This is a repository copy of Trajectory Planning of Multiple Coordinating Robots Using Genetic Algorithms.

White Rose Research Online URL for this paper: http://eprints.whiterose.ac.uk/79939/

Monograph:

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
TRAJECTORY PLANNING OF MULTIPLE COORDINATING ROBOTS USING GENETIC ALGORITHMS

S. SUN¹, A.S. MORRIS² and A.M.S. ZALZALA²

¹Department of Aero-Manufacturing Engineering
Northwestern Polytechnical University
Xi’an 710072, P. R. China

²Robotics Research Group
Department of Automatic Control and Systems Engineering
The University of Sheffield
P.O. Box 600, Mappin Street, Sheffield S1 4DU, United Kingdom

Research Report #574
May 1995
TRAJECTORY PLANNING OF MULTIPLE COORDINATING ROBOTS
USING GENETIC ALGORITHMS

Shudong Sun*, A.S. Morris* and A.M.S. Zalzala*

+ Dept. of Aero-Manufacturing Eng., Northwestern Polytechnical Uni.,
  Xi’an 710072, P.R. China
* Dept. of Automatic Control and System Eng., The University of Sheffield,
  Sheffield S1 3DJ, UK

ABSTRACT
The paper focuses on the problem of trajectory planning of multiple coordinating robots. When multiple robots collaborate to manipulate one object, a redundant system is formed. There are a number of trajectories that the system can follow. These can be described in Cartesian coordinate space by an nth order polynomial. This paper presents an optimisation method based on the Genetic Algorithms (GAs) which chooses the parameters of the polynomial, such that the execution time and the drive torques for the robot joints are minimised. With the robot’s dynamic constraints taken into account, the optimised trajectories are realisable. A case study with two planar-moving robots, each having three degrees of freedom, shows that the method is effective.

Key Words: Robotics, Genetic Algorithms, Trajectory Planning, Coordination, Multiple Robots

1. INTRODUCTION

Genetic Algorithms (GAs) are population-based, stochastic, global search methods. Their performance is superior to that of classical techniques[10,11] and they have been used successfully previously in robot path planning[8,22]. However, there has been very little reported work on applying this optimisation method to trajectory planning of multiple coordinating robots, even though this area has long been recognised as a most interesting research field not only for multiple coordinating robots, but also for multi-fingered and multi-legged systems[13,17,19,24].

Ahmad and Luo[1] considered the coordination problem of a welding system comprising a redundant robot with a positioning table. They proposed a hierarchical model to divide the coordination problem into small subtasks, so that the complexity of the problem is manageable. Jouaneh et. al.[14] also dealt with the coordination problem of a robot with a x-y positioning table. The coordinating path was obtained by moving the two devices simultaneously. Even a sharp cornered path can be followed by these devices. In[15], Jouaneh et. al. used a dynamic programming method to achieve near minimum time and energy trajectories for the two coordinating devices.
The dual-arm coordination problem has been studied extensively for its potential usage in industry. Tabarah et al. [23] used polynomial functions to plan dual-arm system trajectories, the parameters of the polynomial being decided by an optimisation procedure based on the flexible polyhedron search method. Dynamic scaling was used to avoid the recalculation of the robot’s dynamics and ensure that the torque limits were not violated. Ro et al. [21] dealt with the assembly problem using a dual-arm system. The authors used velocity and force ellipsoids to form an objective function. A null-space search method was then used to optimise the posture of the coordinating robots. While one robot holds the male part, the other one grasps the female part, and thus the typical male-female assembly can be finished without the aid of jigs or fixtures.

For the planning and control of multiple coordinating robots, Xi et al. [27] proposed an event-based method. The method is event-based because the authors used a reference function, which is the distance the object moved along a given path $s$, rather than the time $t$ as a reference. The variable $s$ is time independent. Yao and Tomizuke [28] proposed an adaptive control method for multiple coordinating robots. Cell to cell mapping were used by Wang and Pu [25] to plan time-optimal trajectories for multiple coordinating robots, where each cell corresponds to one of the states in the moving path.

