

Multiobjective optimization-based trajectory planning for Laser 3D scanner Robots

Yumeng Huang¹, Guangyu Liu^{1,2*}, Wujia Yu^{1†}, Shen Yu^{1†}

^{1*}School of Automation, HangzhouDianzi University, Baiyang Street,
Hanzghou City, 310018, Zhejiang Province, China.

*Corresponding author(s). E-mail(s): g.liu@hdu.edu.cn;
Contributing authors: myh77954@sina.com; yuwujia@hdu.edu.cn;
nwcd@163.com;

[†]These authors contributed equally to this work.

Abstract

In our industrial material defect detecting processes, the multi criteria is considered in two-level motion planning structure. Firstly, the feed speed of the end-effector should be programmed in optimal time for satisfying the requirement of high efficiency. Secondly, the planned joint velocities and acceleration are characterized by high-order derivatives to guarantee smooth motion, taking into account the kinematic constraints. Last but not least, energy consumption of the robot's movement is a focus during designing trajectories. The Pareto optimal method is applied to solve the trajectory planning problem. The results of the experiments suggest that the Pareto approach can realize effective multi-objective optimization and deliver a group of Pareto solutions for decision makers. Based on the actual requirements, suitable Pareto-optimal trajectory can be achieved and the practical operation of the industrial robot is good.

Keywords: Trajectory planning, Multi-objective optimization, B-spline interpolation, Industrial robots, Kinematic constraints⁴

1 Introduction

In recent years, robots are widely used in automobil industrial manufacturing such as palletizing, hangding, welding and defect detection. Defects detection for machined part surface texture image can obviously enhance the entire efficiency for the automatic

manufacturing. Due to the impact of mass production, the detection time of a single product in the field of automobile manufacturing has strict requirements. In order to compress the inspection time of complex products, the implementation of automated inspection solutions has become more urgent. One of the methods is to install the scanning probe at the end of the robot to make it a robotic system, using the industrial robot to drive the scanner to reach the corresponding position according to the data determined by online or offline programming.

When robots execute feasible trajectories in these production environment, it is necessary to guarantee high accuracy and reatability and safety. As shown in Fig.1, for the 6-Dof manipulator, the robot achieve a fast, collision-free path point according to known environmental information. Then, according to the planned path, the optimal trajectory suitable for the robot handing tasks is planned. The scanning task means that the manipulator should move its end-effector from the initial position to the target position (material toolbox). The characteristics of the industrial manipulator are analyzed as follows in the planning process. 1) Collision-free and smooth trajectories. 2) Various constraints: in addition to task constraints, the manipulator also has joint physical constraints. 3) Different critrion such as jerk-bound. Specifically, in the process of handing operation, industrial robot needs to find the appropriate position to meet the task requirements of production environments such as minimizing operating time[1–3], energy optimization[4], obstacle avoidance planning[5], and jerk elimination[6]. Once the planning is not reasonable, it will not only bring low efficiency energy consumption and other results, but also may occur in the process of working collision wear, damage the dynamic performance of the manipulator. It is worth discussed how to find the relative position relationship between the task and the manipulator in accordance with the set task.

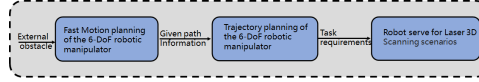


Fig. 1: Motion planning of industrial robot for material hanging scenarios

The trajectory planning of 6-DOF manipulator with multi criteria is a challenging research topic. Since the manipulator is a highly nonlinear and multivariable coupled dynamical system. Trajectory planning is generally transformed into a multi-constraint multi-objective nonlinear trajectory optimization problem. At present, trajectory optimization algorithms fulfill two essential roles in robot motion planning.

Firstly, as a trajectory replanning algorithm, an intuitive trajectory with collisions (such as a simple point-to-point straight path) can be transformed into a local optimal path without collisions and with low-cost. In such methods, trajectory optimization is invariant to reparametrization and these parameters are optimized in ways that include covariant Hamiltonian optimization[5, 7, 8] and derivative-free stochastic optimization[9]. The most important characteristic of optimization-based methods is that they take advantage of the fact that the physical (or simulation) environment changes are always continuous in actual production (or design) scenarios. Secondly, facing on complex industrial environment, optimal trajectory planning

mainly concentrates on the production of offline motions to execute a previously known task and a determined environment. Such methods is decoupled into two more tractable and simpler subproblems : 1) generate a sequence of points in robot configuration space, and 2) optimized trajectory operation parameters. Within a scientific literature on trajectory planning problems, it is potential to identify various optimization criteria; the most critical are the minimal execution time[10–12], the minimal energy[13] (or actuator effort) as well as minimal j bumps[14]. In this paper, we adopted decoupled trajectory optimization method and consider multi criteria for our robotic manipulators.

Various optimization algorithms and parametric curves have been thoroughly developed and employed in the relevant fields of industrial manipulators. For instance, in [15] a classification method was presented that classifies optimization strategies for yielding time-optimal trajectories in the position-velocity phase plane into three families: Dynamic programming [16], Numerical integration [17, 18], convex optimization [19]. In [20, 21], an energy minimized point-to-point trajectory is taken into account with upper limits set on the joint velocity and the amplitude of the control signal.

Taking into account only a single-objective function may not be appropriate to satisfy the diverse requirements of real-world applications. Numerous diverse technical criteria are defined to fulfil the requirements of task. It then becomes a constrained multi-objective optimization (CMOP) problem. CMOP for robot manipulators [22] is typically transformed into the unconstrained problem through the addition of a penalty function associated with constraints in the aforementioned research or the addition of linear weighting to objective function [23]. However, there exists internal conflicts among objectives, any modification of the Pareto optimum in an objective must result in a deterioration of a minimum of one of the other objectives. The optimal trade-off between such objectives can be described in terms of Pareto optimality [24].

At present, in the research methods of multi-objective optimization, the multi-objective evolutionary algorithms (MOEA) [25] generally exploit non-dominated ranking to give the decision maker some Pareto solution, instead of converting all the objectives into a mono-objective function. NSGA-II [26] ranks the solutions through non-dominated evaluation. MOEA/D [27] disaggregates a multi-objective optimization problem into a group of problems with various sampling weights as the sub-problems. As a result, this optimization technology is more preferred by researchers [28].

Our primary contribution is to exploit non-dominance ranking to give decision makers some trajectories of Pareto solutions for their selection of problem-specific solutions. In the present paper, the torque rate has been included as another target function, together with time and jerk, in order to realize a higher localization accuracy. We demonstrated the performance of NSGA-II and the Linear weighted methods in generating trajectories in our problems.

