

## Research Article

# A New Methodology for Solving Trajectory Planning and Dynamic Load-Carrying Capacity of a Robot Manipulator

Wanjin Guo,<sup>1</sup> Ruifeng Li,<sup>1</sup> Chuqing Cao,<sup>2,3</sup> Xunwei Tong,<sup>1</sup> and Yunfeng Gao<sup>1</sup>

<sup>1</sup>State Key Laboratory of Robotics and System, Harbin Institute of Technology,  
92 West Dazhi Street, Nan Gang District, Harbin 150001, China

<sup>2</sup>School of Mechanical Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing 210094, China

<sup>3</sup>Wuhu HIT Robot Technology Research Institute Co., Ltd., Electronic Industrial Park, Block E, Jiujiang District, Wuhu 241007, China

Correspondence should be addressed to Wanjin Guo; [guowanjin0102@gmail.com](mailto:guowanjin0102@gmail.com)

Received 25 June 2016; Revised 31 August 2016; Accepted 23 October 2016

Academic Editor: Alessandro Gasparetto

Copyright © 2016 Wanjin Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new methodology using a direct method for obtaining the best found trajectory planning and maximum dynamic load-carrying capacity (DLCC) is presented for a 5-degree of freedom (DOF) hybrid robot manipulator. A nonlinear constrained multiobjective optimization problem is formulated with four objective functions, namely, travel time, total energy involved in the motion, joint jerks, and joint acceleration. The vector of decision variables is defined by the sequence of the time-interval lengths associated with each two consecutive via-points on the desired trajectory of the 5-DOF robot generalized coordinates. Then this vector of decision variables is computed in order to minimize the cost function (which is the weighted sum of these four objective functions) subject to constraints on joint positions, velocities, acceleration, jerks, forces/torques, and payload mass. Two separate approaches are proposed to deal with the trajectory planning problem and the maximum DLCC calculation for the 5-DOF robot manipulator using an evolutionary optimization technique. The adopted evolutionary algorithm is the elitist nondominated sorting genetic algorithm (NSGA-II). A numerical application is performed for obtaining best found solutions of trajectory planning and maximum DLCC calculation for the 5-DOF hybrid robot manipulator.

## 1. Introduction

The most primary consideration in designing and accomplishing a fundamental required task of a robot manipulator is to plan the trajectory for this robot manipulator moving from its current placement to a series of specified placements while carrying a payload. During these operations of the robot manipulator, the problem of finding such a path is referred to as the trajectory planning problem. Usually, solving an optimization problem is defined to deal with the trajectory planning problem. There are a number of popular methods for transcribing the trajectory planning problem into an optimization problem. Essentially, these transcriptions fall into two categories: direct (global) methods (such as [1–4]) and indirect (decoupled) methods (e.g., [5–8]).

Solving trajectory planning for the robot manipulator is difficult because of the highly coupled nonlinear robot manipulator dynamics and complex system constraints

(which include limitations of actuator forces/torques and payload constraint). A feasible and efficient method must be able to find an appropriate trajectory such that it minimizes a multicriterion cost function and satisfies the limitations. This multicriterion cost function can include travel time, energy, actuator torques, and joint jerks. In order to achieve the greatly improved productivity in industrial situations, it is necessary to minimize the total traveling time of the robot manipulator. Many research works have concentrated on minimum time trajectories [9–13]. The robot manipulator trajectory planning using the energy criteria is reported in several literatures [14–18]. This kind of manipulator trajectory planning can produce smooth trajectories and reduce the stresses between the actuators and the robot manipulator structure. Moreover, it can save energy, which may be desirable in several applications, such as aerial, space, or submarine robots with a capacity limitation of energy source. Both optimal traveling time and the minimum energy of the

actuators are taken together as objective functions in some literatures [19–22]. Several examples of minimizing joint jerks and acceleration in the objective function are studied in [23–27]. Minimization of the joint jerks and acceleration usually contributes to the reduction of errors during trajectory tracking and stresses between the actuators and structure of the robot manipulator. Also, excitation frequencies of the robot manipulator can be limited.

Many attempts to determine the optimal trajectory of robot manipulators are mainly conventional optimization techniques such as sequential unconstrained minimization technique, sequential quadratic programming, dynamic programming, and numerical iterative procedure. However, these conventional techniques have some drawbacks, such as easily falling into local minima or involving calculating the gradient and the Hessian of the objective function and constraints. Usually, traditional techniques cannot solve discontinuous terms (e.g., friction) in physical models. In order to overcome these drawbacks, the evolutionary methods can be applied to handle the optimization trajectory problems [23, 28]. There are several advantages of using evolutionary techniques, which include (a) population-based approach search, thus discovering the global optimal solution is possible, (b) complex and multimodal problems which can be solved for global optimality, (c) no need to provide any auxiliary relationships (like gradients, derivatives, etc.), and (d) problem independence; that is, they are suitable for solving all types of optimization problems.

The dynamic load-carrying capacity (DLCC) is another important term, which is an important characteristic of the robot manipulators. It is mandatory to consider the load-carrying capacity for a robot manipulator and the payload constraint for the trajectory planning. DLCC has received considerable attention in research literature. Wang and Ravani [29, 30] determined the DLCC of a given trajectory for a serial manipulator as an optimization problem according to the state space of the dynamics equation and the torque-speed features of the actuators. Korayem et al. proposed several methods in order to find DLCC of a wheeled mobile manipulator, such as a method which is a combination of open and closed loop optimal control [31], a method based on hierarchical optimal control [32], and a method using open-loop optimal control approach and considering stability on zero-moment point criterion [33].

Several classical optimization approaches have been used to deal with the trajectory planning and the DLCC calculation. Dynamic programming approach is applied to solve optimization problem with nonlinear constraints and the sequential quadratic programming formulation is used to increase the speed of convergence for both fixed and mobile base mechanical manipulators [34]. The iterative linear programming method and the discrete augmented Lagrangian technique are used to solve the nonlinear constrained optimization problem for cable driven robots [35] and parallel kinematic machines [21], respectively.

In this paper, a nonlinear constrained multiobjective optimization problem is formulated with four objective functions (namely, travel time, total energy involved in the motion, joint jerks, and joint acceleration), considering constraints of joint

positions, velocities, acceleration, jerks, forces/torques, and payload. The sequence of the time-interval lengths associated with each two consecutive via-points is defined as the vector of decision variables. Then the cost function (which is the weighted sum of these four objective functions) is minimized by optimizing this vector of decision variables. Using a direct method, two approaches are proposed to deal with the trajectory planning problem and the calculation of maximum dynamic load-carrying capacity for a 5-DOF robotic manipulator, respectively. An evolutionary optimization technique, that is, an elitist nondominated sorting genetic algorithm (NSGA-II), is adopted to handle these optimization problems, taking into account these objective functions and constraints. The NSGA-II algorithm was originally proposed by Deb et al. [36]. It has been proved to be a highly effective and generally successful choice for optimization problems and becomes one of the mainstream solutions to multiobjective optimization questions [37–39].

The primary advantage of the proposed approaches in this paper is that the vector of decision variables (i.e., the sequence of the time-interval lengths associated with each two consecutive via-points) is optimized step by step during the evolution process (where the unfit individuals are replaced by the fit individuals through the crossover and mutation operator in each generation). The best found solutions of the joint trajectory displacement, velocity, acceleration, jerk, and force/torque can be obtained by these time-interval lengths. Additionally, the time intervals with variable lengths are more advantageous than the ones with same lengths (which are adopted by some classical optimal control approaches) because the former is more relatively likely to generate the smooth trajectory and avoid the suddenly change of the trajectory. Other advantages of the proposed approaches include the following: (a) both kinematics and dynamics aspects of the robot manipulator are considered, (b) they take several important decision criteria into account for trajectory planning (which are travel time, total energy involved in the motion, joint jerks, and joint acceleration), (c) adopting the different weights for the decision criteria, the required and suitable solutions that will satisfy various purposes can be obtained, (d) they consider payload mass constraint, (e) they ensure that the resulting best found trajectory is smooth and leads to minimum actuator power and energy, (f) they are easy to program and implement, and (g) they can also be extended to obtain the best found trajectory and the maximum DLCC for other types of robots.

The rest of this paper is organized as follows. The kinematics and dynamics of the studied 5-DOF robotic manipulator are described in Section 2. The problem formulation is presented in Section 3. An approach is proposed for the trajectory planning problem of the robot manipulator in Section 4. The maximum DLCC of the robot manipulator is calculated by a proposed approach in Section 5. The effectiveness of two proposed approaches for the trajectory planning and the maximum DLCC calculation is verified via a numerical example of this 5-DOF hybrid robot manipulator in Section 6. Finally, conclusions are drawn in Section 7.

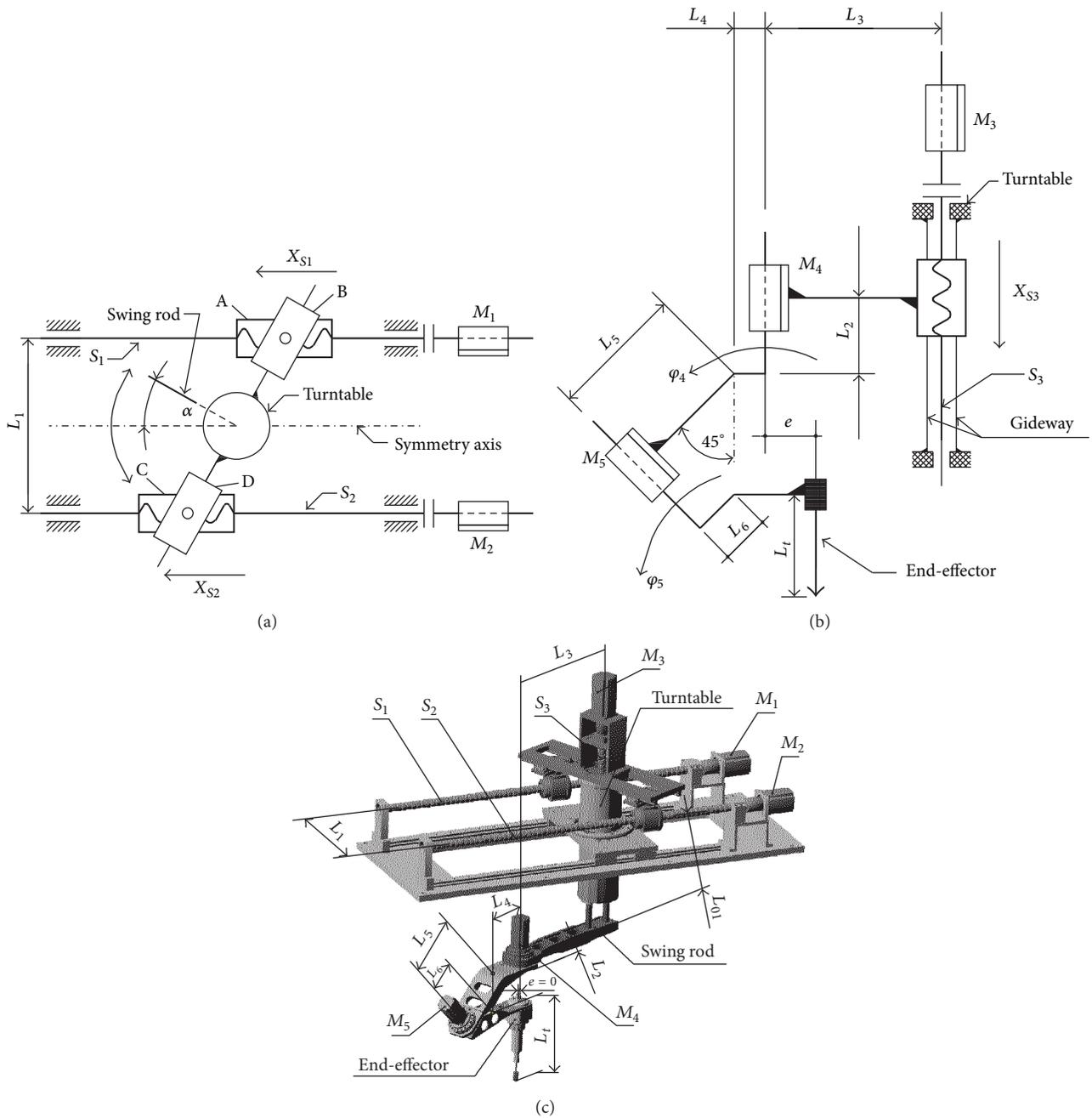


FIGURE 1: Robot manipulator: (a) top view; (b) front view; and (c) 3D model.

## 2. Kinematic and Dynamic Modelling of the 5-DOF Robotic Manipulator

In this section, the kinematic and dynamics modelling of a 5-DOF robot manipulator are described. The structure of this robotic manipulator is shown in Figure 1. This robotic manipulator, consisting of a 3-DOF (3T) parallel module and a 2-DOF (2R) serial module, is a new 3T2R hybrid robot manipulator. Translational motion axes  $J_1$ ,  $J_2$ , and  $J_3$  (consisting of ball screws) are driven by motors  $M_1$ ,  $M_2$ , and  $M_3$ , respectively. Rotational motion axes  $J_4$  and  $J_5$  are driven by motors  $M_4$  and  $M_5$ , respectively. The axis  $J_6$  is the end-effector which contains a high-speed motor spindle. More

details of the structure description for this robot manipulator are described in [40].

**2.1. Kinematic Modelling.** Based on the Denavit-Hartenberg method [41], the reference frames and the kinematic modelling for this robot manipulator are presented.

**2.1.1. Forward Kinematics.** The reference frames for this robot manipulator are assigned as shown in Figure 2(a). The vector  $\mathbf{r}_{ci}$  (shown in Figure 2(b)) is the location of the center of mass of a link  $i$  relative to the frame representing the link. The rotation angle of  $J_1$ ,  $J_2$ ,  $J_3$ ,  $J_4$ , and  $J_5$  is represented by  $\varphi_1$ ,  $\varphi_2$ ,  $\varphi_3$ ,  $\varphi_4$ , and  $\varphi_5$ , respectively. The stroke of the nut along  $S_1$ ,

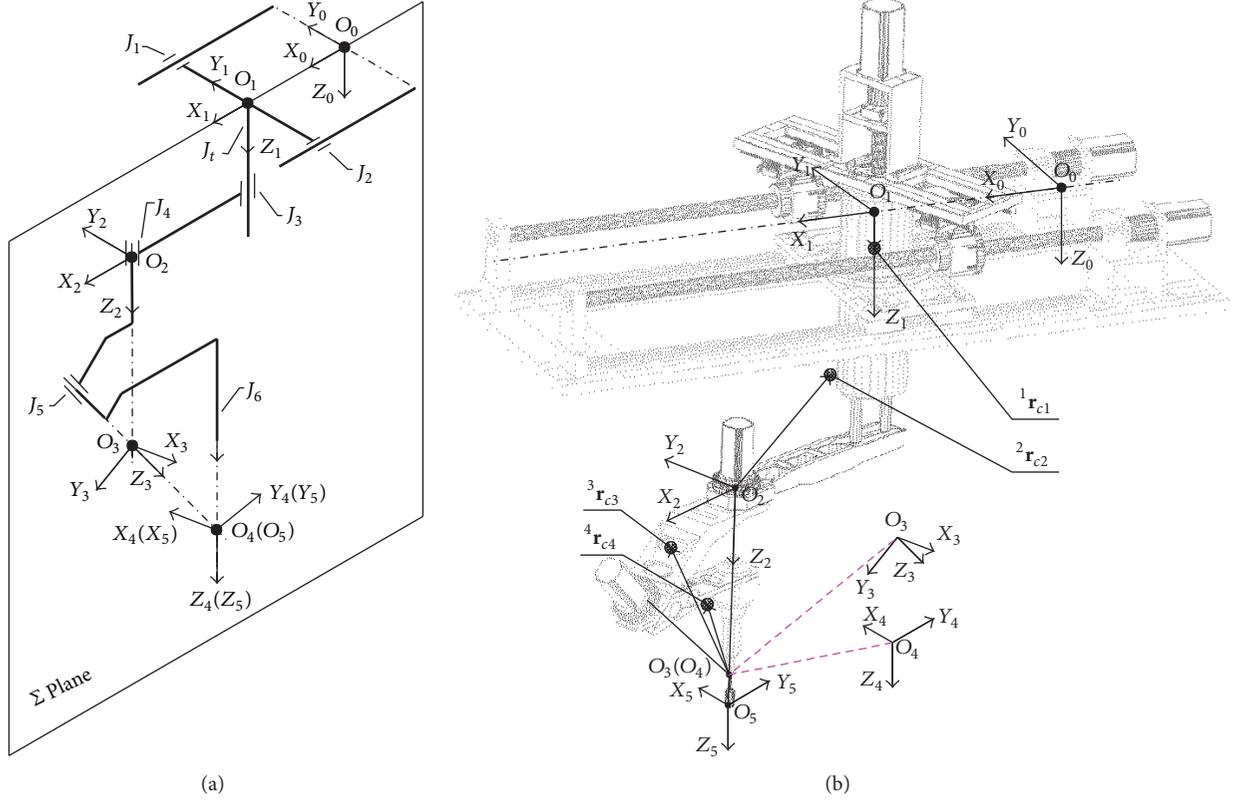


FIGURE 2: The assigned reference frame: (a) D-H reference frame and (b) the location of the center of mass.

$S_2$ , and  $S_3$  is represented by  $X_{S1}$ ,  $X_{S2}$ , and  $X_{S3}$ , respectively. The length of the rod  $i$  is denoted by  $L_i$  ( $i = 1, 2, \dots, 6$ ). The distance between  $J_6$  and  $J_4$  in the horizontal direction is denoted as  $e$ . The rotation angle of  $J_t$  and link 1 is denoted as  $\alpha$ .  $L_{01}$  represents the distance between point  $O_2$  and point  $O_1$  in the vertical direction when the nut is located at the top starting point along  $S_3$ .

The detailed structure parameters of this robot manipulator are listed in Table 1. Considering the reduction ratio of the speed reducer, the pitch, the nominal torque of the AC servomotor, and the suitable safety coefficients, the constraints of displacement, velocity, acceleration, jerk, and force/torque (i.e., generalized force) are given in Table 2. Some structure parameter variables are related by

$$\alpha = \arctan\left(\frac{X_{S2} - X_{S1}}{L_1}\right), \quad (1)$$

$$X_{S1} = \frac{s_x \varphi_1}{2\pi}, \quad (2)$$

$$X_{S2} = \frac{s_x \varphi_2}{2\pi}, \quad (3)$$

$$X_{S3} = \frac{s_z \varphi_3}{2\pi}, \quad (4)$$

where the constant  $s_x$  represents the pitch of  $S_1$  and  $S_2$ , the constant  $s_z$  represents the pitch of  $S_3$ , and the available range of the rotation angle is  $\alpha \in [-\pi/4, \pi/4]$  because of the mechanical constraints.

TABLE 1: Robot manipulator structure parameters (mm) and the lower and upper  $m_p$ .