The paper presented here focuses on the trajectory planning of multiple coordinate robots. First, in a world reference frame, an $n$th order polynomial of time $t$ is supposed to be the system’s trajectory, with its parameters undecided. Then, an optimisation problem is formed with the execution time and the drive torques of the robot joints as the objective function, and with the dynamic equations of the system as constraints. Next, a procedure based on GA is proposed to solve the optimisation problem. Finally, an example of two coordinating arms handling a rectangular object and moving in a plane is given to demonstrate the effectiveness of the method. Simulation results show that the higher the order of the polynomial is, the shorter the execution time will be. This is similar to the result obtained in [23]. However, the computational cost will increase as the order of the polynomial become larger. The highest order of polynomial used in the simulation is four: this is because, with the fourth order of polynomial, the jerk of the system’s trajectory will be continuous and can be controlled. Jerk has been shown to be important in the trajectory planning and control of a robot [16].

The organisation of the paper is as follows. In section two, the problem of the trajectory planning of multiple coordinate robots is formulated. The optimisation method based on GA is proposed in section three. Then, a case study of trajectory planning for two coordinating robots is presented in section four and finally conclusions are drawn in section five.

### 2. PROBLEM FORMULATION

#### 2.1 Dynamic Model of the Coordinating System

Consider $n$ robots manipulating one object, as shown in Fig. 1, each robot having $n_i (n_i \leq 6)$ degrees of freedom. One of the robots (master) holds the object firmly, while the other robots (slaves) can move their position along the border of the object. All the contacts between slave robots and the object are supposed to be point contacts. Let $F_o$ be the world reference frame. $F_e$ is the object-fixed frame, with its origin at the mass center of the object. $F_o$ is the end-effector frame of the $i$th robot, with its origin located at the contact point.

The dynamic equations for each robot in $F_o$ can be expressed as follows:
\[ D_i(q_i)\ddot{q}_i + c_i(q_i, \dot{q}_i) + J_i^T F_i = \tau_i \quad (i=1,2,...,n) \] 

where \( q_i, \dot{q}_i, \ddot{q}_i \in R^n \) are the vectors of the \( i \)th robot’s joint position, velocity and acceleration respectively.

\( D_i(q_i) \in R^{n \times n} \) is the \( i \)th robot inertia matrix.

\( c_i(q_i, \dot{q}_i) \in R^n \) is the vector including coriolis, centripetal and gravity forces.

\( J_i \in R^{6 \times n} \) is the Jacobian matrix of \( i \)th robot.

\( F_i \in R^6 \) is the force vector exerted by the \( i \)th robot on the object.

\( \tau_i \in R^n \) is the vector of \( i \)th robot’s joint torques.

The dynamic equations of the object in \( F \) are given by:

\[ M_E(p) \ddot{p} + c_E(p, \dot{p}) = F \] 

where \( p, \dot{p}, \ddot{p} \in R^6 \) are the vectors of the object posture (position and orientation), velocity and acceleration respectively.

\( M_E \in R^{6 \times 6} \) is the object inertia matrix.

\( c_E(p, \dot{p}) \in R^6 \) is the vector of centripetal and gravity forces of the object.

\[ F = \sum_{i=1}^{n} J_E^T T_{ei} F_i \] 

**Figure 1 Multiple Robots Coordinating to Manipulate one Object**

\( F \) is the vector of the resultant force exerted onto the object by the end-effectors of the robot, and is given by:

\[ F = \sum_{i=1}^{n} J_E^T T_{ei} F_i \]
In equation (3), $J_e \in R^{6 \times 6}$ is the Jacobian matrix of the object. $T_{fi} \in R^{6 \times 6}$ is the force-moment transformation matrix from $F_e$ to $F_i$ [7].

2.2 Kinematic Relationship of the Coordinate System

From Fig.1, the homogeneous transformation between $F_e$, $F_x$ and $F_o$ are as follows[20]:

$$T^0_e = T^0_i T^i_e \quad (i=1,2,\cdots,n)$$  \hspace{1cm} (4)

where $T^0_e$ is the transformation matrix from $F_0$ to $F_e$, given by

$$T^0_e = \begin{bmatrix} R_{3 \times 3} & p_x \\ p_y & p_z \\ 0 & 1 \end{bmatrix}$$ \hspace{1cm} (5)

$$R_{3 \times 3} = Rot(z, p_{0z}) \cdot Rot(y, p_{0y}) \cdot Rot(x, p_{0x})$$ \hspace{1cm} (6)

$p_x, p_y, p_z$ are the position of the origin of $F_e$ in $F_0$ and $p_{0x}, p_{0y}, p_{0z}$ are the angles of axes between $F_e$ and $F_0$.

