Dynamic Modelling and Genetic-Based Motion Planning of Mobile Manipulator Systems with Nonholonomic Constraints

M W Chen and A M S Zalzala

Robotics Research Group
Dept of Automatic Control and Systems Engineering
University of Sheffield, UK
email: rrg@sheffield.ac.uk

Research Report No.600

September 1995

Abstract

An approach for modelling and motion planning of a mobile manipulator system with non-holonomic constraint is presented in this paper. The Newton-Euler equations are used to obtain the complete dynamics of the system. Given the trajectory of the end-effector of the manipulator, near-optimal trajectories for mobile platform and manipulator joints are obtained by using an efficient genetic algorithm with torque and manipulability optimisation and obstacle avoidance. An obstacle avoidance scheme is presented by applying geometric analysis. Various simulations of a platform with a 3-link onboard manipulator are presented to show the effectiveness of the presented method.

Keywords: robotics, motion planning, genetic algorithms, obstacle avoidance.

1. Introduction

A mobile manipulator system in this paper is a mobile robot with an onboard robotic manipulator. A typical characteristics of a mobile manipulator is its high degree of kinematic redundancy created by the addition of the mobile platform's degrees of freedom to the manipulator's. A basic task of a mobile manipulator is to move the platform and the manipulator simultaneously for the end-effector to follow a predefined trajectory. The redundancy of a mobile manipulator, quite desirable for dexterous manipulation and transport functions in cluttered environments, allows the system to generate optimal trajectories for the mobile platform and the manipulator.

Considerable effort is being devoted to the motion planning of mobile robots or base fixed manipulators in the literature[1-6], while studies on mobile manipulators are very limited. What makes the motion planning of a mobile manipulator system difficult is that a wheeled mobile platform is subject to non-holonomic constraints while a manipulator is usually unconstrained.

Carriker et al[7] presented a path planning algorithm for mobile manipulators for multiple task execution by using simulated annealing. Zhao et al[8] solved a similar problem by a genetic algorithm. Seraji[9] utilised the extra degrees of freedom of a mobile manipulator to meet user defined tasks. However, Most of previous papers neglected the dynamics and non-holonomic aspects of mobile manipulators.

Dynamic modelling of mechanical systems with non-holonomic constraints is richly documented by work ranging from Neimark and Fufaev's comprehensive book[10] to more recent developments[11]. Yamamoto and Yun[12] derived the dynamic modelling of a mobile manipulator, but they did not consider the dynamics of the onboard manipulator. While Jagannathan et al[13][14] obtained the dynamic equations of a composite mobile robot arm system from Lagrange's equations, the actuator torques can not be obtained directly from their dynamic equations when the trajectory of the end-effector is specified in the Cartesian space.

Traditional approaches to the motion planning and control of mobile manipulators require the manipulator to be fixed during base motion, and the base to be anchored during manipulator motion[12]. These requirements are unreasonable from the point of view of practical uses. Therefore, trajectory generation for the simultaneous motion of the mobile base and the manipulator of a mobile manipulator with non-holonomic constraints is important.

Because of the redundancy of the mobile manipulator system's degree of freedom, there are many feasible motions between the starting and ending points when only the end-effector's trajectory is specified. Therefore, it is meaningful to search for optimal motions of a mobile manipulator system in meeting certain criteria. The optimal joint trajectory search problem, actually, is a multi-criteria non-linear optimisation problem. Due to the competition of various criteria, multi-criteria optimisation problem often exhibits local minima and many traditional methods fail in tackling this problem[15]. Some are very sensitive to the initial guess and fail to converge unless the guess is sufficiently close to the correct solution, and some are very easy to get stuck in local minima. Genetic algorithms(GAs) are robust search and optimisation methods based on natural selection. They search for the optimum globally and therefore can avoid being trapped in local minima[19]. Moreover, it is easy to combine new criteria into the cost function. Many results[20-24] have shown that genetic-based algorithms perform better than traditional optimisation methods.

The major difficulties with a genetic algorithm are its complexity and slow convergence speed. Usually, in a genetic algorithm, a large population is used and a large number of generations are required to achieve a good result and generated paths need to be smoothed in order to speed up the convergence. Moreover, in most existing motion planning problems which use GAs, trajectory via-points are encoded, and in order to meet the accuracy requirement, many via-points are used. Hence the strings or chromosomes used in a genetic algorithm are very long and thus put a heavy computation in the search process. In order to avoid the above problems and improve the efficiency of genetic algorithms, a different method is used in this paper. The trajectories of a motion planning problem are represented by a polynomial, and hence smoother trajectories are produced and a lower cost is expected. The proposed genetic algorithm requires a smaller population and converges to a near optimum in fewer generations. And in stead of encoding the via points, we encode the parameters of a polynomial. Therefore, the length of a string is determined only by the order of the proposed polynomial, which is much shorter than that represented by via-points.

