Robotica (2012) volume 30, pp. 53–65. © Cambridge University Press 2011
doi:10.1017/S0263574711000336

Maximum load determination of nonholonomic mobile manipulator using hierarchical optimal control

M. H. Korayem*, V. Azimirad, H. Vatanjou and A. H. Korayem

Robotic Research Laboratory, School of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran

(Received in Final Form: March 16, 2011. First published online: April 26, 2011)

SUMMARY

This paper presents a new method using hierarchical optimal control for path planning and calculating maximum allowable dynamic load (MADL) of wheeled mobile manipulator (WMM). This method is useful for high degrees of freedom WMMs. First, the overall system is decoupled to a set of subsystems, and then, hierarchical optimal control is applied on them. The presented algorithm is a two-level hierarchical algorithm. In the first level, interaction terms between subsystems are fixed, and in the second level, the optimization problem for subsystems is solved. The results of second level are used for calculating new estimations of interaction variables in the first level. For calculating MADL, the load on the end effector is increased until actuators get into saturation. Given a large-scale robot, we show how the presenting in distributed hierarchy in optimal control helps to find MADL fast. Also, it enables us to treat with complicated cost functions that are generated by obstacle avoidance terms. The effectiveness of this approach on simulation case studies for different types of WMMs as well as an experiment for a mobile manipulator called Scout is shown.

KEYWORDS: Mobile manipulator; Maximum allowable dynamic load; Hierarchical optimal control; Obstacle avoidance.

Nomenclature

M(q)	inertial matrix of manipulator
C(q, $\dot{q}$)	centrifugal and Coriolis matrix of manipulator
τ, U	control inputs to joints or wheels
q	generalized coordinates of manipulator
X_i	state variable of ith subsystem
ϕ	head angle of nonholonomic mobile base
w^i_{jk}	weighting matrix related to obstacle avoidance of parts in cost function
Z^i	interaction vector of ith subsystem
J	cost function
k	weight of convergence
n	number of generalized coordinates
(X_e, Y_e, Z_e)	coordination of end effector in the world coordination system
O	center of the world coordination system
θ_l	angular displacement of left wheel
r	radius of each wheel
d	distance from C to O
l_2	length of second link
$P_o(X_{obs}, Y_{obs})$	coordination of obstacle in the world coordination system
l_i	inseparable part of cost function of each subsystem
h_i	separable part of subsystem's cost function
h	overall cost function
H	Hamiltonian of general system
R, Q	weighting matrixes of states and control variables
η	states related to velocities
λ_i	adjoin vector of the ith subsystem
x^*	desired state vector
u^*	desired control vector
p_i	Lagrange multiplayer of the ith subsystem
N	number of subsystems
(X_c, Y_c, Z_c)	coordination of center of mass mobile platform in the world coordination system
C	center of mass of mobile platform
θ_r	angular displacement of right wheel
m_p	mass of payload
b	distance between driving wheels and axis of symmetry
l_1	length of first link
e	error value

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

1. Introduction

Using mobile manipulator is an efficient solution for operation in wide areas but the path planning problem of them is more difficult than path planning of separate mobile base or manipulator. The reason is that optimal movement needs coordination between mobile base and manipulator. An effective way is to consider it as a large-scale system and optimize the trajectory by a hierarchical method. One of path planning methods is to find maximum load path. On maximum load path robot can carry maximum allowable dynamic load (MADL). Among the infinite number of possible paths, there exists a path on which the manipulator can carry maximum load on end

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

effector, which is called maximum load path, and the corresponding load is considered as its MADL. Researchers studied different methods: B-spline polynomial is used for payload maximization of fix robots.[1] Ghariblu developed a computational algorithm to find a dynamic motion trajectory of a mobile manipulator with flexible links and joints that allows the robot to carry a maximum load between two specified points using linearized state space form of kinematically modeled mobile base[2] but in this method the initial guess of the solution must be very close to the final optimal solution; otherwise, divergence happens or the suboptimal solutions are obtained. Korayem *et al*.[3] studied dynamic load carrying capacity of mobile manipulator based on optimal control method, but they did not study obstacle avoidance. More recently, they proposed an algorithm to determine maximum load carrying capacity of mobile manipulator considering tip over stability in presence of obstacles.[4]

Although above works for path planning of wheeled mobile manipulator (WMM) is done but all of them suffer from high order of equations that could not be solved easily. One of the best ways for optimization of large-scale robots is hierarchical configuration in which a given unit-level coordinates the units on the level below and, in turn, is coordinated by the units above it. Some authors studied hierarchical path planning. Mohri *et al*. used a suboptimal trajectory planning method based on order of priority for WMMs.[5] But their method was time consuming because the whole dynamic of nonholonomic mobile manipulator is used at the same time in one processor. Sadati *et al*. proposed gradient-based approach for decomposition of system and gradient updating for generating new coordination variables. But linearized equations and high amount of gradient processes attempting to converge make this method inaccurate.[6] Sadati *et al*. used fuzzy method for interaction prediction coordination of large-scale systems. But needing high amount of calculation makes this procedure week in application. Also, the cost function must be separable and there is no guarantee for complete satisfaction of final positions of the states.[7] Papadopoulos *et al*.[8] presented a trajectory planning strategy for nonholonomic mobile manipulators in presence of multiobstacles. They used a polynomial function to describe mobile robot trajectory science. Obstacle avoidance has been allowed by increasing the order of polynomial. Gonzalez *et al*.[9] presented an analytical algorithm to collision-free motion planning of wheeled mobile robots based on potential field method. They only considered kinematic model of mobile robot and assumed circular obstacles in the environment.

Although there have been some methods for hierarchical path planning of robots but none of them can treat with inseparable cost functions (which are generated for obstacle avoidance and specified path of end effector). Also, all of them used direct methods (instead of indirect methods of optimal control) so they may have numerical explosion for large-scale robots.[10] This paper presents a new method based on hierarchical optimal control to find MADL. This method can be applied easily on separable and inseparable cost functions and because of using indirect method of optimal control, it is applicable on any large-scale robot without numerical explosion. At first, the optimal control and hierarchical manner of it by considering the main system as a combination of several subsystems is discussed. Then in order to investigate, the efficiency of the proposed method to find the maximum payload of wheeled mobile manipulator, some simulation, and experimental studies are performed. Procedure is the combination of optimal control and hierarchical path planning for finding MADL that is suitable for inseparable cost functions and path planning in environment with presence of obstacles. Obstacle avoidance is done by defining artificial potential field in the cost function.

Simulations are done in a computer with X9650 Intel processor and with 3 GB of RAM.

The rest of paper is configured as follows: hierarchical optimal control problem and the new hierarchical algorithm for finding MADL are presented in Section 2. In Section 3, the hierarchical dynamic modeling of two groups of large-scale robotic systems contains serial manipulator and nonholonomic mobile manipulator is considered. Some case studies including simulations for a two-link planar WMM, three-links spatial WMM as a large-scale robot and experiments for a WMM named Scout are done in Section 4. Finally Section 5 concludes the paper.

2. Hierarchical Optimal Control and MADL

Tracking the optimal path is a suitable way for reducing energy consumption. However, finding MADL path is quite a bit difficult problem for large-scale robots, as they have complicated and massive equations of motion. An efficient way to solve this problem is to use a hierarchical optimization algorithm. Although there may be infinite ways to transform an optimization problem to a multilevel problem, but they are all essentially combination of two different approaches that may be termed as model-coordination method and the goal-coordination method.[11] The model-coordination method converts the integrated optimization problem into two-level problems by fixing the interaction variables. But the goal-coordination method cuts the interaction variables in order to create a first-level problem that can be decentralized easily or decomposed into some independent subproblems. The second level of this method attempts to force the first-level independent subproblems to arrive at a solution for which the interaction balance principle holds.[12] The proposed method uses a mixed procedure of these approaches and has two levels. At first-level, interaction parameters are estimated, and at second level, the optimal control of each subsystem is solved using estimated parameters. New calculated states and control variables at second level are used to generate interaction parameters in first level and this action is repeated until the difference between desired states and estimated ones becomes less than specified value. Figure 1 shows the architecture of hierarchical optimal control.

