CHAPTER 5

B-SPLINE BASED TRAJECTORY PLANNING OPTIMIZATION

5.1 STAGE 8: MULTI-OBJECTIVE (3 OBJECTIVES) TRAJECTORY PLANNING IN THE PRESENCE OF FIXED OBSTACLES

5.1.1 Introduction

The same problem in stage 5 (STANFORD robot trajectory planning) is considered with the following changes: 1). Cubic B-spline curve is used to represent the trajectory and 2). Only the initial and final points are known to construct the trajectory. The optimal robot manipulator trajectory is the one that optimizes the objective functions without any obstacle collision in the workspace. The problem formulation, kinematic and dynamic models and checking of obstacle avoidance are same as in stage 5.

5.1.2 Formulation of Trajectories

Only the initial and final points to construct the joint trajectories are given (Figure 5.1). A uniform cubic B-spline is used to define the trajectory. A polynomial third degree function ‘f’ with the following properties defines the B-spline trajectory.

1. The knots are uniformly spaced, i.e., δ = t_{j+1} - t_j (constant)
2. For each interval \([t_j, t_{j+1}]\) the function \(f\) is equal to the cubic polynomial.

\[
f_j(t) = q_j(t) = \gamma^1_{j,3}b_3(t) + \gamma^1_{j,2}b_2(t) + \gamma^1_{j,1}b_1(t) + \gamma^1_{j,0}b_0(t) \quad (5.1)
\]

\[
b_0(t, t_j) = \left[\mu^3_j(t)\right]/6; \quad t \in I_j = [t_j, t_{j+1}]; \quad \mu_j(t) = (t - t_j)/\delta \quad (5.2)
\]

\[
b_1(t, t_{j+1}) = \left[1 + 3\mu_{j,1}(t) + 3\mu^2_{j,1}(t) - 3\mu^3_{j,1}(t)\right]/6;
\]

\[
t \in I_{j+1} = [t_{j+1}, t_{j+1} + \delta]; \quad \mu_{j+1}(t) = (t - t_{j+1})/\delta \quad (5.3)
\]

\[
b_2(t, t_{j+2}) = \left[4 - 6\mu^2_{j,2}(t) + 3\mu^3_{j,2}(t)\right]/6; \quad t \in I_{j+2} = [t_{j+2}, t_{j+2} + \delta];
\]

\[
\mu_{j+2}(t) = (t - t_{j+2})/\delta \quad (5.4)
\]

\[
b_3(t, t_{j+3}) = \left[1 - 3\mu_{j,3}(t) + 3\mu^2_{j,3}(t) - \mu^3_{j,3}(t)\right]/6;
\]

\[
t \in I_{j+3} = [t_{j+3}, t_{j+3} + \delta]; \quad \mu_{j+3}(t) = (t - t_{j+3})/\delta \quad (5.5)
\]

where, \(\gamma_j^1\) are the coefficients of the B-spline approximation for \(q_j(t)\) in the interval \(I_j\).

Figure 5.1 Trajectory of gripper
The derivatives of $q_i(t)$ with respect to $t$ are
\[
\frac{d^k q_i(t)}{dt^k} = \gamma_i^1 b_1^{(k)}(t) + \gamma_i^2 b_2^{(k)}(t) + \gamma_i^3 b_3^{(k)}(t) + \gamma_i^4 b_4^{(k)}(t)
\]
(5.6)
where,
\[
b_j^{(k)} = \frac{d^k (b_j)}{dt^k}
\]

For the problems with free terminal time a new time variable $\tau = t/T$ is introduced. This way the interval $[0,T]$ is replaced by non-dimensional interval $[0,1]$. In this case, the trajectories $q_i(\tau)$ are obtained with knots $0 = \tau_0 < \tau_1 < \ldots < \tau_m = T$.

\[
q_i(\tau) = \sum_{j=0}^{m+1} q_i^j(\tau), i = 1, \ldots, n
\]
(5.7)
where, $q_i^j(\tau)$ is given by Equation (5.1).

In the optimization of the trajectories the decision variables are the B-spline coefficients $\gamma_i^j$. The dimensions of the design variable vector is $n(m+2)$. Here 10 knot points are considered. So, total number of variables is 72.

5.1.3 Numerical Example

The optimal trajectory planning for STANFORD robot with six DOF is dealt. In this application the aim is to obtain an optimal trajectory ($\eta_1$) of the end-effectors considering fixed obstacles. The initial and final trajectory points of the end-effectors: $q_i = [0.30\text{rd}, 0.50\text{rd}, -0.20\text{rd}, -0.05\text{rd}, 0.05\text{rd}, 0.10\text{rd}]$ and $q_{f_i} = [0.51\text{rd}, -0.42\text{rd}, 0.58\text{rd}, 0.638\text{rd}, 0.835\text{rd}, 0.20\text{rd}]$. 
The values of the parameters of NSGA-II and MODE techniques are the same as in stage 5. \( N_1 = 4 \) and \( N_2 = 270 \) and \( N_3 = 1 \) are normalizing parameters of objective functions (Average value of individual objective functions).

5.1.4 Results and Discussion

For the sample case \((w_1 = 0.3, w_2 = 0.3\) and \(w_3 = 0.4\)), the optimal displacement \((Q \text{ - rad or m})\), velocity \((V \text{ - rad/s or m/s})\) and acceleration \((W \text{ - rad/s}^2 \text{ or m/s}^2)\) of all the robot links for the best solution selected by normalised weighted objective functions method and average fitness factor method combinedly from the Pareto optimal fronts obtained from NSGA-II and MODE are shown by Figures 5.2 and 5.3. From Figures 5.2 and 5.3, it is noted that the robot joints displacement, velocity, acceleration are within their limiting values. So the obtained trajectories are smoother. Optimum value of the objective functions of the best solution for the sample case \((w_1 = 0.3, w_2 = 0.3\) and \(w_3 = 0.4\)) from SUMT (Saramago et al 2001), NSGA-II and MODE are given in Table 5.1. It is noted that MODE gives better result (minimum \(f_c\)) than NSGA-II and SUMT (Saramago et al 2001). All algorithms give safer optimal trajectories (there is no collision between robot and obstacles, since \(z_3\) is zero). The Pareto optimal fronts obtained from NSGA-II and MODE are given in Figure 5.4. It is observed that NSGA-II gives more number of non-dominated solutions than MODE. Also according to distance metric, the Pareto optimal front from NSGA-II is better than that of the one from MODE. So NSGA-II technique is the best one for this multi-criterion optimization problem, if the user wants more number of solutions for his choice. The results from NSGA-II and MODE are listed in Tables 5.2, 5.3 and 5.4. It is observed that NSGA-II gives maximum ratio of non-dominated solutions (RNI) and minimum solution spread measure (SSM) than those of MODE. MODE gives minimum combined objective function \((f_c)\) and maximum average fitness factor value \((F_{avg})\) than those of NSGA-II and SUMT (Saramago et al 2001).
Also MODE technique is much better than NSGA-II in terms of minimum optimiser overhead (OO) and minimum algorithm effort. It is observed that MODE technique converges quickly than NSGA-II. Also the computational time to find Pareto optimal front in MODE is $1/3^{rd}$ of that of in NSGA-II. So MODE is faster than NSGA-II.

