Journal of Engineering Science and Technology
Vol. 2, No. 1 (2007) 32-54
© School of Engineering, Taylor's University College

OPTIMAL TRAJECTORY PLANNING OF MANIPULATORS: A REVIEW

ATEF A. ATA
Mechatronics Engineering Department, Faculty of Engineering,
International Islamic University Malaysia
P. O. Box 10, 50728, Kuala Lumpur, Malaysia
Email: atef@iiu.edu.my

Abstract

Optimal motion planning is very important to the operation of robot manipulators. Its main target is the generation of a trajectory from start to goal that satisfies objectives, such as minimizing path traveling distance or time interval, lowest energy consumption or obstacle avoidance and satisfying the robot's kinematics and dynamics. Review, discussion and analysis of optimization techniques to find the optimal trajectory either in Cartesian space or joint space are presented and investigated. Optimal trajectory selection approaches such as kinematics and dynamics techniques with various constraints are presented and explained. Although the kinematics approach is simple and straight forward, it will experience some problems in implementation because of lack of Inertia and torque constraints. The application of Genetic Algorithms to find the optimal trajectory of manipulators especially in the obstacle avoidance is also highlighted. Combining the Genetic Algorithms with other classical optimization methods proves to have better performance as a hybrid optimization technique.

Keywords: Optimal trajectory, Minimum time, Energy, Obstacle avoidance, Genetic Algorithms, Generalized Pattern Search

1. Introduction

Trajectory planning is one of the fundamental issues in the design and development of manipulators. The trajectory is normally determined to satisfy a certain criterion optimally. Optimal performance means different things to different people such as minimum time, minimum kinetic energy, and obstacle avoidance. Optimisation is

Nomenclatures

t_f	Travelling time s
t_b	Blending time s
T	Total number of different parameter combinations
Q_i	Number of different values that parameter; i can attain with fine enough sampling
N_{par}	Number of different parameters
$b_i(s)$	Piecewise-cubic-B-spline basis function
J	Motor inertia
V_a	Motor volt, and
K_e	Back *emf* constant
K_t	Torque constant
$K.E_{loss}$	Losses in Kinetic Energy
Greek Symbols	
$\theta(t)$	Angular displacement of the joint rad
$\dot{\theta}$	Angular velocityof the joint rad/s
$\ddot{\theta}$	Angular acceleration of the joint rad/s^2
θ_m	Motor's angular displacement
γ_n	Nonnegative scalar that minimizes the function in the direction of the gradient.
v_i	Are specified path vertices
$J(v)$	Cost function
Subscripts	
0	initial condition
f	Final condition

normally performed in the presence of constraints. In addition to the dynamic system equations acting as constraints, there may be bounds on the inputs as well as constraints on some of the states. The constraints are of two types: The system constraints imposed by the manipulator itself and task constraints given by the task. The problem is how to calculate feasible trajectories from a given path with simultaneous utilization of the maximal capabilities of the manipulator.

A time-trajectory may be generated in joint space or Cartesian space. In joint space trajectories, trajectories are specified for each independent joint. The actual Cartesian position of the end-effector is only known at the initial and goal position. On the other hand, Cartesian trajectories are easy to specify and tip motion of the manipulator is completely specified. Joint motion is obtained via the velocity Jacobian. Since trajectories are not generated in joint space, care must be taken that the trajectories do not pass, or close to, singularities. Trajectories are chosen to fairly smooth to allow reasonable time for the manipulator to accelerate and to decelerate. The trajectories in both joint and Cartesian space schemes can be chosen in a number of ways. The three most common are:

1. Polynomials in time, cubic polynomial, and splines in time (Fig. 1).

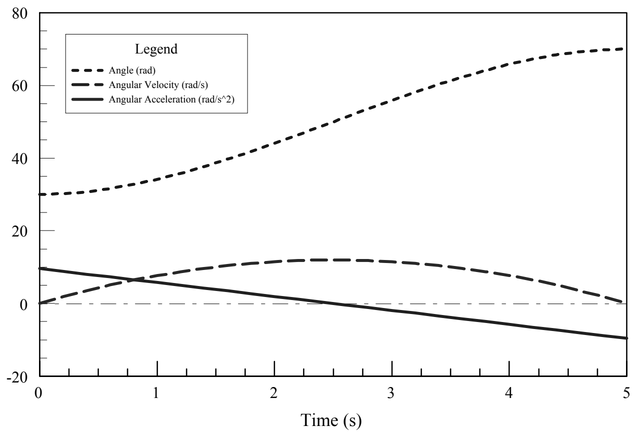

Fig.1. Joint Position, Velocity and Acceleration Using Polynomial Profile.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad (1)$$

$$a_0 = \theta_0, \quad a = 0, \quad a_2 = \frac{3}{t_f^{\,2}}(\theta_f - \theta_0), \qquad a_3 = \frac{2}{t_f^{\,3}}(\theta_f - \theta_0)$$

2. Linear interpolation with smoothing and linear function with parabolic blends LSPB (Fig. 2).

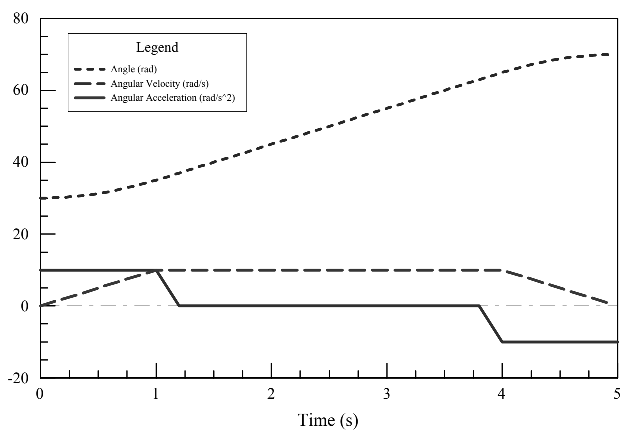

Fig. 2. Joint position, velocity and acceleration using LSPB

$$\theta_i^{\,d}(t)\begin{cases}\theta_0+\dfrac{\ddot{\theta}}{2}t^2 & 0\le t\le t_b\\ \dfrac{\theta_f+\theta_0-\dot{\theta}\times t_f}{2} & t_b<t\le t_f-t_b\\ \theta_f-\dfrac{\ddot{\theta}\times t^2{}_f}{2}+\ddot{\theta}\times t_f t-\dfrac{\ddot{\theta}\times t^2{}_f}{2} & t_f-t_b<t\le t_f\end{cases} \tag{2}$$

$$t_b=\frac{\theta_0-\theta_f+\dot{\theta}\times t_f}{\dot{\theta}}$$

3. Optimal control methods like the shooting method

The objective of this paper is to investigate the different optimization techniques for trajectory selection of manipulators. This paper is organized as follows: Section (2) presents the normal optimization techniques and Section (3) focuses on minimum time trajectory selection including kinematics, dynamics and online approaches. Section (4) emphasizes on minimum kinetic energy while section (5) gives a briefing on Genetic Algorithms and investigates on obstacle avoidance trajectory planning.