Considering the robot dynamic in the following form:

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

where x is the vector of states, u is the vector of inputs, and t shows time. It's possible to define this system as a nonlinear

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

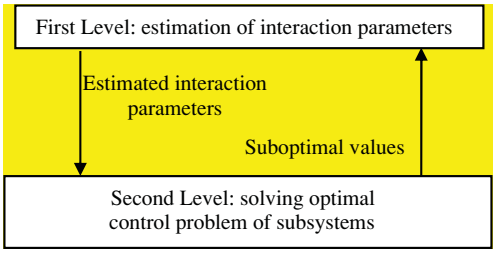


Fig. 1. Architecture of hierarchical optimal control.

interconnected system

$$\dot{x}_i = f(x_i, u_i, z_i, t), \quad i = 1, 2, \ldots, N, \tag{2}$$

where N is the number of subsystems and z_i is the vector of interactions between subsystems and is calculated from dynamic equations. If subsystems are composed of consecutive joints and links, interaction terms will be functions of control variables and states.

$$z_i = g_i(x_1, x_2, \ldots, x_N, u_1, u_2, \ldots, u_N), \tag{3}$$

where x_i is the n_i-dimensional state vector of the ith subsystem, u_i is the m_i-dimensional control vector, and z_i is the n_i-dimensional vector of interactions. In the case of path planning in present of obstacles and path planning in specified path of end effecter in which the cost function is not separable, it could be written as a set of separable and nonseparable terms. Hence, formulation of optimal control for set of subsystems is defined as Eq. (4)

$$\begin{aligned} \min \quad & J_i = \int_0^t [h_i(x_i, u_i) + l_i(x, u)]\mathrm{d}t \\ \text{subject to} : \; & \dot{x}_i = f(x_i, u_i, z_i, t), \quad i = 1, 2, \ldots, N, \end{aligned} \tag{4}$$

$$z_i = g_i(x, u), \tag{5}$$

where h_i is the separable part of subsystem's cost function and l_i is the function of nonseparable terms. The overall cost function is sum of cost functions regarding to each subsystem J_i

$$\sum_{i=1}^{N} J_i = J. \tag{6}$$

States and controls in the inseparable part of the cost function (l_i) are replaced by predicted values for states and controls (x^* and u^*). Separable parts of cost function makes no problem in optimal control solution of each subsystem as they are independent from other subsystems but inseparable parts of cost functions can be solved by fixing terms regarding to other subsystems. This job is done by substituting states and inputs of other subsystems by guessed values of them in previous level. For convergence of the system, another additional term is used. So, the new cost function of each subsystem will be defined as

$$J_i' = \int_0^t \left[h_i(x_i, u_i) + l_i(x^*, u^*) + k\|x_i - x_i^*\|_{I_n}^2\right] \mathrm{d}t, \tag{7}$$

where k is a positive constant and I_i is the n_i-dimensional identity matrix, and x^* is the n_i-dimensional "guessed" state vector. Inserting $x = x^*$ in equations of motion results in desired control vectors. Now by substituting x^* and u^* (guessed terms) into Eq. (3), interconnection terms are achieved and then by substituting them into Eq. (1), the dynamic equations will be decoupled to N independent subsystems. Now the problem becomes in the form of Eq. (8)

$$\begin{aligned} \min \quad & J_i' \\ \textit{subject to} : \; & \dot{x}_i = f(x_i, u_i, z_i, t), \quad i = 1, 2, \ldots, N. \end{aligned} \tag{8}$$

It should be noted that when $x = x^*$, this new problems solution and that of the original problem are identical. Now the problem is to iteratively find $x_i^*(i = 1, 2, \ldots, N)$ such that it approaches x_i. In order to do this, the Hamiltonian is written as

$$\begin{aligned} H = \sum_{i=1}^{N} H_i = \sum_{1}^{N} \{ & h_i(x_i, u_i) + l_i(x^*, u^*) + k\|x_i - x_i^*\|_{I_n}^2 \\ & + \lambda_i^T [f(x_i, u_i, g_i(x^*, u^*), t] + p_i(x_i - x_i^*)\}, \end{aligned} \tag{9}$$

where λ_i^{T} is the n_i-dimensional adjoin vector and p_i is n_i-dimensional Lagrange multiplier that has been added to satisfy the constraint $x = x^*$ at the optimum solution. To hold optimality conditions, the differentiation of H with respect to its variables must be vanished. MADL is the load for which actuators hit their saturation limits. For calculating MADL, system inputs are considered bounded. Introducing this constraint as Eq. (10)

$$U_i^- \le u_i \le U_i^+, \tag{10}$$

where U^+ and U^- are the upper and lower bands of actuators. Now each subsystem with bounded input torques could be solved by optimal control method. Using Eq. (9), the essential equations of optimality for each subsystem will be as follows:

$$\frac{\partial H_i}{\partial u_i} = 0, \tag{11}$$

$$\begin{gathered} \frac{\partial H_i}{\partial \lambda_i} = \dot{x}_i, \; \frac{\partial H_i}{\partial x_i} = -\dot{\lambda}_i, \\ x_i(0) = 0, \lambda_i(T) = 0, \end{gathered} \tag{12}$$

$$TU_i = \begin{cases} U_i^+ & U_i \ge U_i^+, \\ U_i & \text{otherwise}, \\ U_i^- & U_i \le U_i^-, \end{cases} \tag{13}$$

$$\frac{\partial H}{\partial x^*} = 0, \quad \frac{\partial H}{\partial p} = 0, \tag{14}$$

$$\begin{bmatrix} x^* \\ p \end{bmatrix}^{k+1} = \begin{bmatrix} x^*(x, u, \lambda) \\ p(x, u, \lambda) \end{bmatrix}^k. \tag{15}$$

Equations (11)–(12) show the essential conditions for optimization of subsystems that comes from optimal control theory and Pontriyagin's minimum principle. Equation (13) shows the bound of actuators, where U_i is the input control calculated from subsystems optimization, U_i^- and U_i^+ are