The rest of paper is structured as described below. Section 2 presents a description of multi-objective trajectory planning problem. Section 3 describes in detail the dynamical and kinematic analysis of a 6-DoF industrial robot and discusses the generic formulation of multi-degree B-splines. In Section 4, a mathematical model for time-energy jerk optimization is built, joint trajectory interpolation is carried out

with fifth-order B-spline functions, and two multi-objective optimization algorithms are employed for trajectory optimization. In Section 5, results of experiments and result analysis are presented for a 6DoF serial robot with involute joints in EC66 configuration. Lastly, in Section 6, some conclusions are presented.

2 Problem formulation

In the present paper, a 6-DoF manipulator was devised as a workpiece defect detecting platform to complete the task of material scanning as is shown in Fig. 2. The end effector of the robot is a 3D laser scanner, a beam of laser light emitted from the side head irradiates the workpiece, and after being diffusely reflected by the surface of the object, it receives the optical fiber back-transmitted by the object to obtain the geometric information of the measured object. In the process of moving 3D Laser scanner driven by the manipulator, the workpiece is fixed on operating table. The task demands that the manipulator passes through these specified trajectories. The application of a 6-DoF manipulator guarantees that the end-effector can achieve an arbitrary posture and complete the task. During this process, the manipulator should satisfy task constraints of the initial position $P_0 = (p_{0x}, p_{0y}, p_{0z})$ and target position $P = (p_x, p_y, p_z)$. The motion planning of our industrial manipulator includes two stages. The first stage is the no-collision path planning. The second stage is the multiobjective trajectory optimization.

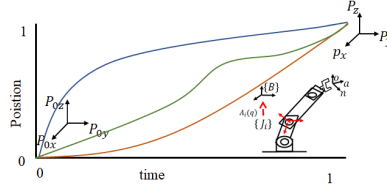


Fig. 2: Feasible Trajectories for a robotic manipulator

We consider the problem of generating a optimal trajectory, i.e. a time-parameterized function $\pi(t) : [t_0, t_f] \in \chi$. Let $\Pi(\chi, T) \in \mathbb{R}$ denote the set of all continuous functions $[t_0, t_f] \rightarrow \chi$ and $x_{init} \in \chi$ be the initial configuration of trajectory. The goal region is $x_{goal} \subseteq \chi$. All trajectory at time $t \in [t_0, t_f]$ is denoted as $x_{free}(t)$. Trajectory constraints are expressed as $D(q, \dot{q}, \ddot{q}, \dots)$. Under these assumptions, the optimal trajectory planning problems generally expressed as: given a 6 tuple $(\chi_{free}, x_{init}, X_{goal}, D, F, T)$, find a trajectory satisfied task requirements.

$$\begin{aligned}
 \arg \min \quad & F(\pi(t)) = F_1(\pi(t)), \dots, F_m(\pi(t)) \\
 & \pi(t_0) = x_{init} \quad \text{and} \quad \pi(t_f) \in x_{goal} \quad \forall t \in [0, T] \\
 s.t. \quad & \pi(t) \in \chi_{free} \quad \forall t \in [0, T] \\
 & D(\pi(t), \dot{\pi}(t), \ddot{\pi}(t) \dots)
 \end{aligned} \tag{1}$$

trajectory of the robot is planned in the joint space, the joint angle vector of the initial configuration is denoted by $q_0 = [\theta_{10}, \theta_{20}, \dots, \theta_{60}]$, the joint angle vector of the target configuration is denoted by $q_f = [\theta_{1f}, \theta_{2f}, \dots, \theta_{6f}]$, and the joint speed vector of the target configuration at moment t_f is defined by $\dot{q}_f = [\dot{\theta}_{1f}, \dot{\theta}_{2f}, \dots, \dot{\theta}_{6f}]$, time-parameterized function can be represented by $q(t)$. Then the multi-objective trajectory planning problem for the 3D Laser scanners robot of equation (1) can be stated as follows:

$$\begin{aligned} \arg \min \quad & F(q(t)) = F_1(q(t)), \dots, F_m(q(t)) \\ \text{s.t.} \quad & q(t_0) = q_0 \quad \text{and} \quad q(t_f) = q_f \\ & \dot{q}(0) = 0 \quad \text{and} \quad \dot{q}(t_f) = \dot{q}_f \end{aligned} \quad (2)$$

3 Preliminary

3.1 Kinematics and dynamics modeling

The application of trajectory planning begins with the creation of a mathematical model of robot to address dynamics and kinematics and to compute a viable joint configuration for the robot. The design structure of trajectory planning subsequently optimizes the robot joint configuration to identify the optimal configuration in terms of minimum jerk, energy consumption and time. The robot dynamics and kinematics model has been produced to allow analysis of the torques and forces on robot joints during robot movement. As an illustration, we have selected the EC66 industrial robot. EC66 is a lightweight collaborative robot having six rotating joints comprising three wrist joints, elbow, shoulder, base. The robot's dimensions and structure are presented in Fig. 3(a). Kinematic modelling first needs to build the robot's kinematic parameters and allocate joint frames. The Denavit-Hartenberg (D-H) approach[29] with modifications was employed to model the robot's kinematic parameters. From link base to link wrist, the frames have been assigned as shown in Fig.3(b). EC66 robot can be described by for quantities $\alpha_i, a_i, \theta_i, d_i$ for each link.

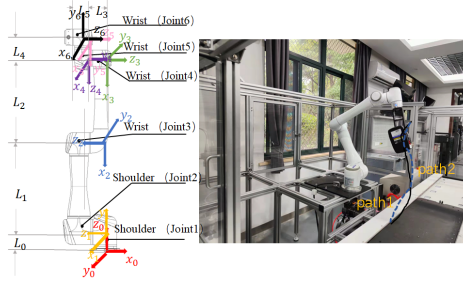


Fig. 3: (a)Physical packing -robot platform (b)Joint structure model of the 6-DoF manipulator

According to the Standard D-H parameters, a transformation matrix depicting the orientation together with position of the robot's end-effector relative to its base is shown in equation(2).

$${}^0T(\theta_1 \dots \theta_6) = {}^0_1T(\theta_1){}_2^1T(\theta_2) \dots {}^5_6T(\theta_6) \quad (3)$$

This transformation matrix depends on D-H parameters and is defined as follows:

$$T_-^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & L_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & L_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Supposing that the posture of the end-effector relative to the base is $[\mathbf{R} \ \mathbf{P}]$, where $\mathbf{P}$ denotes as the position coordinates $[P_x \ P_y \ P_z]^T$ and $\mathbf{R}$ is defined from a group of Euler angles $[n \ o \ a]^T$ designating the orientation. Then the kinematic equation of the manipulator is

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0_1T(\theta_1){}_2^1T(\theta_2) \dots {}^5_6T(\theta_6) \quad (5)$$

The methods of solving inverse kinematic equation is often used closed-form solution called algebraic[30].