$L_1$	$L_2$	$L_3$	$L_4$	$L_5$	$L_6$	$e$	$L_t$	$L_{01}$	$s_x$	$s_z$	$m_{pl}$	$m_{pu}$
420	50	450	180	180	80	0	225	400	10	10	0	10

Link 1 is driven in parallel by  $M_1$  and  $M_2$ , as shown in Figures 1 and 2, and a coupled movement is generated. While link 1 is rotating about  $Z_1$  and  $J_t$ , its moving direction along the axis  $X_1$  is always parallel to  $X_0$ . The base reference frame 0 is defined to be the inertial reference frame. Based on this movement relationship, the homogeneous transformation matrix  ${}^0\mathbf{T}_1$  which refer to the inertial reference frame is directly given as

$${}^0\mathbf{T}_1 = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & \frac{X_{S1} + X_{S2}}{2} \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where

$$\cos(\alpha) = \frac{L_1}{\sqrt{(X_{S2} - X_{S1})^2 + L_1^2}}, \quad (6)$$

$$\sin(\alpha) = \frac{X_{S2} - X_{S1}}{\sqrt{(X_{S2} - X_{S1})^2 + L_1^2}}. \quad (7)$$

TABLE 2: Limiting parameters.

Joint	$QC_{\min}$	$QC_{\max}$	VC	WC	JC	UC
1	0 mm	1000 mm	250 mm/s	5000 mm/s <sup>2</sup>	100000 m/s <sup>3</sup>	500 N
2	0 mm	1000 mm	250 mm/s	5000 mm/s <sup>2</sup>	100000 m/s <sup>3</sup>	500 N
3	0 mm	500 mm	250 mm/s	5000 mm/s <sup>2</sup>	100000 m/s <sup>3</sup>	500 N
4	$-\pi$ rad	$\pi$ rad	1.96 rad/s	60 rad/s <sup>2</sup>	1200 rad/s <sup>3</sup>	17 N·m
5	$-\pi$ rad	$\pi$ rad	1.96 rad/s	60 rad/s <sup>2</sup>	1200 rad/s <sup>3</sup>	17 N·m

TABLE 3: Geometric parameters of links.

Link $i$	$\theta_i$	$\alpha_i$	$d_i$	$a_i$
2	0	0	$X_{S3} + L_{01}$	$L_3$
3	$-\pi/2 + \varphi_4$	$\pi/4$	$L_2 + L_4 + \sqrt{2}L_5$	0
4	$\pi + \varphi_5$	$\pi/4$	$\sqrt{2}e$	0
5 (tool)	0	0	$L_p$	0

Let  $L_p$  be defined as the tool compensation, which is the value of the tool frame point  $O_5$  in  $Z_4$  axis given as

$$L_p = L_t - (L_4 + \sqrt{2}L_6 + e). \quad (8)$$

Based on the D-H reference frame, using the attached reference frames to each link, the four parameters that locate one frame relative to another are defined as follows:  $a_i$  is the distance from  $Z_{i-1}$  to  $Z_i$  along  $X_{i-1}$ ,  $\alpha_i$  is the angle from  $Z_{i-1}$  to  $Z_i$  about  $X_{i-1}$ ,  $d_i$  is the distance from  $X_{i-1}$  to  $X_i$  along  $Z_i$ , and  $\theta_i$  is the angle from  $X_{i-1}$  to  $X_i$  about  $Z_i$ . These geometric parameters of this manipulator for the links 2, 3, 4, and 5 (tool) shown in Figures 1 and 2 are listed in Table 3.

One vector is represented by  $\mathbf{V}_{i-1}$  in the frame  $i-1$  and  $\mathbf{V}_i$  in the frame  $i$ , respectively; the relationship between  $\mathbf{V}_{i-1}$  and  $\mathbf{V}_i$  can be expressed by  $\mathbf{V}_{i-1} = {}^{i-1}\mathbf{T}_i \mathbf{V}_i$ . The homogeneous transformation matrix  ${}^{i-1}\mathbf{T}_i$  is conducted to provide a representation of reference frame  $i$  relative to reference frame  $i-1$  according to the D-H reference frames and the geometric parameters. It is given as

$${}^{i-1}\mathbf{T}_i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

where  $i = 2, 3, 4, 5$ .

By successive multiplications of the transformation matrices, the equivalent homogeneous transformation matrix  ${}^0\mathbf{T}_5$  can be obtained. Let a vector be represented as  $\mathbf{r}_5$  in the end-effector reference frame; then this vector related to the inertial reference frame can be expressed as  $\mathbf{r}_0 = {}^0\mathbf{T}_5 \mathbf{r}_5$ .

Here,  $\mathbf{r}_0$  relates the inertial reference frame. The matrix  ${}^0\mathbf{T}_5$  is obtained as

$${}^0\mathbf{T}_5 = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 {}^3\mathbf{T}_4 {}^4\mathbf{T}_5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$= \mathbf{T}(q),$$

where the vector  $\mathbf{p} = [p_x, p_y, p_z]^T$  is the position vector of the origin point  $O_5$  relative to the inertial reference frame. The vectors  $\mathbf{a} = [a_x, a_y, a_z]^T$ ,  $\mathbf{o} = [o_x, o_y, o_z]^T$  and  $\mathbf{n} = [n_x, n_y, n_z]^T$  are the approach vector, orientation vector, and normal vector of the end-effector reference frame 5 relative to the inertial reference frame, respectively. The rotation matrix of the end-effector reference frame 5 relative to the inertial reference frame can be expressed as  ${}^0\mathbf{R}_5 = [\mathbf{n} \ \mathbf{o} \ \mathbf{a}]$ . The details are as follows:

$$\mathbf{p} = [p_x, p_y, p_z]^T, \quad (11)$$

$$p_x = \frac{X_{S1} + X_{S2}}{2} - e \cos(\alpha + \varphi_4) + L_3 \cos(\alpha) + L_p \left[ \frac{\cos(\alpha + \varphi_4)(\cos(\varphi_5) - 1)}{2} - \frac{\sin(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}} \right], \quad (12)$$

$$p_y = L_3 \sin(\alpha) - e \sin(\alpha + \varphi_4) + L_p \left[ \frac{\sin(\alpha + \varphi_4)(\cos(\varphi_5) - 1)}{2} + \frac{\cos(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}} \right], \quad (13)$$

$$p_z = L_2 + L_4 + \sqrt{2}L_5 + X_{S3} + L_{01} + e + \frac{L_p(1 + \cos(\varphi_5))}{2}, \quad (14)$$

$$\mathbf{a} = [a_x, a_y, a_z]^T, \quad (15)$$

$$a_x = \frac{\cos(\alpha + \varphi_4)(\cos(\varphi_5) - 1)}{2} - \frac{\sin(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}}, \quad (16)$$

$$a_y = \frac{\sin(\alpha + \varphi_4)(\cos(\varphi_5) - 1)}{2} + \frac{\cos(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}}, \quad (17)$$

$$a_z = \frac{1 + \cos(\varphi_5)}{2}, \quad (18)$$

$$\mathbf{o} = [o_x, o_y, o_z]^T, \quad (19)$$

$$o_x = \frac{\sin(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}} - \frac{\cos(\alpha + \varphi_4)(1 + \cos(\varphi_5))}{2}, \quad (20)$$

$$o_y = \frac{-\cos(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}} - \frac{\sin(\alpha + \varphi_4)(1 + \cos(\varphi_5))}{2}, \quad (21)$$

$$o_z = \frac{1 - \cos(\varphi_5)}{2}, \quad (22)$$

$$\mathbf{n} = [n_x, n_y, n_z]^T, \quad (23)$$

$$n_x = -\sin(\alpha + \varphi_4)\cos(\varphi_5) - \frac{\cos(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}}, \quad (24)$$

$$n_y = \cos(\alpha + \varphi_4)\cos(\varphi_5) - \frac{\sin(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}}, \quad (25)$$

$$n_z = -\frac{\sin(\varphi_5)}{\sqrt{2}}. \quad (26)$$

Let the reference point of the end-effector be represented by the vector  $[x_e, y_e, z_e]^T$  relative to the end-effector reference frame 5; the inertial position coordinates  $(x, y, z)$  of the end-effector can be obtained by

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = {}^0\mathbf{T}_5 \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix}; \quad (27)$$

namely,

$$\begin{aligned} x &= p_x + n_x x_e + o_x y_e + a_x z_e \\ y &= p_y + n_y x_e + o_y y_e + a_y z_e \\ z &= p_z + n_z x_e + o_z y_e + a_z z_e. \end{aligned} \quad (28)$$

The results of the inertial position coordinates  $(x, y, z)$  can be represented by the generalized coordinates  $\mathbf{q} = [X_{S1}, X_{S2}, X_{S3}, \varphi_4, \varphi_5]^T$  through substituting the above related equations into (28).

*2.1.2. Inverse Kinematics.* The structure of this 5-DOF robot manipulator satisfies the Peiper criterion [42], so that closed-form solutions can be obtained through an algebraic method. The inverse kinematics and its concise expressions are given as follows:

$$\varphi_5 = \arctan 2(-\sqrt{2}n_z, 2a_z - 1), \quad (29)$$

where the arctan 2 function is the four-quadrant arctangent function; for example,  $\arctan 2(Y, X) = \arg(X + jY) \in (-\pi, \pi]$ ,  $j = \sqrt{-1}$ ; that is, it returns the argument of the complex number with real part  $X$  and imaginary part  $Y$ .

$$\begin{aligned} \varphi_4 &= \arctan 2(a_y(a_z - 1) + a_x n_z, a_x(a_z - 1) - a_y n_z) \\ &\quad - \arctan 2(A_1, B_1), \end{aligned} \quad (30)$$

where

$$\begin{aligned} A_1 &= p_y - L_p a_y + \frac{e(a_y(a_z - 1) + a_x n_z)}{(a_z - 1)^2 + n_z^2}, \\ B_1 &= \sqrt{1 - \left[ p_y - L_p a_y + \frac{e(a_y(a_z - 1) + a_x n_z)}{(a_z - 1)^2 + n_z^2} \right]^2}, \\ X_{S3} &= p_z - (L_2 + L_4 + \sqrt{2}L_5 + L_{01} + e + a_z L_p), \\ X_{S1} &= p_x + e \cos(\alpha + \varphi_4) - L_3 \cos(\alpha) \\ &\quad - L_p \left[ \frac{\cos(\alpha + \varphi_4)(\cos(\varphi_5) - 1)}{2} \right. \\ &\quad \left. - \frac{\sin(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}} \right] - \frac{L_1 \tan(\alpha)}{2}, \\ X_{S2} &= p_x + e \cos(\alpha + \varphi_4) - L_3 \cos(\alpha) \\ &\quad - L_p \left[ \frac{\cos(\alpha + \varphi_4)(\cos(\varphi_5) - 1)}{2} \right. \\ &\quad \left. - \frac{\sin(\alpha + \varphi_4)\sin(\varphi_5)}{\sqrt{2}} \right] + \frac{L_1 \tan(\alpha)}{2}. \end{aligned} \quad (31)$$

Since the available range of the rotation angle is  $\alpha \in [-\pi/4, \pi/4]$  as mentioned above,

$$\begin{aligned} \alpha &= \arctan 2\left(\sin(\alpha), \sqrt{1 - \sin^2(\alpha)}\right), \\ \sin(\alpha) &= \frac{p_y - L_p a_y + e(a_y(a_z - 1) + a_x n_z) / ((a_z - 1)^2 + n_z^2)}{L_3}. \end{aligned} \quad (32)$$

*2.1.3. Representation of Desired End-Effector Inertial Position Trajectory.* Assume that the coordinates of the desired inertial position trajectory of the 5-DOF robot end-effector denoted by  $(x_d, y_d, z_d)$  are specified in the following two forms (see example in Section 6).

(a) *Implicit Form*

$$z_d(t) = f(x_d(t), y_d(t)) \quad t \in [t_0, t_f], \quad (33)$$

where  $f(x_d(t), y_d(t))$  is a given smooth function and the initial and final positions  $(x_d(t_0), y_d(t_0), z_d(t_0))$  and  $(x_d(t_f), y_d(t_f), z_d(t_f))$  are given.

The implicit form is used in the formulation of the basic multiobjective optimization problem (see Section 3.1).

 (b) *Parametric Form*

$$\begin{aligned} x_d(b) &= f_1(b), \\ y_d(b) &= f_2(b), \\ z_d(b) &= f_3(b), \end{aligned} \quad (34)$$

$$b \in [b_0, b_f],$$

where  $b$  is a parameter,  $f_i(b)$ ,  $i = 1, 2, 3$ , are given smooth functions, and the initial and final positions  $(x_d(b_0), y_d(b_0), z_d(b_0))$  and  $(x_d(b_f), y_d(b_f), z_d(b_f))$  are given.

The parametric form is used in the formulation of the finite dimensional multiobjective optimization problem (see Section 3.2).

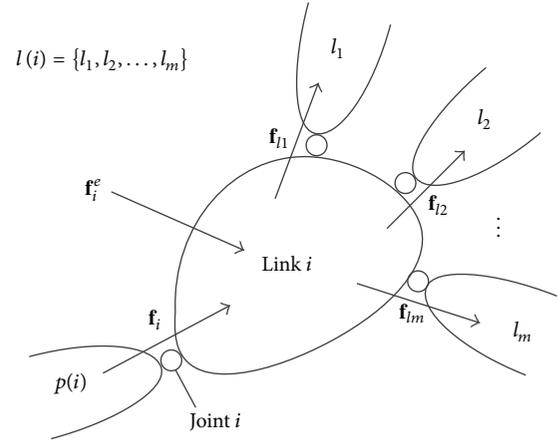
Given a 5-DOF robot motion trajectory  $q(t)$   $t \in [t_0, t_f]$ , then the resulting actual inertial position of the end-effector, denoted by  $(x(t), y(t), z(t))$   $t \in [t_0, t_f]$ , is computed by using (28).

Assume that the actual inertial position of the end-effector  $(x(t), y(t), z(t))$   $t \in [t_0, t_f]$  matches the desired inertial position trajectory specified in implicit form; see (33). Then, it follows that  $(x(t), y(t), z(t))$   $t \in [t_0, t_f]$  satisfies the following constraint equation:

$$\begin{aligned} z(t) &= f(x(t), y(t)) \quad t \in [t_0, t_f], \\ (x(t_0), y(t_0), z(t_0)) &= (x_d(t_0), y_d(t_0), z_d(t_0)), \\ (x(t_f), y(t_f), z(t_f)) &= (x_d(t_f), y_d(t_f), z_d(t_f)). \end{aligned} \quad (35)$$

**2.2. Dynamic Modelling Using the RNEA.** Because of the need for computational efficiency, the robotics community has focused on this problem of the robotic manipulator dynamics. For inverse dynamics, the recursive Newton–Euler algorithm (RNEA) proposed by Luh et al. [43] remains the most important algorithm [44]. The RNEA uses a Newton–Euler formulation of the problem, and this formulation is especially amenable to the development of efficient recursive algorithms for dynamics computations.

The RNEA is an algorithm with calculation complexity of  $O(n)$  (where  $n$  is the number of degrees of freedom in the mechanism), which is aimed to calculate recursive dynamics of fixed base open-loop robot. The calculation divides into four steps shown as follows, with two steps in each of the two recursions.


 FIGURE 3: Forces acting on link  $i$ .

*Step 1.* Calculate the velocity and acceleration of each link in sequence, beginning with the known velocity and acceleration of the fixed base and calculating towards the end link of robot. The velocity of each link in the robot is obtained by the recursive formula

$$\mathbf{v}_i = \mathbf{v}_{p(i)} + \Phi_i \dot{\mathbf{q}}_i, \quad (\mathbf{v}_0 = \mathbf{0}), \quad (36)$$

where  $\mathbf{v}_i$  is the velocity of link  $i$ ,  $\Phi_i$  is the motion matrix of joint  $i$ , and  $\dot{\mathbf{q}}_i$  is the vector of joint velocity variables for joint  $i$ .

The equivalent formula for acceleration is given by differentiating (36):

$$\mathbf{a}_i = \mathbf{a}_{p(i)} + \Phi_i \ddot{\mathbf{q}}_i + \dot{\Phi}_i \dot{\mathbf{q}}_i, \quad (\mathbf{a}_0 = \mathbf{0}), \quad (37)$$

where  $\mathbf{a}_i$  is the acceleration of link  $i$  and  $\ddot{\mathbf{q}}_i$  is the vector of joint acceleration variables.

Through initializing  $\mathbf{a}_0$  to  $-\mathbf{a}_g$  instead of zero (where  $\mathbf{a}_g$  is the gravitational acceleration vector and the gravity is simulated by a fictitious base acceleration), the effect of a uniform gravitational field on robot can be simulated. In this case,  $\mathbf{a}_i$  is not the true acceleration of link  $i$ , but the sum of its true acceleration and  $-\mathbf{a}_g$ .

*Step 2.* Calculate the equation of motion for each link. This step computes the forces required to cause the acceleration calculated in Step 1. The equation of motion for link  $i$  is shown as

$$\mathbf{f}_i^a = \mathbf{I}_i \mathbf{a}_i + \mathbf{v}_i \times \mathbf{I}_i \mathbf{v}_i, \quad (38)$$

where  $\mathbf{I}_i$  is the spatial inertia of link  $i$  and  $\mathbf{f}_i^a$  is the net force acting on link  $i$ .

*Step 3.* Calculate the spatial force on each joint. According to Figure 3, the force acting on link  $i$  is as follows:

$$\mathbf{f}_i^a = \mathbf{f}_i^e + \mathbf{f}_i - \sum_{j \in c(i)} \mathbf{f}_j, \quad (39)$$

TABLE 4: The mass, moment of inertia matrix, and center of mass and of the robot manipulator.

Link	Mass (kg)	Moment of inertia matrix (kg·(mm) <sup>2</sup> )	Center of mass (mm)
1	83.025	$\begin{bmatrix} 5095170.782 & -61.008 & 1019.065 \\ -61.008 & 3291706.841 & -15.858 \\ 1019.065 & -15.858 & 2367692.766 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -44.182 \end{bmatrix}$
2	21.333	$\begin{bmatrix} 1475990.573 & -3970.014 & -748196.633 \\ -3970.014 & 2366543.574 & -8327.223 \\ -748196.633 & -8327.223 & 911968.216 \end{bmatrix}$	$\begin{bmatrix} -304.563 \\ -1.306 \\ -278.925 \end{bmatrix}$
3	8.022	$\begin{bmatrix} 142265.977 & -0.114 & 0.002 \\ -0.114 & 45016.139 & 46002.096 \\ 0.002 & 46002.096 & 109393.601 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -63.686 \\ -468.138 \end{bmatrix}$
4	9.120	$\begin{bmatrix} 119573.680 & 27.342 & -0.849 \\ 27.342 & 36160.108 & -18452.465 \\ -0.849 & -18452.465 & 94076.578 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -52.909 \\ -261.241 \end{bmatrix}$

where  $\mathbf{f}_i$  is the force transmitted across joint  $i$ ,  $\mathbf{f}_i^e$  is the sum of all relevant external forces acting on link  $i$ , and  $l(i)$  is the connected set of link  $i$ . Rearranging (38) gives the recursive formula for calculating the joint forces:

$$\mathbf{f}_i = \mathbf{f}_i^a - \mathbf{f}_i^e + \sum_{j \in l(i)} \mathbf{f}_j, \quad (40)$$

where  $i$  iterates from  $N_B$  to 1.