The rest of the paper is organised as follows. In the next section, the complete dynamic model of a mobile manipulator system is obtained by using Newton-Euler equations. The mobile base is represented by three links with two fictitious ones. In Section 3, the mobile manipulator system trajectory problem is formulated as an optimisation problem with torque minimisation, manipulability maximisation and obstacle avoidance. An obstacle avoidance scheme is presented by using geometric analysis. In Section 4, an efficient genetic algorithm is proposed to search for optimal trajectories for both the mobile platform and the manipulator. Various simulations for a system including a three-link manipulator mounted on a mobile platform are presented in Section 5 to demonstrate the central ideas and the efficiency of the proposed method. Finally, some conclusions are drawn in Section 6.

2. Dynamic modelling of the mobile manipulator system

Consider a mobile platform with an onboard manipulator as shown in Figure 1. The manipulator has one rotational link and two planar links. The platform has two driving wheels(the centre ones) and four passive ones(the corner ones). The two driving wheels are independently driven by two motors.

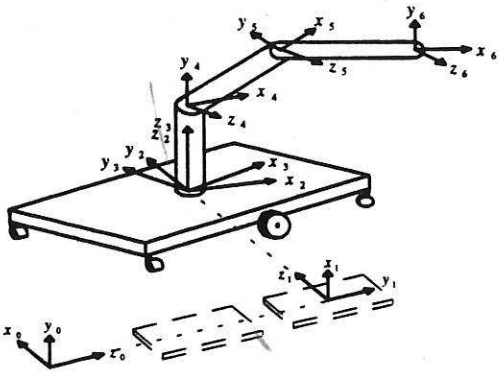


Figure 1 A mobile platform with an onboard manipulator

Consider the inertial reference frame in the $(\mathbf{Z}_0, \mathbf{X}_0)$ plane and choose a point P along the axis of the driving wheels on the mobile platform whose frame is $(\mathbf{X}_3, \mathbf{Y}_3)$ in this plane. The mobile platform at point P can be described by three variables(z, x, θ), where (z, x) denotes the Cartesian position and θ describes the heading angle measured between $\mathbf{X}_3$ and $\mathbf{Z}_0$(see Figure 2) respectively in the world frame. For the manipulator, the joint angles of the three links are $\theta_4, \theta_5, \theta_6$. Define the generalised coordinates

$$\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)^T \tag{1}$$

where q_1, q_2 and q_3 denote the position(z, x) and the heading angle θ of the platform, respectively, and q_4, q_5, and q_6 denote the joint angles θ_4, θ_5 and θ_6, respectively.

The platform is subject to the following non-holonomic constraints:

$$-\dot{z}\sin\theta + \dot{x}\cos\theta = 0 \tag{2}$$

i.e., the platform must move in the direction of the axis of symmetry. Note that the position (z, x) and the heading angle θ of the platform are not independent of each other due to the non-holonomic constraint.

In order to apply the Newton-Euler equations to obtain the dynamic equations, it is convenient to visualise the platform as a planar joint having three degrees of freedom. Any multiple degrees of freedom joint, such as the planar joint can be synthesised by an appropriate number of single degree of freedom joints with zero link length and zero link mass[16]. Here, the platform is modelled as a serial chain of two prismatic joints and one revolute joint, as shown in Figure 1.

The coordinate systems for the composite mobile manipulator system are given in the Figure 1. Given the parameters of the system, the transformation matrix can be obtained accordingly. By applying the Newton-Euler equations[17], all the joint torques can be obtained by iterations. The manipulator joint torques are calculated from following equations:

$$\tau_i = \mathbf{n}_i^T \mathbf{z}_{i-1}, \quad i = 4, 5, 6. \tag{3}$$

where $\mathbf{n}_i$ is the moment exerted on link i by link i-1 at the coordinate frame($\mathbf{x}_{i-1}, \mathbf{y}_{i-1}, \mathbf{z}_{i-1}$).

Because the platform is subject to a non-holonomic constraint, a centripetal force is exerted on the two fictitious prismatic links. Therefore, the input forces at these two links are

$$f_{pz} = \mathbf{f}_1^T \mathbf{z}_0 + N_c \sin\theta \tag{4}$$

$$f_{px} = \mathbf{f}_2^T \mathbf{z}_1 - N_c \cos\theta \tag{5}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the external forces exerted on link 1 and link 2, f_{pz} and f_{px} are the input forces at the two fictitious prismatic links in z-direction and x-direction, respectively. N_c, as shown in Figure 2, is the centripetal force due to the non-holonomic constraint.