2. Minimum Seeking Algorithms

Optimization can be defined as the process of adjusting the inputs to or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result." The process or function is called cost function, objective function, fitness function or evaluation function. Since cost is something to be minimized, optimization becomes minimization. But by just adding minus sign in front of fitness function, the function can be maximized if maximization of function is required [1]. Searching all possible function values for the minimum cost lies at the heart of all optimization routines. Normally fitness function has many peaks, valleys and ridges meaning that there are so many local minima. An optimization algorithm will try to find minimum attitude throughout the area. In finding that, some methods will stack in local minimum instead of finding global minimum. Some optimization methods have better strength in finding global minimum and some methods are better for local search. The following are some of the minimum seeking algorithms [1].

2.1 Exhaustive search

This approach requires checking an extremely large but finite solution space with the number of combinations different parameter values given by

$$T=\sum_{i=1}^{Npar}Q_i \tag{3}$$

Exhaustive search will not get stuck in local minima and work for either continuous or discontinuous parameters. It is only practical for a small number of parameters in a limited search space since it takes an extremely long time to reach global minima.

2.2 Analytical optimization

An extremum is found by setting the first derivative of a cost function to zero or undefined and solving for the parameter value. If the second derivative is greater than zero, the extremum is a minimum and conversely, if the second derivative is less than zero, the extremum is a maximum. It quickly finds a single minimum, but requires a search scheme to find the global minimum. But continuous functions with analytical derivatives are required. It works well when the minimum is nearby, but cannot deal with cliffs or boundaries, where the gradient cannot be calculated.

In the eighteenth century, Lagrange introduced a technique for incorporating the quality constraints into the cost function. The method now is known as Lagrange multiplies finds the extrema of a function f(x,y,…) with constraints g_m (x,y,…) = 0 by finding the extrema of the new function.

The disadvantage of the calculus approach and the numerical technique is that they often find a local minimum rather than the global minimum. And also they do not work well with discrete parameters.

2.3 Nedler-Mead downhill simplex method

Linear programming concerns the minimization of a linear function of many variables subject to constraints that they are linear equations and equalities. In 1947, Dantzig introduced the Simplex method, which has been the workforce for solving linear programming problems. It does not require calculation of the derivatives.

A simplex is the most elementary geometrical figure that can be formed in dimension, n and has (n +1) sides (e.g., a triangle in two-dimensional space). Each iteration generates a new vertex for the simplex. If this new point is better than at least one of the existing vertices, it replaces the worst vertex. In this way the diameter of the simplex gets smaller and the algorithm stops when the diameter reaches a specific tolerance.

Although it is not popular for it is speed of processing, it has certain robustness that makes it attractive. It can be combined with the random search algorithm to find global minimum. In 1965, Box extended the simplex method and called it the complex method, which stands for constrained simplex method. This approach allows the addition of inequality constraints, uses up to 2n vertices, and expands the polyhedron at each normal reflection.

2.4 Optimisation based on line minimisation

The largest category of optimization methods fall under the general title of successive line optimization method. An algorithm begins at some random point on the cost surface, chooses a direction to move, then moves in that direction until the cost function begins to increase. Next, the procedure is repeated in another direction.

Devising a sensible direction to move is critical to algorithm convergence and has spawned a variety of approaches.

The steepest descent algorithm starts at an arbitrary point on the cost surface and minimizes along the direction of the gradient.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} - \gamma_n \nabla f\left(x_n, y_n\right) \tag{4}$$

By definition, the new gradient formed at current iteration is orthogonal to the previous gradient. If the valley is narrow means that ratio of maximum to minimum eigenvalue is large, then this algorithm bounces from side to side for many iterations before reaching the bottom.

2.5 Genetic algorithms

Genetic Algorithms are search procedures based on the mechanics of natural genetics and natural selection. They combine an artificial survival of the fittest with genetic operators abstracted from nature to form a surprisingly robust search mechanism that is suitable for a variety of search problems [2].

The operation of GAs is similar to that of natural species. Organisms in nature live for a certain period and produce offspring in that period of time. In each successive generation the fittest survive and the weaker ones die off. The offspring produced by their parents inherit some, but not all, of the characteristics of their parents. They may be fitter or weaker. Not only their parents determine their characteristics, other external effects also may influence them. GA borrowed the idea from the nature: “Survival of the Fittest” within a population of competing potential solutions. The generation-by-generation evolution of the population of potential solutions results in a convergence towards the “best” possible solution.

3. Minimum Time Trajectory

The determination of time optimal path for manipulators is an important problem in robot trajectory planning. This problem attracted great interest from many researchers especially in the last two decades. One major motivation for the use of robot is to increase productivity by increasing speeds of robots. A more feasible approach is to minimize the motion time needed to perform a given task subject to physical constraint. It is worth mentioning that, in industrial applications the most widely used criterion is the minimum time criterion where the robot travels through the given path in the shortest possible time. It is needed to reduce the production time and to increase the profit. So, it is of general interest to perform the path motion as fast as possible. In such cases, the manipulator performance is the limiting factor. Consequently, we have to minimize the execution time, i.e., find the maximal feasible path velocity concerning the task requirement and manipulator limitations. On the other hand, there are some cases where the minimum time criteria are not advisable when a nice smooth path for the motion is needed. When using minimum time criteria we get a spiky trajectory which can cause unwanted shock and vibrations. For example when handling non-rigid materials with intrusive (needle) grippers, a quick change in the trajectory can move or shake off the handled object from the gripper.

The cubic order polynomial functions for generating smooth robotic trajectories are well known in the literature. The most popular type of interpolation is algebraic splines. An *n* th order algebraic spline consists of piecewise-continuous *n* th order algebraic polynomials which have continuous derivatives up to order (*n-2*) or less depending on the details of the formulation. The cubic spline is the lowest degree approximation to a function obtained when the magnitude and the derivatives at the two end points are specified [3].

Actually, there are two main approaches in dealing with this problem namely kinematics and dynamics approaches. In the kinematical approach one is to assume that a desired joint trajectory is specified in terms of the trajectory end points and a set of knots between the end points. The trajectory through the knots is then generated to satisfy limits on the joint velocities, accelerations and jerks and some other performance indices. An alternative approach is to combine the dynamics or load characteristics, torque limits on the robotic arm and trajectory performance indices to obtain the optimal trajectories.

3.1 Kinematics approach

In purely kinematics approaches, the objective is to find the sequence of time interval that minimizes the total time spent on moving between two points in space. The main feature of this approach is that the dynamics of the arm is not considered at all, so that the constraints consist of sets of bounds on position, velocity and acceleration. These bounds are imposed by the weakest configuration of the arm, so that the motion in other areas of the work space is sub-optimal. In addition, most of the algorithms require a set of knot points which defines a path. This path is not necessarily the best possible one [4].The optimal time trajectory selection of robot manipulator can be summarized as follows:

Suppose that the three dimensional Cartesian path of the end-effector is represented with uniform cubic-spline polynomials. The parameterization for motion in the x spatial dimension is given by:

$$x(s) = \sum_{i=-1}^{n+1} v_i b_i(s) \qquad 0 \le s \le n \tag{5}$$

For any s, only four of the basis function $b_i(s)$ are nonzero. The optimal path planning problem can be written as one of minimizing the total path traversal time [5]

$$J(v) = t_f = \int_0^n \frac{ds}{\dot{s}} \tag{6}$$

subjected to some constraints in terms of either kinematics configuration or dynamics equations of motion.