*Step 4.* Calculate the joint force variables,  $\boldsymbol{\tau}_i$ . They are given by the equation

$$\boldsymbol{\tau}_i = \boldsymbol{\Phi}_i^T \mathbf{f}_i. \quad (41)$$

In this section, the dynamics problem (i.e., inverse dynamics problem) of this 5-DOF robot manipulator is solved based on this very efficient RNEA. The velocity and acceleration of each link and the resultant forces on each link are first computed through an outward recursion in turn, starting with the known velocity and acceleration of the fixed base and working towards the tips. A second, inward recursion uses the force balance equations at each link to compute the spatial force across each joint and the value of each joint torque/force variable. Note that, without loss of generality, only the case of the rigid robot manipulator is stressed here; other contributing factors in the dynamic description of the robot manipulator (which may include the dynamics of the actuators, joint and link flexibility, friction, noise, and disturbances) are not considered in this paper.

Define that  ${}^j\boldsymbol{\omega}_i$ ,  ${}^j\mathbf{v}_i$ ,  ${}^j\dot{\boldsymbol{\omega}}_i$ ,  ${}^j\dot{\mathbf{v}}_i$ ,  ${}^j\mathbf{f}_i$ , and  ${}^j\mathbf{n}_i$  are the angular velocity, the linear velocity, the angular acceleration, the linear acceleration, the force, and the moment of the reference frame  $i$  with respect to the inertial reference frame and expressed in the reference frame  $j$ , respectively. Also define that  ${}^j\dot{\mathbf{v}}_{ci}$ ,  ${}^j\mathbf{f}_{ci}$ , and  ${}^j\mathbf{n}_{ci}$  are the linear acceleration of the center of mass of link  $i$ , the inertial force, and the inertial moment acting on the center of mass of link  $i$  in the reference frame  $i$

with respect to the inertial reference frame and expressed in the reference frame  $j$ , respectively.

The vector  ${}^j\mathbf{z}_i$  is defined as the axis  $Z_i$  of the reference frame  $i$  expressed in the reference frame  $j$ . The vector  ${}^j\mathbf{p}_i$  is defined as the origin of the reference frame  $i$  with respect to the reference frame  $j$  and expressed in the reference frame  $j$ . The rotation matrix  ${}^j\mathbf{R}_i$  is defined as the reference frame  $i$  with respect to the reference frame  $j$  and expressed in the reference frame  $j$ .

The mass ( $m_i$ ), center of mass ( ${}^i\mathbf{r}_{ci}$ ), and moment of inertia about the center of mass ( ${}^{ci}\mathbf{I}_i$ ) of link  $i$  expressed in reference frame  $i$  are listed in Table 4.

### 2.2.1. Outward Recursion

(a) *Link 1.* As shown in Figures 1 and 2, link 1 is driven in parallel by  $M_1$  and  $M_2$ , and a coupled movement is generated. The angular and linear velocity and acceleration of reference frame 1 representing link 1 with respect to the inertial reference frame and expressed in the inertial reference frame can be directly given as follows:

$$\begin{aligned} {}^0\boldsymbol{\omega}_1 &= [0, 0, \dot{\alpha}]^T, \\ {}^0\mathbf{v}_1 &= \left[ \frac{\dot{X}_{S1} + \dot{X}_{S2}}{2}, 0, 0 \right]^T, \\ {}^0\dot{\boldsymbol{\omega}}_1 &= [0, 0, \ddot{\alpha}]^T, \\ {}^0\dot{\mathbf{v}}_1 &= \left[ \frac{\ddot{X}_{S1} + \ddot{X}_{S2}}{2}, 0, -g \right]^T, \end{aligned} \quad (42)$$

where  $g = 9.80665 \text{ m/s}^2$ , according to (1), yielding

$$\dot{\alpha} = \frac{L_1 (\dot{X}_{S2} - \dot{X}_{S1})}{L_1^2 + (X_{S2} - X_{S1})^2},$$

$$\ddot{\alpha} = \frac{L_1 (\ddot{X}_{S2} - \ddot{X}_{S1})}{L_1^2 + (X_{S2} - X_{S1})^2} - \frac{2L_1 (X_{S2} - X_{S1}) (\dot{X}_{S2} - \dot{X}_{S1})^2}{(L_1^2 + (X_{S2} - X_{S1})^2)^2}. \quad (43)$$

The angular and linear velocity and acceleration of the reference frame 1 and the linear acceleration of the center of mass of link 1 and the inertial force and the inertial moment acting on the center of mass of link 1 in the reference frame 1, with respect to the inertial reference frame and expressed in the reference frame 1, are derived as

$${}^1\boldsymbol{\omega}_1 = {}^1\mathbf{R}_0 {}^0\boldsymbol{\omega}_1, \quad (44)$$

$${}^1\mathbf{v}_1 = {}^1\mathbf{R}_0 {}^0\mathbf{v}_1, \quad (45)$$

$${}^1\dot{\boldsymbol{\omega}}_1 = {}^1\mathbf{R}_0 {}^0\dot{\boldsymbol{\omega}}_1, \quad (46)$$

$${}^1\dot{\mathbf{v}}_1 = {}^1\mathbf{R}_0 {}^0\dot{\mathbf{v}}_1, \quad (47)$$

$${}^1\dot{\mathbf{v}}_{c1} = {}^1\dot{\mathbf{v}}_1 + {}^1\dot{\boldsymbol{\omega}}_1 \times {}^1\mathbf{r}_{c1} + {}^1\boldsymbol{\omega}_1 \times ({}^1\boldsymbol{\omega}_1 \times {}^1\mathbf{r}_{c1}), \quad (48)$$

$${}^1\mathbf{f}_{c1} = m_1 {}^1\dot{\mathbf{v}}_{c1}, \quad (49)$$

$${}^1\mathbf{n}_{c1} = {}^{c1}\mathbf{I}_1 {}^1\dot{\boldsymbol{\omega}}_1 + {}^1\boldsymbol{\omega}_1 \times ({}^{c1}\mathbf{I}_1 {}^1\boldsymbol{\omega}_1), \quad (50)$$

where the rotation matrix  ${}^1\mathbf{R}_0$  is given by

$${}^1\mathbf{R}_0 = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (51)$$

(b) *Link 2.* The angular and linear velocity and acceleration of the reference frame 2 and the linear acceleration of the center of mass of link 2 and the inertial force and the inertial moment acting on the center of mass of link 2 in the reference frame 2, with respect to the inertial reference frame and expressed in the reference frame 2, are derived as

$${}^2\boldsymbol{\omega}_2 = {}^2\mathbf{R}_1 {}^1\boldsymbol{\omega}_1, \quad (52)$$

$${}^2\dot{\boldsymbol{\omega}}_2 = {}^2\mathbf{R}_1 {}^1\dot{\boldsymbol{\omega}}_1, \quad (53)$$

$${}^2\dot{\mathbf{v}}_2 = {}^2\mathbf{R}_1 [{}^1\dot{\mathbf{v}}_1 + {}^1\dot{\boldsymbol{\omega}}_1 \times {}^1\mathbf{P}_2 + {}^1\boldsymbol{\omega}_1 \times ({}^1\boldsymbol{\omega}_1 \times {}^1\mathbf{P}_2) + 2{}^2\boldsymbol{\omega}_2 \times {}^2\mathbf{z}_1 \dot{X}_{S3} + {}^2\mathbf{z}_1 \ddot{X}_{S3}], \quad (54)$$

$${}^2\dot{\mathbf{v}}_{c2} = {}^2\dot{\mathbf{v}}_2 + {}^2\dot{\boldsymbol{\omega}}_2 \times {}^2\mathbf{r}_{c2} + {}^2\boldsymbol{\omega}_2 \times ({}^2\boldsymbol{\omega}_2 \times {}^2\mathbf{r}_{c2}), \quad (55)$$

$${}^2\mathbf{f}_{c2} = m_2 {}^2\dot{\mathbf{v}}_{c2}, \quad (56)$$

$${}^2\mathbf{n}_{c2} = {}^{c2}\mathbf{I}_2 {}^2\dot{\boldsymbol{\omega}}_2 + {}^2\boldsymbol{\omega}_2 \times ({}^{c2}\mathbf{I}_2 {}^2\boldsymbol{\omega}_2), \quad (57)$$

where

$$\begin{aligned} {}^1\mathbf{P}_2 &= [L_3, 0, X_{S3} + L_{01}]^T, \\ {}^2\mathbf{z}_1 &= [0, 0, 1]^T, \\ {}^2\mathbf{R}_1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (58)$$

(c) *Link 3.* The angular and linear velocity and acceleration of the reference frame 3 and the linear acceleration of the center of mass of link 3 and the inertial force and the inertial moment acting on the center of mass of link 3 in the reference frame 3, with respect to the inertial reference frame and expressed in the reference frame 3, are calculated as follows:

$${}^3\boldsymbol{\omega}_3 = {}^3\mathbf{R}_2 {}^2\boldsymbol{\omega}_2 + {}^3\mathbf{z}_2 \dot{\varphi}_4, \quad (59)$$

$${}^3\dot{\boldsymbol{\omega}}_3 = {}^3\mathbf{R}_2 ({}^2\dot{\boldsymbol{\omega}}_2 + {}^2\boldsymbol{\omega}_2 \times {}^3\mathbf{z}_2 \dot{\varphi}_4) + {}^3\mathbf{z}_2 \ddot{\varphi}_4, \quad (60)$$

$${}^3\dot{\mathbf{v}}_3 = {}^3\mathbf{R}_2 [{}^2\dot{\mathbf{v}}_2 + {}^2\dot{\boldsymbol{\omega}}_2 \times {}^2\mathbf{P}_3 + {}^2\boldsymbol{\omega}_2 \times ({}^2\boldsymbol{\omega}_2 \times {}^2\mathbf{P}_3)], \quad (61)$$

$${}^3\dot{\mathbf{v}}_{c3} = {}^3\dot{\mathbf{v}}_3 + {}^3\dot{\boldsymbol{\omega}}_3 \times {}^3\mathbf{r}_{c3} + {}^3\boldsymbol{\omega}_3 \times ({}^3\boldsymbol{\omega}_3 \times {}^3\mathbf{r}_{c3}), \quad (62)$$

$${}^3\mathbf{f}_{c3} = m_3 {}^3\dot{\mathbf{v}}_{c3}, \quad (63)$$

$${}^3\mathbf{n}_{c3} = {}^{c3}\mathbf{I}_3 {}^3\dot{\boldsymbol{\omega}}_3 + {}^3\boldsymbol{\omega}_3 \times ({}^{c3}\mathbf{I}_3 {}^3\boldsymbol{\omega}_3), \quad (64)$$

where

$$\begin{aligned} {}^2\mathbf{P}_3 &= [0, 0, L_2 + L_4 + \sqrt{2}L_5]^T, \\ {}^3\mathbf{z}_2 &= \left[ 0, \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right]^T, \\ {}^3\mathbf{R}_2 &= \begin{bmatrix} \sin(\varphi_4) & -\cos(\varphi_4) & 0 \\ \frac{\cos(\varphi_4)}{\sqrt{2}} & \frac{\sin(\varphi_4)}{\sqrt{2}} & \frac{\sqrt{2}}{2} \\ -\frac{\cos(\varphi_4)}{\sqrt{2}} & -\frac{\sin(\varphi_4)}{\sqrt{2}} & \frac{\sqrt{2}}{2} \end{bmatrix}. \end{aligned} \quad (65)$$

(d) *Link 4.* The angular and linear velocity and acceleration of the reference frame 4 and the linear acceleration of the center of mass of link 4 and the inertial force and the inertial moment acting on the center of mass of link 4 in the reference frame 4, with respect to the inertial reference frame and expressed in the reference frame 4, are given as follows:

$${}^4\boldsymbol{\omega}_4 = {}^4\mathbf{R}_3 {}^3\boldsymbol{\omega}_3 + {}^4\mathbf{z}_3 \dot{\varphi}_5, \quad (66)$$

$${}^4\dot{\boldsymbol{\omega}}_4 = {}^4\mathbf{R}_3 ({}^3\dot{\boldsymbol{\omega}}_3 + {}^3\boldsymbol{\omega}_3 \times {}^4\mathbf{z}_3 \dot{\varphi}_5) + {}^4\mathbf{z}_3 \ddot{\varphi}_5, \quad (67)$$

$${}^4\dot{\mathbf{v}}_4 = {}^4\mathbf{R}_3 [{}^3\dot{\mathbf{v}}_3 + {}^3\dot{\boldsymbol{\omega}}_3 \times {}^3\mathbf{P}_4 + {}^3\boldsymbol{\omega}_3 \times ({}^3\boldsymbol{\omega}_3 \times {}^3\mathbf{P}_4)], \quad (68)$$

$${}^4\dot{\mathbf{v}}_{c4} = {}^4\dot{\mathbf{v}}_4 + {}^4\dot{\boldsymbol{\omega}}_4 \times {}^4\mathbf{r}_{c4} + {}^4\boldsymbol{\omega}_4 \times ({}^4\boldsymbol{\omega}_4 \times {}^4\mathbf{r}_{c4}), \quad (69)$$

$${}^4\mathbf{f}_{c4} = m_4 {}^4\dot{\mathbf{v}}_{c4}, \quad (70)$$

$${}^4\mathbf{n}_{c4} = {}^{c4}\mathbf{I}_4 {}^4\dot{\boldsymbol{\omega}}_4 + {}^4\boldsymbol{\omega}_4 \times ({}^{c4}\mathbf{I}_4 {}^4\boldsymbol{\omega}_4), \quad (71)$$

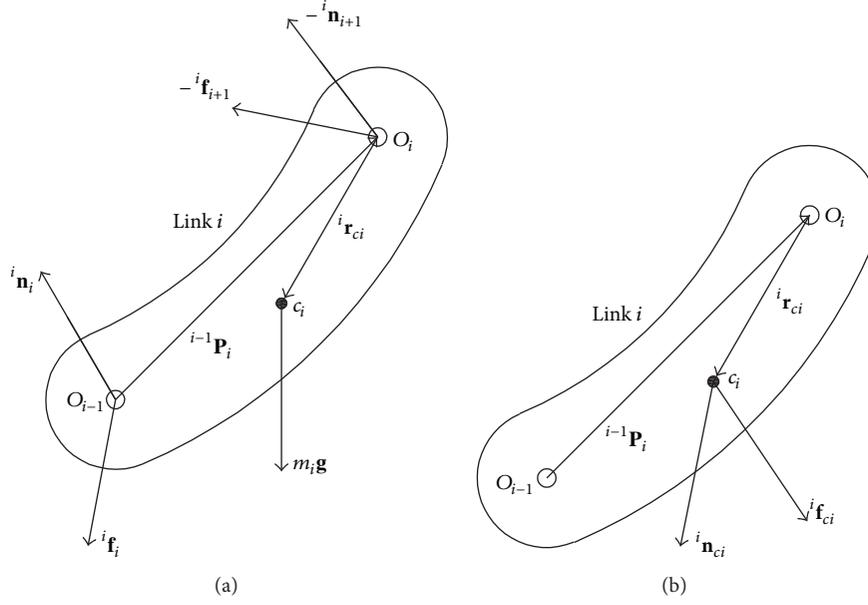


FIGURE 4: External forces and resultant forces of link  $i$ : (a) external forces and moments acting on link  $i$  and (b) resultant force and moment at the center of mass of link  $i$ .

where

$$\begin{aligned} {}^3 \mathbf{P}_4 &= [0, 0, \sqrt{2}e]^T, \\ {}^4 \mathbf{z}_3 &= \left[ 0, \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right]^T, \\ {}^4 \mathbf{R}_3 &= \begin{bmatrix} -\cos(\varphi_5) & -\sin(\varphi_5) & 0 \\ \frac{\sin(\varphi_5)}{\sqrt{2}} & \frac{-\cos(\varphi_5)}{\sqrt{2}} & \frac{\sqrt{2}}{2} \\ \frac{-\sin(\varphi_5)}{\sqrt{2}} & \frac{\cos(\varphi_5)}{\sqrt{2}} & \frac{\sqrt{2}}{2} \end{bmatrix}. \end{aligned} \quad (72)$$

**2.2.2. Inward Recursion.** For each link of robot manipulator, except the gravitational force, the forces and moments acting one link belong to internal forces and moments acting between the connected links; that is, the external forces and moments relationships between connected links are forces/moments and reaction forces/moments. A diagram is given for displaying the external forces and resultant forces (shown in Figure 4). External forces and moments acting on link  $i$  are shown in Figure 4(a). Here, the force and moment of link  $i-1$  acting on link  $i$  are resultant  ${}^i \mathbf{f}_i$  and  ${}^i \mathbf{n}_i$ , respectively; also, the force and moment of link  $i+1$  acting on link  $i$  are resultant  ${}^i \mathbf{f}_{i+1}$  and  ${}^i \mathbf{n}_{i+1}$ , respectively;  $m_i \mathbf{g}$  is the gravitational force of link  $i$  (where  $m_i$  is the mass of link  $i$  and  $\mathbf{g}$  is gravitational acceleration vector). The resultant force and moment at the center of mass of link  $i$  caused by these external forces and moments acting on link  $i$  based on static balancing conditions are  ${}^i \mathbf{f}_{ci}$  and  ${}^i \mathbf{n}_{ci}$  (shown in Figure 4(b)), respectively.

(a) *Link 4.* When a robot performs a manipulation task (e.g., polishing a surface), the robot establishes physical contact with external environment; the relevant external forces and/or moments will act on the end-effector. In order to express this influence in the dynamic equations of motion, suppose that the resultant of all relevant external forces and moments applied on the origin of the end-effector reference frame 5 is  ${}^5 \mathbf{f}_{\text{tool}}$  and  ${}^5 \mathbf{n}_{\text{tool}}$ , respectively. Thus, the force and moment of link 4 applied on the origin of the reference frame 5 are  ${}^5 \mathbf{f}_5 = -{}^5 \mathbf{f}_{\text{tool}}$  and  ${}^5 \mathbf{n}_5 = -{}^5 \mathbf{n}_{\text{tool}}$ , respectively. The payload mass  $m_p$  can be entered as an external force  $\mathbf{f}_{\text{tool}}$ .