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

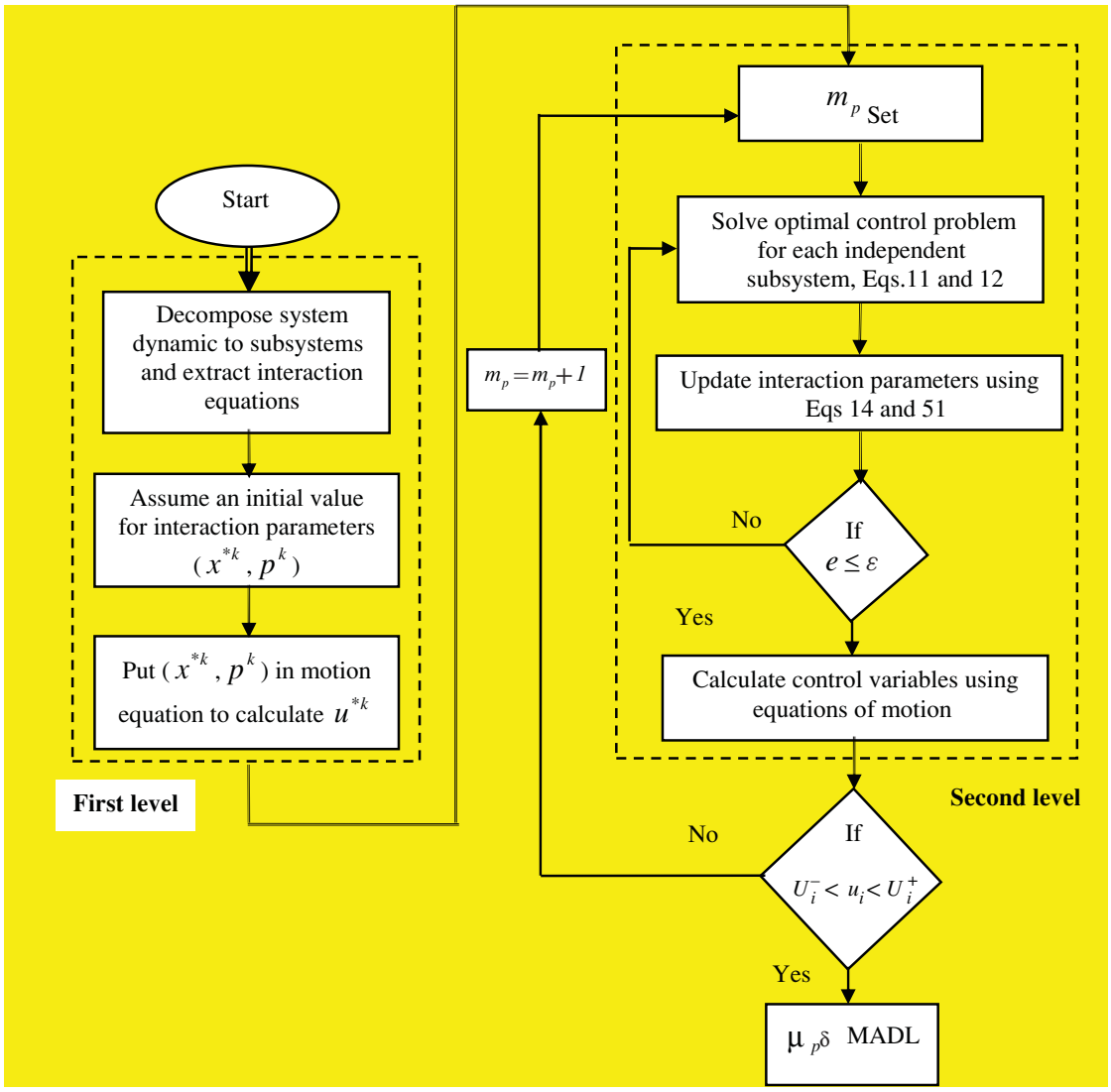


Fig. 2. The two-level algorithm for calculating MADL by a hierarchical optimal control.

bounds of inputs, and TU_i is the final calculated control input considering actuator limits. Equations (14)–(15) are used to update interaction parameters in first level.

The two-level hierarchical algorithms for calculating MADL is shown in Fig. 2. At first, overall system is decomposed into smaller scale subsystems, and then, all interaction states in equation of motion of subsystems are replaced by estimated values (x^{*k}). Using this estimated values and equations of motion, values for interaction inputs (u^{*k}) are calculated. Substituting x^{*k} and u^{*k} into Eq. (3), the vector of interaction variables for each subsystem is achieved. Now subsystems are independent from each other. By fixing the payload (m_p), optimal control problem for each subsystem is solved and optimum states and controls for each subsystem are generated. These new values for states and controls are used in Eqs. (14)–(15) to update values for x^{*k} and u^{*k} (second level). This action is iterated until the error between two following iterations becomes less than a presented value (ε). The error value is calculated from Eq. (16)

$$e = |x - x^*|, \tag{16}$$

when the error value becomes less than desired value, overall input variables are calculated from equations of motion, comparing this values with input limits determines if the preset value for m_p is MADL or not, if the control inputs are not saturated then the algorithm repeats the action with an increased value of m_p and if they are saturated, the value of MADL will be equal to last m_p.

But to use the hierarchical method, one needs to separate the overall system correctly. The next section is about converting two most common groups of large-scale robots with holonomic and nonholonomic constraints to a set of subsystems.

3. Dynamic Modeling

The most versatile mode of operation for mobile manipulator is simultaneous action of manipulator and mobile platform; in another word to increase manipulability, the coordinated dynamic of platform and manipulator must be considered. Assuming a robot as a large-scale system comes from high amount of subsystems containing links, joints, and complicated nonlinear interaction between

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

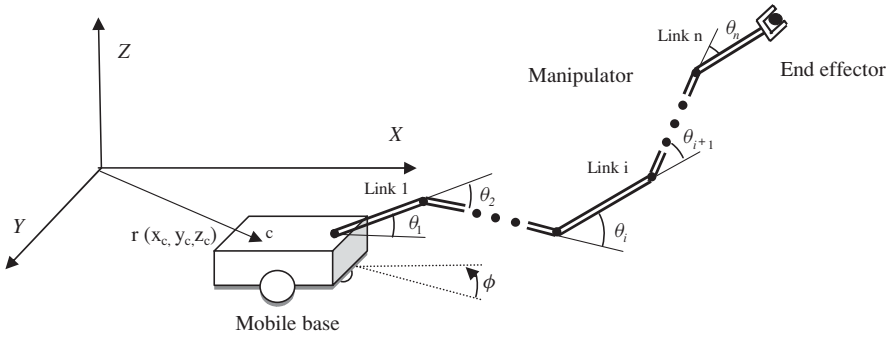


Fig. 3. Nonholonomic mobile platform with a serial link manipulator.

subsystems. Successful analysis of large-scale systems involves decomposition of the overall system into a number of subsystems with their own goals and constraints. In this section, to show the applicability of proposed method, two main groups of complicated robotic systems contain high DOF serial links as well as nonholonomic mobile manipulators are formulated. Other robotics systems can be formulated easily by using this method.

3.1. Serial manipulator

Dynamic equations of a serial manipulator can be written as Eq. (17)

$$M(q)\ddot{q} + C(q, \dot{q}) = U, \tag{17}$$

where $q \in R^{n\times 1}$ is the vector of generalized coordinates, $M \in R^{n\times n}$ is the inertia matrix, and $C \in R^{n\times 1}$ is the vector of nonlinear terms of Coriolis, centrifugal, and gravity forces. Also, $U \in R^{n\times 1}$ is the vector of input torques. So state space equations of system can be written in the following form:

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}, \tag{18}$$

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \tag{19}$$

where

$$\begin{aligned} F_1 = X_2; F_2 = D(X) + B(X_1)U, \\ B(X_1) = M^{-1}; D(X) = -M^{-1}C. \end{aligned} \tag{20}$$

For decomposing the robot, each link and consecutive joint is considered as a subsystem. So the dynamic equations of subsystems in state space can be exploited as Eqs. (21)–(23).

$$X^i = \begin{bmatrix} X_1^i \\ X_2^i \end{bmatrix} = \begin{bmatrix} q_i \\ \dot{q}_i \end{bmatrix} \Rightarrow \dot{X}^i = \begin{bmatrix} F_1(i) \\ F_2(i) \end{bmatrix}, \tag{21}$$

$$F_1(i) = X_2^i, \tag{22}$$

$$F_2(i) = D_i(x^i, z^i) + B_i(x_1^i, z^i)U. \tag{23}$$

In this formulation, the index "i" refers to the subsystem number. Also, interaction terms between subsystems are represented by the term means the effect of other subsystems on the ith subsystem.

