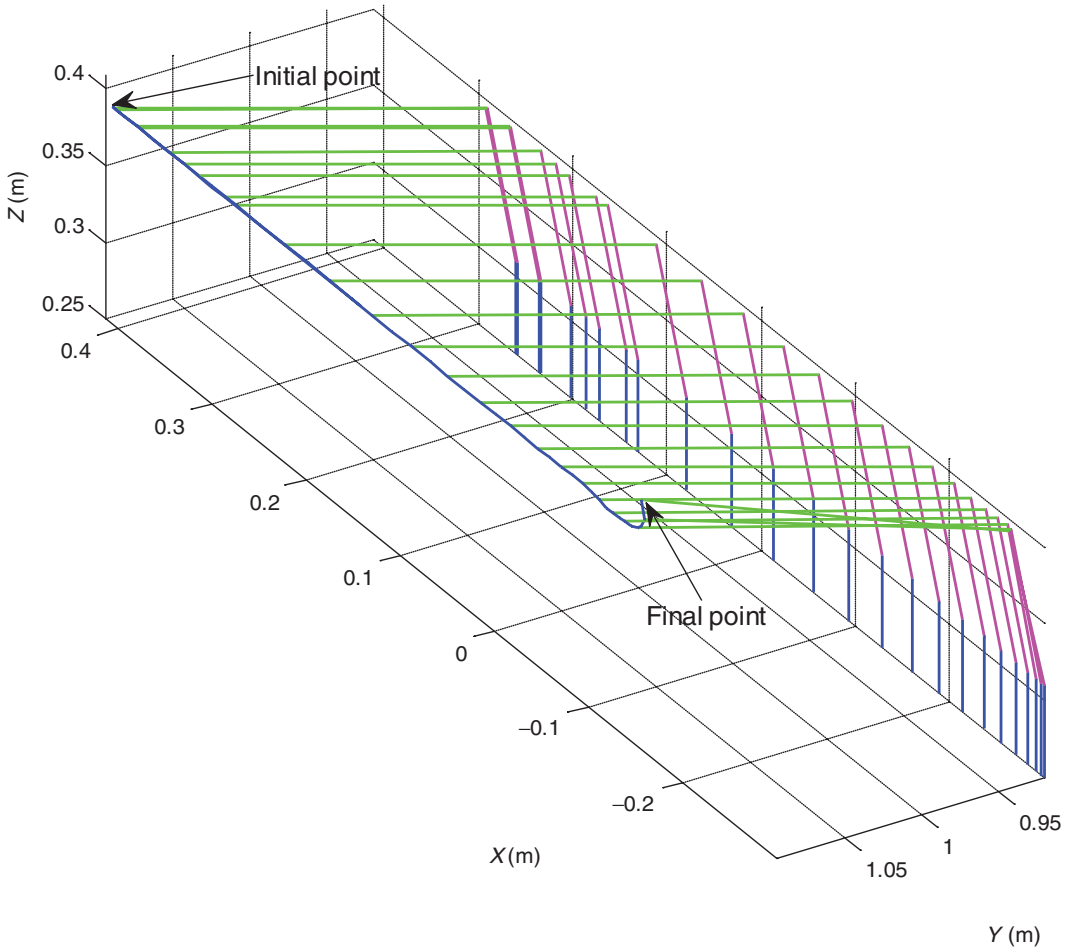


Fig. 46. (Colour online) Configuration of Scout robot (second simulation).

The results of second simulation show that the smoothness of resultant trajectory and prevention from sudden changes in angular positions and velocities can be controlled via further limitation on angular velocities (Eq. (8)). Also, results imply that the end-effector reaches to its final position with allowable 0.1 m accuracy. As mentioned in previous section, the algorithm has no meaningful result for smaller accuracy value in this particular simulation. Unlike the methods that use indirect optimization,[5] the proposed method is systematic and can be applied on any other system without using complex mathematical works that is required in indirect methods. Also, contrary to linear direct optimization methods,[2] no initial guess is needed for optimal trajectory. These advantages express that the proposed method is more suitable than previous works for trajectory planning and DLCC calculation.

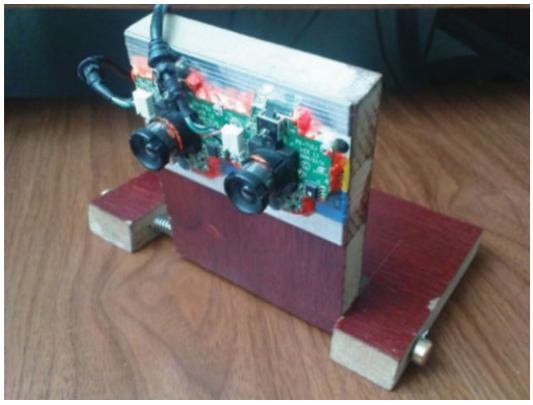


Fig. 47. (Colour online) Stereo vision set up.

