Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: http://www.tandfonline.com/loi/geno20

Task-based design optimization of serial robot manipulators

Hatem Al-Dois , A. K. Jha & R. B. Mishra

Engineering Optimization, 2013
Vol. 45, No. 6, 647–658, http://dx.doi.org/10.1080/0305215X.2012.704027

Taylor & Francis
Taylor & Francis Group

Task-based design optimization of serial robot manipulators

Hatem Al-Dois*, A.K. Jha and R.B. Mishra

Department of Mechanical Engineering, Institute of Technology, Banaras Hindu University, Varanasi, U.P. 221005, India

(*Received 2 November 2011; final version received 28 May 2012*)

In this article a direct search non-gradient design optimization method is proposed to determine the optimal task time and optimal parameters of serial robot manipulators. The design variables are treated as continuous variables associated with the motion trajectory profile and robot's link lengths and masses. The method adopted in this study can be applied to any serial redundant or non-redundant manipulator that has rigid links and known kinematic and dynamic models with free motions or motions along specified paths. Trajectory can be planned in Cartesian or joint space under kinodynamic constraints. The cost function to be minimized is weighted terms of task time and joint torques quadratic average. Examples on 3-R planar and spatial serial robot manipulators are given to illustrate the method where the kinematic and dynamic models and the optimization method are developed in MATLAB®.

Keywords: 3-R serial manipulator; robot design; optimal performance; trajectory planning; direct search optimization

1. Introduction

The rapid growth of manufacturing technologies demands optimal design of the machinery and components involved. The design of a robot manipulator for optimal performance is a challenging task owing to the large number of factors included in the design process and the highly non-linear governing equation of motion (Angeles 2007). Researchers have adopted many important design criteria, such as condition number of Jacobian matrix (Salisbury and Craig 1982, Gosselin and Angeles 1991), manipulability (Yoshikawa 1985), acceleration radius (Graettinger and Krogh 1988), inertial and acceleration characteristics (Tourassis and Neuman 1985, Klein and Blaho 1987, Khatib and Bowling 1996) and actuator efforts (Ma 1996, Yunong *et al.* 2004). Methods that can be used to find the optimal solutions for a given performance criterion considering a robot's kinematics and dynamics can be grouped into two categories: gradient-based and non-gradient-based methods. The gradient-based methods have a fast asymptotic convergence and an accurate approximation but they require accurate calculations of derivatives, which are a cumbersome task in many engineering optimization problems (Rao 2010). A large number of different non-gradient methods exists. However, the non-gradient-based methods do not rely explicitly on gradient calculations; thus, they are general methods but sometimes suffer from low convergence and a limit size of the problem (Kolda *et al.* 2003, Pettersson and Olvander 2007) and can be attracted to local minima. Examples of non-gradient direct search methods include

*Corresponding author. Email: haldois@yahoo.com

the Simplex method, Box's Complex method, pattern search, parallel direct search, simulated annealing and genetic algorithm stochastic methods.

There has been a substantial amount of research in the time-optimal control of robotic manipulators. Kahn and Roth (1971) were the first to introduce time-optimal control of robotic manipulators. The first efficient algorithm for finding the optimal trajectory was developed independently by Bobrow *et al.* (1985) and Shin and McKay (1985). This algorithm was later simplified and modified (Pfeiffer and Johanni 1986, Shiller and Dubowsky 1991). More details on time-optimal control can be found in many references, such as Haddad *et al.* (2007) and Siciliano *et al.* (2009).

Robot design problems can be broadly classified into general purpose designs and task-based designs. The approach taken in this study falls into the second category, where the robot and trajectory parameters are optimized towards motions typical to those in particular applications such as arc and spot welding, material handling and machine tending. Task-based design optimization has been of considerable interest of researchers in robotics, especially in areas such as modular and configurable robot systems and special purpose manipulators (Kim and Khosla 1993, Park *et al.* 2003, Kim 2006). The goal of this article is to apply the Complex direct search method to optimize the task time and joint torques of serial robot manipulators for a particular application. The considered parameters are candidates of the motion trajectory and manipulator links lengths and masses. The kinematic and dynamic models of the manipulator are derived symbolically to allow the use of different parameter values in the simulation program. A multiobjective function is implemented which is a weighted form between the total task time and the quadratic average of joint torques. Numerical examples of the proposed method are given on the 3-R planar and spatial serial manipulators, with a free-motion task given in joint space and a path-following task in Cartesian space.