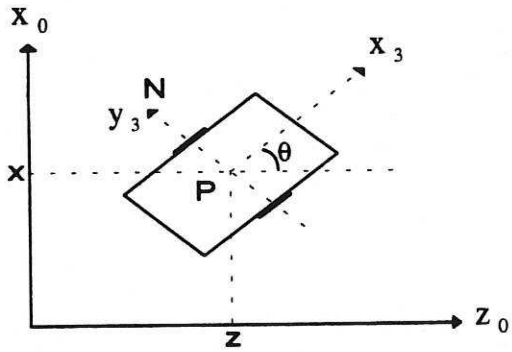


Figure 2 Top view of the mobile platform

The dynamic torque for the rotation of the platform is:

$$\tau_p = I\ddot{\theta} \tag{6}$$

where I is the rotational inertia of the platform including the onboard manipulator about the $\mathbf{Z}_3$ axis.

Considering that the platform has one driving wheel on each side, the input forces and torque for the platform are determined by

$$f_{pz} = \frac{\cos\theta(\tau_r + \tau_l)}{r} \tag{7}$$

$$f_{px} = \frac{\sin\theta(\tau_r + \tau_l)}{r} \tag{8}$$

$$\tau_p = \frac{R}{2r}(\tau_r - \tau_l) \tag{9}$$

where τ_r, τ_l denote the input torques at the right and left wheels, r is the radius of the wheels, R is the width of the mobile platform.

Considering that the two fictitious links are massless, by Newton-Euler iteration, we have

$$\mathbf{f}_1{}^T \mathbf{z}_0 = \cos\theta \ f_{3x} - \sin\theta \ f_{3y} + m_3 \ddot{z} \tag{10}$$

$$\mathbf{f}_2{}^T \mathbf{z}_1 = \sin\theta \ f_{3x} + \cos\theta \ f_{3y} + m_3 \ddot{x} \tag{11}$$

where f_{3x}, f_{3y} are the projections of the force $\mathbf{f}_3$ onto the $\mathbf{X}_3$, $\mathbf{Y}_3$ axis, respectively, m_3 is the mass of the platform.

Inserting (7)(8) and (10)(11) into (4) and (5), we obtain

$$m_3 \ddot{z} = \frac{\cos\theta(\tau_r + \tau_l)}{r} - \cos\theta \ f_{3x} + \sin\theta \ f_{3y} - N_c \sin\theta \tag{12}$$

$$m_3 \ddot{x} = \frac{\sin\theta(\tau_r + \tau_l)}{r} - \sin\theta \ f_{3x} - \cos\theta \ f_{3y} + N_c \cos\theta \tag{13}$$

Differentiating (2), multiplying (12) and (13) by $-\sin\theta$ and $\cos\theta$ respectively, and adding, we obtain

$$N_c = m_3(\dot{z}\cos\theta + \dot{x}\sin\theta)\dot{\theta} + f_{3y} \tag{14}$$

3. Problem formulation for motion planning

The motion planning problem of a mobile manipulator with non-holonomic constraints is defined as follows:

a. The kinematic relations for a mobile manipulator is given by

$$\mathbf{X}_e(t) = \mathbf{X}_p(t) + \mathbf{X}_{m/p}(\mathbf{\Phi}(t)) \tag{15}$$

where $\mathbf{X}_e(t)$ is the given Cartesian trajectory of the end-effector of the manipulator in the world frame, $\mathbf{X}_p(t)$ is the Cartesian position trajectory of the platform in the world frame, $\mathbf{X}_{m/p}(\mathbf{\Phi}(t))$ represents the vector of the position of the end-effector with respect to the platform reference frame and $\mathbf{\Phi}$ is the vector of manipulator joint angles.

b. It can be assumed in general that the components of the joint angles of the manipulator are constrained by independent upper and lower bounds specified by the vectors $\mathbf{\Phi}_l$, $\mathbf{\Phi}_u$. Thus, the constraints can be described by

$$\mathbf{\Phi}_l \le \mathbf{\Phi} \le \mathbf{\Phi}_u \tag{16}$$

where the vector inequalities are applied componentwise.

c. The platform is subject to the no slipping nonholonomic constraint,

$$-\dot{z}\sin\theta + \dot{x}\cos\theta = 0 \tag{17}$$

d. Cost function

Various optimisation criteria can be applied in the motion planning of the mobile manipulator system. In this study, total actuator torque minimisation and manipulator manipulability measure maximisation are used.

The total actuator torque over the time interval starting at $t = t_0$ and ending at $t = t_f$ is defined as

$$f_1 = \int_{t_0}^{t_f} \tau^T \tau \, dt \tag{18}$$

where τ is the actuator torque vector and is constrained by independent upper and lower bounds,

$$\tau_l \leq \tau \leq \tau_u \tag{19}$$

where τ_l and τ_u are the lower and upper bound vectors of the actuator torque vector τ.

The manipulator manipulability measure is defined as [12]

$$\omega = \sqrt{\det(J(\Phi)J^T(\Phi))} \tag{20}$$

where $J(\Phi)$ is the manipulator Jacobian matrix.