The force and the moment of the reference frame 4 with respect to the inertial reference frame and expressed in the reference frame 4 and the required joint actuator torque of joint 5 are derived as

$${}^4 \mathbf{f}_4 = {}^4 \mathbf{R}_5 {}^5 \mathbf{f}_5 + {}^4 \mathbf{f}_{c4}, \quad (73)$$

$${}^4 \mathbf{n}_4 = {}^4 \mathbf{R}_5 {}^5 \mathbf{n}_5 + {}^4 \mathbf{n}_{c4} + {}^4 \mathbf{r}_{c4} \times {}^4 \mathbf{f}_{c4} + {}^4 \mathbf{P}_5 \times ({}^4 \mathbf{R}_5 {}^5 \mathbf{f}_5), \quad (74)$$

$$\tau_5 = {}^4 \mathbf{z}_3^T {}^4 \mathbf{n}_4, \quad (75)$$

where

$${}^4 \mathbf{P}_5 = [0, 0, L_p]^T,$$

$${}^4 \mathbf{R}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (76)$$

Note that, for link 4, the inertial force  ${}^4 \mathbf{f}_{c4}$  and the inertial moment  ${}^4 \mathbf{n}_{c4}$  (which are expressed in (70) and (71), resp.) are represented as the term  ${}^4 \mathbf{n}_{c4} + {}^4 \mathbf{r}_{c4} \times {}^4 \mathbf{f}_{c4}$  in (74) in order to apply the influence of the center of mass of link 4 on the origin

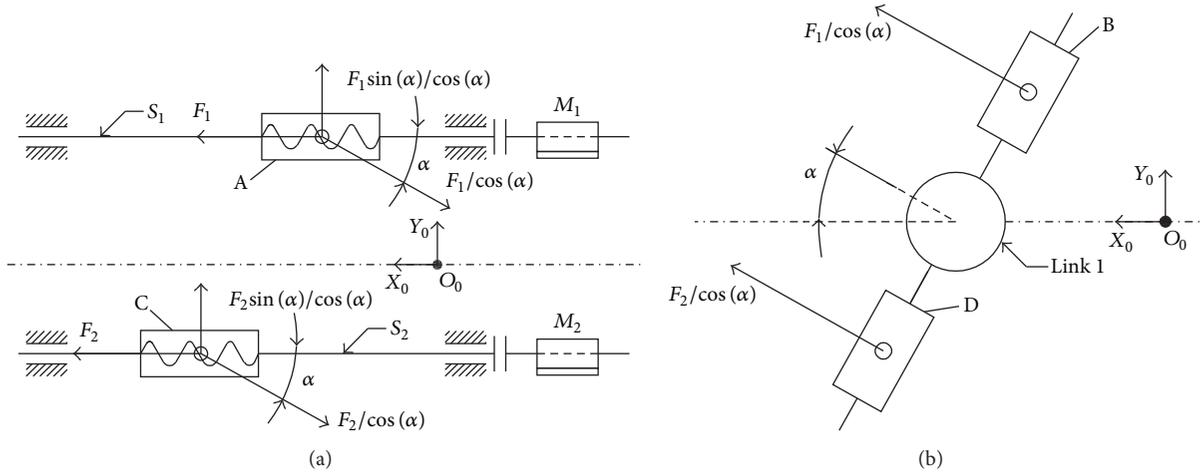


FIGURE 5: The interaction forces: (a) screw joints and (b) link 1.

of the reference frame 4. Other influences of the center of mass of link  $i$  ( $i = 1, 2, 3$ ) have the same representation forms.

(b) *Link 3.* The force and the moment of the reference frame 3 with respect to the inertial reference frame and expressed in the reference frame 3 and the required joint actuator torque of joint 4 are obtained as

$${}^3\mathbf{f}_3 = {}^3\mathbf{R}_4 {}^4\mathbf{f}_4 + {}^3\mathbf{f}_{c3}, \quad (77)$$

$${}^3\mathbf{n}_3 = {}^4\mathbf{R}_3^T {}^4\mathbf{n}_4 + {}^3\mathbf{n}_{c3} + {}^3\mathbf{r}_{c3} \times {}^3\mathbf{f}_{c3} + {}^3\mathbf{P}_4 \times ({}^4\mathbf{R}_3^T {}^4\mathbf{f}_4), \quad (78)$$

$$\tau_4 = {}^3\mathbf{z}_2^T {}^3\mathbf{n}_3. \quad (79)$$

(c) *Link 2.* The force and the moment of the reference frame 2 with respect to the inertial reference frame and expressed in the reference frame 2 and the required joint actuator torque of joint 3 are expressed as follows:

$${}^2\mathbf{f}_2 = {}^2\mathbf{R}_3 {}^3\mathbf{f}_3 + {}^2\mathbf{f}_{c2}, \quad (80)$$

$${}^2\mathbf{n}_2 = {}^3\mathbf{R}_2^T {}^3\mathbf{n}_3 + {}^2\mathbf{n}_{c2} + {}^2\mathbf{r}_{c2} \times {}^2\mathbf{f}_{c2} + {}^2\mathbf{P}_3 \times ({}^3\mathbf{R}_2^T {}^3\mathbf{f}_3), \quad (81)$$

$$F_3 = {}^2\mathbf{z}_1^T {}^2\mathbf{f}_2. \quad (82)$$

(d) *Link 1.* The force and the moment of the reference frame 1 with respect to the inertial reference frame and expressed in the reference frame 1 are given as follows:

$${}^1\mathbf{f}_1 = {}^1\mathbf{R}_2 {}^2\mathbf{f}_2 + {}^1\mathbf{f}_{c1}, \quad (83)$$

$${}^1\mathbf{n}_1 = {}^1\mathbf{R}_2 {}^2\mathbf{n}_2 + {}^1\mathbf{n}_{c1} + {}^1\mathbf{r}_{c1} \times {}^1\mathbf{f}_{c1} + {}^1\mathbf{P}_2 \times ({}^1\mathbf{R}_2 {}^2\mathbf{f}_2). \quad (84)$$

The interaction forces between link 1 and screw joints (consisting of ball screws) are shown in Figure 5. Combining Newton's and Euler's equations for the link 1 leads to

$$\frac{L_1 (F_2 - F_1)}{2 \cos^2(\alpha)} = {}^1\mathbf{z}_0^T ({}^0\mathbf{R}_1 {}^1\mathbf{n}_1) \quad (85)$$

$$F_2 + F_1 = {}^1\mathbf{x}_0^T ({}^0\mathbf{R}_1 {}^1\mathbf{f}_1),$$

where  ${}^1\mathbf{z}_0 = [0, 0, 1]^T$  and  ${}^1\mathbf{x}_0 = [1, 0, 0]^T$ .

The required joint actuator forces of joint 1 and joint 2 are derived by (85); that is,

$$F_1 = \frac{{}^1\mathbf{x}_0^T ({}^0\mathbf{R}_1 {}^1\mathbf{f}_1)}{2} - \frac{{}^1\mathbf{z}_0^T ({}^0\mathbf{R}_1 {}^1\mathbf{n}_1) \cos^2(\alpha)}{L_1}, \quad (86)$$

$$F_2 = \frac{{}^1\mathbf{x}_0^T ({}^0\mathbf{R}_1 {}^1\mathbf{f}_1)}{2} + \frac{{}^1\mathbf{z}_0^T ({}^0\mathbf{R}_1 {}^1\mathbf{n}_1) \cos^2(\alpha)}{L_1}. \quad (87)$$

Ignoring transmission power loss, the resulting five applied motor torques (while the five reaction torques are equal in magnitude and opposite in direction) can be obtained as

$$T_{M1} = \frac{F_1 s_x}{2\pi},$$

$$T_{M2} = \frac{F_2 s_x}{2\pi},$$

$$T_{M3} = \frac{F_3 s_z}{2\pi}, \quad (88)$$

$$T_{M4} = \tau_4,$$

$$T_{M5} = \tau_5.$$

Thus, given a 5-DOF robot trajectory  $(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t))$   $t \in [t_0, t_f]$ , then the applied forces and torques  $\mathbf{U} = [F_1 \ F_2 \ F_3 \ \tau_4 \ \tau_5]^T$  required to produce the above-mentioned trajectory are computed as follows.

*Step 1.* Calculate the velocity and acceleration of each link in turn, starting with the known velocity and acceleration of the

fixed base, and working towards the end. They are computed by (44)–(47), (52)–(54), (59)–(61), and (66)–(68).

*Step 2.* Calculate the equation of motion for each link. They are determined by (50), (57), (64), and (71).

*Step 3.* Calculate the spatial force on each joint. They are given by (74), (78), (81), and (84).

*Step 4.* Calculate the joint force variables (i.e., the applied forces and torques  $\mathbf{U} = [F_1 \ F_2 \ F_3 \ \tau_4 \ \tau_5]^T$ ). They are obtained by (75), (79), (82), (86), and (87).

*2.3. Dynamic Modelling Using the Lagrange Equations.* Besides the RNEA, the robot dynamic equations of motion can also be derived by the Lagrange equations. The dynamic equations can be deduced by calculating the kinetic and potential energies of the links and by defining the Lagrangian and by differentiating the Lagrangian equation with respect to the joint variables [45].

*2.3.1. Kinetic Energy.* As mentioned above, the link 1 is driven in parallel by  $M_1$  and  $M_2$ , and a coupled movement is generated. In order to separately express this coupled movement, an addition virtual link with zero mass and zero length is introduced between the base and link 1 to produce a translational motion. Thus, a translational motion of the virtual link along the axis  $X_1$  and a rotation motion of the link 1 about the axis  $Z_1$  can be decoupled from this coupled movement. The corresponding transformation matrix  ${}^0\mathbf{T}_1$  can be represented as

$$\begin{aligned} {}^0\mathbf{T}_1 &= \mathbf{A}_1\mathbf{A}_2, \\ \mathbf{A}_1 &= \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{A}_2 &= \begin{bmatrix} \cos(\phi_2) & -\sin(\phi_2) & 0 & 0 \\ \sin(\phi_2) & \cos(\phi_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (89)$$

where

$$\begin{aligned} d_1 &= \frac{X_{S1} + X_{S2}}{2}, \\ \phi_2 &= \alpha. \end{aligned} \quad (90)$$

Other transformation matrices  ${}^{i-1}\mathbf{T}_i$  for the link  $i$  ( $i = 2, 3, 4$ ) are represented by  $\mathbf{A}_j$  ( $j = 3, 4, 5$ ), respectively; that is,

$$\mathbf{A}_j = {}^{i-1}\mathbf{T}_i \quad (i = 2, 3, 4 \ j = i + 1), \quad (91)$$

where the joint variables are given as follows:

$$\begin{aligned} d_3 &= X_{S3} + L_{01}, \\ \phi_4 &= \frac{-\pi}{2} + \varphi_4, \\ \phi_5 &= \pi + \varphi_5. \end{aligned} \quad (92)$$

The transformation between the reference frame  $i$  and the inertial reference frame can be written as

$$\begin{aligned} {}^0\mathbf{T}_i &= {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \cdots {}^{i-1}\mathbf{T}_i = \mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_i \\ & \quad i = 1, 2, \dots, 5. \end{aligned} \quad (93)$$

Using local generalized coordinates  $\boldsymbol{\psi} = [d_1, \phi_2, d_3, \phi_4, \phi_5]^T$  to represent these joint variables ( $d_1$  and  $d_3$  for prismatic joints and  $\phi_1, \phi_4$  and  $\phi_5$  for revolute joints), the derivative of  $\mathbf{A}_i$  matrix for a revolute joint or a prismatic joint can be expressed as

$$\frac{\partial \mathbf{A}_i}{\partial \psi_i} = \rho_i \mathbf{A}_i \quad i = 1, 2, \dots, 5, \quad (94)$$

where

$$\begin{aligned} \rho_i \text{ (revolute)} &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ \rho_i \text{ (prismatic)} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (95)$$

Extending the same differentiation principle to the  ${}^0\mathbf{T}_i$  matrix of (93) with multiple joint variables differentiated with respect to only one variable  $q_j$  gives

$$\begin{aligned} \Gamma_{ij} &= \frac{\partial {}^0\mathbf{T}_i}{\partial \psi_j} = \frac{\partial (\mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_j \cdots \mathbf{A}_i)}{\partial \psi_j} \\ &= \mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{Q}_j\mathbf{A}_j \cdots \mathbf{A}_i \quad j \leq i \leq 5. \end{aligned} \quad (96)$$

Higher-order derivatives can be formulated similarly from

$$\Gamma_{ijk} = \frac{\partial \Gamma_{ij}}{\partial \psi_k}. \quad (97)$$

Using  $\mathbf{r}_i$  to represent a point on any link  $i$  of the robot relative to frame  $i$ , the position of the point can be expressed by premultiplying the vector with the transformation matrix representing its frame as follows:

$$\mathbf{p}_i = {}^0\mathbf{T}_i\mathbf{r}_i. \quad (98)$$

Differentiating (98) with respect to all the local variables  $\psi_j$  yields the velocity of the point:

$$\begin{aligned} \mathbf{V}_i &= \frac{d}{dt} ({}^0\mathbf{T}_i\mathbf{r}_i) = \sum_{j=1}^i \left( \frac{\partial {}^0\mathbf{T}_i}{\partial \psi_j} \frac{d\psi_j}{dt} \right) \mathbf{r}_i \\ &= \sum_{j=1}^i \left( \Gamma_{ij} \frac{d\psi_j}{dt} \right) \mathbf{r}_i. \end{aligned} \quad (99)$$

The kinetic energy of an element of mass  $m_i$  on a link is

$$dK_i = \frac{1}{2} \text{Trace} \left[ \left( \sum_{p=1}^i \left( \Gamma_{ip} \frac{d\psi_p}{dt} \right) \mathbf{r}_i \right) \cdot \left( \sum_{r=1}^i \left( \Gamma_{ir} \frac{d\psi_r}{dt} \right) \mathbf{r}_i \right)^T \right] dm_i, \quad (100)$$

where the mathematical symbol “Trace” is the trace of a matrix. Integrating this equation and rearranging terms yield the total kinetic energy:

$$K_i = \int dK_i = \frac{1}{2} \text{Trace} \left[ \sum_{p=1}^i \sum_{r=1}^i \Gamma_{ip} \left( \int \mathbf{r}_i \mathbf{r}_i^T dm_i \right) \Gamma_{ir}^T \dot{\psi}_p \dot{\psi}_r \right]. \quad (101)$$

The Pseudo Inertia Matrix, representing the  $\int \mathbf{r}_i \mathbf{r}_i^T dm_i$  terms, can be written as

$$\mathbf{J}_i = \begin{bmatrix} \frac{(-I_{xx} + I_{yy} + I_{zz})_i}{2} & I_{ixy} & I_{ixz} & m_i x_{lci} \\ I_{ixy} & \frac{(I_{xx} - I_{yy} + I_{zz})_i}{2} & I_{iyz} & m_i y_{lci} \\ I_{ixz} & I_{iyz} & \frac{(I_{xx} + I_{yy} - I_{zz})_i}{2} & m_i z_{lci} \\ m_i x_{lci} & m_i y_{lci} & m_i z_{lci} & m_i \end{bmatrix}, \quad (102)$$

where  $\mathbf{r}_{lci} = [x_{lci}, y_{lci}, z_{lci}]^T$  is the location of the center of mass of the link  $i$  relative to the frame representing the link. For the local generalized variables  $\boldsymbol{\psi} = [d_1, \phi_2, d_3, \phi_4, \phi_5]^T$ , the corresponding Pseudo Inertia Matrices are  $\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3, \mathbf{J}_4$ , and  $\mathbf{J}_5$ , respectively. The Pseudo Inertia Matrices and the location of the center of mass of the studied robot manipulator are listed in Table 5.

Substituting (102) into (101) gives the final form for kinetic energy of the studied robot manipulator (here, the kinetic energy of the actuators is not stressed in this paper):

$$K = \frac{1}{2} \sum_{i=1}^5 \sum_{p=1}^i \sum_{r=1}^i \text{Trace} \left( \Gamma_{ip} \mathbf{J}_i \Gamma_{ir}^T \right) \dot{\psi}_p \dot{\psi}_r. \quad (103)$$

**2.3.2. Potential Energy.** The potential energy of the system is the sum of the potential energies of each link and can be written as

$$P = \sum_{i=1}^5 P_i = \sum_{i=1}^5 \left[ -m_i \mathbf{g}^T \left( {}^0 \mathbf{T}_i \mathbf{r}_{lci} \right) \right], \quad (104)$$

where  $\mathbf{g}^T = [g_x \ g_y \ g_z \ 0]$  is the gravity matrix. Notice that the values in the gravity matrix are dependent on the orientation of the reference frame.

**2.3.3. The Lagrangian.** The Lagrangian of the robot manipulator is then expressed as

$$L = K - P = \frac{1}{2} \sum_{i=1}^5 \sum_{p=1}^i \sum_{r=1}^i \text{Trace} \left( \Gamma_{ip} \mathbf{J}_i \Gamma_{ir}^T \right) \dot{\psi}_p \dot{\psi}_r - \sum_{i=1}^5 \left[ -m_i \mathbf{g}^T \left( {}^0 \mathbf{T}_i \mathbf{r}_{lci} \right) \right]. \quad (105)$$

**2.3.4. The Lagrangian Dynamics for the Robot's Equations of Motion.** The Lagrangian of the robot manipulator can be differentiated for local generalized coordinates  $\boldsymbol{\psi} = [d_1, \phi_2, d_3, \phi_4, \phi_5]^T$  in order to form the dynamic equations of motion:

$$T = \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\boldsymbol{\psi}}} \right) - \frac{\partial L}{\partial \boldsymbol{\psi}}. \quad (106)$$

The final equations of motion for all local generalized variables  $\boldsymbol{\psi} = [d_1, \phi_2, d_3, \phi_4, \phi_5]^T$  can be derived as follows:

$$T_i = \sum_{j=1}^5 D_{ij} \ddot{\psi}_j + \sum_{j=1}^5 \sum_{k=1}^5 D_{ijk} \dot{\psi}_j \dot{\psi}_k + D_i \quad (107)$$

$$i = 1, 2, \dots, 5,$$

where

$$D_{ij} = \sum_{p=\max(i,j)}^5 \text{Trace} \left( \Gamma_{pj} \mathbf{J}_p \Gamma_{pi}^T \right),$$

$$D_{ijk} = \sum_{p=\max(i,j,k)}^5 \text{Trace} \left( \Gamma_{pj} \mathbf{J}_p \Gamma_{pi}^T \right), \quad (108)$$

$$D_i = \sum_{p=i}^5 -m_p \mathbf{g}^T \Gamma_{pi} \mathbf{r}_{lcp}.$$

The following derivations of transformation relationship, between equations of motion  $\mathbf{U} = [F_1, F_2, F_3, \tau_4, \tau_5]^T$  (represented by the generalized coordinates  $\mathbf{q} = [X_{S1}, X_{S2}, X_{S3}, \phi_4, \phi_5]^T$ ) and equations of motion  $\mathbf{U} = [F_1, F_2, F_3, \tau_4, \tau_5]^T$  (represented by the local generalized variables  $\boldsymbol{\psi} = [d_1, \phi_2, d_3, \phi_4, \phi_5]^T$ ), can be yielded as follows.

TABLE 5: The mass ( $m_i$ ), Pseudo Inertia Matrix ( $J_i$ ), and the location of the center of mass ( $\mathbf{r}_{lci}$ ).