$T^0_i$ is the transformation matrix from $F_0$ to $F_i$.

$T^i_e$ is the transformation matrix from $F_e$ to $F_i$. It is dependent on the position of the $i$th robot’s end-effector and has the following form:

$$T^i_e = \begin{bmatrix} (R_{3 \times 3})_i & r^i_e \\ 0 & 1 \end{bmatrix}$$ \hspace{1cm} (7)

$(R_{3 \times 3})_i$ is the rotational matrix from $F_e$ to $F_i$, which has the same form as Eq.(6). $r^i_e$ is the vector of the origin of $F_e$ in $F_i$.

If the coordinates of the robot end-effector are selected properly, then

$$(R_{3 \times 3})_i = \text{constant} \quad (i=1,2,\cdots,n)$$ \hspace{1cm} (8)

The motion constraints for the slave robots, in $F_e$, are given by:

$$\varphi(r^i_e) = 0$$ \hspace{1cm} (9)

Suppose the start and end points of the object are known a priori, and the object is freely moving in the space enclosing start and end points.

Let the trajectory of the object be expressed as an $n$th order of polynomial of $t$, that is:

$$p = CT$$ \hspace{1cm} (10)

where $p$ has the same meaning as in Eq.(2), $C \in R^{6 \times n}$ is the parameter matrix to be determined, and $T = (1, t, t^2, \cdots, t^{n-1})^T$.

If the object trajectory is decided by Eq. (10), then we can decide each robot’s end-effector trajectory by Eqs. (4), and (9). Since $n_i \leq 6$, the motion of each robot is settled.
2.3 Objective and Constraint Functions

Because the coordinating system is an redundant system, there are numerous C’s to satisfy Eq.(10) and the related constraints. Here, the matrix C is selected in such a manner that the following objective function is minimized:

\[ f = wT_f + \sum_{i=1}^{n} W_i \tau_i \]  \hspace{1cm} (11)

where \(T_f\) is the time for the object moving from the start point to the end point, \(\tau_i\) has the same meaning as in Eq.(1), \(w\) and \(W_i\) are weight coefficients.

The trajectory planning problem for the coordinating multiple robots can now be expressed as the following optimisation problem:

\[ \min \quad f = wT_f + \sum_{i=1}^{n} W_i \tau_i \]  \hspace{1cm} (12)

s.t.  \hspace{1cm} \text{Eqs. (1), (2), (4) and}

\[ q_j^\text{min} \leq q_j \leq q_j^\text{max} \]
\[ \dot{q}_j^\text{min} \leq \dot{q}_j \leq \dot{q}_j^\text{max} \]
\[ \ddot{q}_j^\text{min} \leq \ddot{q}_j \leq \ddot{q}_j^\text{max} \]
\[ \tau_j^\text{min} \leq \tau_j \leq \tau_j^\text{max} \]  \hspace{1cm} (j = 1,2,\ldots,n; i = 1,2,\ldots,n)  \hspace{1cm} (13)

3. OPTIMISATION METHOD

As stated above, if the optimal trajectory of the object is decided, then each robot’s trajectory can be easily decided from Eq.(4). Since the selected objective function given in Eq.(12) includes each robot’s drive torques, the trajectory obtained from kinematics constraints should be the optimal trajectory for each robots by the principle of energy consumption.

The GA procedure proposed to optimise the object trajectory and each robot’s trajectory is shown in Fig.2.

In Fig.2, gen stands for how many generations have been evaluated, C(gen) and C(gen-1) are the populations of present and last generation.

In the procedure, the coding method for the parameter matrix C is binary coding. Because the movement of the object is limited by its start and end points, and the variables of robot joints are also bounded, binary coding has been shown to be the most effective coding method in this kind of parameter optimisation[18].

The fitness function of the optimisation is selected as being the same as the objective function Eq.(12) with linear ranking[5].