3.2. Nonholonomic mobile manipulator

A mobile manipulator is composed of a serial manipulator and a mobile platform. But the path planning is difficult and time consuming, since high redundancy, nonholonomic constraints of mobile platform and complicated dynamic interactions between the manipulator and the mobile platform.[13] Recent tendency is to make mobile robots much more independent in path planning level from origin, means to do the path planning level fast and independently. Complicated and large-scale dynamic equations are two main hinders toward this goal. To study this matter, a serial manipulator mounted on a differentially driven nonholonomic wheeled mobile platform is considered. Wheels are independently driven by motors. The schematic of a WMM with attached coordinates to its links is shown in Fig. 3.

The system is decomposed into two subsystems, the nonholonomic platform and the holonomic manipulator. Dynamic equations of each subsystem are given as follows. Dynamic equations of mobile base (first subsystem) are in the form[13]

$$\begin{aligned} &M_{v1}(q_v)\ddot{q}_v + V_{v1}(q_v, \dot{q}_v) + V_{v2}(q_r, q_v, \dot{q}_r, \dot{q}_v) \\ &\quad = E_v\tau_v - A^T\lambda - M_{v2}(q_r, q_v)\ddot{q}_v - R(q_r, q_v)\ddot{q}_r, \end{aligned} \tag{24}$$

where M_{v1} and V_{v1} are the mass inertia and the velocity dependant vector of the platform, respectively, M_{v2} and V_{v2} represents the inertial term and centrifugal and Coriolis terms due to the presence of the manipulator, respectively, τ_v is the input torque to the vehicle, E_v is the constant matrix, λ denotes the Lagrange multipliers corresponding to the kinematic constraints, and R_v represents the inertia matrix that reflects dynamical effect of the arm motion on the vehicle. Also, dynamic equation for the mounted arm is

$$\begin{aligned} &M_r(q_r)\ddot{q}_r + V_{r1}(q_r, \dot{q}_r) + V_{r2}(q_r, \dot{q}_r, \dot{q}_v) \\ &\quad = \tau_r - R_r(q_r, q_v)\ddot{q}_v \end{aligned}, \tag{25}$$

where q_r denotes the n-dimensional Lagrangian coordinates of the manipulator, M_r is the inertia matrix, V_{r1} represents Coriolis and centrifugal terms, V_{r2} denotes centrifugal and Coriolis terms caused by the angular motion of the platform, τ_r is the input torque for the manipulator, and R_r is the inertial matrix that represents effects of the vehicle motion on the manipulator.[13] Next, the dynamic equations of subsystems could be written in state space form; for mobile base, these

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

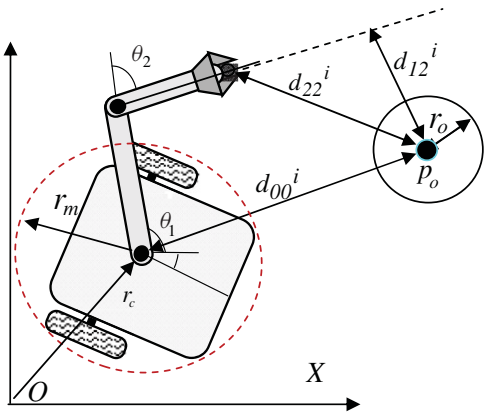


Fig. 4. Two-link planar WMM in presence of obstacle.

Table I. Properties of two-link planar WMM.

Parameter	Value	Unit
Length of links	$l_1 = l_2 = 0.5$	m
Mass of links	$m_1 = 5, m_2 = 3$	Kg
Moment of inertia of links	$I_1 = 0.416, I_2 = 0.0625$	Kg.m^2
Mass of base	94	Kg
Mass of wheels	5	Kg
Moment of inertia of base	6.609	Kg.m^2
Moment of inertia of wheels	$\begin{matrix} 0.131 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.313 \end{matrix}$	Kg.m^2
b	0.171	m
r	0.075	m
d	0	m

Table II. Maximum error value for states in the last iteration (two-link planar WMM).

Error number	Time (s)	Maximum error value
e_1	1.282	0.002
e_2	1.336	0.004
e_3	2.536	0.016
e_4	0.900	0.050
e_5	1.855	0.064
e_6	1.991	0.025
e_7	1.964	0.023
e_8	2.673	0.563
e_9	2.673	0.265
e_{10}	2.373	0.112
e_{11}	2.400	0.051

equations become as Eq. (26)

$$X^1 = \begin{bmatrix} x_1^1 \\ x_2^1 \end{bmatrix} = \begin{bmatrix} q_v \\ \eta \end{bmatrix} \Rightarrow \dot{X}^1 = \begin{bmatrix} Z_1 \\ N_1 \end{bmatrix} + \begin{bmatrix} 0 \\ N_2 \end{bmatrix} \tau_v,$$
$$q_v = \begin{bmatrix} x \\ y \\ \varphi \\ \theta_r \\ \theta_l \end{bmatrix}; \eta = \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix}, \tag{26}$$

where q_v is a (5×1) vector of Lagrange coordinates of nonholonomic base and η is of order (2×1). Also

$$Z_1 = S\left(x_1^1\right) x_2^1, \tag{27}$$

$$\begin{aligned} N_1 = &\left(S^{\mathrm{T}} M_{v1}\left(x_1^1\right) S\right. \\ &\left. + S^{\mathrm{T}} M_{v1}\left(x_1^1, x_1^1\right) S\right)^{-1} - S^{\mathrm{T}} M_{v1}\left(x_1^1\right) \dot{S} x_2^1 \\ &- S^{\mathrm{T}}\left(C_{v1}\left(x_1^1, x_2^1\right) - C_{v2}\left(z_1^1, x_1^1, z_2^1, x_2^1\right)\right) \\ &- S^{\mathrm{T}} M_{v1}\left(x_1^1, z_1^1\right) \dot{S} x_2^1 - S^{\mathrm{T}} R_v\left(x_1^1, z_1^1\right) z_3^1 \end{aligned}, \tag{28}$$

$$N_2 = \left(S^{\mathrm{T}} M_{v1}\left(x_1^1\right) S + S^{\mathrm{T}} M_{v2}\left(x_1^1, z_1^1\right) S\right)^{-1} E_v, \tag{29}$$

where

$$\begin{aligned} Z^1 = \begin{bmatrix} z_1^1 \\ z_2^1 \\ z_3^1 \end{bmatrix} = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \dot{x}_2^2 \end{bmatrix}; x_1^2 = q_r = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \\ x_2^2 = \dot{q}_r = \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix}, \end{aligned} \tag{30}$$

where M_{v1} is of order (5×5) and $S$is of order (2×5), then X^1 in Eq. (26) is of order (7×1), and extracting orders of Z_1, N_1, N_2 the right-hand side of Eq. (26) is of order (7×1). And for manipulator arm state space equations are as Eqs. (31)–(34)

$$\begin{aligned} X^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} q_r \\ \dot{q}_r \end{bmatrix}, \\ \Rightarrow \dot{X}^2 = \begin{bmatrix} x_2^2 \\ M_1 \end{bmatrix} + \begin{bmatrix} 0 \\ M_2 \end{bmatrix} \tau_r, \\ q_r = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \end{aligned} \tag{31}$$

where

$$\begin{aligned} M_1 = M_r^{-1}\left(x_1^2\right)\left(V_{r1}\left(x_1^2, x_2^2\right) - V_{r2}\left(x_1^2, x_2^2, z_2^2\right)\right. \\ \left. - R_r\left(x_1^2, z_1^2\right)\left(S z_3^2 + \dot{S} z_2^2\right)\right) \end{aligned}, \tag{32}$$

$$M_2 = M_r^{-1}\left(x_1^2\right), \tag{33}$$

$$Z^2 = \begin{bmatrix} z_1^2 \\ z_2^2 \\ z_3^2 \end{bmatrix} = \begin{bmatrix} x_1^1 \\ x_2^1 \\ \dot{x}_2^1 \end{bmatrix}; \quad x_1^1 = q_v = \begin{bmatrix} x \\ y \\ \phi \\ \theta_r \\ \theta_l \end{bmatrix}; \tag{34}$$
$$x_2^1 = \eta = \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix},$$

where q_r is of order $(n \times 1)$, M_r is of order $(n \times n)$, V_{r1} and V_{r2} are of order $(n \times 1)$, so X^2 in Eq. (31) is of order $(2n \times 1)$ and order compatibility in this equation holds.