The total cost function is defined as:

$$f = \varepsilon \int_{t_0}^{t_f} \tau^T \tau \, dt + \lambda \int_{t_0}^{t_f} (\omega_m - \omega) dt \tag{21}$$

where ω_m is the possible maximum manipulability measure, ε and λ are the relative weightings between the two criteria which are under the control of users.

e. Obstacle avoidance

Considerable effort has been directed towards the navigation of wheeled mobile robots in the present of obstacles by employing potential field. Another method for obstacle avoidance is to approximate the obstacle edges by a number of consecutive circles. These two methods are computationally intensive. In this report, a geometric analysis method[18] is applied to detect obstacle collision.

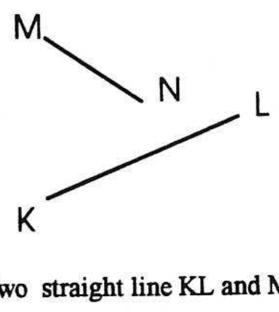


Figure 3 Two straight line KL and MN in a plane

Consider two line segments KL and MN in a plane(as shown in Figure 3). They are specified by their endpoint coordinates (x_M, y_M), (x_N, y_N), (x_K, y_K) and (x_L, y_L). A line can be considered as the interval t=0 to t=1 of an infinite parametric line. Thus, the line KL is represented as

$$x = x_K + t(x_L - x_K) \tag{22}$$

$$y = y_K + t(y_L - y_K) \tag{23}$$

The line MN is represented as

$$x = x_M + s(x_N - x_M) \tag{24}$$

$$y = y_M + s(y_N - y_M) \tag{25}$$

The solution of the set of above 4 simultaneous equations then gives the intersection point as:

$$s = \frac{(x_N - x_M)(y_M - y_K) - (y_N - y_M)(x_M - x_K)}{(x_N - x_M)(y_L - y_K) - (y_N - y_M)(x_L - x_K)} \tag{26}$$

$$t = \frac{(x_L - x_K)(y_M - y_K) - (y_L - y_K)(x_M - x_K)}{(x_N - x_M)(y_L - y_K) - (y_N - y_M)(x_L - x_K)} \tag{27}$$

Only if the values of both s and t are in the range 0 to 1, then the intersection is within both line segments. Otherwise these two lines do not intersect. Therefore, if none of the segments in the obstacles intersects with any of the segments of the platform and no obstacle is enclosed by the platform, the platform does not collide with the obstacles.

f. The optimal trajectory searching problem is stated as follows:

Given the trajectory of the end-effector of the manipulator, search for the trajectories for the platform and the manipulator joints to minimise the total actuator torque and maximise the manipulability measure with nonholonomic constraints and obstacle avoidance.

4. The genetic-based method

To obtain the optimal solution for the above trajectory planning problem, we must solve a number of non-linear equations with nonholonomic constraints. No analytical solution can be obtained and therefore some numerical method must be used. Furthermore, because obstacles are present in the working space of the platform, the solution space for the motion platform usually is discontinuous and exists local minima. Genetic algorithms(GAs) are robust search and optimisation methods based on natural selection. They search for the optimum globally and therefore can avoid being trapped in local minima[19]. Moreover, it is easy to combine new criteria into the cost function. Considering the characteristics of the mobile manipulator system trajectory planning problem, a robust and efficient genetic algorithm is applied.

a. Parameter encoding

In order to search for optimal trajectories by a genetic algorithm, we need to choose a coding scheme to encode the parameters of a mobile manipulator into genetic strings.

Any configuration of a mechanical system can be completely described by an n-vector $\mathbf{q} = (q_1, q_2, \ldots, q_n)^T$ of generalised coordinates, which can be subject to m independent kinematic constraints($m < n$) of the form

$$a_j^T(q)\dot{q} = 0, \; j = 1, 2, \ldots, m. \tag{28}$$

where $a_1, a_2, \ldots, a_m$ are smooth linearly independent vector fields in $\mathbf{R}^n$ and $\dot{q}$ denotes the first time derivative of the generalised coordinates. In this problem the generalised coordinate vector $\mathbf{q}$ contains the platform rotational joint and fictitious joints as well as the manipulator joints as defined in Equation(1).

Given the end-effect trajectory of mobile manipulator system, for a redundant system, the joint angles can not be uniquely determined by the inverse kinematics. If a system has n_r redundant degrees of freedom, then n_r generalised coordinates can be chosen randomly, subject to lower and upper bound constraints and nonholonomic constraints.

The problem of moving the end-effector of a mobile manipulator system from its initial position to the ending position over a time interval along a given trajectory is considered. And the initial and ending positions and velocities of each joint are also given. Suppose a joint trajectory can be represented by a fifth order polynomial as shown as follows:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \quad t_0 \le t \le t_f \tag{29}$$

The angles and velocities at the beginning and end of a joint trajectory are given as follows.

$$\theta_0 = \theta(t_0), \tag{30}$$

$$\theta_f = \theta(t_f), \tag{31}$$

$$\dot{\theta}_0 = \dot{\theta}(t_0) = 0, \tag{32}$$

$$\dot{\theta}_f = \dot{\theta}(t_f) = 0. \tag{33}$$

Two additional constraints are required to determine the above polynomial. Here the accelerations $\ddot{\theta}_0$ and $\ddot{\theta}_f$ of the beginning and end of a joint trajectory are introduced as the constrains required.