Number	$m_i$	$J_i$	$\mathbf{r}_{lci}$
1	0	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$
2	83.025	$\begin{bmatrix} 282114.412 & -61.008 & 1019.065 & 0 \\ -61.008 & 2085578.353 & -15.858 & 0 \\ 1019.065 & -15.858 & 3009592.428 & -3668.210 \\ 0 & 0 & -3668.210 & 83.025 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -44.182 \\ 1 \end{bmatrix}$
3	21.333	$\begin{bmatrix} 901260.608 & -3970.014 & -748196.633 & -6497.242 \\ -3970.014 & 2085578.353 & -8327.223 & -27.860 \\ -748196.633 & -8327.223 & 3009592.428 & -3668.210 \\ -6497.242 & -27.860 & -3668.210 & 21.333 \end{bmatrix}$	$\begin{bmatrix} -304.563 \\ -1.306 \\ -278.925 \\ 1 \end{bmatrix}$
4	8.022	$\begin{bmatrix} 6071.881 & -0.114 & 0.002 & 0 \\ -0.114 & 103321.719 & 46002.096 & -510.889 \\ 0.002 & 46002.096 & 38944.257 & -3755.403 \\ 0 & -510.889 & -3755.403 & 8.022 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -63.686 \\ -468.138 \\ 1 \end{bmatrix}$
5	9.120	$\begin{bmatrix} 5331.503 & 27.342 & -0.849 & 0.501 \\ 27.342 & 88745.074 & -18452.465 & -482.530 \\ -0.849 & -18452.465 & 30828.605 & -2382.517 \\ 0.501 & -482.530 & -2382.517 & 9.120 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -52.909 \\ -261.241 \\ 1 \end{bmatrix}$

The relationship between forces  $F_1$  and  $F_2$ , respectively, produced by the generalized coordinates  $X_{S1}$  and  $X_{S2}$  and  $T_1$  and  $T_2$ , respectively, produced by the virtual generalized coordinate  $d_1$  and  $\phi_2$  can be derived using the principle of virtual work and (90) as

$$F_1 \delta X_{S1} + F_2 \delta X_{S2} = T_1 \delta d_1 + T_2 \delta \phi_2$$

$$\delta d_1 = \frac{\delta X_{S1} + \delta X_{S2}}{2} \quad (109)$$

$$\delta \phi_2 = \frac{\cos^2(\alpha) (\delta X_{S2} - \delta X_{S1})}{L_1}.$$

Based on (6) and (109), it yields

$$F_1 = \frac{T_1}{2} - \frac{T_2 L_1}{(X_{S2} - X_{S1})^2 + L_1^2}, \quad (110)$$

$$F_2 = \frac{T_1}{2} + \frac{T_2 L_1}{(X_{S2} - X_{S1})^2 + L_1^2}.$$

Similarly, based on (92), it gives

$$\begin{aligned} F_3 &= T_3, \\ \tau_4 &= T_4, \\ \tau_5 &= T_5. \end{aligned} \quad (111)$$

From the above derivations in this section, the dynamic equations of motion represented by the local generalized variables  $\boldsymbol{\psi} = [d_1, \phi_2, d_3, \phi_4, \phi_5]^T$  and the generalized coordinates  $\mathbf{q} = [X_{S1}, X_{S2}, X_{S3}, \varphi_4, \varphi_5]^T$  for the studied 5-DOF robot manipulator are obtained, respectively.

Silver compared the computational efficiency between Newton–Euler and Lagrangian formulations for manipulator dynamics and proved that they are indeed equivalent to each other [46]. There is in fact no fundamental difference in computational efficiency between two. The former is significantly more efficient than the latter due to two factors: the recursive structure of the computation and the representation chosen for the rotational dynamics. Namely, the former works directly with Newton's and Euler's equations for a rigid body, which are contained within the spatial equation of motion. This formulation is especially amenable to the development of efficient recursive algorithms for dynamics computations. The latter only considers the total robot energy and proceeds via the Lagrangian of the robot mechanism; then the dynamic equations of motion are developed using Lagrange's equation for each defined generalized coordinate.

### 3. Problem Formulation

For a given prespecified trajectory in the task space of this 5-DOF robot manipulator in this paper, the aim is to

optimize a trajectory and calculate the maximum DLCC. This can be formulated as an optimization problem which optimizes four objective functions considering the robot manipulator physical constraints, payload constraint, and actuator limitations.

**3.1. Cost Function and Constraints.** In this paper, the cost function is the weighted sum of four objective functions representing traveling time, total energy involved in the motion of the robot manipulator, joint jerks, and joint acceleration, respectively. The cost function is considered as the evaluation criterion for the optimal trajectory planning problem.

In practice, the results of the multiobjective optimization problems are influenced by the values of the weighting coefficients. Hence, appropriate values are required to characterize the importance of the corresponding decision criteria for the trajectory planning applications. Multiple objectives can be combined into a scalar cost function via a weight vector. Weights may be assigned through the direct assignment, the eigenvector method, the empty method, the minimal information method, the random determination, or the adaptation determination. In this paper, it is considered that minimizing operation time (minimum traveling time) and saving energy (minimizing total energy involved in the motion) take the highest priority over others. The next priority is applied for reducing vibrations of robot manipulator joints (minimizing joint jerks and acceleration). Thus, a corresponding vector of normalized weights,  $w_1 = 0.3$ ,  $w_2 = 0.3$ ,  $w_3 = 0.2$  and  $w_4 = 0.2$ , is assigned for the objective functions in the optimization problem.

Significant differences may be made in the magnitude of each function when the objective functions are simply weighted and used to produce a single cost function, resulting in a dominating effort that is mainly caused by the function corresponding to the largest magnitude. In particular, the larger the magnitude of an objective function, the higher the influence on the cost function. Hence, a feasible and efficient means will be needed to deal with this problem effected by the large magnitude of the objective function. One possibility of avoiding this problem is to make an adequate normalization of the objective functions and formulate the same magnitude of objective functions. It can have a roughly equivalent influence on the objective functions for the cost function. Also normalizing parameters make all objective functions as unitless functions. In this paper, the normalized weighted objective functions method is used to balance the influence on four objective functions and select the best found solution. More details on this method can be referred to [47, 48]. The values of  $\xi_1 = 1$ ,  $\xi_2 = 1000$ ,  $\xi_3 = 10$ , and  $\xi_4 = 1$  are taken as the normalizing parameters of the objective functions  $z_1$ ,  $z_2$ ,  $z_3$ , and  $z_4$ .

As mentioned above, joint variables  $\mathbf{q} = [X_{S1} \ X_{S2} \ X_{S3} \ \varphi_4 \ \varphi_5]^T$  of this robot manipulator form a vector of generalized coordinates. Likewise, the variables  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}$ ,  $\dddot{\mathbf{q}}$ , and  $\mathbf{U} = [F_1 \ F_2 \ F_3 \ \tau_4 \ \tau_5]^T$  are generalized velocities, acceleration, jerks, and forces.

Given the following date, (a) the initial time  $t_0$  ( $t_0 \geq 0$ ); the final time  $t_f$  which is unknown and must be determined

(see below); (b) the coordinates of the desired inertial position trajectory of the 5-DOF robot end-effector specified in two forms (see Section 2.1.3), the resulting desired initial and final conditions of the generalized coordinates  $\mathbf{q}_d(t_0)$  and  $\mathbf{q}_d(t_f)$  are calculated by using (29)–(32); set  $\mathbf{q}(t_0) = \mathbf{q}_d(t_0)$  and  $\mathbf{q}(t_f) = \mathbf{q}_d(t_f)$ ; (c) the initial and final conditions are  $\dot{\mathbf{q}}(t_0)$ ,  $\dot{\mathbf{q}}(t_f)$ ,  $\ddot{\mathbf{q}}(t_0)$ ,  $\ddot{\mathbf{q}}(t_f)$ , and  $\ddot{\mathbf{q}}(t_f)$ ; thus, the basic multiobjective optimization problem is stated as follows.

Compute the final time  $t_f$ , the payload mass  $m_p$ , and a smooth 5-DOF robot motion trajectory ( $\mathbf{q}(t)$ ,  $\dot{\mathbf{q}}(t)$ ,  $\ddot{\mathbf{q}}(t)$ ,  $\ddot{\mathbf{q}}(t)$ )  $t \in [t_0, t_f]$  satisfying the above-mentioned initial and final conditions so as to minimize the cost function  $F_c$  given by

$$F_c = \frac{w_1 z_1}{\xi_1} + \frac{w_2 z_2}{\xi_2} + \frac{w_3 z_3}{\xi_3} + \frac{w_4 z_4}{\xi_4}, \quad (112)$$

Where one has the following:

Total traveling time between initial and final configurations:

$$z_1 = t_f - t_0. \quad (113)$$

Total energy involved in the motion of the robot manipulator:

$$z_2 = \int_{t_0}^{t_f} \sum_{j=1}^5 (|U_j^T(t)| |\dot{q}_j(t)|)^2 dt. \quad (114)$$

Integral of joint jerks squared:

$$z_3 = \int_{t_0}^{t_f} \sum_{j=1}^5 (|\ddot{q}_j(t)|)^2 dt. \quad (115)$$

Integral of joint acceleration squared:

$$z_4 = \int_{t_0}^{t_f} \sum_{j=1}^5 (|\dot{q}_j(t)|)^2 dt. \quad (116)$$

subject to the following constraints:

The final time constraint:

$$t_f \geq t_0. \quad (117)$$

The displacement constraint:

$$QC_{\min.j} \leq q_j(t) \leq QC_{\max.j} \quad (118)$$

$$j = 1, \dots, 5, t \in [t_0, t_f].$$

The velocity constraint:

$$|\dot{q}_j(t)| \leq VC_j \quad j = 1, \dots, 5, t \in [t_0, t_f]. \quad (119)$$

The acceleration constraint:

$$|\ddot{q}_j(t)| \leq WC_j \quad j = 1, \dots, 5, t \in [t_0, t_f]. \quad (120)$$

The jerk constraint:

$$|\dddot{q}_j(t)| \leq JC_j \quad j = 1, \dots, 5, t \in [t_0, t_f]. \quad (121)$$

The force/torque constraint:

$$|U_j(t)| \leq UC_j \quad j = 1, \dots, 5, t \in [t_0, t_f]; \quad (122)$$

that is,  $|F_1(t)| \leq UC_1, |F_2(t)| \leq UC_2, |F_3(t)| \leq UC_3, |\tau_4(t)| \leq UC_4, |\tau_5(t)| \leq UC_5, t \in [t_0, t_f]$ .

The payload constraint:

$$m_{p\_min} \leq m_p \leq m_{p\_max}. \quad (123)$$

The kinematic constraints: that is, (29)–(32) (which are specific specified in Section 2.1).

The dynamic constraints: that is, (75), (79), (82), and (86)–(88) (which are specific specified in Section 2.2 are also considered for the multiobjective optimization problem).

Here,  $QC_{min\_j}$  and  $QC_{max\_j}$  are the  $j$ th joint lower and upper bounds of displacement, respectively.  $VC_j, WC_j, JC_j,$  and  $UC_j$  are the  $j$ th joint upper bounds of velocity, acceleration, jerk, and force/torque, respectively.  $m_{p\_min}$  and  $m_{p\_max}$  are the minimum and maximum allowable load corresponding to the DLCC of the robot manipulator, respectively.

**3.2. Finite Dimensional 5-DOF Robot Motion Trajectory Representation.** The cubic spline in trajectory planning is very useful, because the generated trajectories have continuous values of the acceleration; also, unlike higher-order polynomials, cubic splines would not generate problems such as excessive oscillations and overshoot between any pair of reference points.

In this paper, cubic splines are applied to represent the trajectory of the joint space, given that the desired inertial position trajectory of the 5-DOF robot end-effector is specified in parametric form (see Section 2.1.3). Also, given a positive integer  $m_0$ , select an increasing sequence of  $m_0$  parameter values  $b_1, b_2, \dots, b_{m_0}$  lying in the parameter interval  $[b_0, b_f]$  such that

$$b_0 = b_1 < b_2 < \dots < b_{m_0} = b_f. \quad (124)$$

Thus,  $m_0$  via-points points are obtained on the desired inertial position trajectory of the 5-DOF robot end-effector with coordinates

$$(x_d(b_i), y_d(b_i), z_d(b_i)) \quad i = 1, 2, \dots, m_0. \quad (125)$$

By using the inverse kinematics of the 5-DOF robot (see (29)–(32)),  $m_0$  via-points are obtained on the desired trajectory of the 5-DOF robot generalized coordinates  $\mathbf{q}_d(t)$  at a sequence of  $m_0$  increasing time instants indexed as follows:

$$t_0 = t_1 < t_3 < \dots < t_{m-2} < t_m, \quad (126)$$

where the initial time  $t_0$  is given and  $m = m_0 + 2$ .

An additional two via-points  $\mathbf{q}_d(t_2)$  and  $\mathbf{q}_d(t_{m-1})$ , corresponding to times  $t_2$  and  $t_{m-1}$ , are added in order to accommodate the specified initial and final conditions  $\dot{\mathbf{q}}(t_1), \ddot{\mathbf{q}}(t_1), \dot{\mathbf{q}}(t_m)$ , and  $\ddot{\mathbf{q}}(t_m)$  (see below).

Thus, a total of  $m$  via-points on the desired trajectory of the 5-DOF robot generalized coordinates  $\mathbf{q}_d(t)$  are known and are denoted as follows:

$$q_{j,i} = q_{dj}(t_i) \quad j = 1, 2, \dots, 5, \quad i = 1, 2, \dots, m. \quad (127)$$

What is not known and must be determined is the sequence of  $m - 1$  increasing time instants,

$$t_0 = t_1 < t_2 < t_3 < \dots < t_{m-2} < t_{m-1} < t_m. \quad (128)$$

The sequence of time instants  $t_i, i = 1, 2, \dots, m$ , defines  $m - 1$  consecutive time intervals  $I_i$  given by

$$I_i = [t_i, t_{i+1}], \quad i = 1, 2, \dots, m - 1. \quad (129)$$

Let  $h_i$  denote the length of the time interval  $I_i$ ; that is,  $h_i = t_{i+1} - t_i, i = 1, 2, \dots, m - 1$ .

Given the above-mentioned  $m$  via-points and the time-interval lengths  $h_i, i = 1, 2, \dots, m - 1$ , then the actual motion trajectory of the 5-DOF robot is constructed by using a cubic spline segment to connect every pair of consecutive via-points; see (127).

In this section, a methodology is proposed for computing the unknown time-interval lengths  $h_i, i = 1, 2, \dots, m - 1$ , such that the resulting actual motion trajectory of the 5-DOF robot minimizes a cost function subject to constraints (see later).

Let  $Q_{j,i}(t)$  be the cubic polynomial for the  $j$ th joint defined on the interval  $I_i = [t_i, t_{i+1}]$ . Solving the interpolation problem means to find  $m - 1$  cubic polynomials satisfying the imposed conditions. In order to avoid considering cumbersome calculations with  $4(m - 1)$  unknowns, it can be adopted that the second derivative of  $Q_{j,i}(t)$  is a linear function of  $t$  in the interval  $I_i$ ; namely,

$$\ddot{Q}_{j,i}(t) = \frac{t_{i+1} - t}{h_i} \ddot{Q}_{j,i}(t_i) + \frac{t - t_i}{h_i} \ddot{Q}_{j,i}(t_{i+1}) \quad (130)$$

$$t \in I_i, \quad i = 1, \dots, m - 1, \quad j = 1, 2, \dots, 5.$$

Integrating (130) and setting  $Q_{j,i}(t_i) = q_{j,i}$  and  $Q_{j,i}(t_{i+1}) = q_{j,i+1}$ , the following interpolating functions are obtained:

$$\dot{Q}_{j,i}(t) = \frac{-(t_{i+1} - t)^2 \ddot{Q}_{j,i}(t_i)}{2h_i} + \frac{(t - t_i)^2 \ddot{Q}_{j,i}(t_{i+1})}{2h_i} + \left( \frac{q_{j,i+1}}{h_i} - \frac{h_i \ddot{Q}_{j,i}(t_{i+1})}{6} \right) - \left( \frac{q_{j,i}}{h_i} - \frac{h_i \ddot{Q}_{j,i}(t_i)}{6} \right), \quad (131)$$

$$t \in I_i, \quad i = 1, \dots, m - 1, \quad j = 1, 2, \dots, 5,$$

$$Q_{j,i}(t) = \frac{(t_{i+1} - t)^3 \ddot{Q}_{j,i}(t_i)}{6h_i} + \frac{(t - t_i)^3 \ddot{Q}_{j,i}(t_{i+1})}{6h_i} + \left( \frac{q_{j,i+1}}{h_i} - \frac{h_i \ddot{Q}_{j,i}(t_{i+1})}{6} \right) (t - t_i) + \left( \frac{q_{j,i}}{h_i} - \frac{h_i \ddot{Q}_{j,i}(t_i)}{6} \right) (t_{i+1} - t), \quad (132)$$

$$t \in I_i, \quad i = 1, \dots, m - 1, \quad j = 1, 2, \dots, 5.$$



The values of the velocity at the instant ends are given by

$$\begin{aligned}\dot{Q}_{j,i}(t_i) &= \frac{-h_i a_{j,i}}{2} + \frac{(q_{j,i+1} - q_{j,i})}{h_i} \\ &\quad - \frac{h_i(a_{j,i+1} - a_{j,i})}{6}, \\ &\quad j = 1, \dots, 5, \quad i = 1, \dots, m-1, \\ \dot{Q}_{j,i}(t_{i+1}) &= \frac{h_i a_{j,i+1}}{2} + \frac{(q_{j,i+1} - q_{j,i})}{h_i} \\ &\quad - \frac{h_i(a_{j,i+1} - a_{j,i})}{6}, \\ &\quad j = 1, \dots, 5, \quad i = 1, \dots, m-1.\end{aligned}\quad (144)$$

Hence, the semi-infinite constraints (26) can be expressed much more accurately as

$$\max \left\{ \left| \dot{Q}_{j,i}(t_{j,i}^*) \right|, \left| \dot{Q}_{j,i}(t_i) \right|, \left| \dot{Q}_{j,i}(t_{i+1}) \right| \right\} \leq VC_j \quad (145)$$

$$j = 1, \dots, 5 \quad i = 1, \dots, m-1$$

which represent a finite number of relationships that must be satisfied for the planned trajectory.

The kinematic constraints on acceleration and jerk can be similarly reduced to a finite form. We note that the acceleration is linear in the interval  $I_i = [t_i, t_{i+1}]$  and therefore it must reach its maximum value at one of two interval ends. Hence, the semi-infinite constraints on the acceleration,

$$\left| \ddot{Q}_{j,i}(t) \right| \leq WC_j \quad (146)$$

$$t \in I_i, \quad j = 1, \dots, 5, \quad i = 1, \dots, m-1,$$

can be expressed in a finite form as

$$\max \left\{ |a_{j,1}|, \dots, |a_{j,m}| \right\} \leq WC_j \quad j = 1, \dots, 5. \quad (147)$$

The jerk can be obtained as follows:

$$\begin{aligned}\ddot{Q}_{j,i}(t) &= \frac{(a_{j,i+1} - a_{j,i})}{h_i}, \\ &\quad t \in I_i, \quad j = 1, \dots, 5, \quad i = 1, \dots, m-1.\end{aligned}\quad (148)$$

As for the jerk, the semi-infinite constraint is given by

$$\left| \ddot{Q}_{j,i}(t) \right| \leq JC_j \quad (149)$$

$$t \in I_i, \quad j = 1, \dots, 5, \quad i = 1, \dots, m-1.$$

Since the jerk is constant on any interval  $I_i = [t_i, t_{i+1}]$ , the finite form of the jerk yields

$$\left| \frac{(a_{j,i+1} - a_{j,i})}{h_i} \right| \leq JC_j \quad (150)$$

$$j = 1, \dots, 5, \quad i = 1, \dots, m-1.$$

As for the force/torque, the semi-infinite constraint gives

$$\left| u_{j,i}(t) \right| \leq UC_j, \quad (151)$$

$$t \in I_i, \quad j = 1, \dots, 5, \quad i = 1, \dots, m-1;$$

that is,  $|F_{1,i}(t)| \leq UC_1$ ,  $|F_{2,i}(t)| \leq UC_2$ ,  $|F_{3,i}(t)| \leq UC_3$ ,  $|\tau_{4,i}(t)| \leq UC_4$ , and  $|\tau_{5,i}(t)| \leq UC_5$ ,  $i = 1, \dots, m-1$ .