An important variation of minimum-time trajectory can be obtained by leaving the final time t_f unspecified and seeking the fastest trajectory between points q_0 and q_1 with a given constant acceleration a. That is sometimes called *Bang-Bang* trajectory since the optimal solution is achieved with the acceleration at its maximum value + a until an appropriate switching time t_s at which time is abruptly switches to its minimum value - a (maximum deceleration) from t_s to t_f. For rest to rest maneuver, symmetry consideration would suggest that the switching time t_s is just half t_f. For nonzero initial and/or final velocities, the solution is more complicated [6].

In 1990, Wang and Horng [7] applied the Flexible Polyhedron search algorithm to generate the near-minimum-time path with velocity, acceleration, and jerk constraints on every intermediate point. The interpolated path via B-spline function is determined via a unique set of virtual knots so that the robot path can pass every intermediate knot and satisfy the boundary conditions. A point-to-point joint trajectory planning scheme which generates a smooth joint trajectory for a given joint velocity profile was introduced by Won *et al*., [8]. The given joint velocity profile is approximated in terms of the Nth partial sum of Fourier series. Coefficients of the Fourier series are computed from the inverse kinematics solution of the given point-to-point task. The proposed method has been applied for the cases of minimum energy and minimum time criteria. A conceptually simple approach to the trajectory planning problem is to generate a joint-space trajectory based on interpolation of a sequence of desired joint angles was proposed [9]. The number of knots points is chosen along the desired Cartesian trajectory in such away that this number is a trade off between the exactness and computational expense. The Cartesian knots are then mapped into joint knots using inverse kinematics. Finally an analytic interpolating curve is fitted to the joint knots. This curve provides the trajectory tracker with joint angles and derivatives at the controller rates. Sakamoto and Kawamura [10] applied the variation method to optimize the trajectory planning for robot manipulator. The reference is considered in joint space and is expressed as a B-spline curve for less memory requirement for the reference path. The performance index includes the trajectory position error and the acceleration term of each joint with weight coefficients. A new design method was proposed to set up the weight coefficients from which the characteristics of the optimized trajectory can be determined. Their approach was verified both theoretically and experimentally through SCARA robot. Minimum time point-to-point motion planning of robot manipulator was investigated considering kinematical constraints on speed and acceleration [11]. A polynomial trajectory is chosen and the parameters of the trajectory for minimal time motion were found. Although the simple expression previously obtained is extended to actuator constraints, the obtained results are not applicable directly to an industrial robot. An algorithm for determining the optimal placement of a robotic manipulator within a work cell for minimum time coordinated motion was proposed [12]. The algorithm uses a simple principle of coordinated motion to estimate the time of a joint interpolated motion. Specifically, the coordinated motion profile is limited by the slowest axis. Two and six degrees of freedom manipulators examples were also presented for verification. Although many researchers applied the B-spline interpolation , a B-spline does not go through the knots of the path though they are suitable for on-line computation. In case of cubic-spline trajectory interpolation, joint angles are well calculated beforehand for certain points on the surface as such the ned-effector strictly remains in contact with the surface at those particular points while manoeuvring [13]. A new approach to constrained time-efficient and smooth trajectory using cubic-spline was presented by the same authors [14]. A suitable objective function to combine the minimization of travelling time with smoothness of the robot trajectory is selected. The effects of weights in the objective function and the control knots on smoothness, time optimality and path deviation error are chosen mainly by simulation with a SCARA robot.

In the last decade, some researchers applied the genetic algorithm to search for the minimum time trajectory for manipulators. A genetic algorithm is a stochastic search algorithm which can optimize nonlinear functions using the mechanics of natural genetics and natural selection [2]. A genetic search algorithm to schedule the time interval between each pair of adjacent knots such that the total travelling time is minimized subject to physical constraints on joint velocities, accelerations and jerks was addressed [15]. Modified heuristic crossover and a scaled and normed performance measure are applied to the genetic algorithmic searching procedure. Experiments with different combinations of crossover rates and mutation rates were carried out and corresponding results outweigh the constrained minimum time were obtained from the trial and error polyhedron search method. A general formulation for path modelling using cubic B-splines was presented [16]. Two different strategies are used: the first one corresponds to a given initial and final points and the second one deals with the task imposes the trajectory to contain a given number of points. An algorithm was given for each case and simple numerical examples were also illustrated. Yano and Tooda [17] applied a genetic algorithm to solve the position and movement of the end-effector of a two joints robot arm. The objective functions were defined in both Cartesian and joint spaces and then combined to optimize the robot trajectory. Optimum solutions with smooth trajectories and minimal joint rotation were obtained.

3.2 Dynamics approach

In 1985, a general solution for the minimum time path of robots was investigated [4]. Their solution incorporates the full dynamics of movements and actuator constraints and involves joint-space tessellation, dynamic time-scaling algorithm and graph search. They discovered that the optimal paths are not straight lines but rather curve in joint space that utilizes the dynamics of the arm and gravity to help in moving the arm faster to its destination. Another approach for a robot arm with actuator constraints was proposed [3]. The approach combines the control theory and the exhaustive search approaches to yield faster response. The method aims at parameterising the path in the Cartesian space, then, given a path using control theory to determine the minimum time trajectory subject to the actuator torque constraints. Finally searching among all possible paths to find the minimum time path. The perturbation trajectory improved algorithm (PITA), which can generate the joint positions, velocities and torques under very general torque constraints was introduced [18]. The algorithm starts with a non-optimal trajectory which meets all the required torque constraints, and perturbs the trajectory in such away as to always decrease the traversal time. The torque constraints may be expressed in terms of quantities related to torque rather than the torque itself. The proposed method is applied to the first three joints of the Bendix PACS Arm.

Schoenwald *et al.* [19] analyzed a minimum-time control scheme for a two-link flexible robot. An offline optimization routine determines a minimum-time, straight-line tip trajectory which stays within the torque constraints of the motors and ends with no variational transient. An efficient finite-element model was utilized in the optimization to approximate the flexible arm dynamics. The results indicate that combination of model based and error-driven control strategies achieves a closer tracking of the desired trajectory. A computational technique for minimum time trajectories which considers limitations to link movements due to design constraints were taken into consideration was addressed [20]. Numerical examples based on a two-link planar robot arm shows the feasibility of the proposed technique. Another approach which is applicable to both rigid robots and flexible joint robots is to use polynomials approximation to obtain parametric optimization [21]. An optimality result on the structure of the time optimal solution was presented showing that the minimum time solution for flexible joint robots has similar properties as rigid solution, in the sense that at least one torque is at the limit.

Vincent [22] presented an inverse dynamic-based dynamic programming (IDBDP) for optimal point-to-point robot trajectory planning. Compared with the conventional dynamic programming, this method eliminates the interpolation requirement and requires only inverse dynamics computations. Thus the requirement to integrate equations of motion is thus avoided. The reliability and efficiency of the IDBDP have been verified via computer simulation.