![Graphs of optimal motions of the robot joints obtained from NSGA-II](image)

Figure 5.2 Optimal motions of the robot joints obtained from NSGA-II
Figure 5.3  Optimal motions of the robot joints obtained from MODE
Figure 5.4 Pareto optimal fronts obtained from NSGA-II and MODE

Table 5.1 Results obtained from SUMT (Saramago et al 2001), NSGA-II and MODE algorithms

<table>
<thead>
<tr>
<th>Technique</th>
<th>Time ($z_1$) (sec)</th>
<th>Energy ($z_2$) (Nm)</th>
<th>Penalty for obstacle avoidance ($z_3$)</th>
<th>Combined objective function ($f_c$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMT</td>
<td>10.5</td>
<td>29200</td>
<td>0</td>
<td>33.23194</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>5.2</td>
<td>276.42</td>
<td>0</td>
<td>0.697133</td>
</tr>
<tr>
<td>MODE</td>
<td>3.64</td>
<td>284.3</td>
<td>0</td>
<td>0.588889</td>
</tr>
</tbody>
</table>

Table 5.2 Average fitness factor obtained from various algorithms

<table>
<thead>
<tr>
<th>$z_1$ max</th>
<th>$z_1$ min</th>
<th>$z_2$ max</th>
<th>$z_2$ min</th>
<th>$F_{avg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SUMT</td>
<td>NSGA-II</td>
<td>MODE</td>
</tr>
<tr>
<td>12.84</td>
<td>2.52</td>
<td>0.075581</td>
<td>0.580081</td>
<td>0.630378</td>
</tr>
</tbody>
</table>

Table 5.3 Algorithm effort obtained from NSGA-II and MODE algorithms

<table>
<thead>
<tr>
<th>Technique</th>
<th>Simulation time $T_{run}$ (sec)</th>
<th>$N_{eval}$</th>
<th>Algorithm effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>2</td>
<td>92</td>
<td>0.02174</td>
</tr>
<tr>
<td>MODE</td>
<td>2</td>
<td>138</td>
<td>0.01449</td>
</tr>
</tbody>
</table>
Table 5.4  SSM, RNI and OO obtained from NSGA-II and MODE algorithms

<table>
<thead>
<tr>
<th>Technique</th>
<th>SSM</th>
<th>RNI</th>
<th>OO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>0.86238</td>
<td>0.16</td>
<td>0.1354</td>
</tr>
<tr>
<td>MODE</td>
<td>0.95746</td>
<td>0.15</td>
<td>0.0257</td>
</tr>
</tbody>
</table>

5.1.5 Limitations

This stage considers only three objective functions i.e. travelling time, mechanical energy and penalty for obstacle avoidance. But if we want a more realistic optimal trajectory plan, we have to consider all the important objective functions of trajectory planning. Also, this stage considers only stationary obstacles in the robot’s workspace. But in real world situation all types of obstacles may be in a robot’s workspace. So in the next stage, all types of obstacles (stationary and moving (rotating and oscillating)) are considered in a robot’s workspace.

Saramago and Junior (2000) proposed a method based on SUMT to obtain an optimal motion planner for the STANFORD manipulator in the presence of fixed, moving, and oscillating obstacles. Their objective function was a weighted balance of transfer time, mechanical energy of the actuators, and penalty for collision-free motion. They have taken into account the physical constraints, such as joint displacements, velocities, accelerations, jerks and torques, and collision avoidance.

The following are the major limitations of the their works:

- They have only used conventional optimization technique i.e. SUMT.
They have not considered all the important objective functions (Minimization of travelling time, energy, joint jerks, joint accelerations, penalty for obstacle avoidance and maximization of manipulability measure) in a combined manner.

To overcome the above drawbacks, in the next stage, two evolutionary optimization techniques namely NSGA-II and DE are proposed to do minimum cost trajectory planning for STANFORD and PUMA 560 robots.

5.2 STAGE 9: COMBINED OBJECTIVE (6 OBJECTIVES) TRAJECTORY PLANNING IN THE PRESENCE OF FIXED, MOVING AND OSCILLATING OBSTACLES

The aim is to obtain an optimal trajectory planning for intelligent robot manipulators (PUMA 560 and STANFORD robots) in the presence of fixed, moving and oscillating obstacles considering the actuator constraints, joint limits and obstacle avoidance constraint and all objective functions (Minimization of travelling time, energy, joint jerks, joint accelerations, penalty for obstacle avoidance and maximization of manipulability measure) as well. The same problem in stage 7 with the following change is considered: cubic B-spline curve is used to represent the trajectory. The trajectory representation is the same as in stage 8. 5 knot points are considered for cubic B-spline curve. So, total number of variables is 42.

5.2.1 Results and Discussion

5.2.1.1 PUMA 560 Robot

The optimal displacement (Q - rad), velocity (V - rad/s) and acceleration (W - rad/s$^2$) of all the links of robot obtained from NSGA-II and
DE are shown by Figures 5.5 and 5.6 for application 1 (without obstacles), Figures 5.7 and 5.8 for application 2 (with fixed and moving obstacles), and Figures 5.9 and 5.10 for application 3 (with fixed and oscillating obstacles). Table 5.5 compares the optimum results obtained from various techniques. From Table 5.5, it is observed that DE gives better results than NSGA-II. So DE is superior to NSGA-II.

**Figure 5.5** Displacement, velocity and acceleration for optimal trajectories by NSGA-II for application 1 (PUMA 560 robot)
Figure 5.6 Displacement, velocity and acceleration for optimal trajectories by DE for application 1 (PUMA 560 robot)
Figure 5.7 Displacement, velocity and acceleration for optimal trajectories by NSGA-II for application 2 (PUMA 560 robot)
Figure 5.8 Displacement, velocity and acceleration for optimal trajectories by DE for application 2 (PUMA 560 robot)
Figure 5.9 Displacement, velocity and acceleration for optimal trajectories by NSGA-II for application 3 (PUMA 560 robot)
Figure 5.10 Displacement, velocity and acceleration for optimal trajectories by DE for application 3 (PUMA 560 robot)
Table 5.5 Comparison of optimum results (PUMA 560 robot)

<table>
<thead>
<tr>
<th></th>
<th>APPLICATION 1</th>
<th>APPLICATION 2</th>
<th>APPLICATION 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z₁</td>
<td>z₂</td>
<td>z₃</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>5.15</td>
<td>26342</td>
<td>0</td>
</tr>
<tr>
<td>DE</td>
<td>1.88</td>
<td>24681</td>
<td>0</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>10.2</td>
<td>14624</td>
<td>0</td>
</tr>
<tr>
<td>DE</td>
<td>8.97</td>
<td>12253</td>
<td>0</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>5.34</td>
<td>23483</td>
<td>0</td>
</tr>
<tr>
<td>DE</td>
<td>3.8</td>
<td>19341</td>
<td>0</td>
</tr>
</tbody>
</table>

5.2.1.2 STANFORD Robot

The optimal displacement (Q – rad or m), velocity (V – rad/s or m/s) and acceleration (W – rad/s² or m/s²) of all the links of robot obtained from NSGA-II and DE, are shown by Figures 5.11 and 5.12 for application 1 (without obstacles), Figures 5.13 and 5.14 for application 2 (with fixed and moving obstacles), and Figures 5.15 and 5.16 for application 3 (with fixed and oscillating obstacles). Table 5.6 compares the optimum results obtained from various techniques. From Table 5.6, it is observed that DE gives better results than NSGA-II. So DE is superior to NSGA-II.
Figure 5.11 Displacement, velocity and acceleration of optimal trajectories by NSGA-II for application 1 (STANFORD robot)
Figure 5.12 Displacement, velocity and acceleration of optimal trajectories by DE for application 1 (STANFORD robot)
Figure 5.13 Displacement, velocity and acceleration of optimal trajectories by NSGA-II for application 2 (STANFORD robot)
Figure 5.14 Displacement, velocity and acceleration of optimal trajectories by DE for application 2 (STANFORD robot)
Figure 5.15 Displacement, velocity and acceleration of optimal trajectories by NSGA-II for application 3 (STANFORD robot)
Figure 5.16 Displacement, velocity and acceleration of optimal trajectories by DE for application 3 (STANFORD robot)
From Figures 5.5 – 5.16, it is noted that the robot joints displacement, velocity, acceleration and jerk are within their limiting values. So the resulting trajectories are smoother. Table 5.6 shows the optimum results obtained from various techniques for comparison. From Table 5.6, it is observed that the proposed technique DE gives better results than NSGA-II. Table 5.7 compares the optimum results obtained from proposed techniques (NSGA-II and DE) and literature results (SUMT (Saramago et al 2000)) for \(w_1=0.3\), \(w_2 =0.4\), \(w_3=0.3\). From Table 5.7, it is observed that the proposed techniques (NSGA-II and DE) give better results than SUMT (Saramago et al 2000).