The rest of the article is arranged as follows. The problem is formulated in Section 2. The calculation of task time is explained in Section 3. The Complex optimization method implemented in this study is outlined in Section 4. Sections 5 and 6 discuss the trajectory planning and the optimization algorithm. Numerical examples using 3-R planar and spatial serial manipulators are given in Section 7 and the article is summarized in Section 8.

2. Problem formulation

Considering a serial robot manipulator with n degrees of freedom, the kinematic model can be extracted from the well-known D-H convention (Denavit and Hartenberg 1955). The governing equations of motion can be derived using the Newton–Euler method or Lagrange–Euler method (Angeles 2007), which appear in the form

$$\tau(t) = M(q)\ddot{q} + h(q, \dot{q}) + G(q) \tag{1}$$

where $\tau(t)$ is the joint torque vector, M is the joint space symmetric inertia matrix, h is the vector of centrifugal and Coriolis forces, G is the gravity force vector, and $q, \dot{q}, \ddot{q}$ are the joint position, velocity and acceleration vectors, respectively. A multiobjective cost function is implemented which considers the task time T and the quadratic average of actuator efforts QAT (Haddad *et al.* 2007) while the robot is required to move freely or along a specified path from the initial point $P_{initial}$ to a destination point P_{final}, *i.e.*

$$\text{Min } F_{obj} = \mu T + (1 - \mu)QAT \tag{2}$$

subject to the following initial and final conditions:

$$\left.\begin{aligned} q_{(t=0)} &= q_{initial} \\ \mathrm{q}_{(t=T)} &= q_{final} \\ \dot{q}_{(t=0)} &= 0 \\ \dot{q}_{(t=T)} &= 0 \\ \ddot{q}_{(t=0)} &= 0 \\ \dot{q}_{(t=T)} &= 0 \end{aligned}\right] \tag{3}$$

and the following constraints:

$$\left.\begin{aligned} |q_i(t)| &\leq q_i^{max} \\ |\dot{q}_i(t)| &\leq \dot{q}_i^{max} \\ |\ddot{q}_i(t)| &\leq \ddot{q}_i^{max} \\ |\tau_i(t)| &\leq \tau_i^{max} \end{aligned}\right] \tag{4}$$

where μ is a weighting factor ranging from 0 to 1 corresponding to the relative importance of the cost function term, T is the task time, QAT is the quadratic average joint torques, calculated by

$$QAT = \int_0^T \sum_{i=1}^{n} \left(\frac{\tau_i(t)}{\tau_i^{max}} \right)^2 dt \tag{5}$$

and i denotes the ith robot joint. τ_i^{max} is introduced in the QAT term (5) to normalize the torques at each robot joint. The squared term casts the problem of optimizing actuator torques as constrained time-varying quadratic programming (QP). The QP function describes the physical costs better than norms and provides, with a positive definite Hessian, a global solution for the optimization problem for which many efficient solvers exist. QP has been widely used in robotic trajectory and control optimization problems (*e.g.* Rao and Rawlings 2000, Stumper and Kennel 2011).

Minimizing time is desirable to increase productivity, but the system acceleration and deceleration are always along extremes, which means maximum torques. Minimizing joint torques produces a smoothing effect, which is preferable for joint motors, and helps to avoid excitation (Pfeiffer and Johanni 1986). Weighting factors are used to define the importance associated with different objectives and constraints as well as to normalize different physical units. The value of μ depends on the geometry of the path. Selecting a value of $\mu = 1$ gives all importance to minimizing the travel time; selecting a value of $\mu = 0$ gives all importance to minimizing joint torques. With a proper weighting of μ a smoothing effect of path velocity and joint torques can be accomplished.