4. Simulations and Experimental Studies

To study the feasibility of this method, some case studies are done. First case study is about complicated cost function in

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

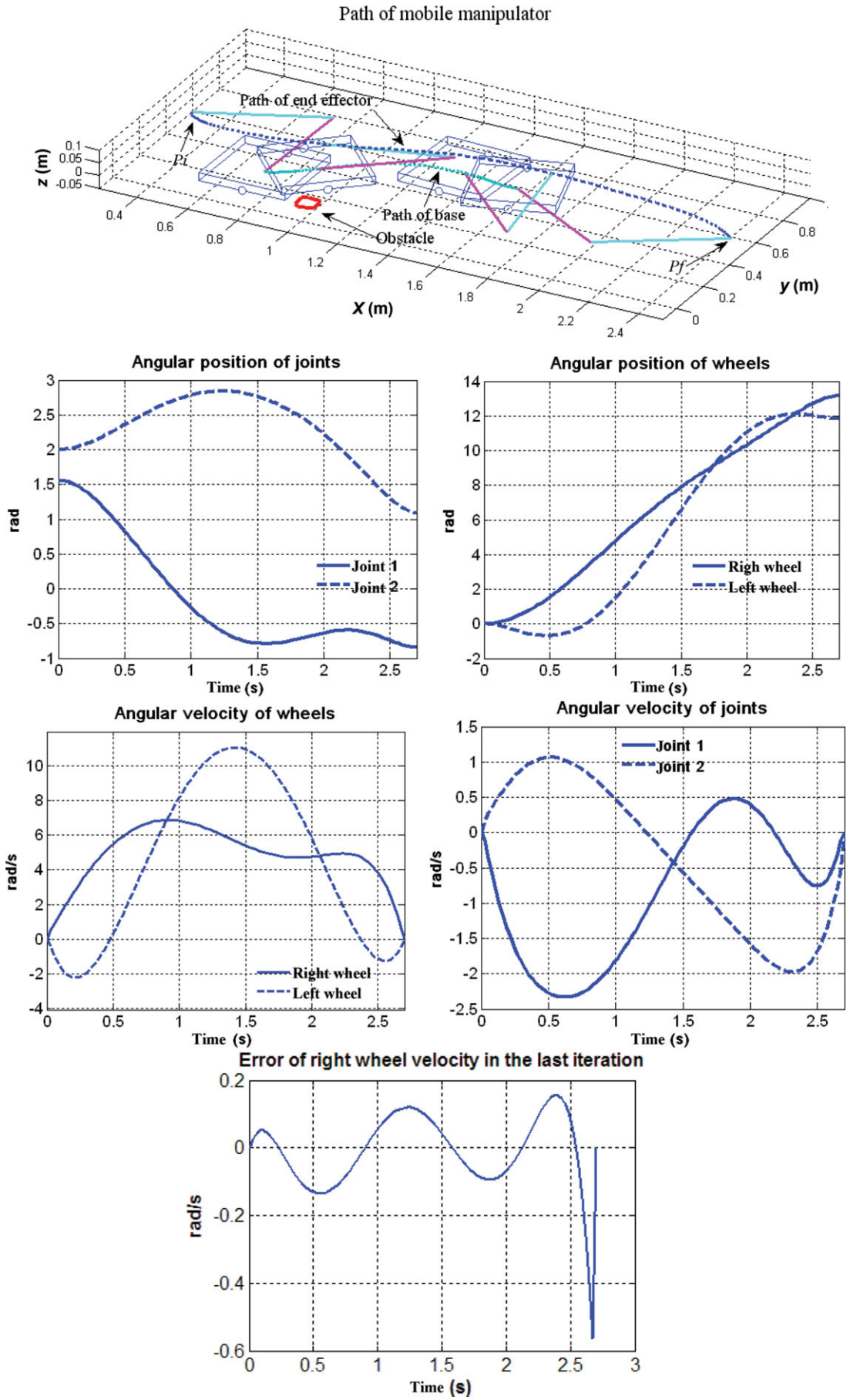


Fig. 5. Hierarchical optimal path planning in presence of obstacle.

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

Table III. Properties of 3p WMM.

Parameter	Value	Unit
Length of links	$l_1 = 0.4, l_2 = l_3 = 0.5$	m
Mass of links	$m_1 = 12, m_2 = 10, m_3 = 5$	kg
Moment of inertia of link	$I_1 = 0.2, I_2 = 0.2, I_3 = 0.1$	kg.m^2

which obstacle avoidance considered to show that the method is able to solve optimal control problems with inseparable cost functions. Second one is about finding MADL for a three-link WMM as a large-scale robot, and finally, MADL of robot named Scout is calculated and verified by experimental results.

4.1. Two arm nonholonomic mobile manipulator in point-to-point motion in presence of obstacle

This case study is about path planning of a planar WMM in presence of obstacle (Fig. 4), but to avoid more complexity, finding MADL is ignored and it is done for a known load on end effector. One method for optimal path planning in presence of obstacle is utilizing artificial potential field in the cost function with velocity regarded states and control inputs, but it causes the cost function to be inseparable due to nonlinear terms of distances. The path planning problem in presence of obstacle is accurately solvable using the new hierarchical algorithm.

For a given WMM in presence of obstacle as shown in Fig. 4, the distance between center of mobile base and center of obstacle will be as

$$d^i_{00} = \|r_c - P_o\| = \sqrt{(X_c - X_{obi})^2 + (Y_c - Y_{obi})^2}. \quad (35)$$

The distance between first link and center of obstacle is

$$d^i_{11} = \frac{\left| det \begin{bmatrix} l_1 Cos(\phi + \theta_1) & X_c - X_{obi} \\ l_1 Sin(\phi + \theta_1) & Y_c - Y_{obi} \end{bmatrix} \right|}{l_1}, \quad (36)$$

the distance between first joint and center of obstacle is

$$d^i_{21} = \sqrt{\begin{matrix} (X_c + l_1 Cos(\phi + \theta_1) - X_{obi})^2 \\ +(Y_c + l_1 Sin(\phi + \theta_1) - Y_{obi})^2 \end{matrix}}, \quad (37)$$