For the majority of robots, the maximal available acceleration and velocity varies along the path due to the state-dependent dynamics and joint actuator limitations. For the robot manipulator, the dynamic equations can be derived through the application of the Natton-Euler or Langerand-Euler formulas in the joint space, in the following general form:

$$\tau(\ddot{\theta}, \dot{\theta}, \theta) = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) \quad (6)$$

in which $\tau \in \mathbb{R}^n$ represents the joint force vector; $M(\theta) \in \mathbb{R}^{n \times n}$ denotes inertia matrix; $C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times n}$ represents the Coriolis and centripetal force matrix; $G(\theta) \in \mathbb{R}^n$ denotes the gravity term vector. Given a trajectory point, $\theta, \ddot{\theta}$ and $\dot{\theta}$, we can find the required vector of joint torques, by Newton-Euler algorithm. The algorithm presented is based upon the method published by Luh, Walker, and Paul in [31].

3.2 Trajectory planning in joint space utilizing fifth-order B-splines

We recall here that a B-spline $p(u)$ of order $p = k + 1$ and degree k is represented in the so-called B-form. That is, as a linear combination of polynomials $N_{i,k}(t)$ of degree p , called base or blending functions, which is weighted by some coefficients d_i called control points. The curve is founded on a series of nodes u_i and the basis function is recursively defined through the De Boor formula[32].

$$q(u) = \sum_{i=0}^n d_i N_{i,k}(u) \quad (7)$$

The r -th order derivatives of the k -th degree B-spline in relation to u is

$$\begin{aligned} \frac{d^r q(u)}{du^r} &= \sum_{j=i-k}^{i-r} d_j, r N_{i,k-r}(u), u \in [u_i, u_{i+1}] \\ d_j, l &= \begin{cases} d_j, l = 0 \\ (k-l+1) \frac{d_{j+1,l-1} - d_{j,l-1}}{u_{j+k+1} - u_{j+1}}, l = 1, 2, \dots, r \end{cases} \end{aligned} \quad (8)$$

When $r = 1, 2$ and 3 , the derivatives stands for the vectors of the velocity $\dot{q}(t)$, acceleration $\ddot{q}(t)$ and jerk $\dddot{q}(t)$ of the robot joints, separately, namely

$$\begin{aligned} \dot{q}(t) &= \frac{dq(u)}{du} = \sum_{j=i-k}^{i-r} d_j, 1 N_{i,k-r}(u) \\ \ddot{q}(t) &= \frac{d^2 q(u)}{du^2} = \sum_{j=i-k}^{i-r} d_j, 2 N_{i,k-r}(u), u \in [u_i, u_{i+1}] \\ \ddot{q}(t) &= \frac{d^3 q(u)}{du^3} = \sum_{j=i-k}^{i-r} d_j, 3 N_{i,k-r}(u), u \in [u_i, u_{i+1}] \end{aligned} \quad (9)$$

In the present paper, we utilize $k = 5$. Thus, each of the interval will be a polynomial of degree 5, and it is ensured to be located in the convex hull of its 4 control point 6 control points $\{d_j, d_{j+1}, d_{j+2}, d_{j+3}, d_{j+4}, d_{j+5}\}$.

4 Multiobjective optimal trajectory planning

The problem shown as equation(1) is transformed to a problem that producing an energy-,jerk- and time- optimal path via a given set of waypoints for robot configurations, including constraints on joint positions, jerks, torques and velocities, while fulfilling the various dynamical and kinematic constraints relevant the robot as is shown in table 1.

According to task requirements, a no-collision path-time nodes of each joint (t, q_k) is generated by RRT* algorithm. Given the set points $q_k, k = 0, 1, \dots, n$ to be approximated for time instants u_k within a specified tolerance, it is possible to employ a quintic spline within minimum energy, smoothing and minimum time of equation (7), its control points are computed as a tradeoff between two or more suitable targets: 1.To find a well-fitted method for a specific via-points. 2.To achieve a trajectory that is as smooth as possible (acceleration/curvature as small as possible). 3.To acquire a trajectory with as short an execution time as possible. 4.To get a trajectory with as little energy as possible.

we have designed a motion trajectory for these six joints, utilizing a fifth-order B-spline curve as equation (8). Initial and final state control points for the packaging process are premised on the hypothesis that the robot is at a standstill initially and stops completely at the end of trajectory. The selection of control points can be carried out via solving the multi-objective optimization problem, aimed to make each sub-objective function as optimal as possible:

$$\begin{aligned}
\min \quad & F(x) = \{f_1(x), \dots, f_m(x)\} \\
& g_j(x) \leq 0, j = 1, \dots, J \\
s.t. \quad & h_k(x) = 0, k = 1, \dots, K \\
& x \in \Omega \subseteq \mathbb{R}^n
\end{aligned} \tag{10}$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an n -dimensional decision vector in the decision space Ω , and $f : \Omega \rightarrow \Theta \subseteq \mathbb{R}^m$ represents an objective vector comprising m objective functions, mapping Ω to the realizable objective space Θ , and $g_j(x)$ and J , separately, stand for the J -th inequality constraint together with the number of inequality constraint; and $h_k(x) = 0$ and K , separately denoting the K -th equality constraint together with the number of equality constraint. A solution is considered to be a viable solution when it meets all the constraints; otherwise, it is an unviable solution.

The total structure of trajectory planning method consists of 5 modules as shown in Fig.4:1. The geometric path that is created by the upper path planner is presented as a series of waypoints in Cartesian space, which stand for the continuous directions and positions of the manipulator's end effectors; 2. Quintic B-splines curves are employed for fitting the trajectory, where the trajectory is re-parameterized relative to time in order to create movement patterns and geometric paths designed to achieve robot jerk, acceleration and velocity of continuity; 3. Multi-objective optimization algorithm produces certain Pareto-optimal trajectories for the robot, through linking temporal information to geometric paths, considering several optimality criteria and some kinematic limitations; 4. The decision-makers choose a solution for the specific problem according to the Pareto solutions.