Conditions (3) equate the joint positions at the beginning and end of the trajectory to the required initial and final positions with zero initial and final joint velocities and accelerations. Inequalities (4) set the constraints on joint positions, velocities, accelerations and torques, respectively. Motion constraints are introduced since not all motions are tolerable and power resources are limited. Higher values of permissible motion limits allow the robot to finish the task in a shorter time but with higher values of actuator efforts. Heavy link masses force the robot to move slower if actuator powers are limited, otherwise they exert more torques to move in higher motions. Torques due to inertial forces also become higher for higher values of inertial coefficients (due to longer or heavier links) or if the links accelerate or decelerate more rapidly. Only the T and QAT terms were considered in the objective function because of the trade-off relation between minimizing them; however, more than two objective terms can be handled. The manipulability measure, for example, can be added as the third term in the cost function with a proper weighting to optimize the manipulator ability to accelerate and decelerate in different directions in the work space. This is practically useful while optimization is required to find the optimal path with high manipulability measure (Saravanan and Ramabalan 2008). Calculations of τ are straightforward at each time step using (1). The following section describes the method adopted in the calculation of optimal task time T.

3. Calculation of task time

To facilitate the stochastic optimization method, normalization of the timescale is introduced, where the problem is solved for a fixed final time (Hollerbach 1983), *i.e.* for $q(x) : 0 \leq x \leq 1$. Equation (1) becomes

$$\tau_i(x) = \frac{1}{T^2}\left[\sum_{j=1}^{n} M_{ij}(q(x))q_j''(x) + h_i(q(x), q'(x))\right] + G_i(q(x)) \tag{6}$$

Note that the gravity component is position dependent and not affected by the time scaling. This process allows the geometric, kinematic and dynamic constraints to be translated into bounds on admissible values of the optimal transfer time T_q of the parameter (x), for instance,

$$T_v = \max_{i=1,\ldots,n}\left[\max_{x\in[0,1]} \frac{|q'(x)|}{\dot{q}_i^{\max}}\right] \tag{7}$$

$$T_a = \max_{i=1,\ldots,n}\left[\max_{x\in[0,1]} \frac{|q''(x)|}{\ddot{q}_i^{\max}}\right]^{1/2} \tag{8}$$

where T_v and T_a are the minimum times due to constraints on joint velocities and accelerations, respectively. T_L and T_R, which are the time bracketing interval due to joint torque limitations, are calculated based on the values of τ_{a_i}, τ_{b_i} and $H_i(x)$ given in Table 1, where

$$\tau_{a_i}(x) = -\tau_i^{\max} - G_i(x) \tag{9}$$

$$\tau_{b_i}(x) = \tau_i^{\max} - G_i(x) \tag{10}$$

$$H_i(x) = \sum_{j=1}^{n} M_{ij}(q(x))q_j''(x) + h_i(q(x), q'(x)) \tag{11}$$

$$q' = \frac{\partial q(x)}{\partial x} \quad \text{and} \quad q'' = \frac{\partial^2 q(x)}{\partial^2 x} \tag{12}$$

Table 1. Calculations of T_L and T_R.

Condition		$T_{L_i}(x)$	$T_{R_i}(x)$
$\tau_{a_i}(x) < 0$	$H_i(x) > 0$	$\left(\frac{H_i(x)}{\tau_{b_i}(x)}\right)^{1/2}$	$+\infty$
$\tau_{b_i}(x) > 0$	$H_i(x) = 0$	0	$+\infty$
	$H_i(x) < 0$	$\left(\frac{H_i(x)}{\tau_{a_i}(x)}\right)^{1/2}$	$+\infty$
$\tau_{b_i}(x) > 0$	$H_i(x) > 0$	$\left(\frac{H_i(x)}{\tau_{b_i}(x)}\right)^{1/2}$	$\left(\frac{H_i(x)}{\tau_{a_i}(x)}\right)^{1/2}$
$\tau_{a_i}(x) > 0$	$H_i(x) \leq 0$	Unfeasible motion	Unfeasible motion
$\tau_{b_i}(x) < 0$	$H_i(x) \geq 0$	Unfeasible motion	Unfeasible motion
$\tau_{a_i}(x) < 0$	$H_i(x) < 0$	$\left(\frac{H_i(x)}{\tau_{a_i}(x)}\right)^{1/2}$	$\left(\frac{H_i(x)}{\tau_{b_i}(x)}\right)^{1/2}$