The distance between second link and center of obstacle is

$$d^i_{12} = \frac{\left| det \begin{bmatrix} l_2 Cos(\phi + \theta_1 + \theta_2) & x_1 + l_1 Cos(\phi + \theta_1) - X_{obi} \\ l_2 Sin(\phi + \theta_1 + \theta_2) & x_2 + l_1 Sin(\phi + \theta_1) - Y_{obi} \end{bmatrix} \right|}{l_2}, \quad (38)$$

and the distance between end effector and center of obstacle is

$$d^i_{22} = \sqrt{(X_e - X_{obi})^2 + (Y_e - Y_{obi})^2}. \quad (39)$$

Hence, assuming the links as a line, the distance parts of robot from ith obstacle d^i_{jk} can be determined from the geometry and positions,[4] where d^i_{jk} with j, $k = 0$ is about distance mobile body from ith obstacle, and d^i_{jk} with $j = 1$ is about distance of links ($k = 1$ for first link and $k = 2$ for second link) from ith obstacle and with $j = 2$ is about the joints of first and second links. So, the path planning algorithm must minimize the following cost function:[4]

$$\left(\frac{1}{2} \|X_v\|^2_Q + \frac{1}{2} \|U\|^2_R + J = \int_{t_0}^{t_f} \frac{1}{2} \left\| \frac{1}{r_o + r_m + s_p - d^i} \right\|^2_{w^i_{00}} + \frac{1}{2} \left\| \frac{1}{r_o + s_p - d^i_{jk}} \right\|^2_{w^i_{jk}} \right) dt, \quad (40)$$

where $\|X_v\|^2_Q = X^{\mathrm{T}}_v Q X_v$ is the sum of squares of joint and wheel velocities, Q is the symmetric, positive semi-definite weighting matrix, R is the symmetric, positive definite matrix, r_m is the radius of mobile base, r_0 is the radios of obstacle (Fig. 4), and s_p is the safety factor related to specifications of robot. First terms in Eq. (40) minimize the angular velocity in order to avoid large joint velocities. Second term in Eq. (40) is about minimizing the torques of wheels and joints, so one can decide on the relative importance among the angular velocities of joints, torques of motors, and obstacle avoidance of each part by the choice of weighting matrixes R, Q, and w^i_{jk}. As it is seen, the cost function is combined from complicated functions of states;

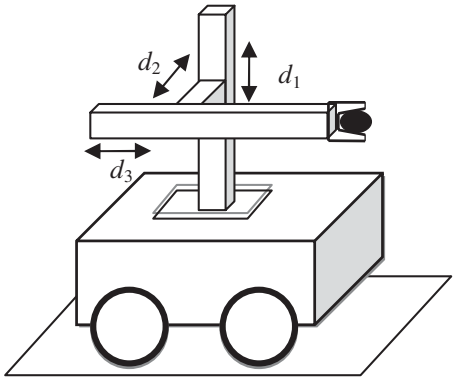


Fig. 6. Nonholonomic mobile platform with three-link spatial prismatic manipulator.

Table IV. Maximum error value for states in the last iteration (3p spatial nonholonomic WMM).

Error number	Time (s)	Maximum error value
e_1	3.737	0.011
e_2	3.566	0.007
e_3	1.286	0.031
e_4	1.384	0.212
e_5	3.737	0.185
e_6	2.882	0.005
e_7	2.980	0.006
e_8	2.929	0.084
e_9	0.556	0.246
e_{10}	4.495	0.237
e_{11}	2.071	0.002
e_{12}	3.333	2.694
e_{13}	2.576	0.089

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

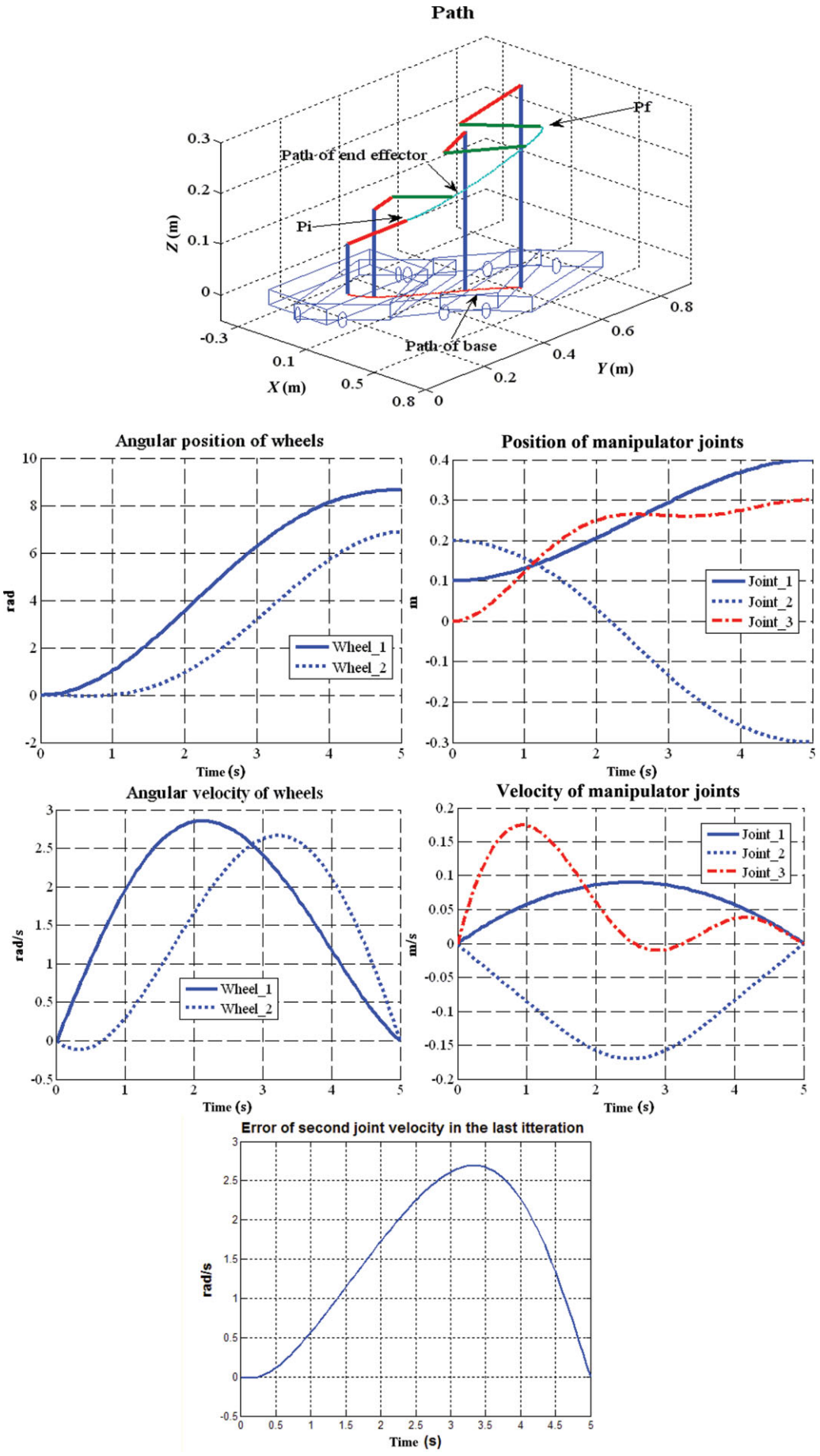


Fig. 7. Simulation results for 3p spatial nonholonomic WMM.

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

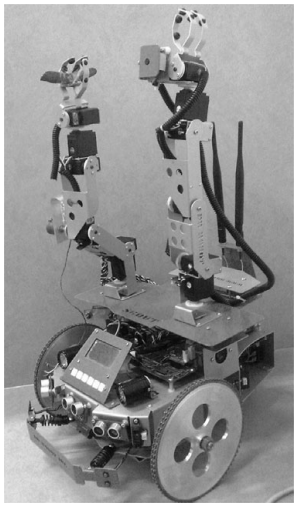


Fig. 8. Nonholonomic Scout robot.

hence, it is impossible to separate them into terms that are only functions of states of each subsystem. So using the proposed method, these terms can be considered within l_i in Eq. (4). By replacing predicted values of states and controls in the inseparable part of each subsystem's cost function, they will become independent from other subsystems, and then, calculations can be done in parallel manner. Simulation results for this study in which there is obstacle in environment are shown in Fig. 5, which is the reason that the method has ability to solve complicated problems with inseparable cost functions. The number of iterations for this case study is four and the computing time is about 2 h. The characteristics of parts are shown in Table I.