A Genetic-based Algorithm for the minimum time trajectory planning of articulated robotic manipulator was introduced [23]. The planning procedure is performed in the Cartesian space and includes all physical constraints and non-linear dynamics. Kim *et al.* [24] suggested a minimum-time trajectory planning method along with tracking control scheme for manipulators. The problem was treated in two levels, finding the minimum time trajectory by optimizing cubic polynomial joint profile using the Evolution Strategy (ES). The second level is tuning the sliding mode controller parameters for the manipulators to track precisely the trajectories that were found in the previous step. A global optimization approach based on hybrid genetic/interval algorithm to obtain minimum time cubic-spline trajectory subjected to torque constraints was introduced by Bianco and Guarino [25]. The feasibility of their solution has been proved by comparison with alternative optimization technique.

The variational approach by Agrawal and Xu [26] represents a global optimum path planning scheme for redundant space robotic arm. Two optimum path planning problems are considered: first, given the end-effector trajectory, find the optimum trajectories of the joints and second, given the terminal conditions of the end-effector, find the optimum trajectories for the end-effector and the joints. This formulation leads to a system of differential and algebraic equations and a set of terminal conditions. A numerical scheme is presented for forward integration, an iterative shooting method is used to satisfy the terminal conditions.

While several time-optimal trajectory planning techniques have been developed for continuous non-linear systems, there has been little discussion on the subject for discrete nonlinear systems. Furukawa [27] proposed a technique to search for time suboptimal trajectory for discrete non-linear systems. In his technique, the control inputs with respect to time are partitioned into piecewise continuous functions. The piecewise constant functions and the time step interval, which is used in the discretisation of the system, are then searched by a general-purpose non-linear programming optimization method. Two examples of time sub-optimal trajectory planning were demonstrated for verification.

Most commonly, researchers in an early attempt to solve the trajectory selection problem have linearised the dynamics in order to apply standard techniques of linear optimal control theory to the solution [28]. The derived sub-optimal control inputs were expressed in terms of switching curves based on the linearised equations of motion. However, this method is effective only when the system motion is confined to a small region near the terminal configuration where the linearity assumption is valid [27]. Unfortunately, the trajectory planning can then become very complicated due to inter-joint nonlinearities and the couples characteristics of the robot manipulator. Also the exact dynamics of a robot arm can only be obtained by identification. However, for a class of industrial robots with large reduction (gear) ratios, the coupled nonlinear dynamics are not significant.

3.3 On-line trajectory selection

The on-line algorithms for continuous path (CP) in robotics applications appear in operations such as arc welding, flame cutting, routing, etc. Cartesian space trajectory planning has the advantage of being straightforward concept. It is easy to perceive the desired end-effector configuration in Cartesian space. The actual physical constraints are the applied torques/forces at the joints, as the control of manipulator is done in the joint space. Due to the high nonlinear mechanical structure of manipulator, it is very difficult to convert torque/force constraints in the joint space into corresponding velocity and acceleration bounds in Cartesian space. To execute Cartesian space motion it also requires computing the transformation between joint and Cartesian coordinates in real time which adds high computational complexity.

Chang *et al.* [29] derived an on-Line algorithm that will generate approximate joint path for any curve in Cartesian space provided proper knot points are selected. The error between the desired and approximate paths is required to be within some prescribed tolerance. Two on-line interpolation schemes Simple Quartic Spline Interpolation (SQSI) and Modified Quartic Spline Interpolation (MQSI) with quartic spline functions for least-squared error fit have been applied and the algorithm seems reasonable. Bestaoui [30] presented a new joint reference trajectories taking into account the manipulator dynamics were calculated to be more adapted to the situation detected by the position and velocity sensors. This motion generation method uses the resolution of a minimum time optimization. The constraints are first expressed in terms of joint velocities and torque bounds and are then transformed into joint velocities, accelerations and jerk limitations. In 1997, an on-line algorithm for computing robot manipulator's trajectory in Joint space with velocity and acceleration constraints was proposed [31]. This algorithm is based on optimizing the minimum possible time for velocity and acceleration constraints using cubic-spline. Simulation results for the proposed algorithm have been presented. Xu *et al.* [32] presented an on-line method for planning Continuous-Path motion trajectory with minimum time in Cartesian space. The time interval required to compute the path was divided into m segments, and the coefficients of polynomial at each segment can be obtained in recurrence form. The travelling path could be specified by a group of parameters in Cartesian coordinates.

For trajectory planning in Cartesian space, the transformation from Cartesian to joint coordinates in real time is required. As control of robot motion is done at joint level, the computational complexities involved in trajectory planning and coordinate transformations have hindered the on-line implementation of Cartesian-based path planning. Due to recent advances in computer, on-line trajectory selection has received recent interest. It should be noticed that On-line algorithms can be done either under kinematics or dynamics constraints.

The method is based on the hydrolysis of tributyrin by lipase, and titrating the butyric acids produced with 0.05 NaOH in distilled water [7]. The alkali consumption was registered as a function of time at pH 7.0 using an auto titrator (Metrohm 794 Basic Titrino, Swaziland).

4. Minimum Energy Consumption

In some fields the time is not the most important, but the consumed energy is considered as the primal criteria. This can be the case where the amount of available energy is scarce. Many applications require a motion control system to move an object (load) starting from rest to a specified final angle θ_f (or distance) in a specified final time t_f, and to return the load to rest at that time. This kind of motion is suitable to robotics applications and is known as rest to rest maneuvering. A motion control system using a DC motor to perform this task can be designed for optimal performance by properly selecting the following [33]:

1. The commanded motion profile.
2. The coupling ratio connecting the motor with the load.
3. The motor and amplifier.

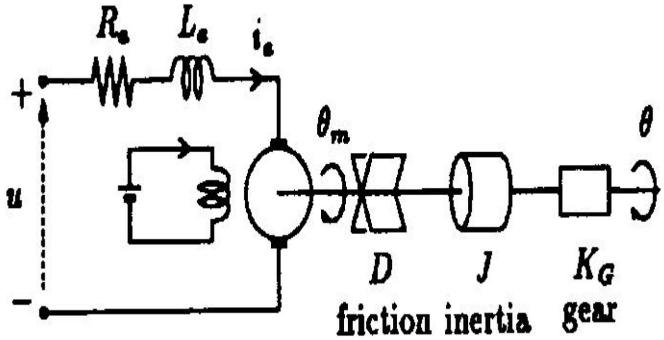

Fig.3. A Field Controlled DC Motor [19].

The energy consumption E per cycle is the energy dissipated in the motor resistance R during the time $0 < t < t_f$. In most applications, the motion cycle is repeated continuously, and so the energy consumption, while perhaps small for one cycle, can be very significant over many cycles. The electric power which is applied to the dc servo motor as shown in Figure (1) can be calculated by the following integral:

$$K.E_{loss} = \frac{R_a J^2 K_G{}^2}{K_t{}^2} \int_0^{t_f} \ddot{\theta}^2 dt \qquad (7)$$

The damping coefficient can be assumed small which is often true in practice. From equation (7), it is understood that the integral of the square of acceleration becomes a part of loss in armature resistance and this term needs to be minimized.