As shown in Fig.2, a certain number of chromosomes are selected dead for each generation, with the same number of new chromosomes being added to the population to improve the efficiency of GA[18]. The number of dead is controlled by a parameter called the generation gap, GGAP (0<GGAP<1). The selection of the dead is based on a stochastic universal sampling method[2].
procedure \textbf{OMCRT}

\textbf{BEGIN}

\texttt{begin} \hspace{1cm} \% Start GA

\texttt{gen} \leftarrow 0;

\texttt{initialize} \texttt{C(gen);}

\texttt{evaluate} \texttt{C(gen);}

\texttt{while ( gen \leq MAXGEN ) do}

\texttt{begin}

\texttt{gen} \leftarrow \texttt{gen} + 1;

\texttt{select parents from} \texttt{C(gen-1);}

\texttt{select dead from} \texttt{C(gen-1);}

\texttt{form} \texttt{C(gen): reproduce the parents;}

\texttt{evaluate} \texttt{C(gen);}

\texttt{end}

\texttt{end} \hspace{1cm} \% End GA

\texttt{decide each robot's trajectory using Eq.(4);}

\textbf{END}

\textbf{Figure 2 Procedure to optimise multiple coordinating robots' trajectory}

Single-point crossover with probability $p_c$ is used to form the new generation of $C(\text{gen})$. Also, mutation with a low probability $p_m$ is used as a background operator to provide a guarantee that good genetic material will not be lost through the action of selection or crossover[11]. The termination condition for the GA procedure is the maximum number of generations ($\text{MAXGEN}$).

4. CASE STUDY

4.1 Simulation Problem and Parameters

In the case study, the same problem as in [23] is used, i.e. two identical robots moving in a vertical plane and collaborating to manipulate one rectangular object. The first robot holds the object firmly at the middle point of one edge, while the second robot moves its position along the opposite edge of the object, as shown in Fig.3. The motion of the second robot is supposed to be at constant speed, with the start point at one end corner of the edge, and the end point at the other end corner. The motion time is the same as the time taken by the object to move from its given start point to the end point.

The dynamic equations of the system can be obtained in the same form as in Eqs.(1) and (2)[4]. The motion constraints for the second robot are given as:

\begin{equation}
\begin{aligned}
x_{s2} &= 0.15 \\
y_{s2} &= -0.2 + \frac{0.4}{T_f} t
\end{aligned}
\end{equation}  \text{ (14)}
From Eqs. (5) and (7), the transformation matrix for the system is:

\[
T_E^0 = \begin{pmatrix}
\cos\theta & -\sin\theta & 0 & x_E \\
\sin\theta & \cos\theta & 0 & y_E \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]  \hspace{1cm} (15)

\(x_E\) and \(y_E\) stand for the coordinates of the object mass center in \(F_0\), \(\theta\) is the angle of the \(x\)-axis between \(F_E\) and \(F_0\).

\[
T_E^1 = \begin{pmatrix}
1 & 0 & 0 & -0.15 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]  \hspace{1cm} (16)

\[
T_E^2 = \begin{pmatrix}
-1 & 0 & 0 & x_{E2} \\
0 & -1 & 0 & y_{E2} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]  \hspace{1cm} (17)

From Eq. (4), the transformation matrix \(T_i^0\) can be obtained as

\[
T_i^0 = T_E^0 (T_i^E)^{-1} \hspace{1cm} (i=1,2)
\]  \hspace{1cm} (18)

That is, the trajectory for the first coordinating robot's end-effector is
\[ x_1 = 0.15 \cos \theta + x_E \]
\[ y_1 = 0.15 \sin \theta + y_E \]
\[ \theta_1 = \theta \] (19)

and for the second robot, it is:
\[ x_2 = 0.15 \cos \theta + 0.2 \sin \theta - \frac{0.4}{T_f} t \sin \theta + x_E \]
\[ y_2 = 0.15 \sin \theta - 0.2 \cos \theta + \frac{0.4}{T_f} t \cos \theta + y_E \]
\[ \theta_2 = \theta + \pi \] (20)

Table 1  Physical Parameters of the Coordinate System

<table>
<thead>
<tr>
<th></th>
<th>Dimension (m)</th>
<th>Mass (Kg)</th>
<th>Inertia (Kgm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>0.3 × 0.4</td>
<td>2</td>
<td>0.04</td>
</tr>
<tr>
<td>robot link 1</td>
<td>1.0</td>
<td>12</td>
<td>0.8</td>
</tr>
<tr>
<td>robot link 2</td>
<td>0.8</td>
<td>6</td>
<td>0.2</td>
</tr>
<tr>
<td>robot link 3</td>
<td>0.4</td>
<td>3</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 2  Upper and Lower Bounds of the Robot Joints