The minimum task time T_q of the selected trajectory $q(x)$ should satisfy

$$T_{\mathrm{q}} \geq T_v;\ T_q \geq T_a; \quad \text{and} \quad T_q \in [T_L, T_R] \tag{13}$$

Applying this method inside an optimization algorithm allows for testing of different trajectory profiles with different design parameters, such as link lengths and masses, and for efficient scanning of the solution space for the optimization objective. In the following section the Complex direct-search method adopted in this study is briefly described, as given in Krus and Andersson (2003). More details can be found in Krus and Andersson (2003) and Rao (2010).

4. The Complex optimization method

The Complex method was developed by Box (1965), and since then has been modified to be a simple and an easy-to-implement method. The method can be used to maximize (or minimize) the function $F(x_1, x_2, \ldots, x_N)$ subject to the constraints $g_i < x_i < h_i$, where $i = 1, 2, \ldots, M$. An initial complex of m points is generated. The variables at each point are generated using random numbers:

$$x_{ij} = g_i + r_{ij}(h_i - g_i) \tag{14}$$

where j is an index indicating a point in the complex, i is an index indicating a variable, and r_{ij} is a random number in the interval [0, 1]. If the implicit constraints are not fulfilled, a new point is generated until the implicit constraints are fulfilled. The number of points m in the complex must be such that $m \geq N + 1$, where N is the number of independent variables. The objective function is evaluated at each point. The point with the lowest value is replaced by a point reflected in the centroid of the remaining points by a factor α (Figure 1):

$$x_{ij}\ (\text{new}) = x_{ic} + \alpha(x_{ic} - x_{ij}\ (\text{old})) \tag{15}$$

The centroid is calculated as

$$x_{ic} = \frac{1}{m-1}\sum_{i=1}^{m}(x_{ij} - x_{ij}\ (\text{old})) \tag{16}$$

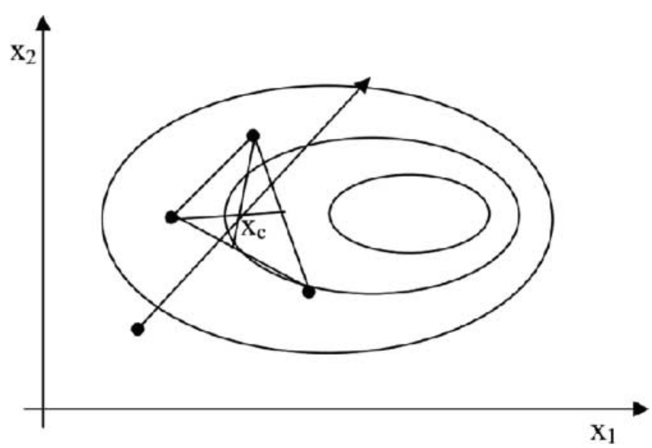

Figure 1. The reflection move of the worst point in the Complex method.

Box (1965) recommended $\alpha = 1.3$. If a point repeats as the lowest value on consecutive trials, it is moved half the distance towards the centroid of the remaining points. In this case,

$$x_{ij}\text{ (new)} = x_{ic} + (x_{ic} - x_{ij}\text{ (new)})/2 \tag{17}$$

5. Trajectory planning

The trajectory describes the positions, velocities and accelerations of either the end effector of a robot manipulator in Cartesian space or the robot joints in joint space. Many planners have been suggested, such as cubic trajectory, quintic trajectory, trapezoidal trajectory, smooth trapezoidal trajectory and cubic spline. The equations governing the smoothed trapezoidal velocity profile (STVP) are given next, which are used frequently in industrial manipulators specifically for minimum-time motions with no obstacles in the workspace. For motion along specified paths with obstacle avoidance a piecewise polynomial such as the cubic spline trajectory can be used. In Section 7, examples of STVP and cubic spline are given with 3-R planar and spatial manipulators.