The above six constraints specify a fifth order polynomial with its coefficients are:

$$a_0 = \theta_0, \tag{34}$$

$$a_1 = \dot{\theta}_0, \tag{35}$$

$$a_2 = \frac{\ddot{\theta}_0}{2}, \tag{36}$$

$$a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3}, \tag{37}$$

$$a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4}, \quad (38)$$

$$a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}. \quad (39)$$

In the genetic algorithm, the initial and ending accelerations of n_r generalised coordinates p_1, p_2, ... p_{n_r} ($p_i \in \mathbf{q}$, i= 1, 2, ..., n_r) are chosen to be encoded, and a binary genetic string is generated as follows:

$$\{a_{10}, a_{1f}, \ldots, a_{i0}, a_{if}, \ldots, a_{n_r0}, a_{n_rf}\} \quad (40)$$

The polynomial trajectory of each encoded joint is determined by the generated acceleration constraints combined with the angle and velocity constraints at the beginning and end of each trajectory.

After the trajectories of encoded joints are determined, the remaining joint trajectories are calculated in the following way. Discretize the time interval starting at t = 0 and ending at t = t_f into N via-points equally. Beginning from the starting point, the joint value at each via-point of each remaining trajectory is calculated by using the inverse kinematics combined with non-holonomic constraints. At the same time, the joint value at every via-point is checked to see whether it meets the lower and upper bound. In the case when there are obstacles in the working space, the geometric method is applied to check whether in any position the platform trajectory collides with obstacles. If at any via-point, any trajectory violates the kinematic constraints or collides with obstacles, the joint value of this trajectory at this via-point is discarded and a new joint value is regenerated as follows until a valid value is produced.

$$p_{i,j} = p_{i,j-1} + r_{i,j}\,\Delta_i \quad (41)$$

where $p_{i,j}$ is the randomly produced new joint value at point j, and $p_{i,j-1}$ is the joint value at point j-1, Δ_i is a small given positive number and $r_{i,j}$ is a randomly generated integer from a given range.

b. Fitness function

A population of initial strings are generated randomly in the way described above. Then a fitness is assigned to each string according to the fitness function. In this problem, the fitness function is defined as

$$F = f_{max} - f - f_p \quad (42)$$

where f is the cost function defined by Equation (21), f_{max} is a properly selected positive real number not less than the maximum value of f, and f_p is a torque penalty which is $f_{max}/20$ when any trajectory violates the upper or the lower torque bounds, otherwise is set to 0.

c. Reproduction

A reproduction approach is applied to select strings for the next generation. In order to reduce the stochastic error associated with the selection, the stochastic remainder sampling scheme without replacement is applied. Other stratedges applied include fitness scaling and random immigration.

d. Crossover

The crossover operation is applied as follows. Members of the newly reproduced strings are paired at random. For each pair of selected strings, with a probability of p_c, a cross-position is selected at random. The two new strings are created by swapping parts of the strings from the selected position to the last position. The result is that two sets of initial and ending accelerations are formed, that is, two sets of joint trajectories are produced.

e. Mutation

The mutation operator changes individual strings on a bit by bit basis, with a very small probability p_m. Once a mutation is performed in a string, a new set of boundary accelerations are formed. The main purpose of mutation is to bring in new information and to protect against loss of some potentially useful genetic materials caused in reproduction and crossover.

After mutation the fitness values of the new population's strings are evaluated, the process of reproduction and crossover and mutation begins again. This process continues until a predefined number of generations is reached.

Because in the proposed genetic algorithm all the joint trajectories are represented by polynomials, smooth trajectories are produced and a lower cost for each trajectory is expected. Therefore, the genetic algorithm requires a smaller population and converges to a near optimum in fewer generations. And because the parameters of a polynomial path other than the via-points in the path are encoded, the length of a genetic string is determined by the order of the proposed polynomial, which is much shorter than that represented by via-points, and hence speed-up is achieved.

Computation complexity is analysed as follows. In each generation there are n strings and each string undergoes conversion from genotype to phenotype, fitness calculation, selection, crossover and mutation. The computation complexity for each string is O(N*m+m), where N is the via points at a joint trajectory and m is number of joints including the platform joints. Hence, the total computation complexity is O(g*n*(N*m+m)), where g is the number of generations.

5. Simulation results

For case study, a system with a puma-like 3-link manipulator mounted on a mobile platform(as shown in Figure 4) is considered.