<table>
<thead>
<tr>
<th>joint</th>
<th>item</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
</tr>
<tr>
<td>upper</td>
<td>2.8</td>
<td>3.0</td>
<td>2.5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>lower</td>
<td>0.3</td>
<td>-3.0</td>
<td>-2.5</td>
<td>-10</td>
<td>-10</td>
<td>-10</td>
<td></td>
</tr>
<tr>
<td>joint</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>item</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
<td>q</td>
</tr>
<tr>
<td>upper</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>200</td>
<td>150</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>lower</td>
<td>-50</td>
<td>-50</td>
<td>-50</td>
<td>-200</td>
<td>-150</td>
<td>-100</td>
<td></td>
</tr>
</tbody>
</table>

Table 3  GA Parameters

| population size | 40 |
| GGAP            | 0.9 |
| probability of crossover \( p_c \) | 0.7 |
| probability of mutation \( p_m \) | 0.01 |
| length of binary coding | 20 bit |
| MAXGEN          | 150 |
The parameters used in the simulation are as shown in Table 1, 2 and 3.

The weights of the objective function are selected as:
\[ w = 1 \quad \text{and} \quad W_1 = W_2 = (1 \ 1 \ 1) \]

In the simulation, a variety of GA parameters have been used to look for the optimum trajectory. The parameters shown in Table 3 correspond to the simulation results shown in section 4.2.

4.2 Simulation Results

The start and end points for the object motion used in the simulation study are as follows:
\[ x_0 = 0.15 \]
\[ y_0 = 0.2 \]
\[ \theta_0 = 0.3 \]

and
\[ x_f = 1.0 \]
\[ y_f = 0.9 \]
\[ \theta_f = 0.8 \]

Using a GA Toolbox[6], which is based on the MATLAB[30], we studied the trajectory optimisation problem using second, third, and fourth order polynomials.

4.2.1 Second Order Polynomial

The optimal trajectory of the object is:
\[ x_e = 0.1500 + 0.7523t - 0.1539t^2 \]
\[ y_e = 0.2000 + 0.2668t + 0.0723t^2 \]
\[ \theta = 0.3000 + 3.7104t - 1.9343t^2 \]

The execution time is \( T_e = 1.7724s \).

The corresponding optimal trajectory for the first robot end-effector can be easily found from Eq.(19) as
\[ x_1 = 0.15 \cos \theta + 0.1500 + 0.7523t - 0.1539t^2 \]
\[ y_1 = 0.15 \sin \theta + 0.2000 + 0.2668t + 0.0723t^2 \]
\[ \theta_1 = 0.3000 + 3.7104t - 1.9343t^2 \]

From Eq.(20), we can get the optimal trajectory for the second robot end-effector as
\[ x_2 = 0.15 \cos \theta + 0.2 \sin \theta - 0.2257t \sin \theta + 0.1500 + 0.7523t - 0.1539t^2 \]
\[ y_2 = 0.15 \sin \theta - 0.2 \cos \theta + 0.2257t \cos \theta + 0.2000 + 0.2668t + 0.0723t^2 \]
\[ \theta_2 = \pi + 0.3000 + 3.7104t - 1.9343t^2 \]
Fig. 4(a) shows the simulation result for the best objective function, (b) shows the object path and orientation, (c) shows joint torques of the master robot, and (d) shows joint torques of the slave robot.

4.2.2 Third Order Polynomial

The optimal trajectory of the object is:
\[
x_x = 0.1500 + 0.6975t + 0.0862t^2 - 0.1184t^3 \\
y_x = 0.2000 + 0.0856t + 0.3584t^2 - 0.0832t^3 \\
\theta = 0.3000 - 5.3443t + 1.1231t^2 + 1.5643^3
\]

The execution time is \( T_f = 1.5770s \).