**Table 5.6 Comparison of optimum results (STANFORD robot)**

<table>
<thead>
<tr>
<th></th>
<th>(z_1)</th>
<th>(z_2)</th>
<th>(z_3)</th>
<th>(z_4)</th>
<th>(z_5)</th>
<th>(z_6)</th>
<th>(f_c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION 1</td>
<td>NSGA-II</td>
<td>9.28</td>
<td>20638</td>
<td>0</td>
<td>0.714</td>
<td>804.5</td>
<td>458.92</td>
</tr>
<tr>
<td></td>
<td>DE</td>
<td>5.91</td>
<td>16630</td>
<td>0</td>
<td>0.7</td>
<td>776.9</td>
<td>425.7</td>
</tr>
<tr>
<td>APPLICATION 2</td>
<td>NSGA-II</td>
<td>8.5</td>
<td>24800</td>
<td>0</td>
<td>0.72</td>
<td>457.4</td>
<td>643.95</td>
</tr>
<tr>
<td></td>
<td>DE</td>
<td>4.8</td>
<td>16630</td>
<td>0</td>
<td>0.71</td>
<td>422.5</td>
<td>567.92</td>
</tr>
<tr>
<td>APPLICATION 3</td>
<td>NSGA-II</td>
<td>7.5</td>
<td>32086</td>
<td>0</td>
<td>0.724</td>
<td>217.5</td>
<td>437.8</td>
</tr>
<tr>
<td></td>
<td>DE</td>
<td>6.3</td>
<td>30312</td>
<td>0</td>
<td>0.712</td>
<td>201.8</td>
<td>411.36</td>
</tr>
</tbody>
</table>

**Table 5.7 Comparison of optimum results (STANFORD robot)**

<table>
<thead>
<tr>
<th>APPLICATION 1</th>
<th>Time((z_1))(sec)</th>
<th>Energy((z_2))(Nm)</th>
<th>(f_c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMT</td>
<td>13.00</td>
<td>36974</td>
<td>36987.300</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>9.28</td>
<td>32086</td>
<td>32088.784</td>
</tr>
<tr>
<td>DE</td>
<td>5.91</td>
<td>30312</td>
<td>30313.773</td>
</tr>
<tr>
<td>APPLICATION 2</td>
<td>SUMT</td>
<td>12.40</td>
<td>27875</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>8.50</td>
<td>24800</td>
<td>24802.550</td>
</tr>
<tr>
<td>DE</td>
<td>6.30</td>
<td>16630</td>
<td>16631.890</td>
</tr>
<tr>
<td>APPLICATION 3</td>
<td>SUMT</td>
<td>9.90</td>
<td>18871</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>7.50</td>
<td>20638</td>
<td>20640.250</td>
</tr>
<tr>
<td>DE</td>
<td>6.30</td>
<td>16630</td>
<td>16631.890</td>
</tr>
</tbody>
</table>
The computational time to find the multicriterion cost function \( f_c \) for all the applications in a HP computer (with configuration of 640MB DDR RAM, 40 GB HDD, Pentium 4 Processor with 3GHz speed) are given in Table 5.8. Here the number of generation is 100. From Table 5.8, we can observe that the computational time for computing the multicriterion cost function in DE is lesser than that of in NSGA-II. So DE technique is faster than NSGA-II.

<table>
<thead>
<tr>
<th>Application No.</th>
<th>STANFORD robot</th>
<th>PUMA560 robot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NSGA-II</td>
<td>DE</td>
</tr>
<tr>
<td>1</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>8</td>
</tr>
</tbody>
</table>

5.2.2 Limitations

In this stage, cubic B-spline functions have been used for constructing the joint trajectories, since they allow a great deal of control of the continuity of the joints motion between adjacent segments. The success of the cubic B-spline approach depends on how a finite-dimensional problem can approximate the original infinite-dimensional problem. If the finite-dimensional problem is formulated with less number of sample points (knots), it may occur that the constraints are violated between those points. Increasing the number of sample points will overcome this difficulty. In the next stage, an investigation of the influence of the number of knots on an optimal trajectory planning is presented.
5.3 STAGE 10: EFFECT OF NUMBER OF KNOTS ON TRAJECTORY PLANNING IN THE PRESENCE OF FIXED, MOVING AND OSCILLATING OBSTACLES

The aim of this stage is to address an important issue of optimal trajectory planning of industrial robots. This stage presents a study about the effect of number of knots on optimal trajectory planning for an industrial robot manipulator in the presence of moving obstacles.

5.3.1 Introduction

The same problem in the previous stage is considered with the following change: only 5 knot points are considered in the previous stage. But here, three different number of knot points (20 in first case, 40 in second case and 60 in the last case) are considered. So, total number of variables (B-spline coefficients) is 132 in first case, 252 in second case and 372 in the last case.

5.3.2 Problem details

The aim is to minimize a combined objective function by considering dynamic equations of motion with various constraints. The constraints are actuator efforts, joint limits and obstacle avoidance. Trajectories are defined by B-Spline functions. The variables are position of the B-spline nodes of the trajectory. The robot model is based on Denavit-Hartenberg parameters. A dynamic model for the manipulator is obtained by writing the Euler-Lagrange’s equations using Lagrange’s energy function in order to evaluate the mechanical energy of the actuator’s action. This problem is a non-linear constrained optimization problem with 6 objective functions, 31 constraints and maximum of 372 variables. The combined objective
function is a weighted balance of transfer time, actuators efforts, penalty function to guarantee collision free motion, singularity avoidance, joint jerks and joint accelerations. All the types of obstacles (fixed, moving and oscillating obstacles) are present in the workspace of the robot. The proposed optimization techniques are explained by applying the same to an industrial robot (STANFORD robot) with a numerical example having three applications. The weightage values given to objective functions are \( w_1 = w_2 = 0.25, \ w_3 = w_4 = w_5 = 0.1, \ w_6 = 0.2 \). \( N_1 = 1, \ N_2 = 1e6, \ N_3 = 1, \ N_4 = 1, \ N_5 = 100 \) and \( N_6 = 10 \) are normalizing parameters of objective functions.

### 5.3.2.1 NSGA-II Parameters

The following are the values of the parameters of NSGA-II technique that have been used to obtain the best optimal results: Variable type = Real variable, Population size = 100, Crossover probability = 0.6, Real-parameter mutation probability = 0.01, Real-parameter SBX parameter = 10, Real-parameter mutation parameter = 100, Total number of generations = 100.

### 5.3.2.2 Differential Evolution Parameters

The following are the values of the parameters of DE technique that have been used to obtain the best optimal results: Strategy = DE/rand/1/bin, crossover constant CR = 0.9, number of population NP = 360, \( F_s = 0.5 \) and total number of generations = 100.

### 5.3.3 Results and Discussion

As a sample, the robot joints optimal displacement (Q - rad or m), velocity (V - rad/s or m/s) and acceleration (W - rad/s\(^2\) or m/s\(^2\)) obtained from DE and NSGA-II for case 1 of application 1 (without obstacles) are shown in Figures 5.17 and 5.18. Figures 5.19 and 5.20 show the robot joints optimal
displacements, velocities and accelerations obtained from DE and NSGA-II for case 2 of application 2 (with fixed and moving obstacles). The robot joints optimal displacements, velocities and accelerations obtained from DE and NSGA-II for case 3 of application 3 (with fixed and oscillating obstacles) are shown in Figures 5.21 and 5.22. From the Figures 5.17 – 5.22, it is noted that the robot joint displacements, velocities and accelerations are within their limiting values. Also from the Figures 5.17 – 5.22, it is observed that for all applications, if we increase number of knot points in between the initial and final configurations, the resultant optimal trajectories are not smooth. They have more ups and downs.