$$q(x) = \begin{cases} q_{initial} + \dfrac{(q_{final} - q_{initial})}{(1 + x_b - x_a)}\left(\dfrac{2x^3}{x_a^2} - \dfrac{x^4}{x_a^3}\right) & 0 \le x < x_a \\ q_{initial} + \dfrac{(q_{final} - q_{initial})(2x - x_a)}{(1 + x_b - x_a)} & x_a \le x \le x_b \\ q_{initial} + \dfrac{(q_{final} - q_{initial})}{(1 + x_b - x_a)}\left(\dfrac{(x - x_b)^4}{(1 - x_b)^3} - \dfrac{2(x - x_b)^3}{(1 - x_b)^2} + (2x - x_b)\right) & x_b < x \le 1 \end{cases} \tag{18}$$

$$\dot{q}(x) = \frac{d(q(x))}{dx^2} \tag{19}$$

$$\ddot{q}(x) = \frac{d^2(q(x))}{dx^2} \tag{20}$$

The variables in the STVP trajectory are x_a and x_b, which define the shape and characteristics of the trajectory profile, while x is the current time in the scaled period as described in Section 3 and $q_{initial}$ and q_{final} are the initial and final joint positions, respectively.

6. The optimization methodology

To find the trajectory candidates and robot parameters that minimize the objective function for a given task using the approach discussed above, the methodology is outlined in the following algorithm, which is used in the Complex optimization method:

1. Read initial position, final position, constraints, μ
2. Initialize F_{best}
3. **While** (generation $\le$ No. of generations)
4. {Generate a trajectory based on random trajectory candidates (x_a and x_b) using (18–20)
5. Calculate τ_{a_i}, τ_{b_i} and $H_i(x)$, using (9–11)
6. Calculate $T_{L_i}(x)$, $T_{R_i}(x)$ using Table (1)
7. Get T_v, T_a from (7, 8)
8. Calculate T_q from (13)
9. Calculate τ_i using (1)
10. Calculate F_{obj} using (2)

11. **If** ($F_{obj} < F_{best}$) then
12. $F_{obj} \leftarrow F_{best}$
13. **Endif}**

If the task is given in Cartesian space then the trajectory is planned for the position and orientation of the robot's end-effector (EE). The corresponding configurations, velocities and accelerations for robot joints are calculated using the inverse kinematics model and *Jacobian* matrix before the **While** statement. The trajectory of EE orientation can be solved using, for example, a rotation about a vector in space an angle of $\boldsymbol{\alpha}$, then the selected trajectory can be applied to $\boldsymbol{\alpha}$. (see Saha 2008 for more details.)

When the robot is asked to move along a path specified by number of points in Cartesian space, additional free points are inserted between the via points. The problem then is turned to be finding the optimal positions of the additional points to construct a joint space trajectory with minimum cost function satisfying all limit conditions. The cubic spline can be used to construct the trajectory which guarantees continuity up to the second order. The process can be started by distributing all points uniformly in the scaled time. Then, the optimization method is applied to find the optimal positions of additional points. In this case, in the algorithm given above, the trajectory is generated based on via points (fixed) and additional points (variables) using cubic spline rather than on x_a and x_b using STVP, step 4 in the algorithm.

7. Numerical examples

7.1. *3-R planar serial manipulator with point-to-point motion in joint space*

This example considers a 3-R planar serial robot manipulator (Figure 2). The robot links are assumed to be rigid thin rods with masses concentrated at centres of gravity. The motors' weights and inertias are neglected. The robot parameters, links inertial terms in link coordinates, and motion constraints are given in Table 2. The robot is required to move between two points in joint space, *i.e.* from [15°, −70°, 20°] to [135°, 95°, −75°], with zero initial and final velocities.