8. Determining Position of the End-Effector via Stereo Vision

Vision-based methods are common for determining the position of the end-effector of a robot. In this paper, a stereo vision strategy is used for determining the position of the end-effector in experimental tests. Using the encoder data and finding the end-effector position via direct kinematic equations is another common way. But output errors of the encoders and geometric errors can affect calculations and result in high error in the end-effector position. This reason leads us to use a direct positioning method. Thus, stereo vision is selected with accuracy less than 1 mm that can be applied in real-time conditions. A view of stereo vision set up is shown in Fig. 47 and the characteristics of the used cameras are presented in Table III.

Modeling of the cameras is based on ref. [16] and considers seven coefficients for radial and tangential distortion, focal length, and principal point. Each camera is calibrated using proposed method in ref. [17]. A red LED is mounted on the

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

Table III. Characteristics of cameras.

Image sensor	$^1/_4$" CMOS, 640 × 480 (350 k pixels)
Frame rate	30 fps @ 320 × 240
Lens	$F = 2.2$, $f = 4.6$ mm
View angle	65 degree
Interface	USB 2.0 port

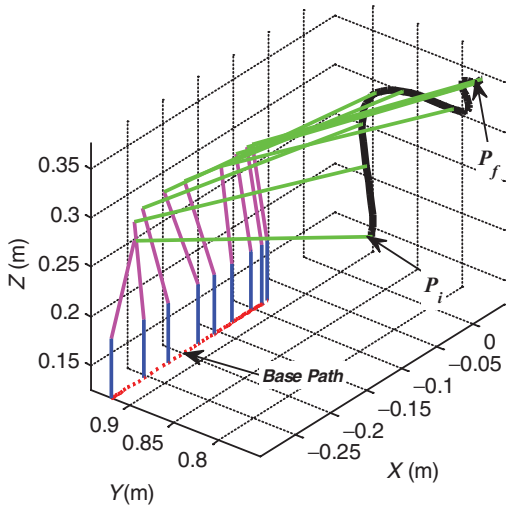


Fig. 48. (Colour online) Configuration of Scout robot (simulation).

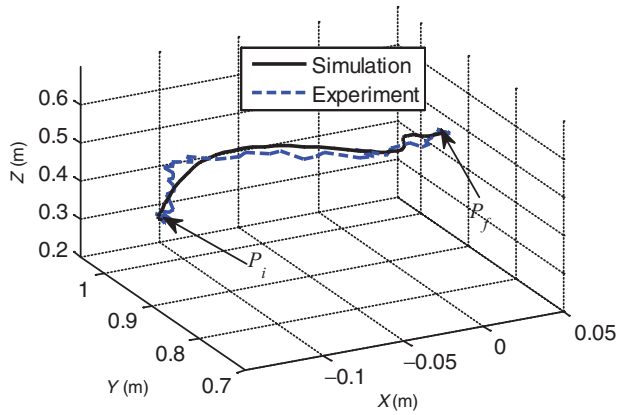


Fig. 49. (Colour online) End-effector position (simulation and experiment).

end-effector and cameras capture photo in 0.2 sec sampling period. Position of the red pixel is searched in left and right images and then 3D position of the LED, which is in fact the position of the end-effector, is calculated using triangulation method.[18]

9. Experimental Results on Scout Robot

An experimental test was done on Scout robot to verify the simulation results. The robot is controlled using software launched on a PC. In this simulation, point-to-point motion is considered between initial point of $P_i(-0.1, 0.75, 0.4)$ and final point of P_f (0.04, 0.78, 0.47) in 5 sec and linear path is considered for mobile base. Optimal trajectory that is extracted through the simulation is shown in Fig. 48 and resultant path from experimental test and simulation are compared in Fig. 49. Simulated and experimental angular

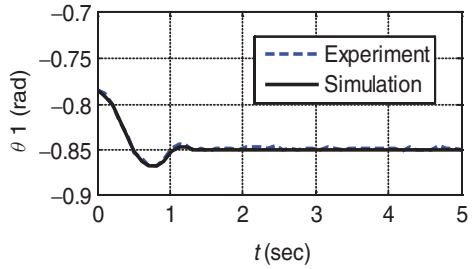


Fig. 50. (Colour online) Angular position of joint 1.

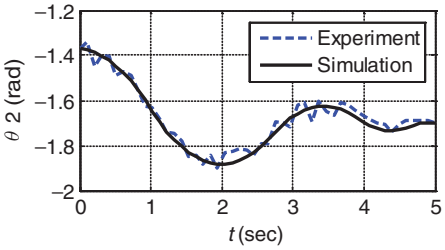


Fig. 51. (Colour online) Angular position of joint 2.

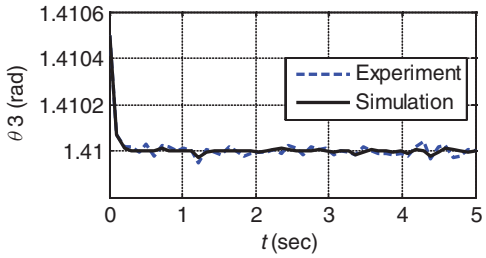


Fig. 52. (Colour online) Angular position of joint 3.

positions of the joints are displayed in Figs. 50–52. The oscillations that occur in Figs. 49 and 51 are results of flexibility at the second joint that is not modeled in dynamic equations.