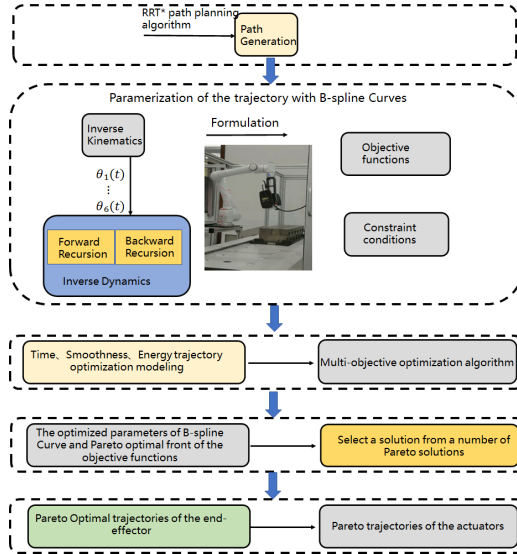


Fig. 4: Overview of the design structure

4.1 Mathematical modeling for multi-objective optimal trajectory planning

Firstly, minimum-time is an important trajectory planning techniques since they were closely connected to the requirements to raise productivity in industrial sector.

$$F_1(t) = T = \sum_{i=0}^{m-1} h_i = \sum_{i=0}^{m-1} (t_{i+1} - t_i) \quad (11)$$

where T is execution time to fulfill the assigned task.

The trajectory planner must produce a law of motion that satisfies the kinematic and dynamic constraints and it must be the case that none of the mechanical resonance patterns are excited. This can be accomplished through the manipulator producing a smooth trajectory, that is, the one with good continuity. Specifically, in order to keep the absolute value of the acceleration with a certain range, it is better to obtain a continuous trajectory of the joint jerk. High values of jerk means high change rate of acceleration. Frequent change of robots's acceleration requires the torque of joint motor as driving torque, which may wear out the robot structure and reducing of trajectory accuracy. Vibration created by a non-smooth trajectory damages the robot actuator and induces large errors when robot performs tasks for example material handling. In addition, trajectories with low jerk can be implemented more precisely and faster. The criteria for smooth trajectories are listed below:

$$F_2(t) = \sum_{i=0}^{m-1} \sqrt{\frac{1}{T} \int_0^T \ddot{q}_{ji}(t)^2 dt} \quad (12)$$

where $i = 0, 1, \dots, m-1$.

The power consumption is computed by a two-step process. Firstly, the power consumed by each joint in a given time interval i is located, and subsequently the total power consumption for that time interval is computed by summing up the power consumption of all joints, as illustrated in equation (14).

$$F_3(t) = W_j = \sum_{i=0}^{m-1} \left| \int_{t_0}^{t_f} |\tau_{ji}(t)| \dot{\theta}_{ji} dt \right| E = \sum_{j=1}^6 E_j \quad (13)$$

where $j = 1 \dots 6$, τ is torque of actuators. For the short time intervals, it can be presumed that the angular velocity together with torque are constant.

Thus, the problem of optimal trajectory planning can be expressed as

$$\begin{aligned} \min \quad & F(t) = \{F_1(t), F_2(t), F_3(t)\} \\ \text{s.t.} \quad & |\dot{q}_{ji}(t)| \leq V_{jmax} \\ & |\ddot{q}_{ji}(t)| \leq A_{jmax} \\ & |\dddot{q}_{ji}(t)| \leq J_{jmax} \end{aligned} \quad (14)$$

in which $i = 1, 2, \dots, n$, $q(t), \dot{q}(t), \ddot{q}(t), \dddot{q}(t)$ denote the displacement, velocity, acceleration and jerk of the j th joint. $Q_{jmax}, V_{jmax}, A_{jmax}, J_{jmax}$ denote the maximal value of displacement, velocity, acceleration as well as jerk.

The system is devised to select joint configurations that need the lowest consumption of energy, the smallest bumps and the shortest time. As the number of potential configurations is restricted to a few solutions; the performance of each configuration is investigated with multi-objective optimization algorithm in an iterative manner via the entire target path and a Pareto non-inferiority is selected.

4.2 Trajectory Optimization algorithm

The developed algorithm utilizes the NSGA-II and weighting method. The weighting method is one of the first basic methods that one thinks of if multiple objectives require optimization at the same time. In the weighting method, the problem (1) is transformed into problem

$$\begin{aligned} \min \quad & \sum_{i=1}^k \omega_i f_i(x) \\ \text{s.t.} \quad & x \in S \end{aligned} \quad (15)$$

where $\omega_i \geq 0$ for all $i = 1, \dots, k$ and typically, $\sum_{i=1}^k \omega_i = 1$. The solution of (12) can be proven to be weakly Pareto optimal and, furthermore, Pareto optimal if we have $\omega_i > 0$ for all $i = 1, \dots, k$ or if the solution is unique.

For our problem, we would like to find out how the weights can be employed to control the solution process when the weights change at various values. As a result, the optimal trajectory planning of robotic manipulator can be expressed as

$$\begin{aligned} \min \quad & \omega_1 F_1(t) + \omega_2 F_2(t) + \omega_3 F_3(t) \\ & |\dot{q}_{ji}(t)| \leq V_{jmax} \\ \text{s.t.} \quad & |\ddot{q}_{ji}(t)| \leq A_{jmax} \\ & |\dddot{q}_{ji}(t)| \leq J_{jmax} \end{aligned} \quad (16)$$

where $F_1(t), F_2(t), F_3(t)$ is defined in equation (12), (13), (14). The trajectory to address this optimization problem have to satisfy the conditions for interpolation of all channel points, together with the original and ultimate conditions for torque, acceleration jerk as well as velocity. In this method, we use GWO algorithm[33] to solve problem(17).

In addition to the weighting methods, The genetic algorithm based on non-domination for multiobjective optimization was presented by Deb et al[26]. NSGA-II algorithm is attend to find numerous Pareto optimal solutions in a multi-objective optimization problem with these characteristics: it utilizes the elite principle, it employs an explicit diversity preservation mechanism, and it stresses nondominated solutions.

5 Application of the 3D areal scanner robotic manipulator

5.1 Simulation environment

To confirm the validity of our framework, two experiments were performed in the environment of MATLAB 2021a operated in a computer with Intel (R) core (TM) i7-10510U CPU @ 1.80GHz, 16.00GB RAM (15.8GB available) and 64 bit operating system. The robot trajectory planning on the basis of NSGA-II and the weighting methods. A simulation environment for the EC66 robot can then be operated with the trajectory planning algorithm mentioned in above section. In this section, Elite EC66 robot is considered. The standard D-H method was employed to model the robot's kinematics and the constraint conditions have been illustrated in Table 1. The parameters from below that are useful for the kinematics analysis were taken into account.

Table 1: Standard DH parameters.