Existing the obstacle terms in the cost function, which are highly nonlinear, makes the other hierarchical methods told before, unable to solve this problem exactly, and in another word, if other methods yet are used, one must first decompose the cost function that is difficult and approximate. The maximum error value of each state and corresponding time of occurrence in the last iteration is shown in the Table II.

4.2. *Spatial prismatic nonholonomic WMM*

A spatial three-link prismatic robot mounted on a nonholonomic base is considered as large-scale WMM, which is shown in Fig. 6, and the new hierarchical algorithm is used for path planning of it.

But because of preventing more complexity, MADL computation is ignored and simulation is done for known load. Suppose that initially the load is at a point with coordinates p_i ($Xe = 0$m, $Ye = 0.2$m, $Ze = 0.1$m) and it must reach to final point with coordinates p_f ($Xe = 2.3$m, $Ye = 2.5$m, $Ze = 0.4$m). The manipulator characteristics are listed in Table III.

The original system is decomposed into two subsystems, the first one is the base and the second one is the system of the arms. First subsystem is of order 7 and the second

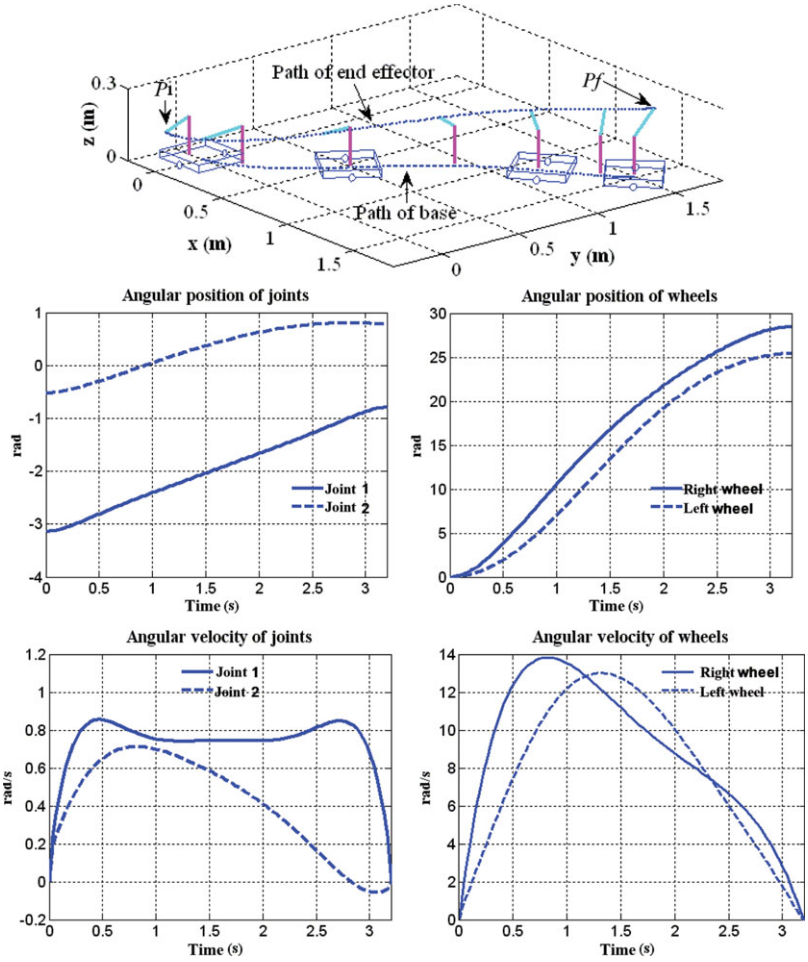


Fig. 9. Simulation results of Scout robot when carrying 80 g load.

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

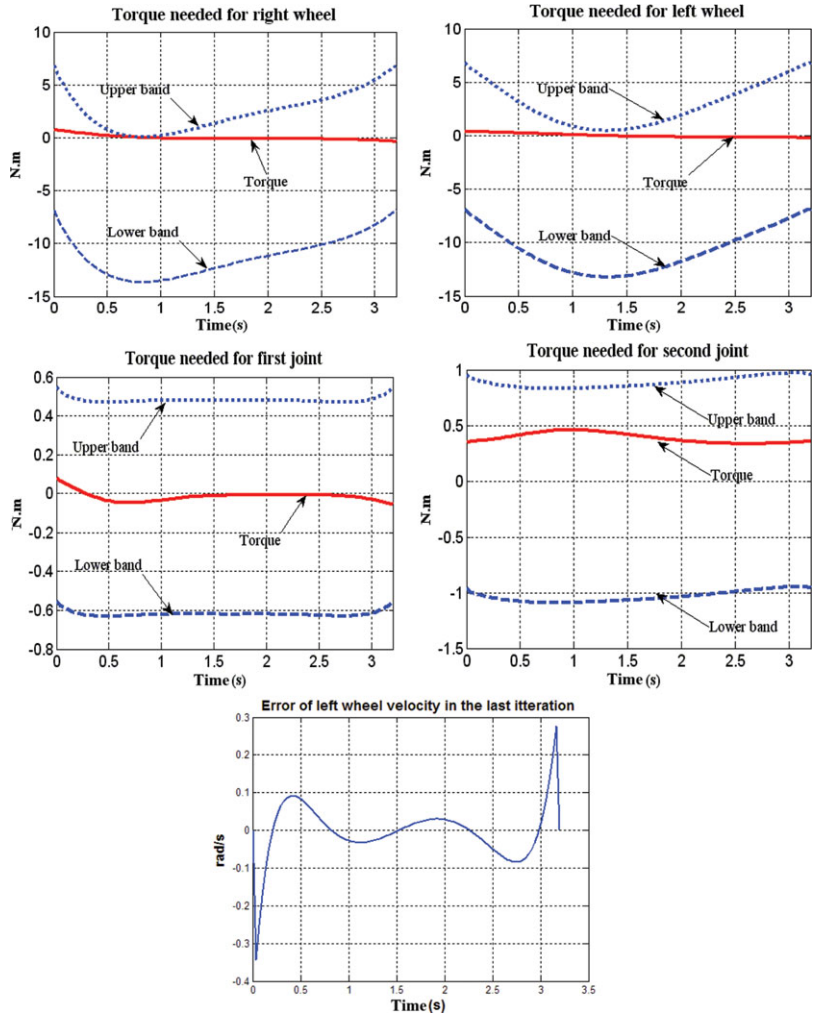


Fig. 9. Continued.

Table V. Maximum error value for states in the last iteration.

Error number	Time (s)	Maximum error value
e_1	0.226	0.003
e_2	0.2877	0.001
e_3	3.038	0.005
e_4	0.259	0.030
e_5	2.974	0.040
e_6	3.006	0.024
e_7	0.356	0.008
e_8	0.032	0.230
e_9	0.032	0.344
e_{10}	3.168	0.248
e_{11}	0.032	0.042

one is order of 6. The weight of convergence (k_1, k_2) is set to 100. The simulation time is 5s and the payload is 14.5kg. The simulation is done with hierarchical method. State trajectories and control variables as well as the path of end effectors and nonholonomic base are given in Fig. 7. This figure also *illustrates* the error of second joint velocity in the last iteration. This simulation iterates two times and the computing time is 1 h.

As the value of convergence weights (k_1, k_2) increase the degree of coincidence increase. The maximum error value of states and corresponding time of occurrence in the last iteration is shown in the Table IV.