10. Conclusion

In this paper, the dynamic programming approach is discussed as a method of trajectory planning for manipulators. All kinematic and dynamic equations and equality and inequality constraints are considered to have nonlinear forms. Formulations of this approach for trajectory planning are discussed in two types: pre-specified path and point-to-point motion problems. Also, performance index that must be minimized is changed to be applicable for these two problems. A new algorithm is proposed to design the optimal trajectory for manipulators and also to calculate the DLCC. Limitations on the value of the motor torques are considered in algorithm and angular velocities of the joints are considered to be bounded. Simulations are performed for a wheeled mobile base manipulator named Scout and a linear tracked Puma. In each case, the optimal trajectory is designed and its related DLCC is calculated using the proposed algorithm. Experimental test has been performed for Scout robot and also stereo vision is used to record 3D position of the end-effector.

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press

Two main advantages of the proposed method are:

- Compared to the ILP[2] method, no initial guess is required. ILP method is very sensitive to the initial guess so it must be very close to the final optimal solution in order to obtain a good result.
- No complex mathematical efforts are required unlike indirect optimization methods such as two-point boundary value problem (2PBVP) that has been used in ref. [14]. The proposed method can be easily implemented for any system with different dynamic equations without complicated computations.

These superiorities represent that the proposed method is suitable for trajectory planning and DLCC calculation of mobile manipulators with nonlinear constraints.

References

1. L. T. Wang and B. Ravani, "Dynamic load carrying capacity of mechanical manipulators: Part 1," *ASME J. Dyn. Syst. Meas. Control* **110**(1), 46–52 (1988).
2. M. H. Korayem, Kh. Najafi and M. Bamdad, "Synthesis of cable driven robots dynamic motion with maximum load carrying capacities: Iterative linear programming approach," *Scientia Iranica: Trans. B, Mech. Eng.* **17**(3), 229–239 (2010).
3. M. H. Korayem, H. Ghariblu and A. Basu, "Maximum allowable load of mobile manipulator for two given end points of end-effector," *Int. J. AMT* **24**(10), 743–751 (2004).
4. M. H. Korayem, S. Firouzy and A. Heidari, "Dynamic Load Carrying Capacity of Mobile-Base Flexible-Link Manipulators Using Finite Element Approach," *Proceedings of IEEE International Conference on Robotics and Biomemitics*, Sanya, China (Dec. 15–18, 2007) pp. 2172–2177.
5. M. H. Korayem, V. Azimirad, A. Nikoobin and Z. Boroujeni, "Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability," *Int. J. Adv. Manuf. Technol.* **46**(5–8), 811–829 (2010).
6. B. Bakker, Z. Zivcovic and K. Ben, "Hierarchical Dynamic Programming for Robot Path Planning" *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS, Edmonton, Canada (Aug. 2–6, 2005) pp. 3720–3725.
7. G. Eisler, G. Wilson and R. Robinett, "Discrete Dynamic Programming for Optimized Path Planning of Flexible Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan (Sep. 28–Oct. 2, 2004) pp. 2918–2923.
8. A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robot. Comput.-Integr. Manuf.* **24**(3), 415–426 (2008).
9. S. Parasuraman, Chiew Mun Hou and V. Ganapathy, "Motion planning for a redundant manipulator using genetic algorithm," *Key Eng. Mater.* **467–469**, 782–787 (2011).
10. M. Radovnikovich, C. Cheok and P. Vempaty, "Comparison of Optimal Path Planning Algorithms for an Autonomous Mobile Robot," *IEEE Conference on Technologies for Practical Robot Applications*, Woburn, MA, USA (Apr. 11–12, 2011) pp. 35–39.
11. M. Stilman, C. G. Atkeson, J. J. Kuffner and G. Zeglin, "Dynamic Programming in Reduced Dimensional Spaces: Dynamic Planning for Robust Biped Locomotion," *IEEE International Conference on Robotics and Automation*, Barcelona, Spain (Apr. 18–22, 2005) pp. 2399–2404.
12. C. Sallaberger and G. D'Eleuterio, "Optimal robotic path planning using dynamic programming and randomization," **35**(2–3), 143–156 (1995).
13. T. T. Lee and C. L. Shih, "A study of the gait control of a quadruped walking vehicle," *IEEE J. Robot. Autom.* **2**(2), 61–69 (1986).
14. M. H. Korayem, A. Nikoobin and V. Azimirad, "Maximum load carrying capacity of mobile manipulators: Optimal control approach," *Robotica* **27**(1), 147–159 (2009).
15. Y. Yamamoto and X. Yun, "Effect of the dynamic interaction on coordinated control of mobile manipulators," *IEEE Trans. Robot. Autom.* **12**(5), 816–824 (1996).
16. J. Heikkilä and O. Silvén, "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico (Jun. 17–19, 1997) pp. 1106–1112.
17. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(11), 1330–1334 (2000).
18. J. F. C. Garcia, H. G. Yao and S. Zheng, "3D reconstruction of objects using stereo imaging," *Opt. Lasers Eng.* **22**, 193–213 (1995).

https://doi.org/10.1017/S0263574712000628 Published online by Cambridge University Press