Many researchers applied the minimum energy consumption criterion for optimal trajectory selection of robot manipulators. Chou and Song [34] proposed a technique based on studying the geometric work of a regular open chained manipulator and applied it to the path planning for minimum energy consumption. The path planning for minimum energy consumption is generated by dynamic programming method and the results were compared with the map of zero geometric work. The presented method can also be used to arrange the pick and place positions for minimum travel energy consumption. Shugen [35] presented a real time algorithm to generate quasi-minimum energy point-to-point control of robotic manipulators. The main objective is to overcome the difficult and computationally very intensive algorithms at that time even when the dynamic equations and parameters of the manipulators are precisely known. While not producing exact minimum-energy solutions, the algorithm can be easily used on-line to approximate minimum-energy control. Numerical experiments were executed to show the effectiveness of the proposed techniques.

Field [36] presented an iterative dynamic programming technique to plan minimum-energy consumption trajectories for robotic manipulators considering joint actuator and time constraints. The algorithm has an inherent parallel structure, allowing for reduced computation time on parallel architecture computers. No limiting assumptions are made about the performance index or function to be optimized. So variety of complex functions and constraints can be handled. The algorithm has been verified experimentally by planning and execution of a minimum energy consumption path for a Reis V15 Industrial manipulator. Zlajpah [37] presented a solution to the optimal trajectory planning problem subject to constraints given by joint torques and velocities and task requirement. He has followed the same approach [17] except that he added more constrainable factors like bounded joint velocities and task requirements. The main result is trajectory planning algorithm which enables the calculation of the optimal path taking into considerations the given requirements as well as the dynamics of the robot. Some examples have been given to illustrate the capabilities of the given trajectory planning algorithm. The variational approach within the B-spline function is introduced for minimization of the consumed electrical energy [38]. The inverse matrix of Jacobian calculation is not required and the minimal trajectory error is guaranteed while minimizing the consumed electrical energy. The secondary criterion named sub-performance index and an auto tuned weight parameter are proposed. The sub-performance index accomplishes the minimization of the input electrical energy and the weight parameter allows optimal balance of each performance index.

Saramago and Steffen [39] considered a solution to the problem of moving a robot manipulator with minimum cost along a specified geometric path. The optimum traveling time as well as the minimum mechanical energy of the actuator were considered together to build a multi-objective function and the obtained results depend on the associated weighting factor. The optimization problem is subject to physical constraints and the mathematical model takes into account the manipulator nonlinear coupled equations of motion. Numerical results for manipulator arms with three and six degrees of freedom have been presented for verification.

A comparison between the optimal time and the constant kinetic energy criterion for continuous path trajectory planning was illustrated [40]. The dynamical equations of motion were determined in Rieman space in such away that it would provide constant kinetic energy. In order to select the suitable criteria, they applied their approach for two robots; the linear portal robot and the polar robot using LabVIEW software. They discovered that the constant kinetic energy trajectory had taken longer time to accomplish than time optimal motion. However, the difference was not so big. When the difference is so small, the constant kinetic energy motion is better because it uses less energy and the trajectory is much smoother (better for handling delicate objects). Ata *et al.*, [41] investigated the problem of optimal joint trajectory for a rigid flexible manipulator during constrained motion. Three trajectories, cubic polynomial, sine and Gaussian profiles are proposed and checked based on the minimum enrgy consumption represented by equation (7). These three profiles represent generic pick and place operation and possess start and ending values but they differ in the rate of increase of velocity. They concluded that Gauusian profile gives the lowest energy followed by sine and cubic polynomial profiles.

The genetic algorithms can be applied also to humanoid robot to find the minimum consumed energy during specific tasks. Capi *et al.*, [42] proposed a genetic algorithm gait synthesis method to generate angle trajectories for walking and gait up-stairs. He applied a Radial Basis Neural Network for the real applications. Lei and Su [43] proposed an adaptive algorithm based on the genetic algorithm for stable, minimum energy trajectory. The proposed trajectory can be applied also to other activities like walking and overcoming obstacles. Another interesting applicationfor genetic algorithm for underwater manipulator was introduced by Shintaku [44]. He proposed a simple technique based on genetic algorithm to approximate the joint trajectory as a polynomial. The genetic algorithm searches for the optimal coefficients of the polynomial to minimize the fitness of the objective function.

5. Obstacle Avoidance

In many today's manufacturing applications such as welding, spray painting, etc. a robot manipulator must traverse with its hand tip a desired curve, while its body avoids collisions with the obstacles in the environment. The problem is very difficult, while little articles from the bibliography face it in this specific form. The purpose of those studies is to plan a sequence of motion for robots. However, the obstacle avoidance considering full manipulator dynamics is pretty difficult because of non-linearity of manipulator dynamics and existence of obstacles. Since infinitely many collision-free paths are usually possible, the criterion used to select one of them is most often minimum-distance, that is, the path with the smallest arc-length . In many cases this solution is not practical, since minimum-distance paths are straight lines with sharp corners near the obstacle intrusions. Robot motion along such paths would require the velocity of the tip to stop or slow down considerably at each corner. This leads to the need for some criterion other than minimum-distance to be used for selection of the collision-free paths.

To solve this problem, Gilbert and Johnson [45] parameterized the motion of each joint of the manipulator arm as a function of time. The joint motions were then optimized while ensuring that at each instant, obstacles are avoided and no actuator input needed for the motion exceeds its bound. Bobrow [5] presented a path planning technique that utilizes the minimum-time rather than the minimum-distance to produce time-optimal manipulator motion in a workspace containing obstacles. The full nonlinear equations of motion are used in conjunction with the actuator limitations to produce optimal trajectories. The Cartesian path of the manipulator is represented with B-spline polynomials and the shape of this path is varied in a manner that minimizes the travel time. Obstacle avoidance constraints are included in the problem through the use of distance function. The example presented showed a reduction in the time required for typical motions. The approach used here is the same [3] in such away that the geometric path of the manipulator is presented as a set of parameterized interpolation functions, and the parameters are varied in such away that the path traversal time required for the motion is minimized. A method which describes the joint trajectories of a manipulator with B-spline curves and optimizes the values of control points using the complex method has been suggested [46]. This method incorporates the full dynamic as well as kinematics constraints for collision-free maneuvering and does not require evaluation of the gradient in the optimization.

It also utilizes the B-spline curve's uniqueness and continuity and is applicable to manipulators with arbitrary degree of freedom. This method achieves the generation of the trajectory which takes priority of the specified constraints into account. For verification, the method was applied to three degrees of freedom manipulator and the results show its effectiveness

In the last two decades, genetic algorithms (*GAs*) have been successfully applied to solve the obstacle avoidance trajectory planning problem of robot manipulators. Genetic Algorithms (GAs) are robust optimization tools based on natural evolution for complex search problems which can cope with discontinuities, non-linearities and even noisy. In genetic algorithms (*GAs*), the space for searching must be translated into a coding string. The string characterizes individuals which mean spatial paths of a manipulator for the current problem. It is very important how the paths are coded, because success in genetic algorithm depends on the coding. The selection of individual (spatial path) as a member of initial group is essential as shown in Figure (4). The initial individuals depend on the variety of final solutions. The initial individuals had better have every possible path including even collision path. The selection of genetic operators namely, mutation and crossover, in genetic algorithms is also very important. Each genetic operator should be connected with actual meaning of the problem. The mutation is a basic operation to get variety of individuals. This genetic operator should be adopted to assure large space of searching. This operator consists on several actions namely, commuting the type of the joint, modifying the link length and change the joint variable. The cross operator also is an essential operator in geneting algorithms. If the individuals are represented as sequence of control points of B-spline and we consider a gap between two control point $\widehat{\theta}^{i}{}_{k}$ and $\widehat{\theta}^{i}{}_{k+1}$ as a gab of the spring, this operator may produce a better solution by connecting two other parts of the path (Yamamoto, *et al.,* 1994). The algorithm gives fairly well multiple solutions to avoid the obstacles. The fairly good solutions in the sense of travelling time are used as initial feasible solutions of the gradient method.