The computational time to find the multicriterion cost function \( f_c \) for all the cases (20, 40 and 60 knots) in all applications in a HP computer (with configuration of 640MB DDR RAM, 40 GB HDD, Pentium 4 Processor with 3GHz speed) is given in Table 5.9. From Table 5.9, we can observe that the computational time for computing the multicriterion cost function increases with the increase in number of knots, since the number of variables and the number of functions to be evaluated by the computational technique are increasing as we increase the number of knot points. Also we can note that DE technique is faster than NSGA-II. Also DE technique converges quickly than NSGA-II technique.

Table 5.9 Number of knots Vs computational time

<table>
<thead>
<tr>
<th>Application No.</th>
<th>Number of knots</th>
<th>Number of decision variables</th>
<th>Number of iteration</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NSGA-II</td>
<td>DE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>132</td>
<td>100</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>132</td>
<td>100</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>132</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>252</td>
<td>100</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>252</td>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>252</td>
<td>100</td>
<td>37</td>
</tr>
<tr>
<td>1</td>
<td>60</td>
<td>372</td>
<td>100</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>372</td>
<td>100</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>372</td>
<td>100</td>
<td>47</td>
</tr>
</tbody>
</table>

Figure 5.17 Displacement, velocity and acceleration of optimal trajectories by DE for application 1 (case1)
Figure 5.18 Displacement, velocity and acceleration of optimal trajectories by NSGA-II for application 1 (case1)
Figure 5.19 Displacement, velocity and acceleration of optimal trajectories by DE for application 2 (case2)
Figure 5.20 Displacement, velocity and acceleration of optimal trajectories by NSGA-II for application 2 (case2)
Figure 5.21 Displacement, velocity and acceleration of optimal trajectories by DE for application 3 (case 3)
Figure 5.22 Displacement, velocity and acceleration of optimal trajectories by NSGA-II for application 3 (case 3)
The Table 5.10 compares the optimum multicriterion cost function \((f_c)\) values obtained for the three cases (20, 40 and 60 knots) in applications 1, 2 and 3 from NSGA-II and DE techniques.

**Table 5.10 Comparison of optimum results \((f_c)\)**

<table>
<thead>
<tr>
<th>Application No.</th>
<th>20 KNOT (case 1)</th>
<th>40 KNOT (case 2)</th>
<th>60 KNOT (case 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DE</td>
<td>NSGA-II</td>
<td>DE</td>
</tr>
<tr>
<td>1</td>
<td>4.192405</td>
<td>4.044207</td>
<td>10.173573</td>
</tr>
<tr>
<td>3</td>
<td>26.361542</td>
<td>15.680773</td>
<td>11.519875</td>
</tr>
</tbody>
</table>

From Table 5.10, it is observed that (1) NSGA-II gives better results than DE in majority of the cases, (2) a minimum multicriterion cost function \((f_c)\) value for application 1 (without obstacles) is obtained in case 1 (20 knots) using NSGA-II. A minimum \(f_c\) value for application 2 (with fixed and moving obstacles) is obtained in case 2 (40 knots) using DE. A minimum \(f_c\) value for application 3 (with fixed and oscillating obstacles) is obtained in case 3 (60 knots) using DE. So it is concluded that we have to consider more number of knots when the complexity of problem increases. When we consider the robot without obstacles (low complexity), we have to consider minimum number of knots (20 knots). When we consider the robot with fixed and moving obstacles (medium complexity), we have to consider medium number of knots (40 knots). When we consider the robot with oscillating obstacles (high complexity), we have to consider maximum number of knots (60 knots). But in a real world situation, there will be some obstacles in the workspace of any industrial robot. So to get a best optimal trajectory planning...
for industrial robots, we need to consider more number of knot points in
between the initial and final configurations of robots.

The next stage presents a new general method for computing the
optimal motions of an industrial robot manipulator (Adeptone XL robot) in
the presence of fixed and moving obstacles.

5.4 STAGE 11: COMBINED OBJECTIVE TRAJECTORY
PLANNING FOR AN INDUSTRIAL ROBOT IN THE
PRESENCE OF FIXED AND MOVING OBSTACLES

The aim of this stage is to present a new general method for
computing the optimal motions of an industrial robot manipulator (Adeptone
XL robot) in the presence of fixed and moving obstacles. 20 knot points are
considered for the cubic B-spline curve that represents the trajectory. The
application 2 (with fixed and moving obstacles) of the problem discussed in
the stage 9 is considered with the following change: Only 5 knot points are
considered in the previous stage. But here 20 knot points are considered. The
target is to move the robot in a workspace avoiding the fixed and moving
obstacles, while minimizing a combined objective function of the robot by
considering the physical constraints and actuator limits. This problem has
6 objective functions, 21 constraints and 88 variables. The variables are
position of the B-spline nodes of the trajectory. Minimization of travelling
time, mechanical energy of the actuators, joint jerks, joint accelerations,
penalty for free collision-motion and maximum manipulability are considered
together to build a performance index. \( w_1 = 0.25, \ w_2 = 0.25, \ w_3 = 0.1, \ w_4 = 0.1, \ w_5 = 0.1, \ w_6 = 0.2 \) are the weightage values given to the objective functions.
\( N_1 = 4.5, \ N_2 = 1e7, \ N_3 = 0, \ N_4 = 0.1, \ N_5 = 150 \) and \( N_6 = 150 \) are normalizing
parameters of the objective functions (average value of individual objective
functions). The problem formulation, kinematic and dynamic models,
trajectory representation and checking of obstacle avoidance are same as in stage 10.

5.4.1 Numerical Application

As this is a multicriterion optimization problem, the cost function is written using normalised weighting objective functions method. Four degrees of freedom Adeptone XL robot manipulator is considered here (Figure 5.23). The geometrical and limiting parameters of Adeptone XL robot are presented in Tables 5.11 and 5.12 respectively (Adept one robot user guide 2005). It is considered that the robot is initially at rest and comes to a full stop at the end of the trajectory. This way $q_1 = q_m = q_e = q_n = 0$ for all robot joints. In this application the end-effectors trajectory $(\psi_1)$ of the Adeptone XL manipulator has to avoid a fixed wall $(\psi_2)$ and two moving and self rotating obstacles $(\psi_3$ and $\psi_4)$ as in Figure 5.24. The goal is to obtain an optimal trajectory under the constraints given in Table 5.12. In this case the initial and final points of the end-effectors: $q = [-2.0048, 0.7522, 0.4230, 1.5043]$ and $q_m = [-0.4511, 0.7152, 0.2150, 1.4305]$. Following are the obstacles dimensions: Prism 1: $(X_1=0.7$ and $X_2=1.1; Y_1=0.6$ and $Y_2=0.8; Z_1=0.0; Z_2=0.30)$ and Prism 2: $(X_1=0.7$ and $X_2=1.1; Y_1=1.27$ and $Y_2=1.47; Z_1=0.0; Z_2=0.21)$.