Link i	$\alpha_i(^{\circ})$	$a_i(m)$	$d_i(m)$	$\theta_i(^{\circ})$	Displacement Limit($^{\circ}$)	Velocity Limit($^{\circ}/s$)	Acceleration Limit($^{\circ}/s^2$)	Jerk Limit($^{\circ}/s^3$)
1	-90	0	0.0955	θ_1	360	100	800	60
2	0	-0.418	0	θ_2	270	100	800	60
3	180	-0.398	0	θ_3	270	100	800	60
4	90	0	-0.122	θ_4	270	100	800	60
5	-90	0	0.098	θ_5	270	100	800	60
6	0	0	0.089	θ_6	360	100	800	60

6 Tables

There are four steps to run the MOP algorithm, which are: 1. Beginning with a specific path in the manipulator space, perform a kinematic inversion for acquiring a series of via-points in robotic joint space. 2. Set the values of kinematic constraints and start the algorithm for iterative optimization by selecting a proper initial solution taking into account the design considerations and structural constraints of the manipulator. 3. Input the expressions for the kinematic constraints and the objective function into the optimization algorithm. 4. The MOP solution is subsequently acquired employing the weighting methods and NSGA-II. The parameters of the GWO algorithms to solve problem (15) are as follows: searchAgents number is 10, max number of iterations is 10. The parameters of the NSGA-II algorithms to solve problem (13) are as follows: population size is 50, mating pool size is 12, tournament scale is 2, crossover probability is 0.8, mutation probability is 0.05, iteration number is 180,200, short distance ratio is 0.1.

The process is discussed in detail below. Firstly, the desired trajectory is obtained by RRT* algorithm in Fig.5(a) and inverse kinematics is employed to get the joint space motion trajectories in Table 2. After the robot trajectories in Table 2 have been acquired, an interpolation function is applied to build the time-joint sequence

in Fig.5(b). Figure.5(c)shows the trajectory of end-effector in the tridimensional space.The disadvantage of this approach, nevertheless, is that the jerk values at the start and end are not specified. Unsmooth jerk value for the industrial robot may lead to abrupt variations in torque, which may threaten the collaborator's safety.

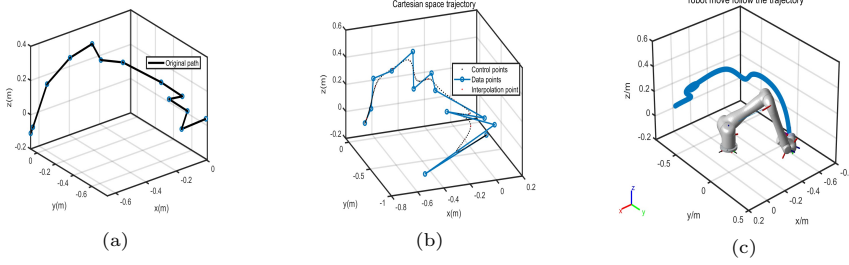


Fig. 5: (a)The given path points in Cartesian space . (b)quintic B-spline curves passing through all path points. (c)ELT EC66robot passes through the Cartesian space trajectory.

Table 2: Robot joint space trajectory

Node	Joint 1(rad)	Joint 2(rad)	Joint 3(rad)	Joint 4(rad)	Joint 5(rad)	Joint 6(rad)
1	1.3840	-1.0507	0.6405	-1.2793	1.6493	0.0506
2	1.1118	-1.0331	0.6417	-1.3247	1.7313	0.1993
3	1.0132	-1.1058	0.5884	-1.1677	1.9146	0.0352
4	0.8016	-1.2078	0.5614	-1.0210	1.9729	0.1052
5	0.7102	-1.1540	0.6475	-0.8477	2.1802	0.1698
6	0.49035	-1.2337	0.4708	-0.6496	2.1555	0.0064
7	0.1477	-1.4881	0.3355	-0.5948	2.0083	-0.1353
8	0.0544	-1.4903	0.2705	-0.5592	1.7672	0.0276
9	-0.0405	-1.5492	0.0876	-0.5059	1.6770	0.1031
10	-0.1625	-1.4811	0.3320	-0.7340	1.7123	0.0868
11	-0.1903	-1.3265	0.5831	-0.9949	1.6491	0.0730
12	-0.2333	-1.1043	0.9517	-1.3523	1.5455	0.0625
13	-0.2356	-1.0751	0.9913	-1.4190	1.5463	0.0506

6.1 Analysis of results

The time interval of trajectory is taken as a decision variable and the best variable value is chosen via two multi-objective optimization algorithms. The weight parameters ω_i of the weighting method is $\omega_1 = 0.1, \omega_2 = 0.6, \omega_3 = 0.3, \omega_1 = 0.7, \omega_2 = 0.2, \omega_3 = 0.1, \omega_1 = 0.2, \omega_2 = 0.3, \omega_3 = 0.5$. The result of weighting method is displayed as Table 4 and Table 5. This priori method assign weight according to expert knowledge before optimization. When weight ω_1 is larger, the pareto optimal solution tend to generate shorter execution time trajectory. However, since the behavior of weights is indirect, the resolution process cannot be easily controlled. For example, we cannot know which

solution can generate trajectory that consume less energy. Especially for MOP problems, the weights that generate some solutions are often not unique, so that remarkably different weights may yield similar solutions. On the other hand, small variations in the weights may also lead to large differences in the objective values.

Table 3: The weighting method experimental results

Number	weighting combination(s)	Decision variable	Fitness Value
1	$\omega_1 = 0.1, \omega_2 = 0.6, \omega_3 = 0.3$	$\Delta t = [2.8469, 2.2046, \dots, 2.6049, 3, 2.6698]$	30.3455
2	$\omega_1 = 0.7, \omega_2 = 0.2, \omega_3 = 0.1$	$\Delta t = [3, 1.6978, 1.4588, \dots, 2.8231, 2.1931]$	30.7623
3	$\omega_1 = 0.2, \omega_2 = 0.3, \omega_3 = 0.5$	$\Delta t = [2.7368, 2.1182, \dots, 2.7263, 3, 1.9991]$	49.4081

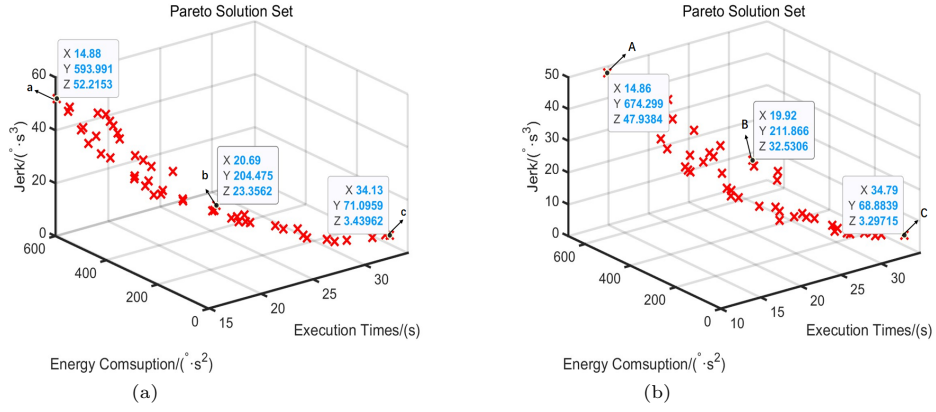


Fig. 6: (a)Pareto front of the NSGA-II method with Gen=180 . (b)Pareto front of the NSGA-II method with Gen=200.