A series of researchers have applied the GA technique to solve the collision free trajectory planning of manipulators. Shiller and Dubowsky [48] proposed a method to solve optimal trajectory with collision-free problem. A small number of candidates of optimal trajectory in a discretised workspace was searched for. Then the trajectory is improved using the gradient method. It is easy predicted that it takes too much time all over the workspace. Yamamoto *et al.* [47] presented a method for time optimal collision free trajectory planning based on both the genetic algorithm and the gradient method using the same approach [17, 46]. The proposed method is basically an iteratively improving one based on a gradient method. Their approach applies two global methods (a genetic algorithm or an exact cell decomposition method) to search multiple initial feasible spatial paths for the gradient method. Next, the gradient method searches time optimal solution locally within the multiple initial feasible solutions. The spatial path is represented only by the control points of B-spline. Therefore, if a boundary condition and dynamical constraints of the problem are specified, the minimum travelling time can be estimated by only the control points.

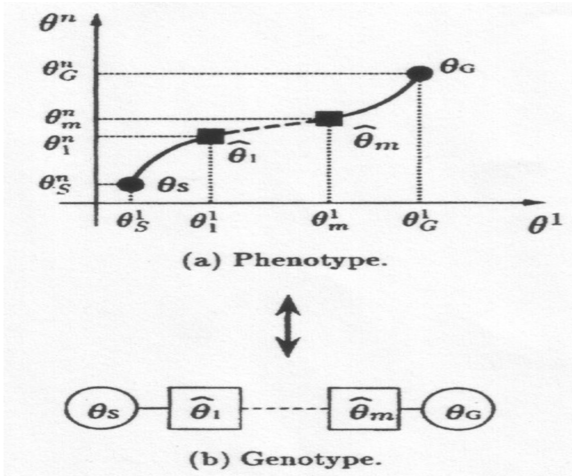

Fig.4. Representation of phenotype and genotype for spatial path $\theta(s)$ by B-spline control points [47]

Rana and Zalzala [49] developed a method to plan a near time-optimal, collision free motion in the case of multi-arm manipulators. The planning is carried out in the joint space and the path is represented as a string of via points connected through cubic spline. Doyle and Jones [50] proposed a path-planning scheme that uses a *GA* to search the manipulator configuration space for optimum path. The *GA* generates good path solutions but it is not sufficiently robust. Lee and Lee [51] proposed a genetic trajectory planning of a robot manipulator producing the optimum trajectory between two points. They investigated the proper genetic trajectory parameterization and have developed an efficient scheme for the implementation of genetic trajectory planner. Pires *et al.* [52] presented a method based on a GA adopting the direct kinematics. The optimal manipulator is the one that minimizes both the path trajectory length and the ripple in the time evolution without any collision with the obstacles in the workspace. Simulation results involving different robot structures and trajectories in the workspace have been carried out to validate the proposed approach.

Nearchou and Aspragathos applied various heuristic techniques to solve the obstacle avoidance problem based on known notions and structures from computational geometry and computer graphics (53-55). A new technique based on the concept of convex-hulls [53] was proposed. The technique guarantees detection of an imminent collision between a robot's link and an obstacle by constructing the convex-hull of the following points. An algorithm for continuous motion on a desired trajectory has been presented [54]. The algorithm computes in each step of the robot movement a small change in the vector of the robot's joints displacement so that to satisfy three criteria: the direction of the motion, the proximity to the desired curve, and the obstacle avoidance. In 1998, they continued by presenting an algorithm for trajectory generation under maximum allowed deviation [55]. The algorithm combines raster techniques and GAs so that the tip traverses a trajectory within a maximum allowed position deviation from the desired curve. Pack *et al.*, [56] proposed a genetic algorithm based method to search for feasible trajectory to avoid point obstacles in the configuration space. Their approach can also be extended to n-dimensional space. Tian and Collins [57] proposed a genetic algorithm using a floating point representation to search for optimal end-effector trajectory for a redundant manipulator. An evaluation function based on multiple criteria such as total displacement of all joints and the uniformity of Cartesian and joint space velocities was introduced. To verify their approach, simulations are carried out in free space and in a workspace with obstacles. While in 2004, they extended their work by developing a novel genetic algorithm for point obstacles avoidance trajectory using a cubic interpolation function [58]. Their algorithm searches several interior points between the starting and target points and uses the hermit cubic interpolation to construct the path. Merchan-Cruz and Morris [59] extended the application of GA for a collision free of a system of two manipulators using potential field approach. In this approach each manipulators is considered as a moving obstacle by the other and collision is avoided. The GA carries parallel optimization to find the best configuration for collision free as well as minimizing the error to their respective goals. Actually, there are two main advantages for genetic algorithm approach. First, genetic algorithm based methods seldom require a priori knowledge of the problem. Furthermore, they do not fall into local optima and proceed toward global optima. However, they have difficulty in handling equality constraints of trajectory boundary conditions because they use probabilistic transition rules to find a solution [47, 52]. Also there are some drawbacks of GA in terms of inconsistency of the solution even if we started from the same population and time consuming. To solve some of these drawbacks, Wetter and Wright [60] suggested that combining genetic algorithm with any method of coordinated search technique will improve the behaviour of the system. Ata and Myo [61] also proved that using a generalized pattern search (GPS) which consists of genetic algorithm as well as direct search will give excellent results in terms of trajectory tracking with minimum tracking error as shown in Figures (5 and 6). This proposed technique which proved to be excellent is being applied for obstacle avoidance with and without artificial potential field.

Ata and Myo [62] continued their work towards applying the Generalized Pattern search for obstacle avoidance either online or offline and the algorithms proved to be very successful.

5. Conclusions

A review of the optimization techniques for the trajectory selection of robot is introduced. Kinematics approaches give interesting results but when comes to reality inertia and torque constraints make it difficult to implement. Dynamical approaches prove to be more realistic in terms of incorporating torque constraints and joints physical limits. Sometimes nonliearities and unmodelled dynamics are still problems in fulfilling the full dynamics model specially for flexible manipulators. It looks like the genetic algorithms are going to have more applications for the coming years. However, a GA may sometimes have difficulties to converge in an optimum solution. This is mainly due to some phenomena such as the deceptive problem and the problem of premature convergence. The GA has some drawbacks in terms of consistency of the solution and time consumed.