5.4.1.1 NSGA-II Parameters

The following are the values of the parameters of NSGA-II technique that have been used to obtain the best optimal results: Variable type = Real variable, Population size=100, Crossover probability =0.7, Real-parameter mutation probability =0.01, Real-parameter SBX parameter =10, Real-parameter mutation parameter =100, Total number of generations=100.
Figure 5.23  Adeptone XL robot manipulator

Figure 5.24  End-effector tridimensional optimal trajectory

Table 5.11  Joint parameters for Adeptone XL robot (Adeptone robot user guide 2005)

<table>
<thead>
<tr>
<th>Joints</th>
<th>a_i (mm)</th>
<th>α_i</th>
<th>d_i (mm)</th>
<th>θ_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>425</td>
<td>180°</td>
<td>876</td>
<td>θ_1</td>
</tr>
<tr>
<td>2</td>
<td>375</td>
<td>0</td>
<td>0</td>
<td>θ_2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>d_3</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 5.12 Limiting values for Adept one XL robot (Adept one robot user guide 2005)*

<table>
<thead>
<tr>
<th>Constraint</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC(rd)</td>
<td>2.6180</td>
<td>2.4435</td>
<td>203</td>
<td>4.7124</td>
</tr>
<tr>
<td>VC(rd/s)</td>
<td>11.3446</td>
<td>16.057</td>
<td>1200</td>
<td>57.5959</td>
</tr>
<tr>
<td>WC(rd/s²)</td>
<td>9.5</td>
<td>9.5</td>
<td>9.5</td>
<td>9.5</td>
</tr>
<tr>
<td>JC(rd/s³)</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>UC(Nm)</td>
<td>400</td>
<td>280</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

*Joint 3 – QC(mm), VC(mm/s), WC(mm/s²), JC(mm/s³), UC(Nm)

5.4.1.2 DE Parameters

The following are the values of the parameters of DE technique that have been used to obtain the best optimal results: Strategy = DE/rand/1/bin, crossover constant CR=0.9, number of population NP=500, F=0.5 and total number of generations=100.

5.4.2 Results and Discussion

The optimal displacement (Q - rad or m), velocity (V - rad/s or m/s) and acceleration (W - rad/s² or m/s²) of all the robot links obtained from DE and NSGA-II are shown in Figures 5.25 and 5.26. From Figures 5.25 and 5.26, it is noted that the robot joints displacement, velocity, acceleration and jerk are within their limiting values. So the resulting trajectories are smoother. The optimum objective functions obtained from NSGA-II and DE algorithm are given in Table 5.13. The result histories of NSGA-II and DE are compared in Figure 5.27. From Table 5.13, it is noted that NSGA-II technique
is superior to DE technique. But from the Figure 5.27, it is noted that DE technique is faster than NSGA-II technique.

Figure 5.25 Displacement, velocity and acceleration of optimal trajectories by DE
Figure 5.26 Displacement, velocity and acceleration of optimal trajectories by NSGA-II

Table 5.13 The optimum solutions

<table>
<thead>
<tr>
<th>Technique</th>
<th>Time (T) (sec)</th>
<th>Multicriterion cost Function (f_c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>4.58</td>
<td>0.527503</td>
</tr>
<tr>
<td>DE</td>
<td>4.57</td>
<td>0.645305</td>
</tr>
</tbody>
</table>

Figure 5.27 Result histories of NSGA-II and DE

5.4.3 Limitations

The following are the limitations of this stage: 1. The problem is approached as a single objective optimisation problem (using weighted objective function method, all objective functions are combined as a single objective function), 2. The proposed approach is not applicable, if the user does not know what weightage is to be given for each objective, 3. The method used by this stage cannot be directly used for treating all objectives individually and 4. Only one optimal solution is got from this stage. But for a real world problem, a Pareto optimal front that offers more number of optimal solutions for user’s choice is most desirable. So the next stage proposes two evolutionary algorithms namely Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi Objective Differential Evolution (MODE) to
do multi-objective dynamic optimal trajectory planning of Adeptone XL robot in the presence of fixed and oscillating obstacles.

5.5 STAGE 12: MULTI-OBJECTIVE TRAJECTORY PLANNING FOR AN INDUSTRIAL ROBOT IN THE PRESENCE OF FIXED AND OSCILLATING OBSTACLES

5.5.1 Introduction

This stage proposes two evolutionary algorithms namely Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi Objective Differential Evolution (MODE) to do multi-objective dynamic optimal trajectory planning of Adeptone XL robot in the presence of fixed and oscillating obstacles. The same problem in the previous stage (stage 11) is considered with the following changes: 1. Fixed and oscillating obstacles are considered and 2. Multi-objective approach is used.

Two methods namely normalized weighting objective functions and average fitness factor are combinedly used to select the best solution trade-offs. Two multi-objective performance measures namely solution spread measure and ratio of non-dominated individuals are used to evaluate the Pareto optimal fronts. Two multi-objective performance measures namely optimizer overhead and algorithm effort are used to find the computational effort of optimization algorithms.

5.5.2 Problem Formulation

An industrial robot manipulator (Adeptone XL robot) with 4 degrees of freedom (dof) is considered. The target is to move the robot in a workspace avoiding the fixed and oscillating obstacles, while minimizing travelling time, mechanical energy of the actuators, joint jerks, joint
accelerations, penalty function - to guarantee free collision-motion and maximizing manipulability measure of the robot considering the physical constraints, actuator limits and obstacle avoidance. This problem has 6 objective functions, 21 constraints and 88 variables (polynomial coefficients \( \lambda_j \) of B-spline functions that represent the trajectories).

The singularity avoidance is chosen in the form of manipulability measure. Maximizing the manipulability measure will force the manipulator away from singularity. The multicriterion optimisation problem is defined as follows:

Minimize travelling time between initial and final configurations = \( z_1 \) \( (5.8a) \)

Minimize Quadratic average of actuator torques: \( z_2 = \int_0^T \sum_{i=1}^n (u_i(t))^2 \, dt \) \( (5.8b) \)

Minimize Penalty parameter for free-collision motion: \( z_3 = f_{\text{dis}} \) \( (5.8c) \)
Maximize Manipulability measure: \( z_4 = |\det(J_m)| \) \( (5.8d) \)

Minimize Integral of squared joint jerks: \( z_5 = \int_0^T \sum_{i=1}^n (q_i^2(t)) \, dt \) \( (5.8e) \)

Minimize Integral of squared joint accelerations: \( z_6 = \int_0^T \sum_{i=1}^n (q_i^2(t)) \, dt \) \( (5.8f) \)

Subject to,

1. Displacement constraint \( \max|q_{ij}(t)| \leq QC_j \) \( (5.9) \)
2. Velocity constraint \( \max|\dot{q}_{ij}(t)| \leq VC_j \) \( (5.10) \)
3. Acceleration constraint \( \max|\ddot{q}_{ij}(t)| \leq WC_j \) \( (5.11) \)
4. Jerk constraint \( \max|\dddot{q}_{ij}(t)| \leq JC_j \) \( (5.12) \)
5. Force/torque constraint
   \[ \max|u_{ij}(t)| \leq UC_j \] for \( j=1,2,...,n \) and \( i=1,2,...,m-1 \) \( (5.13) \)
6. Obstacle avoidance constraint

\[ d_{i_0}(t) > 0 \text{ for } (l, q) \in I \]  

(5.14)

where, \( u_i \) is the generalized forces, \( J_m \) is jacobian matrix of the robot, \( I \) is the set of possibly colliding pairs of parts, \( n \) represents number of the joints and \( m \) represents number of the knots used to construct the trajectory.

5.5.3 Numerical Application

The geometrical and limiting parameters of Adeptone XL robot (Figure 5.23) are presented in Tables 5.11 and 5.12 respectively. It is considered that robot is initially at rest and comes to a full stop at the end of the trajectory. Therefore \( q_i = q_m = q_i = q_m = 0 \) for all robot joints. In this application the end-effectors trajectory \( (\psi_2) \) of the Adeptone XL manipulator has to avoid two fixed obstacles \( (\psi_3 \text{ and } \psi_4) \) and an oscillating obstacle - Pendulum \( (\psi_2) \) as shown in Figure 5.28.

![Figure 5.28 End-effector tridimensional optimal trajectory](image)

The goal is to obtain the optimal trajectory under the constraints given in Table 5.12. Twenty-knot \( (m=20) \) points have been considered for the
trajectory. In this problem, cubic B-spline coefficients of the joint trajectories are considered as variables. Total number of variables is 88.

5.5.4 Implementation of NSGA-II and MODE Algorithms

The steps for running the algorithms are summarized below:

Step 1: The following are the inputs to NSGA-II and MODE algorithms:

1. Details of Adeptone XL robot:
   Geometrical parameters. (Table 5.11)
   Displacement, velocity, acceleration, jerk and torque limits of each robot joint (Table 5.12).