The complex method is applied to find the values that characterize the optimal trapezoidal trajectory in forms of the two points x_a and x_b in the scaled period and their corresponding values e_a and e_b in the optimal time and the robot's link lengths and masses within the following ranges

$$\left.\begin{aligned} x_a \text{ and } x_b \in [0,1];\ x_a \leq x_b \\ L_i \in [0.2\,\text{m},\ 1\,\text{m}] \\ m_i \in [0.5\,\text{kg},\ 1.5\,\text{kg}] \end{aligned}\right] \tag{21}$$

for $i = 1, 2, 3$, where i is the ith robot link.

The number of points used in the Complex method is 10. The maximum number of generations is 300 with three levels of μ, $\mu = 1$, $\mu = 0.5$ and $\mu = 0$. Results obtained using a 1.8 GHz laptop are tabulated in Table 3.

From the data available in Table 3 it is observed that the minimum task time for the given motion is 0.7601 s with $\mu = 1$, *i.e.* with full consideration to the cycle time; however, when the waiting factor μ changes to 0 (full consideration to the quadratic average torques) the minimum task time increases to 0.8024 s, but with almost minimum corresponding link lengths and masses that obviously lead to minimum torques. Figure 3 shows the corresponding STVP profile (position, velocity and acceleration), and joint torques with respect to time at $\mu = 1$ and $\mu = 0$. It can be seen from the figures how the bounds of motion (Table 2) are not exceeded and how torque values are higher with shorter travelling time when $\mu = 1$.

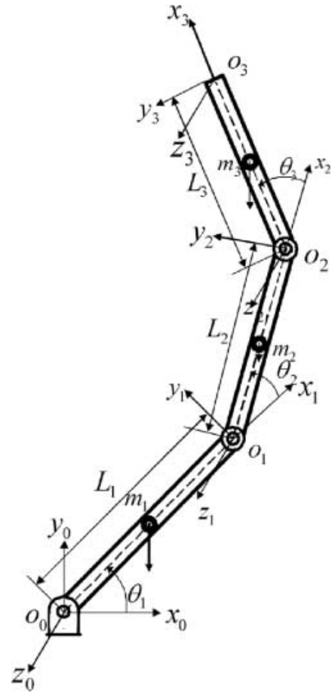

Figure 2. 3-R planar serial robot manipulator.

Table 2. Parameters of 3-R planar manipulator.

Link i	Length (m)	Mass (kg)	I_{xx} (kg.m^2)	I_{yy} (kg.m^2)	I_{zz} (kg.m^2)	$\dot{q}_{i\,\max}$ (rad.s^{-1})	$\ddot{q}_{i\max}$ (rad.s^{-2})	$\tau_{i\max}$ (N.m)
1	L_1	m_1	0	$\frac{1}{12}m_1L_1^2$	$\frac{1}{12}m_1L_1^2$	8	30	20
2	L_2	m_2	0	$\frac{1}{12}m_2L_2^2$	$\frac{1}{12}m_2L_2^2$	8	30	15
3	L_3	m_3	0	$\frac{1}{12}m_3L_3^2$	$\frac{1}{12}m_3L_3^2$	8	30	10

Table 3. Optimal results obtained for point-to-point motion with 3-R planar serial manipulator.

	Weighting factor μ		
	1	0.5	0
F_{obj}	0.7601	1.3565	1.8321
T_{obj}	0.7601	0.7678	0.8024
QAT_{obj}	12.1025	1.9451	1.8321
e_a (s)	0.3724	0.4239	0.4233
e_b (s)	0.3877	0.5742	0.6294
L_1 (m)	0.2459	0.2008	0.2002
L_2 (m)	0.2509	0.2030	0.2019
L_3 (m)	0.3447	0.2018	0.2008
m_1 (kg)	1.0511	0.5434	0.5037
m_2 (kg)	1.0112	0.5089	0.5013
m_3 (kg)	0.5587	0.5017	0.5017
Execution time per generation (s)	0.0366	0.0397	0.0396