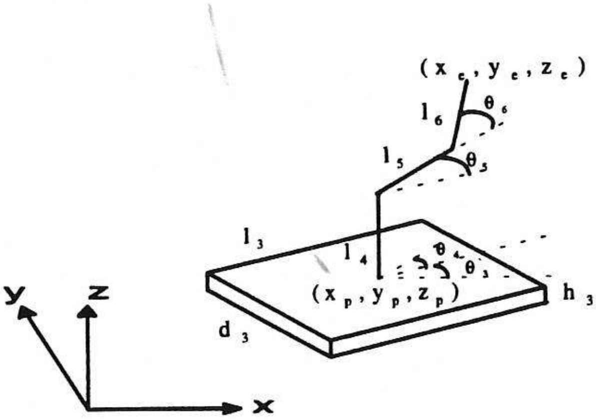


Figure 4 The mobile manipulator for simulation

The forward kinematic equations are

$$x_e = x_p + \cos(\theta_3 + \theta_4)(l_5 \cos\theta_5 + l_6 \cos(\theta_5 + \theta_6)) \tag{43}$$

$$y_e = y_p + \sin(\theta_3 + \theta_4)(l_5 \cos\theta_5 + l_6 \cos(\theta_5 + \theta_6)) \tag{44}$$

$$z_e = z_p + l_4 + l_5 \sin\theta_5 + l_6 \sin(\theta_5 + \theta_6) \tag{45}$$

The platform is subject to the follow non-holonomic constraint:

$$-\dot{x}_p \sin\theta_3 + \dot{y}_p \cos\theta_3 = 0 \tag{46}$$

The parameters of the systems are $l_3 = 1.4$, $d_3 = 1$, $h_3 = 0.2$, $r = .05$, $l_4 = l_5 = l_6 = 1$, z_p=0, and $m_3 = 40$, $m_4 = 10$, $m_5 = 5$, $m_6 = 4$. The path for the end-effector of the manipulator to follow is given in Figure 5, x_e = 2.~2., y_e = 0~2, z_e = 1~2, all along straight line. The trajectory positions in the N points are formed by using cubic spline. Because of the non-holonomic constraint, that is the platform must move in the direction of the axis of symmetry, the platform must turn in order to achieve needed heading angles. The time for the motion is given as T = 5 seconds and the number of via-points N is chosen as 20.

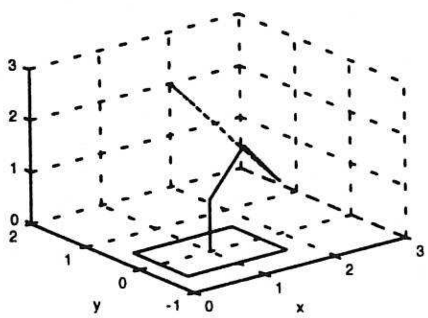


Figure 5 The given path for the end-effector of the onboard manipulator

For the ease of calculating the joint angles and platform positions while meeting the non-holonomic constraint, platform heading angle θ_3 and joint angle θ_5 are encoded. Joint angle θ_6 is calculated directly from Equation(45). θ_4 , x_p and y_p are calculated from Equations (43-44) and non-holonomic constraint Equation (46).

One typical set of genetic parameters used in testing the system was:

population size n = 50;

generation number g = 40;

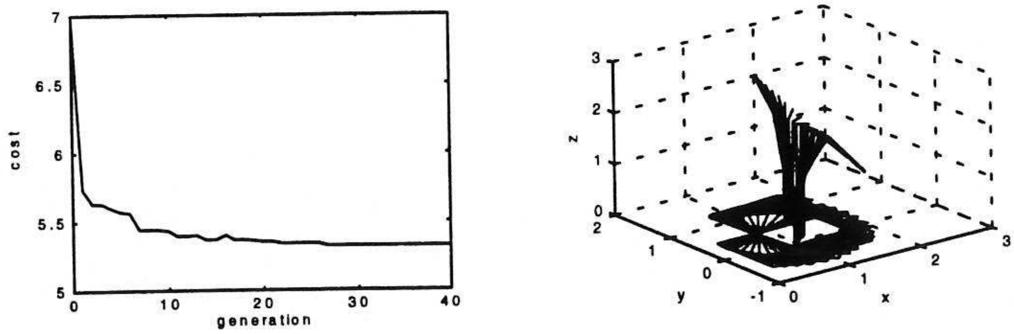
crossover probability $p_c = 0.8$;

mutation probability $p_m = 0.03$;

All the simulations were conducted on a Sun Sparc station and the computation time is less than 100 seconds.