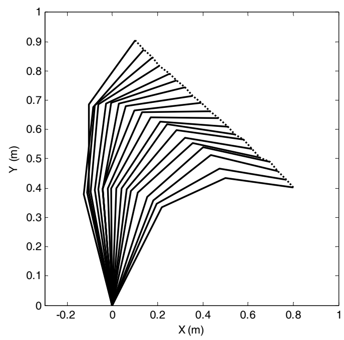

Fig. 5a. End-effector using GA alone

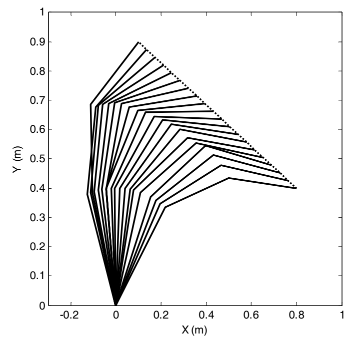

Fig. 6a. End-effector trajectory using GPS

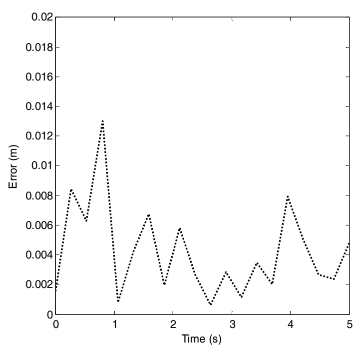

Fig. 5b. Tracking error using GA alone

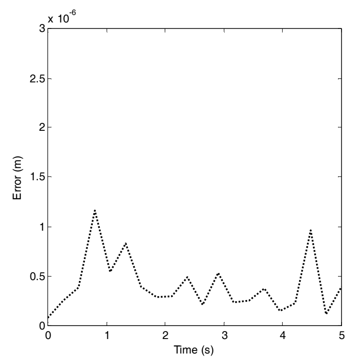

Fig. 6b. Tracking error using GPS

References

1. Haupt, R.L. & Haupt, S.E. (1998). *Practical Genetic Algorithms.* USA: Wiley-Interscience Publication.
2. Goldberg, D. E. (1989). *Genetic Algorithm in Search Optimization, and Machine Learning,* Addison Wesley.
3. Rajan,V. (1985). Minimum time trajectory planning. *IEEE Proceedings of the International Conference on Robotics and Automation*, St. Louis, 2, 759-764.

4. Sahar, G. & Hollerbach, J. (1985). Planning minimum-time trajectories for robot arms. *IEEE Proceedings of the International Conference on Robotics and Automation*, St. Louis, Vol. II, 751-758.
5. Bobrow, J. E. (1988). Optimal robot path planning using the minimum time criterion. *IEEE Transaction of Robotics and Automation*, 43, 443-450.
6. Spong, M. S. & Vidyasagar, M. (1989). *Robot Dynamics and Control*. New York: John Wiley&Sons, Inc.
7. Wang, C-H. & J-G. Horng,(1990). Constrained minimum-time path planning for robot manipulators via virtual knots of the cubic B-spline functions. *IEEE Transactions on Automatic Control*, 35 (5), 573-577.
8. Won, J-H., Choi, B-W & J Chung, M. (1991). Smooth joint trajectory planning for a point-to-point task. *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS 91*, Japan, 1299-1303.
9. Simon, D. (1992). Optimal robot motions for repetitive tasks. *Proceedings of the 31st IEEE Conference on Decision and Control*, Tucson, AZ, Vol. 4, 3130 -3134.
10. Sakamoto, K. & kawamura, A. (1993). Trajectory planning using optimum Solution of variational problem. *Power Conversion Conference*, Yokohama, Japan, 666-671.
11. Pledel, P. (1995). Actuator constraints in point to point motion planning of manipulators. *Proceedings of the 34th IEEE Conference on Decision and Control*, Vol. 2, 1009-1010.
12. Feddema, J. T. (1996). Kinematically optimum robot placement for minimum time coordinated motion. *Proceedings of the 13th IEEE International Conference on Robotics and Automation*, Vol. 4, 3395-3400.
13. Cao, B., Dodds, G. I., & Irwin, G. W. (1994). Time optimal and smooth constrained path planning for robot manipulators, *Proceedings of the International Conference on Robotics and Automation*, Minneapolis, Vol. 43, No. 2, pp 1853-1858.
14. Cao, B., Dodds,G. I. & Irwin, J. W. (1997). Constrained-time efficient and smooth cubic spline trajectory generation for Industrial robots. *IEEE Transactions on Control Theory and Applications*, 144 (5).
15. Tse, K-M & Wang, C-H. (1998). Evolutionary optimization of cubic polynomial joint trajectories for industrial Robots. *Proceedings of the IEEE International Conference on Man, System and Cybernetics*, San Diego,Vol. 4, 3272-3276.
16. Saramago S. F. P. & Junior,V. S. (1999). Using B-splines for optimal trajectory planning. *Science and Engineering Journal,* 1 (17), 106-114.
17. Yano, F. & Tooda, Y. (1999). Preferable movement of a multi-joint robot arm using genetic algorithm, *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision*., Vol. 3837, 80-88.
18. Shin, G. & McKay, N. D. (1986). Minimum-time trajectory planning for industrial robots with general torque constraints. *IEEE International Conference on Robotics and Automation*, San Francisco, Vol. 3, 412-417.
19. Schoenwald, D. A., Feddema, J. T. & Segalman, D. J. (1991). Minimum-time trajectory control of a two-link flexible robotic arm. *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, Vol. 3, 2114-2122.

20. Dissanayake, M. W, Goh, C. J. & Phan-Thien, N. (1991). Time-optimal trajectories for robot manipulators. *Robotica*, 9 (2), 131-138.
21. Dahl, O. (1993). Path constrained motion optimization for rigid and flexible joint robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, Georgia, Vol. 2, 223-229.
22. Vincent, Y. (1995). Inverse dynamic-based programming method for optimal point-to-point trajectory planning of robotic manipulators. *International Journal of Systems Science*, 26 (2), 181-195.
23. Chan K. K. & Zalzala, A. M. S. (1993). Genetic-based minimum-time trajectory planning of articulated manipulators with torque constraints. IEE Colloquium on *Genetic Algorithms for Control Systems Engineering*,London, 4/1 -4/3.
24. Kim, K-W, Kim, H-S, Choi, Y-K. & Park, J-H. (1997). Optimization of cubic polynomial joint trajectories and sliding mode controllers for robots using evolution strategy. *Proceedings of the 23rd International Conference on Industrial Electronics, Control and Instrumentation IECON 97*, Vol. 3, 1444 -1447.
25. Bianco, L. & Guarino, C. (2002). Minimum-time trajectory planning of manipulators under dynamic constraints. *International Journal of Control*, 75 (13), 967-980.
26. Agrawal, O. P. & Xu,Y. (1991). Global optimum path planning for a redundant space robot. Tech. Report, CMU-RI-TR-91-15, Robotics Institute, Carnegie Melon University. http,//www.ri.cmu.edu/pubs/pub_259.html.
27. Furukawa, T. (2002). Time-subminimal trajectory planning for discrete non-linear systems. www.acfr.usyd.edu.au/people/academic/tfurukawa/papers/E02002-1.pdf.
28. Kahn, M. E. & Roth, B. (1971). The near-minimum-time control of open-loop articulated Kinematic Chains. *Transactions of the ASME, Journal of Dynamic Systems, Measurements and Control*, 93 (3), 164-172.
29. Chang, Y., Lee, T. & Lui., C. (1988). On-line Cartesian path trajectory planning for robot manipulators. *IEEE International Conference on Robotics and Automation,* Philadelphia,Vol. 1, 62-67.
30. Bestaoui, Y. (1992). On-line reference trajectory definition with joint torque and velocity constraints. *International Journal of Robotics research*, 11 (1), 75-85.
31. Bazaz, S. A. & Tondo, B. (1997). On-line computing of a robotic manipulator joint trajectory with velocity and acceleration constraints. *Proceedings of the IEEE International Symposium on Assembly and Task Planning, ISATP'97* , California, 1-6.
32. Xu, X-R , Chung, W-J & Choi.,Y-H. (1999). A Method for on-line trajectory planning of robot manipulators in Cartesian space. *In Proceedings of Computational Intelligence in Robotics and Automation, CIRA '99*, Monterey, Vol. 1, 41-46.
33. Palm, W. J., (2001). *Modeling, Analysis and Control of Dynamic Systems.* New York: John Wily & Sons Inc.
34. Chou, L.-S & song, S-M (1990). Geometric work of manipulators and path planning based on minimum energy consumptions, *Transactions of ASME, Flexible mechanism, Dynamics and Robot Trajectories*, 24, 319-326.