2. Details of cubic B-spline curve that defines the trajectory:
   Total number of knot points considered = 20.
   The initial and final configurations of the end-effectors are:
   \( q_i = [0.1682\text{rd}, 1.3849\text{rd}, 1.1362\text{m}, -0.7555\text{rd}, -0.4702\text{rd}, 0.1472\text{rd}] \) and \( q_f = [-0.7610\text{rd}, 0.2450\text{rd}, 1.4366\text{m}, 1.0095\text{rd}, -1.0146\text{rd}] \).

Details of obstacles:
The obstacle dimensions are: prism 1 \((X_1=0.7 \text{ and } X_2=1.1; Y_1=0.6 \text{ and } Y_2=0.8; Z_1=0.0, Z_2=0.30)\) and prism 2 \((X_1=0.7 \text{ and } X_2=1.1; Y_1=1.27 \text{ and } Y_2=1.47; Z_1=0.0, Z_2=0.21)\).
Obstacle avoidance checking formula, (Equation 5.14).

3. Formulae to find the objective functions \((z_1 \text{ to } z_6)\), robot joints displacement \((q)\), robot joints velocity \((\dot{q})\), robot joints acceleration \((\ddot{q})\), robot joints jerk \((\dddot{q})\), robot joints torque \((u_i)\) at each time instant. (Equations (5.1) - (5.13) and (4.22) - (4.26)).
4. Formulae to find multi objective performance measures namely solution spread measure, ratio of non-dominated individuals, optimiser overhead and algorithm effort. (Equations (3.5) - (3.8)).

5. Formulae to find the combined objective function ($f_c$) and average fitness factor value ($F_{avg}$). Based on these values, the best optimal solution from Pareto optimal front is selected. $w_1=w_2=0.25$, $w_3=w_4=0.1$, $w_5=0.15$, $w_6 = 0.15$ are the weightage values given to the objective functions in the sample case. $N_1=8.5$, $N_2=9200000$, $N_3=0$, $N_4=0.1$, $N_5=150$ and $N_6=120$ are normalizing parameters of the objective functions (average value of individual objective functions).

6. The variables limits.

7. The parameters NSGA-II and MODE algorithms. (Given in sections 5.5.4.1 and 5.5.4.2).

Step 2: The software based on NSGA-II and MODE algorithms finds the optimal variables in such a way that:

1. The travelling time, mechanical energy of the actuators, joint jerks, joint accelerations, penalty function - to guarantee free collision motion are minimum and manipulability measure is maximum.

2. All the constraints (joints limits, actuator limits and obstacle avoidance) are satisfied.

Step 3: Step 2 is repeated up to maximum number of iterations.

Step 4: The following are the outputs from NSGA-II and MODE algorithms:
1. Pareto optimal fronts obtained from NSGA-II and MODE. The Pareto optimal front gives number of trade-off solutions. Each solution has optimal objective function value, optimal value of variables and the constraints value. All constraints shall be satisfied by any solution in the Pareto optimal front.

2. The optimal displacement (Q - rad or m), velocity (V - rad/s or m/s) and acceleration (W - rad/s^2 or m/s^2) of all the robot joints.

3. The best optimal solution trade-off selected by the methods namely normalized weighting objective functions and average fitness factor in a combined manner from Pareto optimal fronts. The best optimal solution is the one that gives a safer, faster, economical and smoother optimal trajectory.

4. The strength of Pareto optimal fronts evaluated by the two multi-objective performance measures namely solution spread measure and ratio of non-dominated individuals.

5. The computational effort of NSGA-II and MODE algorithms found by the two multi-objective performance measures namely optimiser overhead and algorithm effort.

5.5.4.1 NSGA-II Parameters

The following are the values of the parameters of NSGA-II technique that have been used to obtain the best optimal results: Variable type = Real variable, Population size=100, Crossover probability =0.7, Real-parameter mutation probability =0.01, Real-parameter SBX parameter =10, Real-parameter mutation parameter =100, Total number of generations=100.
5.5.4.2 MODE Parameters

The following are the values of the parameters of MODE technique that have been used to obtain the best optimal results: Strategy = MODE/rand/1/bin, crossover constant CR=0.9, number of population NP=500, $F_s=0.5$ and total number of generations=100.

5.5.5 Results and Discussion

The optimal solution trade-offs obtained from NSGA-II and MODE are given in Figures 5.29 and 5.30 respectively. From Figures 5.29 and 5.30, it is observed that NSGA-II gives optimal solution trade-offs with more number of non-dominated solutions for user’s choice than MODE. So NSGA-II technique is the best one for this multicriterion optimization problem, if the user wants more number of solution trade-offs for his choice. The results from NSGA-II and MODE are listed in Tables 5.14 – 5.16. From Tables 5.14 – 5.16, it is observed that MODE gives minimum combined objective function ($f_c$), maximum average fitness factor ($F_{avg}$), minimum optimiser overhead (OO) and minimum algorithm effort than those of NSGA-II. But NSGA-II technique gives minimum solution spread measure (SSM) and maximum ratio of non-dominated individuals (RNI) than those of MODE. The two methods (normalized weighting objective functions and average fitness factor methods) are combinedly used to select the best optimal solution trade-offs from the optimal solution trade-offs obtained from NSGA-II and MODE. The best optimal solution trade-offs obtained from NSGA-II and MODE for the sample case $w_1=w_2=0.25$, $w_3=w_4=0.1$, $w_5=0.15$, $w_6 = 0.15$ are shown in Figure 5.31. From Figure 5.31, it is noted that MODE gives best results for five objective functions (Minimum values for $z_1$, $z_3$, $z_5$ and $z_6$ and maximum value for $z_4$). It is also noted that MODE technique converges quickly than NSGA-II. Further, the computational time to find the optimum solution trade-offs in MODE is $1/3^{rd}$ of that of in NSGA-II. So MODE is
faster than NSGA-II. So MODE technique is the best one for this multicriterion optimization problem, if the user wants a best optimal solution trade-off very quickly. Figures 5.32 and 5.33 show the optimal displacement \((Q - \text{rad or m})\), velocity \((V - \text{rad/s or m/s})\) and acceleration \((W - \text{rad/s}^2 \text{ or m/s}^2)\) of all the robot joints for the best solution trade-offs for the sample case \(w_1=w_2=0.25, \ w_3=w_4=0.1, \ w_5=0.15, \ w_6 = 0.15\) obtained from MODE and NSGA-II. From Figures 5.32 and 5.33, it is noted that the robot joints displacement, velocity and acceleration are within their limiting values. So the resulting trajectories are smooth and practical.

Table 5.14 Results obtained from NSGA-II and MODE algorithms

<table>
<thead>
<tr>
<th>(Z_1)</th>
<th>(Z_2)</th>
<th>(Z_3)</th>
<th>(Z_4)</th>
<th>(Z_5)</th>
<th>(Z_6)</th>
<th>(F_{\text{avg}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Z_{\text{max}})</td>
<td>10</td>
<td>9506597.53</td>
<td>0.00005</td>
<td>0.1574</td>
<td>279.932</td>
<td>269.784</td>
</tr>
<tr>
<td>(Z_{\text{min}})</td>
<td>8.48</td>
<td>9178597.58</td>
<td>0.0</td>
<td>0.0483</td>
<td>119.843</td>
<td>107.838</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>8.98</td>
<td>9178597.58</td>
<td>0.0</td>
<td>0.0483</td>
<td>163.238</td>
<td>156.113</td>
</tr>
<tr>
<td>MODE</td>
<td>8.48</td>
<td>9468839.99</td>
<td>0.0</td>
<td>0.1574</td>
<td>119.843</td>
<td>107.838</td>
</tr>
</tbody>
</table>

Table 5.15 Combined objective function and Algorithm effort obtained from NSGA-II and MODE algorithms