Moreover, in the optimization of the trajectory, the time intervals  $h_i$  ( $h_i > 0$ ,  $i = 1, \dots, m-1$ ) which are the true optimization parameters (i.e., the decision variables) must be particularly considered. Since in any interval the condition  $|(q_{j,i+1} - q_{j,i})/h_i| \leq VC_j$  must be satisfied. Hence, the lower bound of the intervals  $h_i$  must be subjected to

$$h_i > w_i = \max_{j=1, \dots, 5} \left\{ \frac{|q_{j,i+1} - q_{j,i}|}{VC_j} \right\} > 0, \quad (152)$$

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

Thus, the finite dimensional multiobjective optimization problem is formulated as follows. Compute  $\mathbf{x} \in R^{m-1}$  so as to minimize the cost function  $F_c$ :

$$F_c = \frac{w_1 z_1}{\xi_1} + \frac{w_2 z_2}{\xi_2} + \frac{w_3 z_3}{\xi_3} + \frac{w_4 z_4}{\xi_4}, \quad (153)$$

where

$$\begin{aligned}z_1 &= \sum_{i=1}^{m-1} h_i, \\ z_2 &= \sum_{j=1}^5 \sum_{i=1}^{m-1} \frac{h_i}{2} \left[ \left( |u_{j,i}^T(t_i)| \left| \dot{Q}_{j,i}(t_i) \right| \right)^2 \right. \\ &\quad \left. + \left( |u_{j,i+1}^T(t_{i+1})| \left| \dot{Q}_{j,i+1}(t_{i+1}) \right| \right)^2 \right], \\ z_3 &= \sum_{j=1}^5 \sum_{i=1}^{m-2} \frac{h_i}{2} \left[ \left( \left| \frac{(a_{j,i+1} - a_{j,i})}{h_i} \right| \right)^2 \right. \\ &\quad \left. + \left( \left| \frac{(a_{j,i+2} - a_{j,i+1})}{h_{i+1}} \right| \right)^2 \right], \\ z_4 &= \sum_{j=1}^5 \sum_{i=1}^{m-1} \frac{h_i}{2} \left[ \left( |a_{j,i}| \right)^2 + \left( |a_{j,i+1}| \right)^2 \right],\end{aligned}\quad (154)$$

subject to the constraints (which are (140), (141), (147), (150)–(152), (29)–(32), (75), (79), (82), (86), and (87)).

The vector of decision variables  $\mathbf{x}$  is defined by the sequence of the time intervals  $h_i$  associated with each two consecutive via-points; thus  $\mathbf{x} = [h_1, h_2, \dots, h_{m-1}]^T$ . When the intervals  $h_i$  and joint displacements (i.e., these two extra via-points and joint displacements  $q_{j,i}$  obtained by solving the inverse kinematics of the 5-DOF robot manipulator) are determined, other functions  $Q_{j,i}(t)$ ,  $\dot{Q}_{j,i}(t)$ ,  $\ddot{Q}_{j,i}(t)$ ,  $\ddot{Q}_{j,i}(t)$ , and  $u_{j,i}(t)$  can also be determined by these time intervals and joint displacements. The initial and final velocity ( $\dot{q}_{j,1}$

and  $\dot{q}_{j,m}$ ) and acceleration ( $\ddot{q}_{j,1}$  and  $\ddot{q}_{j,m}$ ) for all joints  $j = 1, 2, \dots, 5$  are set to zero corresponding to the initial and final time instants. The payload mass  $m_p$  is fixed to a feasible value satisfying (123).

A genetic optimization algorithm NSGA-II is applied in order to try and solve the above-mentioned optimization problem. The algorithm NSGA-II runs for a number of iterations and then stops after meeting some termination criterion, thus producing a vector of decision variables  $\mathbf{x}^*$  in solving trajectory planning and DLCC of the 5-DOF robot. These obtained decision variables  $\mathbf{x}^*$  and other above-mentioned functions can be represented as the best found solution that have been able to find so far.

Actually, the real advantages of NSGA-II used in this application compared to the conventional constrained optimization algorithms (e.g., sequential quadratic programming) are that the former is population-based approach search performing the genetic operations like crossover and mutation for the elements of the decision variables (where the unfit individuals are replaced by the fit individuals through the crossover and mutation operator in population), thus discovering the global optimal solution is possible; also, the former is no need to provide any auxiliary relationships (e.g., gradients and Hessian matrix); moreover, the latter is easy to fall into local minima as the susceptibility to initial values, and it needs to calculate the gradients and the Hessian matrix of the objective functions and constraints resulting in more computational time. In Section 6, a classical nonlinear constrained optimization function (i.e., the function `fmincon` in Matlab Optimization Toolbox) based on sequential quadratic programming algorithm is also applied to solve the above-mentioned optimization problem and compared with the best found solution of the genetic optimization algorithm NSGA-II.

#### 4. The Proposed Approach for Trajectory Planning

In this section, an approach is proposed to deal with the multiobjective optimization problem for a given prespecified trajectory taking into account a feasible DLCC and obtain time intervals corresponding to the via-points, joint displacement, velocity, acceleration, jerk, and force/torque, considering the objective functions and two categories of constraints (inequality linear constraints and equality nonlinear constraints). This feasible DLCC is marked as  $DLCC_{fe}$  and must fall within the DLCC range of reachable values. The detailed calculation for maximum DLCC will be presented in the next section.

Note that the cost function results would converge after a certain number of generations by using the calculation mechanism of the NSGA-II algorithm. In this paper, this stabilized condition is made as the termination criterion, and the corresponding calculation results are taken as the best found solution for the trajectory planning problem and maximum DLCC calculation of this 5-DOF robot manipulator.

The desired joint displacement (i.e., the via-points in the joint space) can be determined by solving the robot manipulator inverse kinematics problem. When the optimization process is performed, the equality nonlinear constraints

and inequality linear constraints described in the previous section are considered in the objective functions. The suitable population size (pop), total number of generations (gen), crossover probability (pc), mutation probability (pm), crossover distribution index (dc), and mutation distribution index (dm) in NSGA-II algorithm are needed to select so that the algorithm leads the robot manipulator to have a smooth trajectory and minimum chattering in profiles of velocity, acceleration, jerk, and force/torque.

The detailed description of the proposed approach for trajectory planning is presented in the following steps.

*Step 1.* Calculate desired joint displacement (i.e., the via-points in the joint space) according to the given desired trajectory in the task space by solving the robot manipulator inverse kinematics problem using (29)–(32).

*Step 2.* Select  $DLCC_{fe}$ , pop, gen, pc, pm, dc, and dm, set the initial and final velocity ( $\dot{q}_{j,1}$  and  $\dot{q}_{j,m}$ ) and acceleration ( $\ddot{q}_{j,1}$  and  $\ddot{q}_{j,m}$ ) for all joints  $j = 1, 2, \dots, 5$ , set the constraints of displacement, velocity, acceleration, jerk, and force/torque, and calculate the lower bound of the time intervals using (152).

*Step 3.* A set of small time-interval lengths  $h_i^0$ ,  $i = 1, \dots, m-3$  (except the time intervals of two extra via-points), are randomly generated using (152) and the constraints  $QC_{min}$ ,  $QC_{max}$ ,  $VC$ ,  $WC$ ,  $JC$ , and  $UC$  need to be satisfied, then, respectively, interpolating randomly at the first and last time-interval lengths (i.e.,  $h_1^0$  and  $h_{m-3}^0$ ) for the time intervals of two extra via-points generates new time-interval lengths  $h_{1,1}^0$ ,  $h_{1,2}^0$ ,  $h_{m-3,1}^0$ , and  $h_{m-3,2}^0$ , and finally assembling  $h_{1,1}^0$ ,  $h_{1,2}^0$ ,  $h_2^0$ ,  $h_3^0, \dots, h_{m-4}^0$ ,  $h_{m-3,1}^0$ , and  $h_{m-3,2}^0$  constructs the initialized time-interval lengths  $h_i$ ,  $i = 1, \dots, m-1$ ; generate two extra via-points using (133), and arrange the displacement  $Q$  in the NSGA-II algorithm.

*Step 4.* If lower bound constraints of the time intervals are satisfied (which are judged by (152)), go back to Step 5. Otherwise, go back to Step 3.

*Step 5.* Calculate  $\dot{Q}$ ,  $\ddot{Q}$ ,  $\dddot{Q}$ , and  $U$  using (131), (137), and (148) and the dynamic equations (75), (79), (82), (86), and (87), respectively. If the constraints (displacement, velocity, acceleration, jerk, and force/torque) are satisfied (which are judged by (140), (145), (147), (150), and (151), resp.), go back to Step 6. Otherwise, go back to Step 3.

*Step 6.* Optimize the time intervals (i.e., the decision variables) and objective functions (which are formulated as (154)) by using the calculation mechanism of the NSGA-II algorithm, compare the corresponding cost function result (i.e., (153)) of each population for the current generation, and find out the minimum one ( $F_c^i = \min\{F_c^{i,1}, \dots, F_c^{i,p}\}$ , where  $i$  denotes the current  $i$ th generation and  $p$  is the number of the total population) and the corresponding best found calculation results (decision variables, i.e., the time intervals, and the corresponding results of objective functions, displacement, velocity, acceleration, jerk, and force/torque).

Then, save and report these selected best found calculation results for each generation. This step continues until all generations are completed.

*Step 7.* Compare the cost function results of each generation, and find out the minimum one ( $F_c = \min\{F_c^1, \dots, F_c^g\}$ , where  $g$  is the total number of generations) and the corresponding other calculation results for the whole calculation process.

Through this detailed calculation process, the required calculation decision variables, that is, the time intervals, and the corresponding best found calculation values (objective functions, cost function, displacement, velocity, acceleration, jerk, and force/torque), taking equality nonlinear constraints and inequality linear constraints into account, are obtained by using the NSGA-II algorithm.

## 5. The Proposed Approach for DLCC Calculation

In many practical applications, the maximum DLCC must be a feasible one to satisfy certain performance requirements of the robot manipulator. It is appropriate to specify a robot manipulator in terms of useful payload as a function of performance rather than just in terms of maximum capacity. Therefore, a feasible maximum DLCC of the robot manipulator, from a safety and reliability point of view, plays a significant role in practical applications. The dominant constraint that limits the DLCC of a robot manipulator is the limitations on the applied forces/torques in the robot manipulator joints. A feasible maximum DLCC, considering this dominant constraint (i.e., the applied forces/torques), is addressed in this section.

Another approach is proposed to calculate the maximum DLCC for a given prespecified desired trajectory which is described in the task space considering the objective functions and two categories of constraints as mentioned above. Firstly, a feasible and suitable DLCC is introduced for the trajectory planning based on the proposed approach in the above section by solving the multiobjective optimization problem for this given trajectory, and the related best found results (its best found time intervals, joint displacement, velocity, acceleration, and jerk) can be obtained. Then these determined results are taken as the initial conditions to calculate the maximum DLCC and also the motor torque requirements using the iterative method. The process of the iterative solution is repeated until the termination condition (i.e., the motor torque requirements are beyond the rated torques) is reached. Here, at each iteration, the DLCC, that is, the allowable payload mass  $m_p$  that is carried by the robot manipulator, is increased by an iteration procedure.  $m_{p,\text{in}}$  and  $\Delta m_p$  are the initial value and the increment of  $m_p$  at each iteration, respectively.

The proposed approach for the maximum DLCC calculation proceeds in nine steps as follows.

*Step 1.* Calculate desired joint displacement (i.e., the via-points in the joint space) according to the given trajectory

in the task space by solving the robot manipulator inverse kinematics problem using (29)–(32).

*Step 2.* Select  $m_{p,\text{in}}$ ,  $\Delta m_p$ , pop, gen, pc, pm, dc, and dm, set  $m_p = m_{p,\text{in}}$ , set the initial and final velocity ( $\dot{q}_{j,1}$  and  $\dot{q}_{j,m}$ ) and acceleration ( $\ddot{q}_{j,1}$  and  $\ddot{q}_{j,m}$ ) for all joints  $j = 1, 2, \dots, 5$ , set the constraints of displacement, velocity, acceleration, jerk, and force/torque, and calculate the lower bound of the time intervals using (152).

*Step 3.* A set of small time-interval lengths  $h_i^0$ ,  $i = 1, \dots, m-3$  (except the time intervals of two extra via-points), are randomly generated using (152) and the constraints  $QC_{\text{min}}$ ,  $QC_{\text{max}}$ ,  $VC$ ,  $WC$ ,  $JC$ , and  $UC$  need to be satisfied, then, respectively, interpolating randomly at the first and last time-interval lengths (i.e.,  $h_1^0$  and  $h_{m-3}^0$ ) for the time intervals of two extra via-points generates new time-interval lengths  $h_{1,1}^0$ ,  $h_{1,2}^0$ ,  $h_{m-3,1}^0$ , and  $h_{m-3,2}^0$ , and finally assembling  $h_{1,1}^0$ ,  $h_{1,2}^0$ ,  $h_2^0$ ,  $h_3^0, \dots, h_{m-4}^0$ ,  $h_{m-3,1}^0$ , and  $h_{m-3,2}^0$  constructs the initialized time-interval lengths  $h_i$ ,  $i = 1, \dots, m-1$ ; generate two extra via-points using (133), and arrange the displacement  $Q$  in the NSGA-II algorithm.

*Step 4.* If lower bound constraints of the time intervals are satisfied (which are judged by (152)), go back to Step 5. Otherwise, go back to Step 3.

*Step 5.* Calculate  $\dot{Q}$ ,  $\ddot{Q}$ ,  $\ddot{Q}$ , and  $U$  using (131), (137), (148) and the dynamic equations (75), (79), (82), (86), and (87), respectively. If the constraints (displacement, velocity, acceleration, jerk, and force/torque) are satisfied (which are judged by (140), (145), (147), (150) and (151), resp.), go back to Step 6. Otherwise, go back to Step 3.

*Step 6.* Optimize the time intervals (i.e., the decision variables) and objective functions (which are formulated as (154)) by using the calculation mechanism of the NSGA-II algorithm, compare the cost function results of each generation, and find out the minimum one ( $F_c = \min\{F_c^1, \dots, F_c^g\}$ , where  $g$  is the total number of generations) and the corresponding other calculation results (decision variables, i.e., the time intervals, and the corresponding results of objective functions, displacement, velocity, acceleration, jerk, and force/torque). Then, save and report these selected best found results.

*Step 7.* Set  $\text{DLCC} = m_p$ ; increase payload  $m_p$  by  $m_p = m_p + \Delta m_p$ .

*Step 8.* Calculate the required motor torques using the dynamic equations (75), (79), (82), (86), and (87) according to the selected best found results (i.e., the best found displacement  $Q$ , velocity  $\dot{Q}$ , and acceleration  $\ddot{Q}$ ) and the current payload mass  $m_p$ . If the constraints of the rated motor torques are satisfied (which are judged by (151)), go back to Step 7. Otherwise, go back to Step 9.

*Step 9.* Save and report the latest DLCC (i.e., the maximum DLCC).

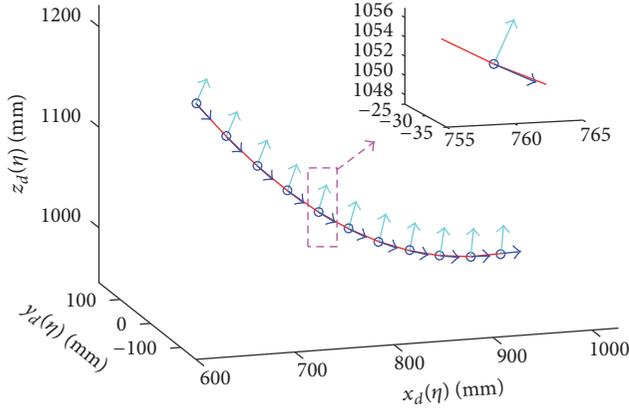


FIGURE 6: The desired end-effector inertial position trajectory.

Through this detailed calculation steps while taking into account equality nonlinear constraints and inequality linear constraints, according to the best found initial conditions (i.e., the selected best found results of displacement  $Q$ , velocity  $\dot{Q}$ , and acceleration  $\ddot{Q}$ ), the maximum DLCC for the given trajectory are determined based on the NSGA-II algorithm and the iterative method.

## 6. Numerical Example

Based on the above proposed approaches, the trajectory planning and the maximum DLCC calculation of this robot manipulator for a given desired end-effector inertial position trajectory are presented in this section.

A saddle line segment is selected as the given desired inertial position trajectory of the 5-DOF robot manipulator end-effector for the trajectory planning and the maximum DLCC calculation. The saddle line segment is defined between initial coordinates ( $x_d = 600$  mm,  $y_d = -150$  mm, and  $z_d = 1200$  mm) and final coordinates ( $x_d = 1000$  mm,  $y_d = 150$  mm, and  $z_d = 950$  mm). In this paper, the equation of this saddle line segment is expressed in implicit form as follows:

$$z_d = f(x_d, y_d) = \frac{(y_d - 100)^2}{20^2} - \frac{x_d^2}{80^2} + 1100. \quad (155)$$

The saddle line segment is also expressed in parametric form as follows:

$$\begin{aligned} x_d &= 600 + 40\eta \\ y_d &= -150 + 30\eta \\ z_d &= \frac{(y_d - 100)^2}{20^2} - \frac{x_d^2}{80^2} + 1100, \end{aligned} \quad (156)$$

where the parameter  $\eta \in [0, 10]$ . In this case  $m_0 = 11$  via-points are chosen on the desired end-effector trajectory leading to the following values for the parameter  $\eta$ :  $\eta_1 = 0, \eta_2 = 1, \dots, \eta_{11} = 10$ .

This saddle line segment (i.e., the prespecified desired trajectory) and each corresponding via-point and orientation in the task space are shown in Figure 6, where the orientation

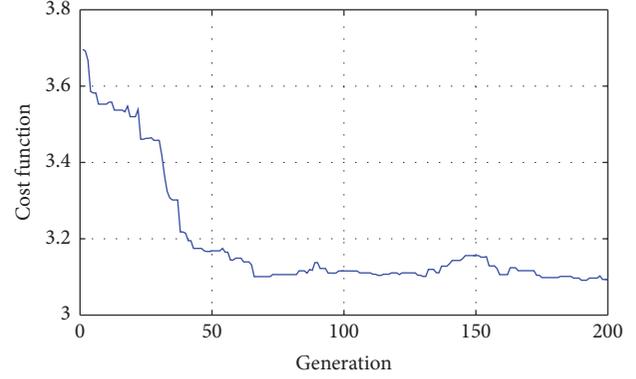


FIGURE 7: The relationship between cost function and generation.

is determined as follows: the vector  $\mathbf{a} = [a_x, a_y, a_z]^T$  is defined along the normal vector of the via-point (cyan arrows as shown in Figure 6),  $\mathbf{o} = [o_x, o_y, o_z]^T$  is defined along the tangent vector of the via-point and points the next via-point (blue arrows as shown in Figure 6), and  $\mathbf{n} = [n_x, n_y, n_z]^T$  is defined by  $\mathbf{n} = \mathbf{o} \times \mathbf{a}$ . The generalized coordinates of this 5-DOF robot manipulator can be represented as  $\mathbf{q} = [X_{S1}, X_{S2}, X_{S3}, \varphi_4, \varphi_5]^T$ . The robot manipulator is at rest initially and comes to a full stop at the end of the trajectory (i.e.,  $\dot{q}_{j,1} = 0, \dot{q}_{j,m} = 0, \ddot{q}_{j,1} = 0$ , and  $\ddot{q}_{j,m} = 0$  for all joints  $j = 1, 2, \dots, 5$ ). In this paper, joint force/torque of this robot manipulator is generated by AC servomotors. The feasible DLCC (i.e.,  $DLCC_{fe}$ ) is taken as a fixed payload mass  $m_p = 10$  kg for solving the trajectory planning problem.