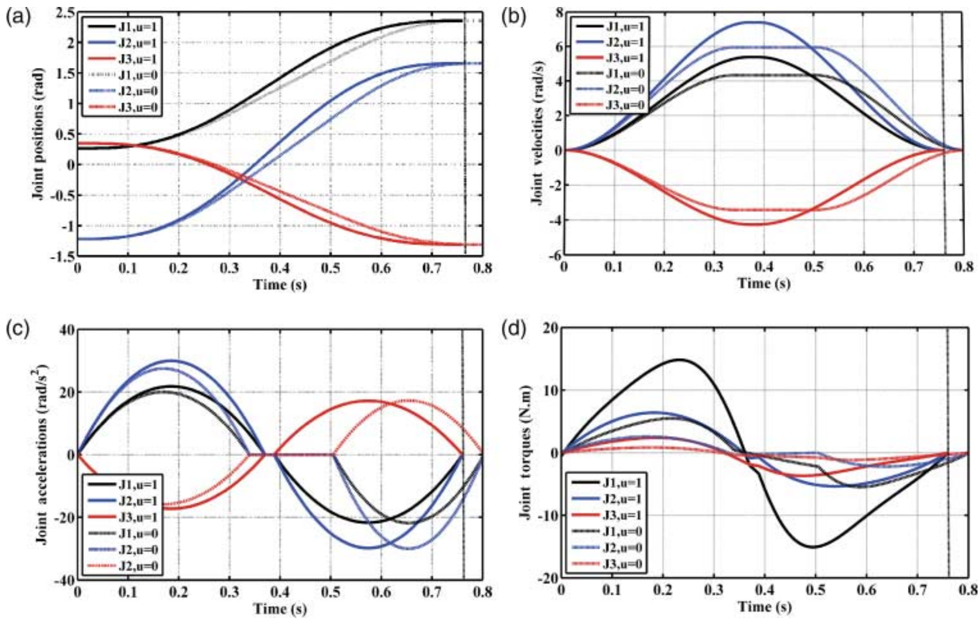

Figure 3. STVP and torque profile at $\mu = 1$ and $\mu = 0$ for 3-R planar serial manipulator: (a) joint positions; (b) joint velocities; (c) joint accelerations; (d) joint torques.

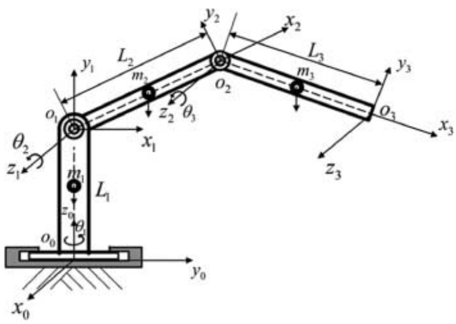

Figure 4. 3-R spatial serial manipulator.

7.2. *3-R spatial serial manipulator with motion along a specified path*

Consider now the 3-R spatial manipulator shown in Figure 4. The robot links are assumed to be rigid thin rods with masses concentrated at centres of gravity. The motors' weights and inertias are neglected. Robot parameters, link inertias in link coordinates and motion constraints are given in Table 4. The robot is required to move from $P_{initial} = [1\,\text{m}, 0, 0]$ to $P_{final} = [0, 1\,\text{m}, 0]$ through via point $P_1 = [0.6\,\text{m}, 0.6\,\text{m}, 0.1\,\text{m}]$. All points' coordinates are given in the operational space. Two additional free points are added: P_2 between $P_{initial}$ and P_1, and P_3 between P_1 and P_{final} (Figure 5).

The objective is to find the optimal positions of P_2 and P_3 and the optimal robot link lengths and masses (for second and third link) so that the constructed cubic spline trajectory possesses minimum travel time when $\mu = 1$ or minimum quadratic average of joint torques when $\mu = 0$ without violating the motion and torque bounds. Note that the length and mass of the first link

Table 4. Parameters of 3-R spatial manipulator.