The optimal trajectory for the robot end-effectors is:
\[
x_1 = 0.15 \cos \theta + 0.1500 + 0.6975t + 0.0862t^2 - 0.1184t^3 \\
y_1 = 0.15 \sin \theta + 0.2000 + 0.0856t + 0.3584t^2 - 0.0832t^3 \\
\theta_1 = 0.3000 - 5.3443t + 1.1231t^2 + 1.5643^3
\]

and
\[
x_2 = 0.15 \cos \theta + 0.15 \sin \theta - 0.2536t \sin \theta + 0.1500 + 0.6975t + 0.0862t^2 - 0.1184t^3 \\
y_2 = 0.15 \sin \theta - 0.2 \cos \theta + 0.2536t \cos \theta + 0.2000 + 0.0856t + 0.3584t^2 - 0.0832t^3 \\
\theta_2 = \pi + 0.3000 - 5.3443t + 1.1231t^2 + 1.5643^3
\]

Fig. 5(a) shows the best objective function, (b) shows the object path and orientation, (c) shows joint torques of the master robot, and (d) shows joint torques of the slave robot.

4.2.3 Fourth Order Polynomial

The optimal trajectory of the object is:
\[
x_x = 0.1500 + 0.7655t + 0.0415t^2 - 0.2107t^3 + 0.1682t^4 \\
y_x = 0.2000 + 1.2111t + 0.0895t^2 - 0.5659t^3 + 0.4020t^4 \\
\theta = 0.3000 + 4.3366t + 0.0485t^2 - 2.8004t^3 + 0.0562t^4
\]

The execution time is \( T_f = 1.2070s \).

The optimal trajectory for the first robot end-effector is:
\[
x_1 = 0.15 \cos \theta + 0.1500 + 0.7655t - 0.0415t^2 - 0.2107t^3 + 0.1682t^4 \\
y_1 = 0.15 \sin \theta + 0.2000 + 1.2111t + 0.0895t^2 - 0.5659t^3 + 0.4020t^4 \\
\theta_1 = 0.3000 + 4.3366t + 0.0485t^2 - 2.8004t^3 + 0.0562t^4
\]

and for the second is:
\[ x_2 = 0.15 \cos \theta + 0.2 \sin \theta - 0.3314r \sin \theta \\
+0.1500 + 0.7655r - 0.0415r^2 - 0.2107r^3 + 0.1682r^4 \\
y_2 = 0.15 \sin \theta - 0.2 \cos \theta + 0.3314r \sin \theta \\
+0.2000 + 1.2111r + 0.0895r^2 - 0.5659r^3 + 0.4020r^4 \\
\theta_2 = \pi + 0.3000 + 4.3366r + 0.0485r^2 - 2.8004r^3 + 0.0562r^4 \]

Figure 4  Simulation Results of Second Order Polynomial
Figure 5  Simulation Results of Third Order Polynomial
Fig. 6 (a) shows the best objective function, (b) shows the object path and orientation, (c) shows joint torques of the master robot, and (d) shows joint torques of the slave robot.

Figure 6  Simulation Results of Fourth Order Polynomial
From Figs.(4),(5) and (6), it can be seen that the execution time decreases from 1.7724 seconds to 1.2070 seconds as the order of the polynomials increases from two to four. All the torques are within their limits. However, there are some peaks which reach the limits of the torque. This kind of condition may be caused by the optimization of the execution time[12]. The trajectory of the object is nearly a straight line, as shown in Fig. 4(b),5(b) and 6(b). The graphs of best objective function show that the GAs work very well, there is no incidence of premature convergence.

5. CONCLUSION

The trajectory planning problem of multiple coordinating robots has been studied in this paper. Based on GAs, an optimisation procedure is proposed to decide the parameters of the polynomial trajectory of the coordinating system. Since the optimised trajectory is independent of robot geometry, it can be used in the control of any kind of multiple coordinating robots. In addition, since the calculation of the polynomial is simple, the optimised trajectory can be used in on-line control of multiple coordinating robots. During the optimisation, the torque limits of the robot joints are taken as constraints, so that the trajectory obtained is both optimal and realistic. A simulation study with two coordinating robots shows that the optimised trajectory can be realised without violating any constraints of the robot actuators.

Acknowledgements: The first author thanks the Department of Automatic Control and System Engineering, The University of Sheffield for the free use of computer facilities and the Chinese Government for the living expenses. The authors thank Prof. P.J. Fleming and Dr. A. Chipperfield for the free use of GA Toolbox.

REFERENCES