Finding a unique optimal solution for the weighted objective function such as problem(15) will exclude some potentially valid compromise solutions. Different from only setting a set of weights, we are more concerned with finding a competitive set of solutions that do not favor the priori method or the posterior method. In the posterior method, the expert implicitly applies the weight by selecting a solution from a group of equally attractive final feasible solutions. The idea of a posteriori method is a typical Pareto optimization strategy like NSGA-II method. Fig.6(a) presents the result of NSGA-II algorithm method. The result in condition that the maximal generation is 200 is shown as Fig.6(b).

When the number of maximal generation is larger, more Pareto solutions will be obtained. In Figure 6(b), three characteristic solution can be used to generate trajectory as Table 4 and Table 5. The vector obtained by point A for the joint space with time variable is $\Delta t = [1.8, 1.07, 1.35, 1.05, 1.1, 1.3, 1, 1.1, 1.5, 1.04, 1.5, 1.05]$,

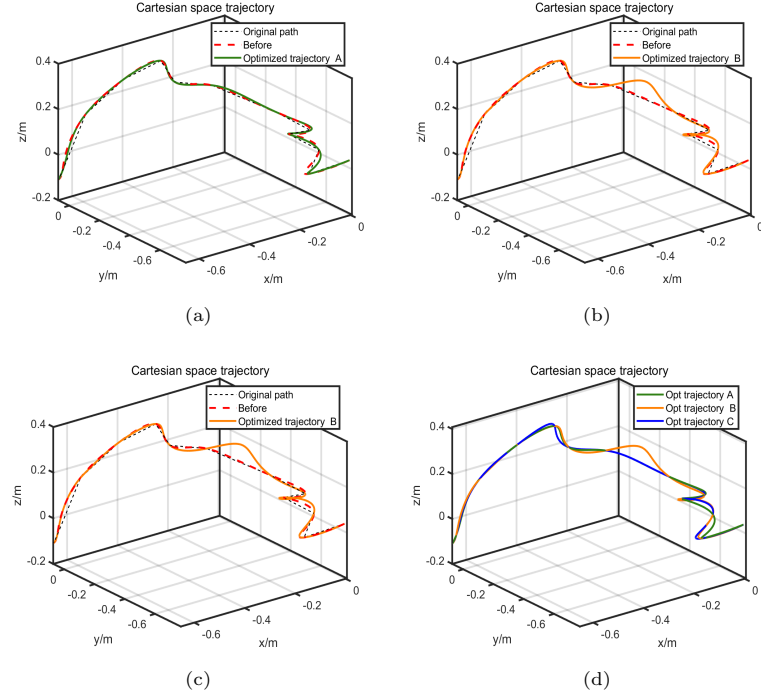


Fig. 7: (a)Optimized trajectory of Pareto solution of point A. (b)Optimized trajectory of Pareto solution of point B.(c)Optimized trajectory of Pareto solution of point A.(d)Trajectory comparison for different optimal criteria.

and the execution time is $14.86(s)$, the jerk value is $47.9384(^{\circ} \cdot s^{-3})$ and the energy consumption is $616.215(J)$. The vector obtained by point B for the joint space with time variable is $\Delta t = [2.5, 1.07, 1.35, 1.42, 1, 1.5, 1, 1.1, 1.5, 2.93, 3.5, 1.05]$, and the execution time is $20.03(s)$, the jerk value is $30.392(^{\circ} \cdot s^{-3})$ and the energy consumption is $244.286(J)$. The vector obtained by point C for the joint space with time variable is $\Delta t = [3, 3, 2.55, 2.81, 2.7, 2.5, 2, 2.6, 4, 2.93, 4, 2.7]$, and the execution time is $34.79(s)$, the jerk value is $3.29715(^{\circ} \cdot s^{-3})$ and the energy consumption is $68.8839(J)$.

From Fig.6(b), a series of solution approaching is obtained by using NSGA-II algorithm. The solution approaching A has poorer energy and smoothing performance and better temporal performance. Conversely, the solution closer to C exhibits poorer temporal performance and better energy and smoothing performance. The solution closer to B gives better energy and smoothing performance as compared to A, while the time performance is worse. Consequently, energy has a positive association with smoothing performance and a negative association with the temporal performance. From Fig.7, Optimized trajectories of three Pareto solution are plotted in Cartesian space. Compared to the trajectory before optimization, optimized trajectory A has shorter execution time. Compared to the optimized trajectory A, optimized trajectory

Point	Excution time(s)	Energy Consumption(J)	Jerk($^{\circ} \cdot s^{-3}$)
a	14.88	593.991	52.215
b	20.69	204.475	23.3562
c	34.13	71.0959	3.43962

Table 4: Optimal function value in Fig.6(a).

Point	Excution time(s)	Energy Consumption(J)	Jerk($^{\circ} \cdot s^{-3}$)
A	14.86	674.299	47.9384
B	19.92	211.866	23.3562
C	34.79	68.8839	3.29715

Table 5: Optimal function value in Fig.6(b).

B is smoother than other trajectories and optimized trajectory C has shorter energy cosumption.

The result get from NSGA-II is a group of optimal solutions, each of which is a non-dominant solution. In this posteriori method, we can select a solution from a group of equally attractive final feasible solutions. For example, we choose different Pareto solution of point A, B and C to generate trajectory. The derivatives of 5-order B-spline curve are shown in Fig.8, 9, 10.

The optimum curves for acceleration, velocity, displacement, jerk together with the optimum torque for each joint derived from NSGA-II allow a continuous of motion trajectories to be achieved via the quintic B-spline curve interpolation. The dynamic and kinematic constraints of the robot are met. In summary, the NSGA-II algorithm can effectively handle our multi-objective trajectory optimization problems in some ways, while a well-distributed Pareto front can be acquired. Compared with the normalized weighting method, multi-objective optimization trajectory planning is to achieve a group of Pareto-optimal trajectories.