35. Shugen, M. (1995). Real-time algorithm for quasi-minimum energy control of robotic manipulators. *Proceedings of IEEE Int. Conf. on Industrial Electronics, Control, and Instrumentation,* Orlando*,* Vol. 2, 1324-1329.
36. Field, G. (1995). Iterative dynamic programming, an approach to minimum energy trajectory planning for robotic manipulators. Minneapolis, *Proceedings of IEEE Int. Conf. on Robotics and Automation*, Vol. 3, 2755-2760.
37. Zajpah, L. (1996). On time optimal path control of manipulators with bounded joint velocities and torques. *Proceedings of IEEE Int. Conf. on Robotics and Automation,* Minneapolis.
38. Hirakawa A. & Kawamura, A. (1997).Trajectory planning of redundant manipulators for minimum energy consumption without matrix inversion. *Proceedings of the IEEE International Conference on Robotics and Automation*, New Mexico, 2415-2420.
39. Saramago, S. F. P. & Junior, V. S. (1998). Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system. *Mechanism and Machine Theory*, 33 (7), 883-894.
40. Zoller, Z. & Zentai, P. (1999). Constant kinetic energy robot trajectory. *Periodica Polytecnia Ser. Mech. Eng*., 43(2), 213-228.
41. Ata, A. A., Salami, M. J. & Johar, H. (2003). Optimum joint profile for constrained motion of a planar rigid-flexible manipulator. *Proceedings of the 19th International Conference on CAD/CAM Robotic and Factories of the Future CARs &FOF*, 22-24 July Kuala Lumpur, Malaysia. Vol. 1, 51-60.
42. Capi, G. Nasu, Y., Barolli, L., Mitobe, K. & Takeda, K. (2001). Application of genetic algorithm for biped robot gait synthesis optimization during walking and going up-stairs. *Advanced robotics*, 15(6), 675-694.
43. Lei, X-S, & Su, J-B. (2004). Running trajectory generation for humanoid robot. *Proceedings of the International Conference on Machine Learning and Cybernetics*, Shanghai, China, Vol. 2, 1002-1007.
44. Shinkatu, E. (1999). Minimum energy trajectory for an underwater manipulator and its simple planning method by using genetic algorithm. *Advanced Robotics*, 13, 115-138.
45. Gilbert E. G. & Johnson, D. W. (1985). Distance function and their applications to robot path planning in the presence of obstacles. *Transactions of IEEE Journal of Robotics and Automation*, Vol. RA-1, 21 -30.
46. Ozaki, H. & Lin, C-J. (1996). Optimal b-spline trajectory generation for collision-free movements of a manipulator under dynamic constraints. *Proceedings of the IEEE International Conference on Robotics and Automation,* Japan, Vol. 4, No. 2, 3592-3597.
47. Yamamoto, M., Isshiki, Y. & Mohri, A. (1994). Collision free minimum time trajectory planning for manipulators using global search and gradient method. *Proceedings of IEEE/Rsj/Gi International Conference on Intelligent Robots and Systems, IROS'94*, Munich,Vol. 3, 2184-2191.
48. Shiller, Z. & Dubowsky, S. (1991). On computing global time-optimal motions of robotic manipulators in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 7(6), 785-797.

49. Rana, A. & Zalzala, A. (1996). An evolutionary planner for near time-optimal collision-free motion of multi-arm robotic manipulators. Exeter, *Proceedings of International Conference on Control*, 29-35.
50. Doyle, A. B. & Jones, D. I. (1996). Robot path planning with genetic algorithm. *Proceedings of 2nd Portuguese Conference on Automatic Control*, 312-318.
51. Lee, Y. D. & Lee, B. H. (1997). Genetic trajectory planner for a manipulator with acceleration parameterization. *Journal of Universal Computer Science*, 3 (9), 1056-1073.
52. Pires, E. J. S., Machado,J. A. T. & Oleviera, P. B. M. (2001). An evolutionary approach to robot structure and trajectory optimization. *ICAR'01-10th International Conference on Advanced Robotics,* 22-25 Aug, Budapest, Hungary, 333-338.
53. Nearchou, A. & Aspragathos, N. (1994). A collision-detection scheme based on Covex-Hulls concept for generating kinematically feasible robot trajectories. *Advances in Robot Kinematics and Computational Geometry*, (J. Lenarcic and B. Ravani editors), Kluwer Academic Publishers, 477-484.
54. Nearchou, A. & Aspragathos, N. (1996). Collision-free continuous path control of manipulators using genetic algorithms. *Journal of Systems Engineering*, 6, 22-32.
55. Nearchou, A. & Aspragathos, N. (1998). Collision-free continuous trajectory generation using raster scanning and genetic algorithms, *Journal of Intelligent Robotic Systems*, 16, 351-377.
56. Pack, D., Toussant, G., & Haupt, R. (1996). Robot trajectory planning using a genetic algorithm. *SPIE,* 2824, 171-182.
57. Tian, L. & Collins, C. (2003). Motion planning for redundant manipulators using a floating point genetic algorithm. *Journal of Intelligent and Robotic Systems, Theory and Applications*, 38(3-4), 297-312.
58. Tian, L. & Collins, C. (2004). An effective robot trajectory planning method using genetic algorithm, *Journal of Mechatronics*, 14, 455-470.
59. Merchan-Cruz, E. A., & Morris, A. S. (2004). GA based trajectory planner for robot manipulators sharing common workspace. *Proceedings of the IASTED International conference on Applied Simulation and Modelling*, 96-101.
60. Wetter, M. & Wright, J. (2003). Comparison of a generalized pattern search and a genetic algorithm optimization method. *Proceedings of the Eighth International IBPSA Conference, Eindhoven,* Netherlands, 11-14 August, 1401-1408.
61. Ata, A. A. & Myo, T. (2005). Optimal point-to point trajectory tracking of redundant manipulators using generalized pattern search, *International Journal of Advanced Robotic Systems*, In Press.
62. Ata, A. A. & Myo, T. R. (2006). Collision-free trajectory planning for manipulators using generalized pattern search. *International Journal of Simulation Modelling*, 5(4), 145-154.