The via-points (i.e., the desired joint displacement) including two extra via-points in the joint space are determined by solving the inverse kinematics of the robot manipulator and given in Table 6. The real variable type is considered in the NSGA-II algorithm. The values of the parameter which is used in algorithm, population size (pop), total number of generations (gen), crossover probability (pc), mutation probability (pm), crossover distribution index (dc), and mutation distribution index (dm), are 100, 200, 0.9, 0.1, 10, and 100, respectively.

The proposed approaches are applied both to trajectory planning problem and maximum DLCC calculation for this 5-DOF robot manipulator, and the related best found solutions are obtained. Note that, without loss of generality, only the case of the rigid robot manipulator is stressed here; other contributions to the dynamic description of the robot manipulator (which may include the dynamics of the actuators, joint and link flexibility, friction, noise, and disturbances) are not considered in this paper. The relationship between the cost function and the total number of generations is shown in Figure 7. It can be found that the cost function value reached the convergence to a stabilized condition from 66th to 200th generation. The relationships between objective functions (total traveling time ( $z_1$ ), total energy involved in the motion ( $z_2$ ), integral of joint jerks squared ( $z_3$ ), and integral of joint acceleration squared ( $z_4$ )) and the corresponding total number of generations (pop) are shown in Figures 8(a), 8(b), 8(c), and 8(d), respectively.

TABLE 6: The desired joint displacement and specified points on desired end-effector inertial position trajectory.

Via-point index $i$	Time instants $t_i$	Joint 1 (mm) $X_{S1,i}$	Joint 2 (mm) $X_{S2,i}$	Joint 3 (mm) $X_{S3,i}$	Joint 4 (rad) $q_{4,i}$	Joint 5 (rad) $q_{5,i}$	$x_d(t_i)$ (mm)	$y_d(t_i)$ (mm)	$z_d(t_i)$ (mm)
1	$t_1$	262.63	142.92	367.10	-1.030	1.028	600	-150	1200
2	$t_2$	264.82	149.16	361.35	-1.0374	1.0253	605.375	-146.070	1194.164
3	$t_3$	279.40	189.61	325.12	-1.082	0.993	640	-120	1157
4	$t_4$	298.23	236.26	288.21	-1.117	0.918	680	-90	1118
5	$t_5$	319.14	284.04	255.31	-1.149	0.838	720	-60	1083
6	$t_6$	341.88	333.02	223.38	-1.179	0.752	760	-30	1052
7	$t_7$	366.21	383.29	201.39	-1.206	0.662	800	0	1025
8	$t_8$	391.95	434.95	180.26	-1.232	0.566	840	30	1002
9	$t_9$	418.93	488.14	162.95	-1.257	0.466	880	60	983
10	$t_{10}$	447.03	543.04	149.39	-1.280	0.361	920	90	968
11	$t_{11}$	476.11	599.93	139.50	-1.303	0.252	960	120	957
12	$t_{12}$	502.37	651.71	134.09	-1.3412	0.2045	993.187	144.873	9.51221
13	$t_{13}$	507.92	662.77	132.92	-1.350	0.197	1000	150	950

TABLE 7: The best found variables (time interval lengths).

$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$	$h_9$	$h_{10}$	$h_{11}$	$h_{12}$
0.330	0.837	0.803	0.569	0.477	0.382	0.348	0.348	0.392	0.552	0.732	0.420

TABLE 8: The best found joint displacement and actual points on end-effector inertial position trajectory.

Via-point index $i$	Time instants $t_i$ (sec)	Joint 1 (mm) $X_{S1,i}$	Joint 2 (mm) $X_{S2,i}$	Joint 3 (mm) $X_{S3,i}$	Joint 4 (rad) $q_4(t_i)$	Joint 5 (rad) $q_5(t_i)$	$x(t_i)$ (mm)	$y(t_i)$ (mm)	$z(t_i)$ (mm)
1	0	262.63	142.92	367.10	-1.030	1.028	599.974	-149.996	1199.998
2	0.330	263.57	145.35	364.92	-1.033	1.027	602.064	-148.530	1197.757
3	1.167	279.40	189.61	325.12	-1.082	0.993	639.988	-120.009	1156.971
4	1.970	298.23	236.26	288.21	-1.117	0.918	680.018	-89.937	1117.988
5	2.530	319.14	284.04	255.31	-1.149	0.838	719.961	-59.994	1082.990
6	3.016	341.88	333.02	223.38	-1.179	0.752	759.998	-30.042	1052.018
7	3.397	366.21	383.29	201.39	-1.206	0.662	799.999	0.022	1025.016
8	3.745	391.95	434.95	180.26	-1.232	0.566	839.951	30.000	1002.038
9	4.093	418.93	488.14	162.95	-1.257	0.466	879.967	59.997	983.048
10	4.485	447.03	543.04	149.39	-1.280	0.361	919.970	89.985	968.014
11	5.037	476.11	599.93	139.50	-1.303	0.252	959.982	119.989	956.999
12	5.769	505.32	667.51	133.42	-1.346	0.1969	996.812	147.566	950.498
13	6.189	507.92	662.77	132.92	-1.350	0.197	999.993	150.042	949.980

The best found variables (i.e., time intervals) obtained from the proposed approach for trajectory planning are given in Table 7. The best found joint displacement is given in Table 8. The best found joint displacement, velocity, acceleration, jerk, and force/torque are shown in Figures 9, 10, 11, 12, and 13, respectively. The actual end-effector inertial position is obtained as shown in Figure 14. The end-effector inertial position tracking error between the actual inertial motion trajectory and the desired inertial position trajectory is shown in Figure 15. The standard deviation tracking error, the variance tracking error, and the maximum tracking error

are 0.0840 mm, 0.0071 mm, and 0.2933 mm (the last one shows the nearby point of the first extra point), respectively.

The best found joint displacement, velocity, and acceleration are taken as the initial conditions (the same as shown in Figures 9, 10, and 11, resp.) to calculate the maximum DLCC and also the motor torque requirements (i.e., requirements of joint force/torque). Note that, without loss of generality, only the case of the rigid robot manipulator is stressed here; other contributions to the dynamic description of the robot manipulator (which may include the dynamics of the actuators, joint and link flexibility, friction, noise, and

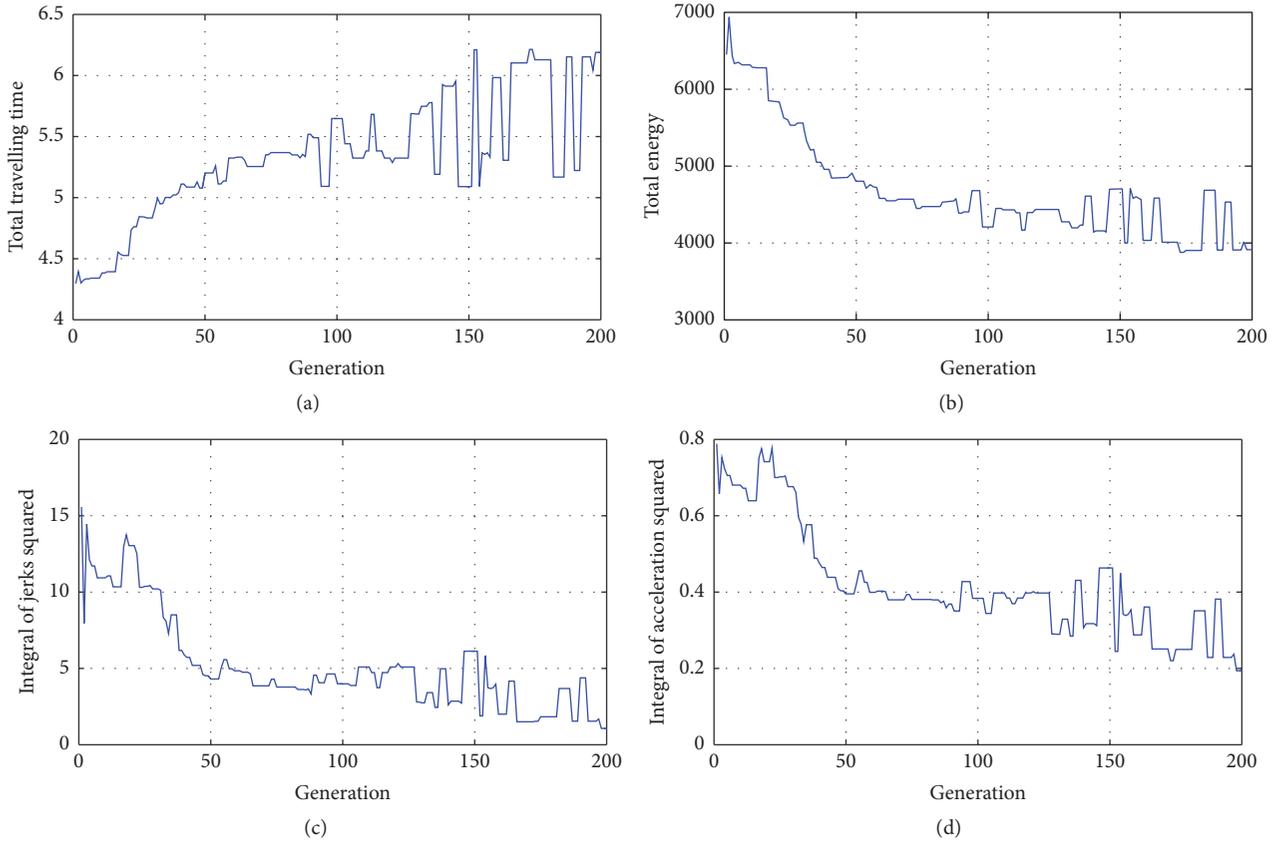


FIGURE 8: The relationship between objective functions and generation: (a) total traveling time; (b) total energy; (c) integral of jerks squared; and (d) integral of acceleration squared.

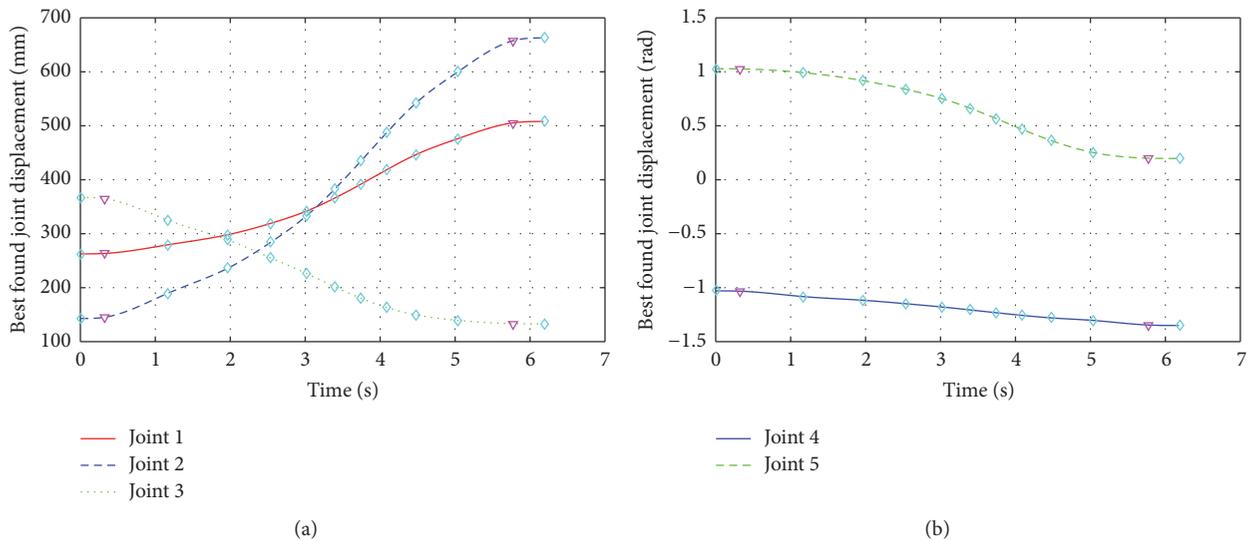
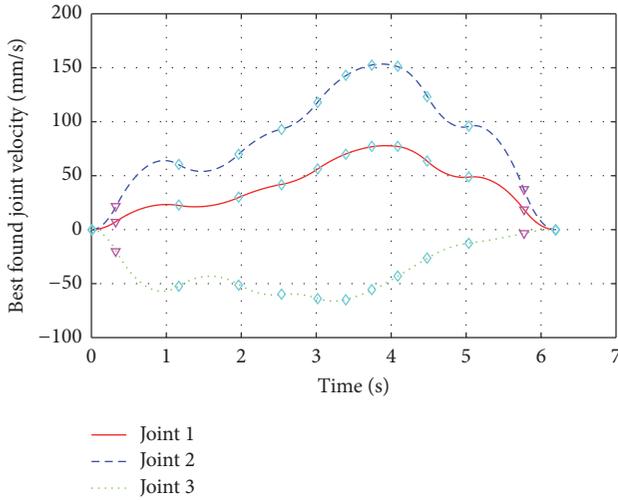


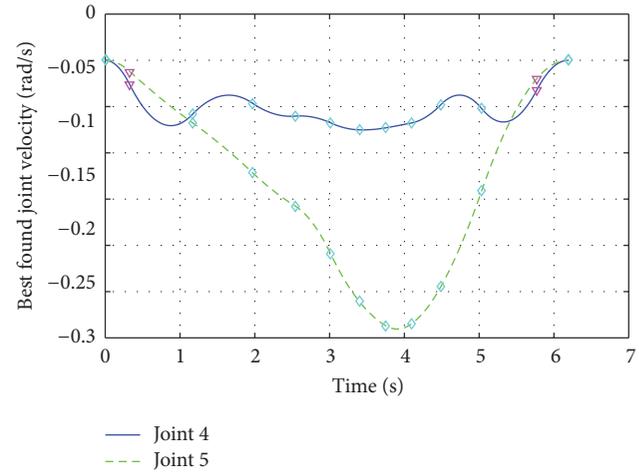
FIGURE 9: The best found joint displacement: (a) joints 1, 2, and 3 and (b) joints 4 and 5.

disturbances) are not considered in this paper. The proposed DLCC calculation approach is applied and the maximum DLCC, that is, the maximum allowable load, is found to be 12.65 kg.

In this paper, the cost function value converges to a stabilized status from 66th to 200th generation, so produced vector of decision variables  $x^*$  (i.e., the time-interval lengths  $h_i, i = 1, 2, \dots, 12$ ) and other above-mentioned functions

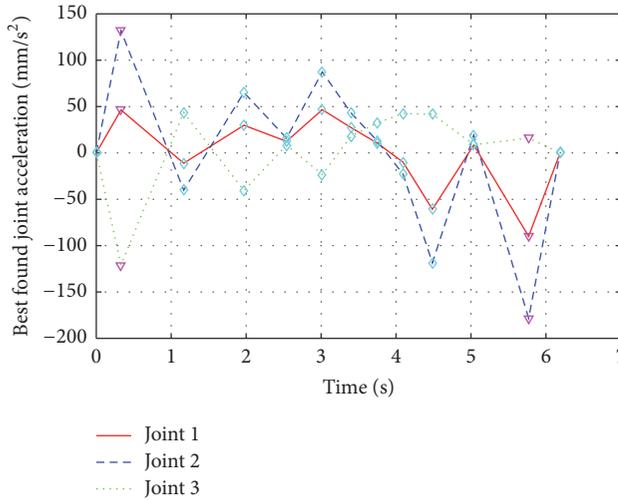


(a)

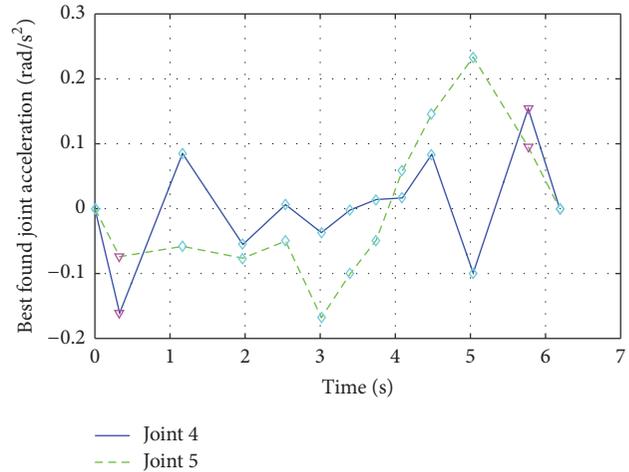


(b)

FIGURE 10: The best found joint velocity: (a) joints 1, 2, and 3 and (b) joints 4 and 5.



(a)



(b)

FIGURE 11: The best found joint acceleration: (a) joints 1, 2, and 3 and (b) joints 4 and 5.

of the last generation are used to represent the solution of trajectory planning and DLCC of the 5-DOF robot. Strictly speaking, it cannot be rigorously proved that this found solution  $\mathbf{x}^*$  is the constrained global minimizer of the cost function  $F_c$  and it only represents the best found solution that have been able to find so far.

The Matlab function of the proposed approach for trajectory planning implementing the optimization algorithm NSGA-II and an example multimodal cost function subject to some nonlinear constraints showing how to call/use the function implementing the NSGA-II algorithm are given in Appendix A and Appendix B (both in Supplementary Materials available online at <http://dx.doi.org/10.1155/2016/1302537>), respectively.

A classical nonlinear programming solver, that is, `fmincon` in Matlab Optimization Toolbox (which is based on

sequential quadratic programming algorithm), is applied to solve the above-mentioned minimum of constrained nonlinear objective functions optimization problem and compared with the best found solution of the genetic optimization algorithm NSGA-II.