<table>
<thead>
<tr>
<th>Proposed Algorithm</th>
<th>(f_c)</th>
<th>Simulation time (T_{\text{run}}) (sec)</th>
<th>(N_{\text{eval}})</th>
<th>Algorithm effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>0.82</td>
<td>2</td>
<td>87</td>
<td>0.02299</td>
</tr>
<tr>
<td>MODE</td>
<td>0.6</td>
<td>2</td>
<td>124</td>
<td>0.01575</td>
</tr>
</tbody>
</table>

Table 5.16 SSM, RNI and OO obtained from NSGA-II and MODE algorithms

<table>
<thead>
<tr>
<th>Technique</th>
<th>SSM</th>
<th>RNI</th>
<th>OO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>0.01023</td>
<td>0.32</td>
<td>0.0769</td>
</tr>
<tr>
<td>MODE</td>
<td>0.02356</td>
<td>0.24</td>
<td>0.0324</td>
</tr>
</tbody>
</table>
Figure 5.29 Optimal solution tradeoffs obtained from NSGA-II

Figure 5.30 Optimal solution tradeoffs obtained from MODE

Figure 5.31 Best optimal solution tradeoffs obtained from NSGA-II and MODE
Figure 5.32 Optimal motions of the robot joints obtained from MODE

Figure 5.33 Optimal motions of the robot joints obtained from NSGA-II
5.5.6 Limitations

Up to this stage, change in payload or payload constraint is not considered. To get a real world trajectory planning, we need to consider the payload constraint as well (Korayem et al 2008). So the next stage presents an optimal trajectory planning for an industrial robot (PUMA560 robot) considering the payload constraint in addition to all other constraints and important objective functions.

5.6 STAGE 13: COMBINED OBJECTIVE TRAJECTORY PLANNER WITH PAYLOAD CONSTRAINTS

This stage presents a new general methodology based on the evolutionary algorithms NSGA-II and DE for optimal trajectory planning of an industrial robot manipulator (PUMA560) considering payload constraints in addition to all other constraints and important objective functions. PUMA560 robot, an industrial robot manipulator with 6 degrees of freedom (dof) is considered (Figure 5.34). The target is to move the robot that carries a payload from an initial configuration to final configuration in its workspace while minimizing a multicriterion cost function of the robot considering the robot physical constraints, payload constraints and actuator limits. This problem has 5 objective functions, 32 constraints and 252 variables (polynomial coefficients $\gamma^i_j$ of B-spline functions that represent the trajectories).

5.6.1 Problem Formulation

Minimization of travelling time, the total energy involved in the motion, joint jerks, joint accelerations and maximization of manipulability are considered together to build a multicriterion cost function. Singularity avoidance is chosen in the form of manipulability measure. Maximizing the manipulability measure will force the manipulator away from singularity.
Multiple objectives are combined into scalar objective via weight vector. Weights may be assigned through: direct assignment, eigenvector method, empty method, minimal information method, randomly determined or adaptively determined. From the literatures, it is noted that the first priority is given for minimizing operation time (minimum travelling time) and saving energy (total energy involved in the motion) (Bobrow et al 1985, Shin et al 1985, Chen 1991, Shin et al 1986, Balkan 1998, Martin et al 1999, Shiller 1996, Chettibi et al 2004, Saramago et al 2000). The next priority is given for obtaining practical trajectory (maximizing manipulability measure) (Lin 2004, Mayorga et al 1987, Lloyd et al 2001). The last priority is given for reducing vibrations of robot joints (minimizing joint accelerations and jerks) (Gasperatto et al 2007, Elnager 2000). So \( w_1 = w_2 = 0.25, w_3 = 0.2, w_4 = w_5 = 0.15 \) are the weightage values given for the objective functions in this problem.

If the objective functions are simply weighted and added to produce a single fitness, the function with largest range would dominate evolution. A poor input value for the objective with larger range makes the overall value much worse than a poor value for the objective with smaller range. To avoid this, all objective functions are normalised to have same range. So \( N_1 = 6, N_2 = 1000, N_3 = 0.005, N_4 = 5 \) and \( N_5 = 20 \) are normalizing parameters of objective functions.

The multi criterion optimisation problem is defined as follows:

Minimize \( f_c = w_1 z_1 / N_1 + w_2 z_2 / N_2 - w_3 z_3 / N_3 + w_4 z_4 / N_4 + w_5 z_5 / N_5 \)  

\[ (5.15a) \]

where,

Total energy involved in the motion: \( z_2 = \int_0^T (u_i(t) \dot{q}_i)^2 \, dt \)  

\[ (5.15b) \]

Manipulability measure: \( z_3 = |\text{det}(J_m)| \)  

\[ (5.15c) \]
Integral of squared link jerks: \[ z_4 = \int_0^T \sum_{i=1}^{n} (\dot{q}_i)^2 \, dt \] (5.15d)

Integral of squared link accelerations: \[ z_5 = \int_0^T \sum_{i=1}^{n} (\ddot{q}_i)^2 \, dt \] (5.15e)

\( f_c \) is the multicriterion cost function,

\( z_1 = 'T' \) is the total travelling time between initial and final configurations,

\( u_i \) is the generalized forces,

\( J_m \) is Jacobian matrix of the robot,

Subject to,

1. Displacement constraint \[ |q_{ji}(t)| \leq q_{ji}^{\max} \] (5.16)

2. Velocity constraint \[ |\dot{q}_{ji}(t)| \leq \dot{q}_{ji}^{\max} \] (5.17)

3. Acceleration constraint \[ |\ddot{q}_{ji}(t)| \leq \ddot{q}_{ji}^{\max} \] (5.18)

4. Jerk constraint \[ |\dddot{q}_{ji}(t)| \leq \dddot{q}_{ji}^{\max} \] (5.19)

5. Force/torque constraint \[ |u(t)| \leq u_{ji}^{\max} \text{ for } j=1,2,\ldots,n \text{ and } i=1,2,\ldots,m-1 \] (5.20)

6. Payload constraint \[ F_{g_{\text{min}}} \leq F_k \leq F_{g_{\text{max}}} \] (5.21)

where, ‘n’ represents the robot joints and ‘m’ represents the knots used to construct the trajectories, \( F_k \) is the grasping forces of fingers (\( F_1 \) or \( F_2 \)) and \( F_{g_{\text{min}}} \) and \( F_{g_{\text{max}}} \) are minimum and maximum grasping forces of the fingers.
Here the kinematic and dynamic models and trajectory representation are same as in the stage 9.

5.6.1.1 Formulation of the Grasping Forces in the Gripper

Usually, industrial robots use two-finger grippers for grasping purpose. So a two-finger gripper is considered here to grasp the object (Figure 5.35). Grasping can be defined as the capability of a mechanical end-effector to establish a contact between its fingers and object. Grasp configurations are achieved such that a static equilibrium exists between the grasping forces by the fingers on the object. The grasping forces can be determined based on the characteristics of the object, such as its weight and shape. However, in most manipulator tasks, inertia forces are also considered depending on the specifications of the motion trajectories (Saramago et al 2002). This stage uses an approach for computing the grasping forces as a function of the inertia forces as well.
Let $F_1$ and $F_2$ be the grasping forces, which are exerted by the fingers. Usually they are not equal, since the contact points A and B are not generally located in same relative position on the two fingers. Similarly, friction can be evaluated at the points A and B through the coefficients $\mu_1$ and $\mu_2$. In Figure 5.35 the grasping configuration of an object with respect to the fingers gives the angles $\psi_1$ and $\psi_2$, which strongly depend on the orientation of the fingers, position of the contact points and shape of the object and fingers. These angles can differ from each other similar to $F_1$ and $F_2$. $r_A$ and $r_B$ represents the distances of A and B, respectively from the squeezing line. $W = m_{\text{object}} g$ is the weight vector of the object and it is oriented with an angle $\psi_w$ with respect to the perpendicular axis to the plane yz and $G_{xyz}$ is a suitable frame fixed on the grasped object as shown in Figure 5.35. $N$ is external torque acting on the object and it includes the inertial actions due to the manipulator movement.
The static equilibrium of a grasped object can be expressed along the directions of the contact and squeezing lines, as outlined by Saramago et al. (2002), in term of forces as

\[
F_1 \cos \psi_1 - F_2 \cos \psi_2 + \mu_1 \frac{F_1 \sin \psi_1 - \mu_2 F_2 \sin \psi_2}{m_{\text{object}}} (g \cos \psi_w \sin \phi_w + a_y) = 0
\]

\[
-F_1 \sin \psi_1 - F_2 \sin \psi_2 + \mu_1 F_1 \cos \psi_1 + \mu_2 F_2 \cos \psi_2 - m_{\text{object}} (g \cos \psi_w \cos \phi_w + a_z) = 0
\]