Case 1: Without obstacle avoidance

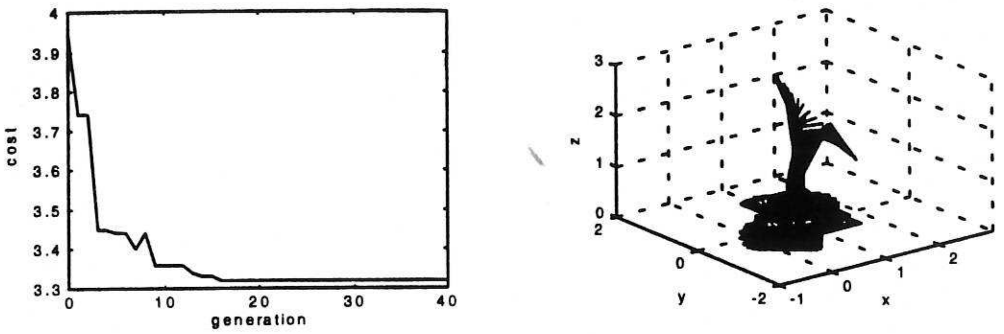
In the first simulation only torque minimisation is considered, that is $\varepsilon = 1$, $\lambda = 0$. The costs versus generation in a GA search is plotted in Figure 6a. and a near optimal motion of the system is obtained as shown in Figure 6b. Its total torque is 5.319, and manipulability measure is 18.925.



(a) (b)

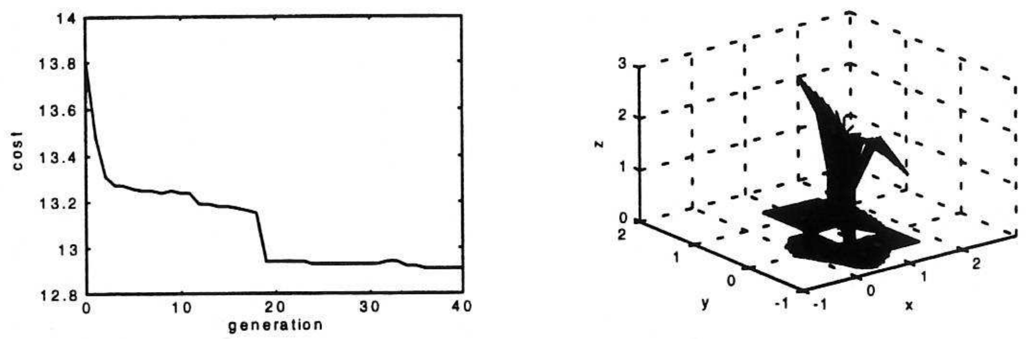
Figure 6 The cost versus generation in a GA search and the motion obtained by the GA(ε=1 ,λ=0)

In Figure 7, only manipulability measure maximisation is considered. The obtained near-optimal motion of the system are shown in Figure 7b. Its total torque is 13.807, and manipulability measure is 26.68, which is much better than in the first simulation but the total torque is much larger. Figure 7b. indicates that the mobile platform moved backward for a short period of time at the very beginning to achieve the maximum manipulability measure and needed heading angle, and then move forward to the pre-specified position. The third simulation with $\varepsilon = 1$ and $\lambda = 1$ showed a compromise between torque minimisation and manipulability measure maximisation as shown in Figure 8 with total torque 8.072 and manipulability measure 25.161.



(a) (b)

Figure 7 The cost versus generation in a GA search and the motion obtained by the GA($\varepsilon = 0$, $\lambda = 1$)

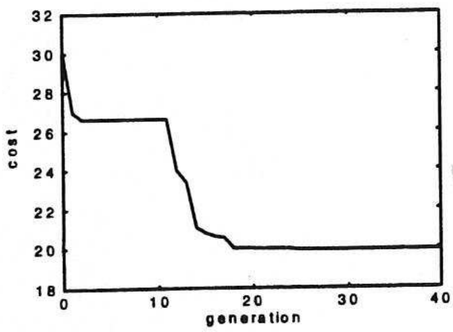


(a) (b)

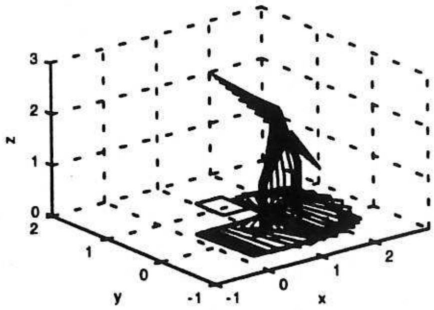
Figure 8 The cost versus generation in a GA search and the motion obtained by the GA($\varepsilon = 1$, $\lambda = 1$)

Case 2: With obstacle avoidance

A rectangle obstacle is placed in the working space of the platform. A near optimal path is obtained by the genetic algorithm. The simulation results are shown in Figure 9 where $\varepsilon = 1$, $\lambda = 1$. The total torque is 13.414 and manipulability measure is 23.494, which are not as good as in the case when no obstacle is considered.