Initializing randomly the time-interval lengths  $h_i$ ,  $i = 1, \dots, m - 1$ , just like the initialized way of Step 3 in Section 4 generates the initial value of the `fmincon` function. The same normalized weights ( $w_i$ ,  $i = 1, 2, 3, 4$ ) and normalizing parameters ( $\xi_i$ ,  $i = 1, 2, 3, 4$ ) as mentioned in Section 3 are used in `fmincon` solver. The same processes of initializing way and solving are performed 100 times and generate 100 solving results with 1032.78 seconds. The mean value of calculation results is listed in Table 9. The best found solution of the algorithm NSGA-II for the above-mentioned optimization problem (which is computed with is 74.55 seconds) is also

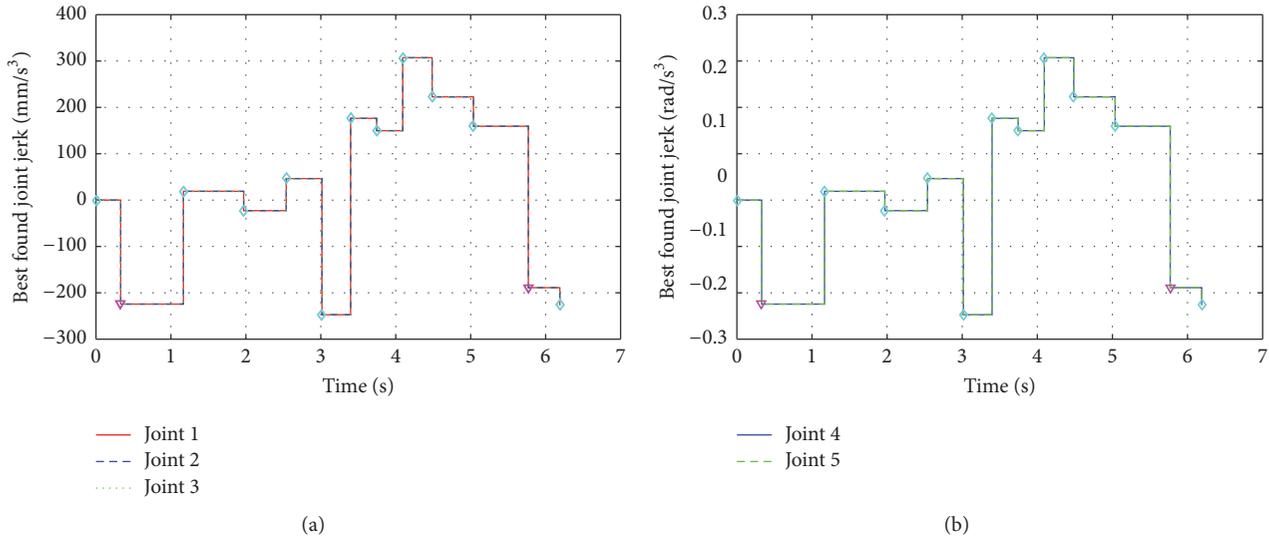


FIGURE 12: The best found joint jerk: (a) joints 1, 2, and 3 and (b) joints 4 and 5.

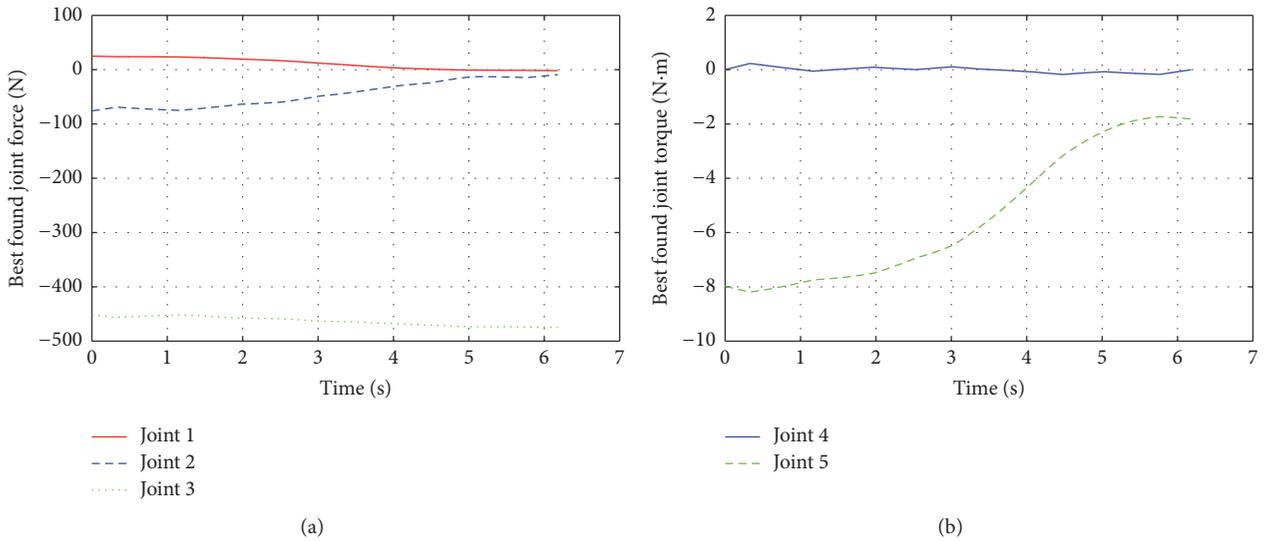


FIGURE 13: The best found joint force/torque: (a) joints 1, 2, and 3 and (b) joints 4 and 5.

listed in Table 9. The results of the cost function and the objective functions in fmincon function solver are given in Appendix C in Supplementary Materials.

Comparing these results, it can be observed that the found solution of the genetic optimization algorithm NSGA-II is better than the fmincon function solver with respect to the cost function ( $F_c$ ), total traveling time ( $z_1$ ), integral of joint jerks squared ( $z_3$ ), and integral of joint acceleration squared ( $z_4$ ). This is because the former is population-based approach which can perform the genetic crossover and mutation operations for the elements of the decision variables and desert the unfit individuals in each generation. Also, these genetic operations can extend and discover the possibly global solution. The computational time of the former is less than the latter, mainly because the algorithm NSGA-II does not need to provide any auxiliary relationships such as the

gradients and Hessian matrix of the objective functions and constraints.

## 7. Conclusions

A new methodology using a direct method is presented for obtaining the best found trajectory planning and maximum dynamic load-carrying capacity (DLCC) for a given prespecified trajectory based on an evolutionary algorithm (i.e., NSGA-II). A cubic spline curve is used to represent the trajectory. This problem is transcribed into a nonlinear constrained multiobjective optimization problem. To summarize, the major contributions of this paper are as follows:

- (i) The nonlinear constrained multiobjective optimization problem is formulated with four objective functions (i.e., total traveling time, total energy involved

TABLE 9: Comparison for the solution of NSGA-II and fmincon solver.

Type	Cost function ( $F_c$ )	Total traveling time ( $z_1$ )	Total energy ( $z_2$ )	Integral of joint jerks squared ( $z_3$ )	Integral of joint acceleration squared ( $z_4$ )
Best found solution of NSGA-II	3.093	6.189	3921.602	1.068	0.194
Mean value solution of fmincon solver	3.220	7.146	3250.239	2.510	0.252

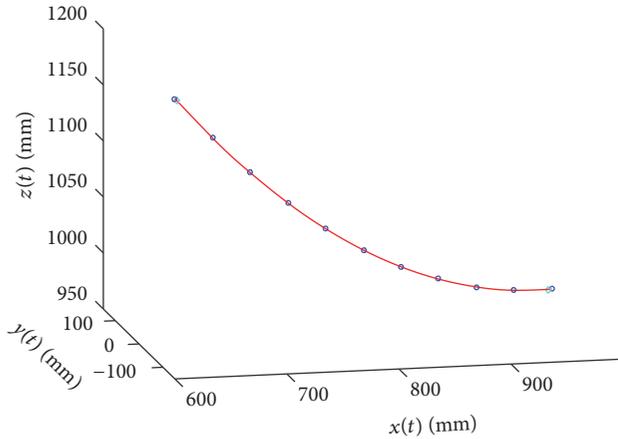


FIGURE 14: Actual end-effector inertial position.

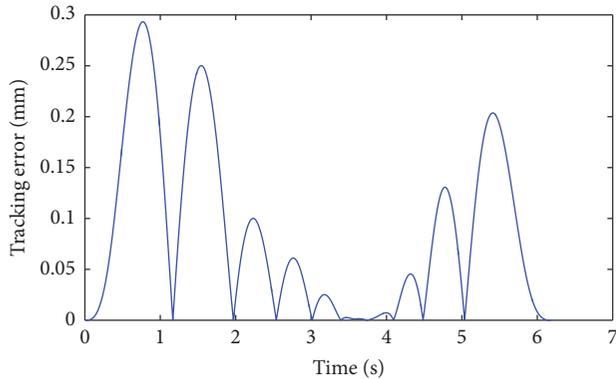


FIGURE 15: End-effector inertial position tracking error.

in the motion, joint jerks, and joint acceleration) and a cost function (which is a weighted sum of these objective functions). These decision criteria can ensure the best found trajectory is smooth and lead to minimum actuator power and energy.

- (ii) Two separate approaches are proposed to deal with trajectory planning problem and maximum DLCC calculation for a 5-DOF robot manipulator, respectively. These approaches are easy to program and implement and can also be extended to handle similar problems for other types of the robots.
- (iii) A numerical example verified the effectiveness of the two proposed approaches and obtained best found solutions for trajectory planning and maximum DLCC calculation of a 5-DOF robot manipulator.

In order to get more accurate and smoother trajectory, the nonuniform rational B-spline (NURBS) curve with more via-points will be used to represent the trajectory in future work. The presented new methodology based on the evolutionary optimization technique and two proposed approaches are expected to provide some insight into the foundational aspects of trajectory planning and maximum DLCC calculation.

### Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

### Acknowledgments

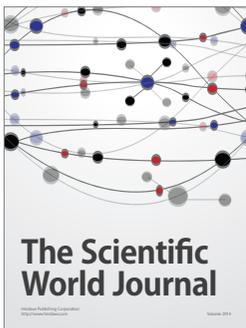
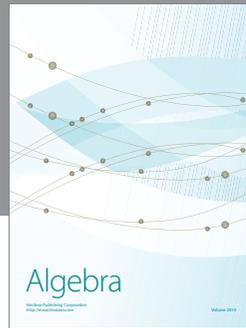
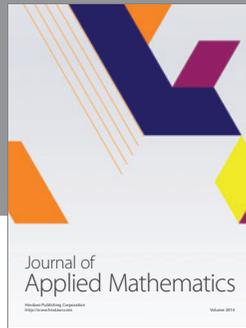
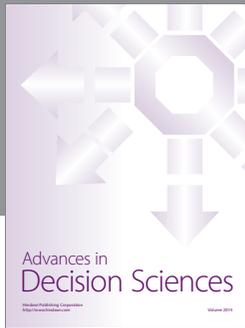
This work was supported by Self-Planned Task (no. SKLRS201609B) of State Key Laboratory of Robotics and System (HIT).

### References

- [1] F. J. Abu-Dakka, F. J. Valero, J. L. Suñer, and V. Mata, "A direct approach to solving trajectory planning problems using genetic algorithms with dynamics considerations in complex environments," *Robotica*, vol. 33, no. 3, pp. 669–683, 2015.
- [2] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [3] F. J. M. Abu-Dakka, *Trajectory Planning for Industrial Robot Using Genetic Algorithms*, Mechanical and Material Engineering Department, Universitat Politècnica de València, Valencia, Spain, 2011.
- [4] T. Chettibi, H. E. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics—A/Solids*, vol. 23, no. 4, pp. 703–715, 2004.
- [5] F. J. Abu-Dakka, F. Rubio, F. Valero, and V. Mata, "Evolutionary indirect approach to solving trajectory planning problem for industrial robots operating in workspaces with obstacles," *European Journal of Mechanics—A/Solids*, vol. 42, pp. 210–218, 2013.
- [6] M. H. Korayem and A. Nikoobin, "Maximum payload for flexible joint manipulators in point-to-point task using optimal control approach," *International Journal of Advanced Manufacturing Technology*, vol. 38, no. 9–10, pp. 1045–1060, 2008.
- [7] E. Bertolazzi, F. Biral, and M. Da Lio, "Real-time motion planning for multibody systems," *Multibody System Dynamics*, vol. 17, no. 2–3, pp. 119–139, 2007.
- [8] S. Behzadipour and A. Khajepour, "Time-optimal trajectory planning in cable-based manipulators," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 559–563, 2006.
- [9] F. Bourbonnais, P. Bigras, and I. A. Bonev, "Minimum-time trajectory planning and control of a pick-and-place five-bar

- parallel robot," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 740–749, 2015.
- [10] A. Gasparetto, A. Lanzutti, R. Vidoni, and V. Zanutto, "Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 164–181, 2012.
- [11] V. Zanutto, A. Gasparetto, A. Lanzutti, P. Boscariol, and R. Vidoni, "Experimental validation of minimum time-jerk algorithms for industrial robots," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 64, no. 2, pp. 197–219, 2011.
- [12] B. H. Cho, B. S. Choi, and J. M. Lee, "Time-optimal trajectory planning for a robot system under torque and impulse constraints," *International Journal of Control, Automation and Systems*, vol. 4, no. 1, pp. 10–16, 2006.
- [13] C. Guarino Lo Bianco and A. Piazzzi, "Minimum-time trajectory planning of mechanical manipulators under dynamic constraints," *International Journal of Control*, vol. 75, no. 13, pp. 967–980, 2002.
- [14] J. Gregory, A. Olivares, and E. Staffetti, "Energy-optimal trajectory planning for the Pendubot and the Acrobot," *Optimal Control Applications & Methods*, vol. 34, no. 3, pp. 275–295, 2013.
- [15] J. Gregory, A. Olivares, and E. Staffetti, "Energy-optimal trajectory planning for robot manipulators with holonomic constraints," *Systems and Control Letters*, vol. 61, no. 2, pp. 279–291, 2012.
- [16] E. Shintaku, "Minimum energy trajectory for an underwater manipulator and its simple planning method by using a Genetic Algorithm," *Advanced Robotics*, vol. 13, no. 2, pp. 115–138, 1999.
- [17] S. F. Saramago and J. Steffen, "Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system," *Mechanism and Machine Theory*, vol. 33, no. 7, pp. 883–894, 1998.
- [18] M. Jin and D. Wu, "Collision-free and energy-saving trajectory planning for large-scale redundant manipulator using improved PSO," *Mathematical Problems in Engineering*, vol. 2013, Article ID 208628, 8 pages, 2013.
- [19] M. Radmanesh and M. Kumar, "Flight formation of UAVs in presence of moving obstacles using fast-dynamic mixed integer linear programming," *Aerospace Science and Technology*, vol. 50, pp. 149–160, 2016.
- [20] C.-T. Chen and T.-T. Liao, "A hybrid strategy for the time- and energy-efficient trajectory planning of parallel platform manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 72–81, 2011.
- [21] A. Khoukhi, L. Baron, and M. Balazinski, "Constrained multi-objective trajectory planning of parallel kinematic machines," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 4–5, pp. 756–769, 2009.
- [22] R. Saravanan, S. Ramabalan, and C. Balamurugan, "Evolutionary multi-criteria trajectory modeling of industrial robots in the presence of obstacles," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 2, pp. 329–342, 2009.
- [23] H.-I. Lin, "A fast and unified method to find a minimum-jerk robot joint trajectory using particle swarm optimization," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 75, no. 3–4, pp. 379–392, 2014.
- [24] R. Béarée and A. Olabi, "Dissociated jerk-limited trajectory applied to time-varying vibration reduction," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 444–453, 2013.
- [25] Y. Chen and B. Li, "A piecewise acceleration optimal and smooth-jerk trajectory planning method for robot manipulator along a predefined path," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, article 50, 2011.
- [26] P. Huang, Y. Xu, and B. Liang, "Global minimum-jerk trajectory planning of space manipulator," *International Journal of Control, Automation and Systems*, vol. 4, no. 4, pp. 405–413, 2006.
- [27] A. Piazzzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 140–149, 2000.
- [28] R. Saravanan and S. Ramabalan, "Evolutionary minimum cost trajectory planning for industrial robots," *Journal of Intelligent and Robotic Systems*, vol. 52, no. 1, pp. 45–77, 2008.
- [29] L. T. Wang and B. Ravani, "Dynamic load carrying capacity of mechanical manipulators, part I: problem formulation," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 110, no. 1, pp. 46–52, 1988.
- [30] L. T. Wang and B. Ravani, "Dynamic load carrying capacity of mechanical manipulators—part ii: computational procedure and applications," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 110, no. 1, pp. 53–61, 1988.
- [31] M. H. Korayem, V. Azimirad, and M. Irani Rahagi, "Maximum allowable load of mobile manipulator in the presence of obstacle using non-linear open and closed loop optimal control," *Arabian Journal for Science and Engineering*, vol. 39, no. 5, pp. 4103–4117, 2014.
- [32] M. H. Korayem, V. Azimirad, H. Vatanjou, and A. H. Korayem, "Maximum load determination of nonholonomic mobile manipulator using hierarchical optimal control," *Robotica*, vol. 30, no. 1, pp. 53–65, 2012.
- [33] M. H. Korayem, V. Azimirad, A. Nikoobin, and Z. Boroujeni, "Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability," *International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5–8, pp. 811–829, 2010.
- [34] M. H. Korayem, M. Irani, A. Charesaz, A. H. Korayem, and A. Hashemi, "Trajectory planning of mobile manipulators using dynamic programming approach," *Robotica*, vol. 31, no. 4, pp. 643–656, 2013.
- [35] M. H. Korayem, K. Najafi, and M. Bamdad, "Synthesis of cable driven robots' dynamic motion with maximum load carrying capacities: iterative linear programming approach," *Scientia Iranica Transaction B: Mechanical Engineering*, vol. 17, no. 3, pp. 229–239, 2010.
- [36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [37] H. Li, W. Niu, S. Fu, and D. Zhang, "Multiobjective optimization of steering mechanism for rotary steering system using modified NSGA-II and fuzzy set theory," *Mathematical Problems in Engineering*, vol. 2015, Article ID 875872, 13 pages, 2015.
- [38] S. Carlucci, G. Cattarin, F. Causone, and L. Pagliano, "Multi-objective optimization of a nearly zero-energy building based on thermal and visual discomfort minimization using a non-dominated sorting genetic algorithm (NSGA-II)," *Energy and Buildings*, vol. 104, pp. 378–394, 2015.
- [39] D. Liu, G. Li, and Y. Shang, "Dynamic modeling and multi-objective optimization of forging manipulator during one forging stroke," *Mechanics Based Design of Structures and Machines*, vol. 40, no. 2, pp. 241–252, 2012.

- [40] W. Guo, R. Li, C. Cao, and Y. Gao, "Kinematics analysis of a novel 5-DOF hybrid manipulator," *International Journal of Automation Technology*, vol. 9, no. 6, pp. 765–774, 2015.
- [41] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics, Transactions of the ASME*, vol. 22, pp. 215–221, 1955.
- [42] D. L. Peiper, *The kinematics of manipulators under computer control in [Ph.D. thesis]*, Stanford University CA Department of Computer Science, 1968.
- [43] J. Y. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," *Journal of Dynamic Systems, Measurement and Control*, vol. 102, no. 2, pp. 69–76, 1980.
- [44] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, 2008.
- [45] S. B. Niku, *Introduction to Robotics: Analysis, Systems, Applications*, Wiley, 2010.
- [46] W. M. Silver, "On the equivalence of lagrangian and newton-euler dynamics for manipulators," *The International Journal of Robotics Research*, vol. 1, no. 2, pp. 60–70, 1982.
- [47] K. Sivakumar, C. Balamurugan, and S. Ramabalan, "Simultaneous optimal selection of design and manufacturing tolerances with alternative manufacturing process selection," *Computer-Aided Design*, vol. 43, no. 2, pp. 207–218, 2011.
- [48] K. Sivakumar, C. Balamurugan, and S. Ramabalan, "Concurrent multi-objective tolerance allocation of mechanical assemblies considering alternative manufacturing process selection," *The International Journal of Advanced Manufacturing Technology*, vol. 53, no. 5–8, pp. 711–732, 2011.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