(5.22)

and in term of torque as

\[
r_A F_1 (\sin \psi_1 - \mu_1 \cos \psi_1) - r_B F_2 (\sin \psi_2 - \mu_2 \cos \psi_2) - N - r_G m_{\text{object}} (g + a_y) \sin \phi_w = 0
\]

(5.23)

where, the acceleration components \(a_y\) and \(a_z\) of the centre point on the manipulator extremity can be computed as

\[
\begin{bmatrix}
a_x \\
a_y \\
a_z \\
1
\end{bmatrix} = \left( \sum_{i=1}^n \left( \frac{\partial T_0^i}{\partial q_j} \right) + \sum_{j=1}^m \left( \frac{\partial \hat{T}_0^j}{\partial q_i} \hat{q}_j \right) \right) \begin{bmatrix}
r_{Gx} \\
r_{Gy} \\
r_{Gz} \\
1
\end{bmatrix}
\]

(5.24)

The distance of the gravity centre of the grasped object is indicated through vector \(r_G\) with components \(r_{Gx}, r_{Gy}, r_{Gz}\) and \(T_0^i\) is the homogeneous transformation matrix. The corresponding velocity components of the gravity centre point can be calculated as

\[
\begin{bmatrix}
v_x \\
v_y \\
v_z \\
1
\end{bmatrix} = \left( \sum_{i=1}^n \left( \frac{\partial T_0^i}{\partial q_j} \right) \right) \begin{bmatrix}
r_{Gx} \\
r_{Gy} \\
r_{Gz} \\
1
\end{bmatrix}
\]

(5.25)
The payload constraints are expressed in terms of feasible range of grasping forces $F_1$ and $F_2$.

5.6.2 Numerical Example

In this stage, DE and NSGA-II are used for the optimal trajectory planning of the PUMA560 robot manipulator in material handling operation. It is considered that initial and final joint velocities and accelerations are null. So $q_i = q_f = q_{i_0} = q_{f_0} = 0$ for all robot joints. A standard six degrees of freedom PUMA 560 robot is considered here, whose geometrical and inertial arm parameters are listed in Table 4.11. The transfer must be done without violating bounce on kinematic and dynamic performances given in Table 4.12. Transfer between initial and final postures is executed in a smooth way without violating imposed constraints. 40-knot points are considered. The dimensions of the design variable vector is $n(m+2)$. So there are 252 variables in this problem. The aim is to obtain an optimal trajectory of the end-effectors. The initial and final trajectory points of the end-effectors: $P_0 = [0.30\text{rd}, 0.50\text{rd}, -0.20\text{rd}, -0.05\text{rd}, 0.05\text{rd}, 0.10\text{rd}]$ and $P_m = [0.51\text{rd}, -0.42\text{rd}, 0.58\text{rd}, 0.638\text{rd}, 0.835\text{rd}, 0.20\text{rd}]$. The grasped object mass (payload) is $m=1\text{kg}$. The payload constraints are expressed in terms of feasible range for the grasping forces given by $F_{g_{\text{min}}} = 0$ and $F_{g_{\text{max}}} = 60\text{N}$. The coefficients associated with dissipation force (Equation 4.26) are adopted as $f_f(N\text{m}) = [0.058, 0.058, 0.058, 0.056, 0.056, 0.056]$ and $f_d (\text{N}\text{m}\text{s}) = [0.0005, 0.0005, 0.000472, 0.000382, 0.000382, 0.000382]$. 

5.6.2.1 NSGA-II Parameters

The following are the values of the parameters of NSGA-II technique that have been used to obtain the best optimal results: Variable
type = Real variable, Population size = 100, Crossover probability = 0.6, Real-parameter mutation probability = 0.01, Real-parameter SBX parameter = 10, Real-parameter mutation parameter = 100, Total number of generations = 100.

5.6.2.2 Differential Evolution Parameters

The following are the values of the parameters of DE technique that have been used to obtain the best optimal results: Strategy = DE/rand/1/bin, crossover constant CR = 0.9, number of population NP = 500, \( F_s = 0.5 \) and total number of generations = 100.

5.6.3 Results and Discussion

From the proposed NSGA-II and DE, the optimal displacement \((Q - \text{rad})\), velocity \((V - \text{rad/s})\) and acceleration \((W - \text{rad/s}^2)\) of all the robot joints are shown in Figures 5.36 and 5.37. From Figures 5.36 and 5.37, it is observed that the resulting trajectories are smooth and the robot joints displacement, velocity and acceleration are within their limiting values.

Table 5.17 shows the optimum results obtained from various techniques for comparison. The computational time to find the multicriterion cost function \((f_c)\) in a HP computer (with configuration of 640MB DDR RAM, 40 GB HDD, Pentium 4 Processor) by NSGA-II and DE are 12 and 7 respectively. From Table 5.17, we can note that the values of \(z_2, z_4\) and \(z_5\) obtained from DE are lesser than those of NSGA-II. So the \(f_c\) value from DE is lesser than that of NSGA-II. Also the computational time for finding \(f_c\) in DE is lesser than that of NSGA-II. So it is concluded that DE technique is faster (less computational time) and gives better results (lower \(f_c\) value) than NSGA-II.
Figure 5.36 Displacement, velocity and acceleration of optimal trajectories by NSGA-II
Figure 5.37 Displacement, velocity and acceleration of optimal trajectories by DE
Table 5.17 Comparisons of Results

<table>
<thead>
<tr>
<th>Optimisation Techniques</th>
<th>$z_1$ (sec)</th>
<th>$z_2$</th>
<th>$z_3$</th>
<th>$z_4$</th>
<th>$z_5$</th>
<th>$f_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>6.50</td>
<td>1159.749</td>
<td>0.004968</td>
<td>17.97366</td>
<td>40.05306</td>
<td>1.2</td>
</tr>
<tr>
<td>DE</td>
<td>5.90</td>
<td>1006.499</td>
<td>0.004968</td>
<td>4.784057</td>
<td>16.49029</td>
<td>0.57</td>
</tr>
</tbody>
</table>

Figure 5.38 shows the result histories of DE and NSGA-II techniques. DE converges quickly and also shows its superior nature by giving better results than NSGA-II.

5.6.4 Limitations

In this stage, cubic B-spline curve is used to represent the trajectory. But NURBS curve has lot of advantages over B-spline curve, as found in the literature review Chapter. So in the next stage, multi-objective trajectory planning using NURBS curves is considered.
This chapter presented a stage-by-stage approach to develop an optimization procedure based on evolutionary algorithms namely NSGA-II, DE and MODE for solving the trajectory planning problem of intelligent robot manipulators (Adeptone XL robot, PUMA560 robot, STANFORD robot) with the prevalence of fixed, moving and oscillating obstacles using cubic B-spline curves. Six objective functions (transfer time, mean average of actuators efforts and power, penalty for collision free motion, singularity avoidance, joint jerks and joint accelerations) have been considered. This chapter also presented an interesting study about effect of number of knots of B-spline curve on trajectory planning of industrial robots in the presence of fixed, moving and oscillating obstacles. DE and MODE techniques give better results than NSGA-II, SQP and SUMT in majority of cases. Next chapter presents a stage-by-stage approach for solving the trajectory planning problem using cubic NURBS curves.