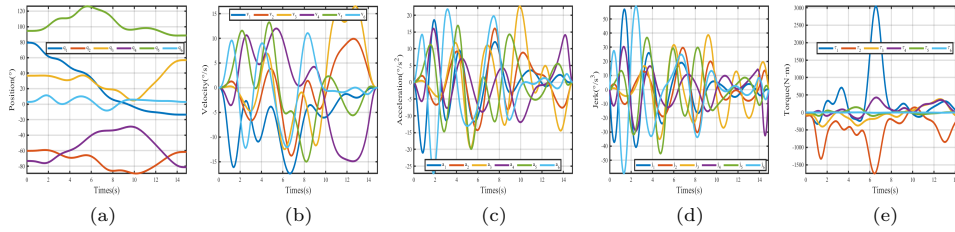


Fig. 8: The derivatives of 5-order B-spline curves of 6 joints for solution of point A (a)Position of manipulaor's 6 joint.(b)Velocity of manipulaor's 6 joint.(c)Accelaration of manipulaor's 6 joint.(d)Jerk of manipulaor's 6 joint.

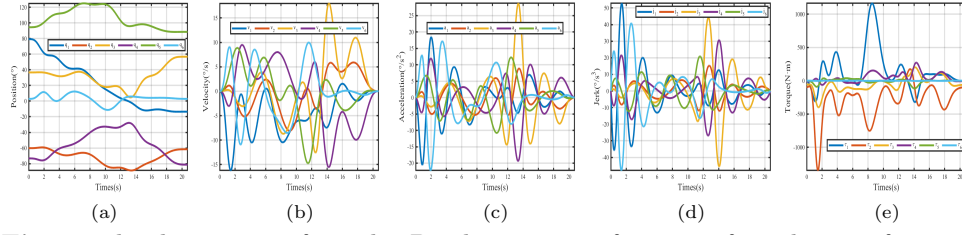


Fig. 9: The derivatives of 5-order B-spline curves of 6 joints for solution of point B (a)Position of manipulaor's 6 joint.(b)Velocity of manipulaor's 6 joint.(c)Acceleration of manipulaor's 6 joint.(d)Jerk of manipulaor's 6 joint.

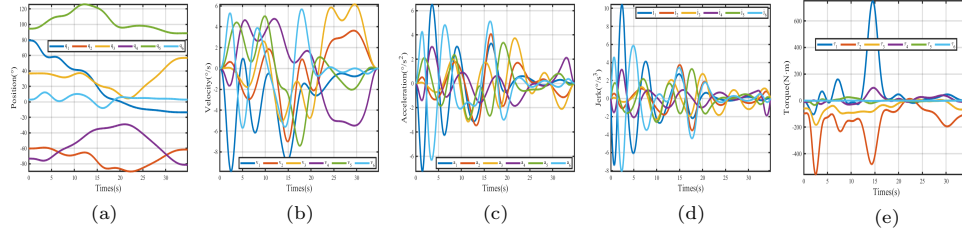


Fig. 10: The derivatives of 5-order B-spline curves of 6 joints for solution of point C (a)Position of manipulaor's 6 joint.(b)Velocity of manipulaor's 6 joint.(c)Acceleration of manipulaor's 6 joint.(d)Jerk of manipulaor's 6 joint.

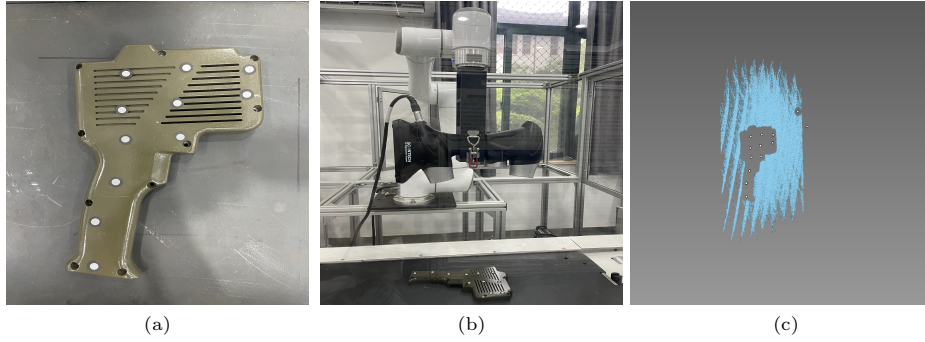


Fig. 11: Experiment on 3D laser scanner robot (a)Figure of strip rolling gun; (b) 3D laser scanner robot; (c) 3D point cloud imaging;

7 Conclusion

Getting a group of Pareto-optimal trajectories for a number of objectives is a more expensive method compared to finding only a single trajectory. Nevertheless, there is a pool of candidates to choose from that makes this method robust to uncertainty and variation. In the present paper, we use a decoupled motion planning structure and consider a multiobjective trajectory optimization problem for our industrial robotic manipulator. Given the velocity, acceleration, and acceleration constraints, the modeling of our problem is finished. For this problem, we identify a group of Pareto-optimal trajectories by doing some simulation experiments. According to some data analysis, it is found that compared with the Weighting method, Perato method can provide decision-makers more choice and has strong robustness and adaptability. It is explored that the application of multiobjective optimization in multitask planning of industrial 3D laser scanner robots as shown in Fig.11. Thus, the trajectory planning framework of the robotic manipulator can be applied industrial environments according to the task requirements. In the future work, dynamic obstacles will be taken into account and it is significant that online trajectory planning should be considered to be adapted to complex planning scenarios.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Grant No. 62273124) and Zhejiang Provincial Science and Technology Plan Project of China (Grant No. 2023C01174 and 2023C01117).

Funding

This work was supported by National Natural Science Foundation of China (Grant No. 62273124) and Zhejiang Provincial Science and Technology Plan Project of China (Grant No. 2023C01174 and 2023C01117).

Author Contribution

The Author Yumeng Huang wrote the matlab codes, perform the simulation experiment, contributed to analysis and manuscript preparation. The corresponding author Guangyu Liu contributed to the conception of the study. The author Wujia Yu contributed to analysis and manuscript revision. The author Shen Yu helped perform the analysis with constructive discussions.

Conflict of Interest

The authors declared that they have no conflicts of interest to this work.

Data Availability Statement

The study case data used to support the findings of this study are available within the article and can be obtained upon request to the corresponding author as well.