(a)



(b)

Figure 9 The cost versus generation in a GA search and the motion obtained by the GA($\varepsilon = 1, \lambda = 1$)

6. Conclusions

Given the trajectory of the endpoint of the manipulator in a mobile manipulator system with non-holonomic constraints, it is important to plan the motion of the platform as well as the manipulator to meet certain criteria, particularly when obstacle avoidance is involved. The optimal trajectory generation problem for a mobile manipulator system is a non-linear multi-criteria optimisation problem with its solution space being discontinuous and containing local minima. A robust genetic algorithm is applied to solve this problem with torque minimisation, manipulability maximisation and obstacle avoidance. More criteria can be easily combined into the cost function. Computational efficiency in the genetic algorithm is achieved by applying a polynomial method. When applying the geometric analysis method to detect obstacle collision, actually, obstacles can be in any shape, in three dimensional and/or blocking the motion of the onboard manipulator.

References

[1] Latombe, J. C. 1991, Robot Motion Planning, Kluwer Academic Publishers, Amsterdam.

[2] Hwang, Y. K. and Ahuja, N. 1992, "Gross Motion Planning, A survey", ACM Computing Surveys, Vol.24, No.3, 1992.

[3] Bobrow, J. E., Dubowsky, S. and Gibson, J. S. 1985, "Time-optimal control of robotic manipulators along specified paths", International Journal of Robotics Research, Vol.4 No.3, pp.3-17.

[4] Hollerbach, J. M. and Suh K. C. 1987, "Redundancy resolution of manipulators through torque optimisation", IEEE Journal of Robotics and Automation, Vol.RA-3, No.4 ,pp.308-316.

[5] Chen T.-H, Cheng, F.-T. and Sun Y.-Y. 1994, "Torque optimisation scheme for kinematically redundant manipulators", Journal of Robotics Systems, Vol.11, No.4, pp.257-269.

[6] Croft, E. A., Benhabib, B. and Fenton, R. G. 1995, "Near-time optimal robot motion planning for on-line applications", Journal of Robotic Systems, Vol.12, No.8, pp.553-567.

[7] Carriker, W. F. , Khosla, P. K. and Krogh, B. H. 1991. "Path planning for mobile manipulators for multiple task execution", IEEE Transaction on Robotics and Automation, Vol.7 No.3, pp.403-408.

[8] Zhao, M., and Ansari, N. and Hou, E. S. H. 1994, "Mobile manipulator path planning by a genetic algorithm", Journal of Robotics Systems, Vol.11 No.3, pp.143-153.

[9] Seraji, H. 1993, "An on-line approach to coordinated mobility and manipulation", Proc. of the IEEE Conference on Robotics and Automation, pp.28-33.

[10] Neimark J. I. and Fufaev, N. A. 1972 Dynamics of Nonholonomic Systems, Providence, RI, American Mathematical Society.

[11] Saha S. K. and Angeles, J. 1991, "Dynamics of nonholonomic mechanical systems using a natural orthogonal complement," ASME, J. Applied Mechanics, Vol.58, pp.238-243, March.

[12] Yamamoto Y. and Yun X. 1994, "Coordinating locomotion and manipulation of a mobile manipulator," IEEE Transaction on Automatic Control, Vol.39, No.6, June, pp.1326-32.

[13] Jagannathan, S. , Lewis, F. L. and Liu K. 1994, "Motion control and obstacle avoidance of a mobile robot with an onboard manipulator," Journal of Intelligent Manufacturing, No.5, 287-302.

[14] Jagannathan, S., Zhu S. Q. and Lewis, F. L. 1994, "Path planning and control of a mobile base with nonholonomic constraints," Robotica, Vol.12, pp.529-539.

[15] Zefran, M., Kumar, V. and Yun, X. 1994, "Optimal trajectories and force distribution for cooperating arms", IEEE International Conference on Robotics and Automation, pp.874-879.

[16] Craig, J. J. 1989, Introduction to Robotics: Mechanics and Control, Addison-Wesley Publishing Company.

[17] Fu, K. S., Gonzalez, R. C. and Lee, C. S. G. 1987, Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill Book Company, New York.

[18] Bowyer, A. and Woodwak, J. 1983, A Programmer's Geometry, Butterworths.

[19] Goldberg, D. E. 1989, Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison Welsey.

[20] Davidor, Y. 1990, Genetic Algorithms and Robotics: A heuristic strategy for optimisation, World Scientific.

[21] Wang, Q. and Zalzala, A. M. S. 1995, "Transputer control system with a GA motion planner for the PUMA 560 industrial robotic manipulator," DCCS'95: 13th IFAC Workshop on Distributed Computer Control Systems, Toulouse-Blagnac, France.

[22] Chen, M. and Zalzala, A. M. S. 1995, "A genetic-based approach to multi-criteria position and configuration optimisation of redundant mobile manipulator systems", International Workshop on Advanced Robotics and Intelligent Machines, Manchester, UK.

[23] Chen, M. and Zalzala, A. M. S. 1995, "Safety considerations in the optimisation of paths for mobile robots using genetic algorithms", First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications.

[24] Miller, D. P. and Parent, R. E. 1994, "An articulated limb motion planner for optimized movement", The Journal of Visualisation and Computer Animation, Vol.5, pp.85-123.