4.3. Spatial two-link nonholonomic WMM

Implementation of the proposed method is done on a commercial, off the shelf WMM, named Scout (Fig. 8). First simulation is done, and then, the method is implemented on Scout. The manipulators have three revolute joints but to get the fast results, one joint is fixed and two other joints are considered in simulation.

Dynamic equations of robot are derived and decomposed into two subsystems containing mobile base and serial manipulator. Robot carries a load from start point $P_i = (-0.18\text{m}, -0.08\text{m}, 0.05\text{m})$ to final point $P_f = (1.70\text{m}, 1.455\text{m}, 0.30\text{m})$ at 3.2 s. So the robot's parameters in these points are $P_i(X = 0\,\text{m}, Y = 0\,\text{m}, \varphi = 0\text{rad}, \theta_1 = -n\text{rad}, \theta_2 = n/6\text{rad})$ and $P_f(X = 1.5\text{m}, Y = 1.5\text{m}, \varphi = 0.84\text{rad}, \theta_1 = -n/4\text{rad}, \theta_2 = n/4\text{rad})$. Diagrams of states and also control inputs are shown in Fig. 9. This figure also illustrates the error of left wheel velocity in the last iteration.

Values for torque limits are extracted from specification catalogue of Scout. Simulation results show saturation of right wheel in the time of about 1 s for load of 80 g. So the simulation predicts the value of MADL to equals 80 g.

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

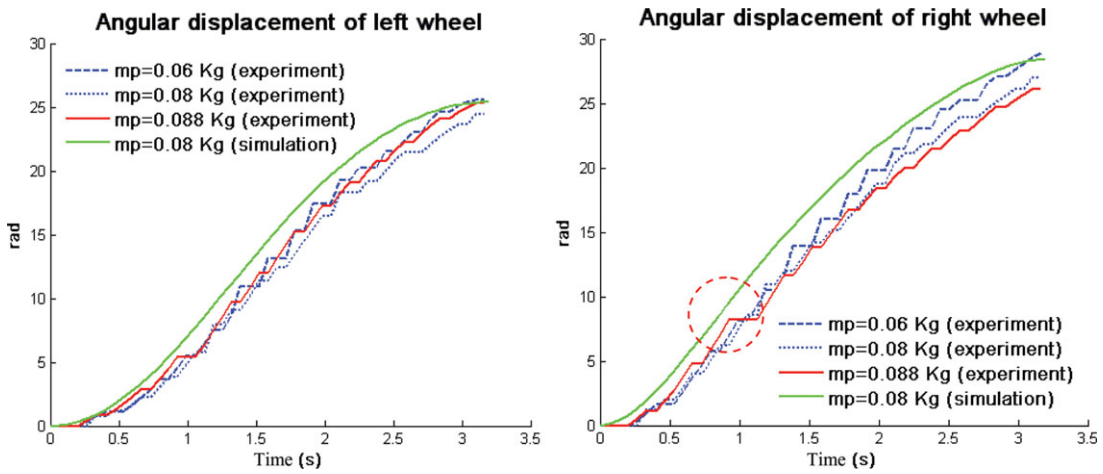


Fig. 10. Comparison of experimental and simulation results for Scout.

The maximum error value of states and corresponding time of occurrence in the last iteration is shown in the Table V. In this case study, the computing time is about 1 h and the algorithm iterates two times to solve the problem.

The robot is controlled using software launched on a PC, which is in contact with the robot in a wireless network mesh. The robot is also shipped with a set of libraries to control it by providing different sets of methods. In our implementation, we controlled the robot using velocity input for wheels and angle positions for link's joints. Offline data are used as input for the program. In each step, one set of wheels' velocity and joints' angles is read from input file and send to the robot's motion controller board. Prior to initiating this process, tuned PID values are provided to the robot. Input velocity data are converted to encoder speed that is calculated based on number of pins on the encoder (in this case is 800). Output files are later produced by saving above data in relevant files. The method is applied on Scout experimentally as it is shown in Fig. 10. The diagrams related to manipulator joints are ignored because their motors are not saturated.

In this work, changing of actuator behaviors is considered as the criterion for experimentally determination of MADL. For this purpose, the load of the end effector is increased until one of the actuators hits the saturation bands. As it is clear in the diagrams of Fig. 10 for the load of 80 g, there is no change of actuators behavior, but when the load is increased to 88 g there is a change in behavior of right motor in about 1 s and saturation in this point happens that is shown by the circle. So there is about 10% error between simulation and experimentally determination of MADL by the new hierarchical method.

5. Conclusions

The main objective of this work was formulating the new hierarchical optimal control for path planning and finding MADL of large-scale robots. The main advantage of this work is ability of treating with inseparable cost functions in comparison to other methods yet. Also, the new method does not have numerical explosion problem of last methods yet. We proposed a two-level algorithm for path planning and finding MADL. At first level, interaction values are estimated, and at second level, the optimal control of each subsystem is solved using estimated parameters. New calculated states and control variables at second level are used to generate interaction parameters in first level. This action is iterated until the error between two following iterations becomes less than a desired value. When the error value becomes less than desired value, overall input variables are calculated from equations of motion, comparing these values with input limits determines if the preset value for the load is the MADL or not, if the control inputs are not saturated then the algorithm repeats the action with an increased value of load, and if they are saturated then the value of MADL will be equal to last step load. The first example demonstrated the efficiency and implementation of algorithm on inseparable cost function. Second, example showed it applicability on large-scale three-link nonholonomic wheeled mobile manipulator. Also, the proposed algorithm implemented on robot named Scout and simulation results compared with experimental results. The MADL of Scout resulted from experiment has 10% difference in comparison with simulation and shows the applicability and effectiveness of proposed method.

References

1. C. E. Wang, W. K. Timoszyk and J. E. Bobrow, "Payload maximization for open chained manipulators: Finding weightlifting motions for a puma 762 robot," *IEEE Trans. Robot. Autom.* **17**(2), 218–224 (2001).
2. H. Ghariblu and M. H. Korayem, "Trajectory optimization of flexible mobile manipulators," *Robotica* **24**(3), 333–335 (2006).
3. M. H. Korayem, A. Nikoobin and V. Azimirad, "Maximum load carrying capacity of mobile manipulators: Optimal control approach," *Robotica* **27**, 147–159 (2009).
4. 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**, 811–830 (2010).
5. A. Mohri, S. Furuno, M. Iwamura and M. Yamamoto, "Sub-Optimal Trajectory Planning of Mobile Manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea 2 (2001) pp. 1271–1276.
6. N. Sadati and A. Babazadeh, "Optimal control of robot manipulators with a new two-level gradient-based approach," *J. Electr. Eng.* **88**, 383–393 (2006).

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press

7. N. Sadati and M. M. Emamzadeh, "Optimal Control of Robot Manipulators Using Fuzzy Interaction Prediction System," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Kunming (2006) pp. 1071–1076.
8. E. Papadopoulos, I. Poulakakis and I. Papadimitriou, "On path planning and obstacle avoidance for nonholonomic mobile manipulators: a polynomial approach," *Int. J. Robot. Res.* **21**, 367–383 (2002).
9. V. J. Gonzalez, R. Parkin, M. L. Para and J. M. Dorador, "A wheeled mobile robot with obstacle avoidance capability", *Mech. Technologia* **1**, 150–159 (2004).
10. T. Chettibi, H. E. Lehtihet, M. Haddad and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *Eur. J. Mech.* **23**, 703–715 (2004).
11. M. Jamshidi, *Large-Scale Systems: Modeling and Control* (North Holland, USA, 1983).
12. D. A. Wismer, *Optimization Methods for Large-Scale Systems with Applications* (McGraw Hill, USA, 1971).
13. 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).

https://doi.org/10.1017/S0263574711000336 Published online by Cambridge University Press