References

- [1] Shin, K.G., Mckay, N.D.: Selection of near-minimum time geometric paths for robotic manipulators. *IEEE Transactions on Automatic Control* **31**(6), 501–511 (1986)
- [2] Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J., Diehl, M.: Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transactions on Automatic Control* **54**(10), 2318–2327 (2009)
- [3] Oftadeh, R., Ghabcheloo, R., Mattila, J.: Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2925–2931 (2014). <https://doi.org/10.1109/ICRA.2014.6907280>
- [4] Saramago, S.F.P., Steffen, V.: Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system. *Mechanism and Machine Theory* **33**(7), 883–894 (1998) [https://doi.org/10.1016/S0094-114X\(97\)00110-9](https://doi.org/10.1016/S0094-114X(97)00110-9)
- [5] Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S.: Chomp: Covariant hamiltonian optimization for motion planning. *The International journal of robotics research* **32**(9-10), 1164–1193 (2013) <https://doi.org/10.1177/0278364913488805>
- [6] Kyriakopoulos, K.J., Saridis, G.N.: Minimum jerk path generation. In: Proceedings, 1988 IEEE International Conference on Robotics and Automation, pp. 364–369 (1988). IEEE
- [7] Ratliff, N., Zucker, M., Bagnell, J.A., Srinivasa, S.: Chomp: Gradient optimization techniques for efficient motion planning. In: the 1988 IEEE International Conference on Robotics and Automation, pp. 489–494 (2009). <https://doi.org/10.1109/ROBOT.2009.5152817>
- [8] Dragan, A.D., Ratliff, N.D., Srinivasa, S.S.: Manipulation planning with goal sets using constrained trajectory optimization. In: 2011 IEEE International Conference on Robotics and Automation, pp. 4582–4588 (2011). <https://doi.org/10.1109/ICRA.2011.5980538>
- [9] Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: Stomp: Stochastic trajectory optimization for motion planning. In: 2011 IEEE International Conference on Robotics and Automation, pp. 4569–4574 (2011). <https://doi.org/10.1109/ICRA.2011.5980280> . IEEE
- [10] Butler, S.D.: General algorithms for the time-optimal trajectory generation problem. PhD thesis, Rice University (2017)

- [11] Shin, K., McKay, N.: A dynamic programming approach to trajectory planning of robotic manipulators. *Journal of Machine Learning Research* **31**(6), 491–500 (2003) <https://doi.org/10.1109/tac.1986.1104317>
- [12] Debrouwere, F., Van Loock, W., Pipeleers, G., Diehl, M., Swevers, J., De Schutter, J.: Convex time-optimal robot path following with cartesian acceleration and inertial force and torque constraints. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* **227**(10), 724–732 (2013)
- [13] Boscariol, R. P. and Caracciolo, Richiedi, D.: Energy optimal design of jerk-continuous trajectories for industrial robots. In: *Advances in Italian Mechanism Science: Proceedings of the 3rd International Conference of IFToMM Italy*, pp. 318–325 (2020). Springer
- [14] Kyriakopoulos, K.J., Saridis, G.N.: Minimum jerk path generation. In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pp. 364–369 (1988). IEEE
- [15] Pham, Q.C.: A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics* **30**(6), 1533–1540 (2014) <https://doi.org/10.1109/TRO.2014.2351113>
- [16] Shin, K., McKay, N.: A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Transactions on Automatic Control* **31**(6), 491–500 (2003) <https://doi.org/10.1109/tac.1986.1104317>
- [17] Pham, Q.C. H. and Pham: A new approach to time-optimal path parameterization based on reachability analysis. *IEEE Transactions on Robotics* **34**(3), 645–659 (2018) <https://doi.org/10.1109/TRO.2018.2819195>
- [18] Shen, P.Y., Zhang, X.B., Fang, Y.C.: Essential properties of numerical integration for time-optimal path-constrained trajectory planning. *IEEE Robotics and Automation Letters* **2**(2), 888–895 (2017) <https://doi.org/10.1109/LRA.2017.2655580>
- [19] Verschuer, D., Demeulenaere, B., Swevers, J., De Schutter, J., Diehl, M.: Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transactions on Automatic Control* **54**(10), 2318–2327 (2009)
- [20] Stryk, O., Schlemmer, M.: Optimal control of the industrial robot manutec r3. *Computational optimal control, International series of Numerical Mathematics* **115**, 367–382 (1994)
- [21] Field, G., Stepanenko, Y.: Iterative dynamic programming: an approach to minimum energy trajectory planning for robotic manipulators. In: *1996 IEEE International Conference on Robotics and Automation*, pp. 2755–2760 (1996).

- [22] Sahu, P.K., Jena, A., Sujan, C., Biswal, B.B., Pati, K.C.: Optimal trajectory planning of industrial robots using geodesic. *International Journal of Robotics and Automation* **5**(3) (2016)
- [23] Zhu, S.Q., Liu, S.G., Wang, X.Y., Wang, H.F.: Time-optimal and jerk-continuous trajectory planning algorithm for manipulators. *Journal of Mechanical Engineering* **46**(3), 47–52 (2010)
- [24] Miettinen, K.: *Nonlinear Multiobjective Optimization* vol. 12. Springer, ??? (1999)
- [25] Deb, K., Sindhya, K., Hakanen, J.: *Multi-objective Optimization*, pp. 161–200. CRC Press, Boston, MA (2016)
- [26] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002)
- [27] Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* **11**(6), 712–731 (2007)
- [28] Huang, J.S., Hu, P.F., Wu, M. K.Y.and Zeng: Optimal time jerk trajectory planning for industrial robots. *Mechanism and Machine Theory* **121**, 530–544 (2018)
- [29] Craig, J.J.: *Introduction to Robotics: Mechanics and Control*. Introduction to Robotics: Mechanics and Control, United States (1986)
- [30] Vasilyev, I.A., Lyashin, A.M.: Analytical solution to inverse kinematic problem for 6-dof robot-manipulator. *Automation and Remote Control* **71**(10), 2195–2199 (2010)
- [31] Luh, J.Y.S., Walker, M.W., Paul, R.P.C.: Newton – euler formulation of manipulator dynamics for computer control11supported by nsf grants eng 76-18567 and apr 77-u533. In: REMBOLD, U. (ed.) *Information Control Problems in Manufacturing Technology* 1979, pp. 165–172. Pergamon, Stuttgart (1980). <https://doi.org/10.1016/B978-0-08-024452-5.50023-0>
- [32] De Boor, C.: *A Practical Guide to Splines*. springer-verlag, New York (1978)
- [33] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Advances in Engineering Software* **69**(3), 46–61 (2014)