Link i	Length (m)	Mass (kg)	I_{xx} (kg.m^2)	I_{yy} (kg.m^2)	I_{zz} (kg.m^2)	$\dot{q}_{i\max}$ (rad.s^{-1})	$\ddot{q}_{i\max}$ (rad.s^{-2})	$\tau_{i\max}$ (N.m)
1	L_1	m_1	$\frac{1}{12}m_1L_1^2$	$\frac{1}{12}m_1L_1^2$	0	5	30	100
2	L_2	m_2	0	$\frac{1}{12}m_2L_2^2$	$\frac{1}{12}m_2L_2^2$	5	30	75
3	L_3	m_3	0	$\frac{1}{12}m_3L_3^2$	$\frac{1}{12}m_3L_3^2$	5	30	50

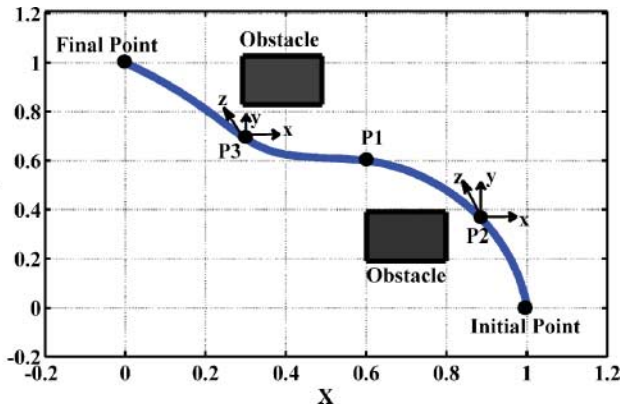

Figure 5. Cubic spline optimal path for 3-R spatial serial manipulator.

Table 5. Optimal results obtained for motion along specified path for 3-R spatial serial manipulator.

	Weighting factor μ	
	1	0
F_{obj}	0.8776	2.1282
T_{obj}	0.8776	0.9490
QAT_{obj}	6.0892	2.1282
P_2 Optimal Cartesian coordinates (m, m, m)	[0.885, 0.365, 0.046]	[0.841, 0.4, 0.066]
P_3 Optimal Cartesian coordinates (m, m, m)	[0.3, 0.69, 0.07]	[0.2139, 0.8, 0]
L_2 (m)	1.134	0.8
L_3 (m)	1.05	1.0482
Execution time per generation (s)	2.304	2.559

of spatial type serial manipulator have no effect on the robot dynamics (Corke and Armstrong-Hélouvry 1995). The search spaces for $P_2$and P_3 are set in ranges so that they do not collide with obstacles. Another alternative to avoid obstacles is to add a Boolean cost term to the multiobjective function which generates *FALSE* if a point in the path lies in the obstacle areas. Assuming that link masses are connected to link lengths with the following proportional ratios: $m_2 = 2L_2$; and $m_3 = L_3$, and the bounds of link lengths are [0.8 m, 1.2 m], the results of this problem are tabulated in Table 5 using the Complex method with 12 points and a maximum of 400 generations on a 1.8 GHz laptop.

From Table 5, it is observed that when full consideration was given to task time with $\mu = 1$, the optimal task time was 0.8776 s, but with link lengths of 1.134 m and 1.05 m and masses of

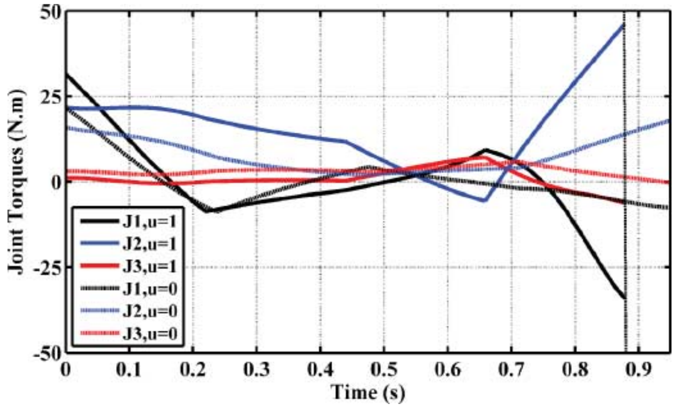

Figure 6. Joint torques at $\mu = 1$ and $\mu = 0$ with cubic spline path for 3-R spatial serial manipulator.

2.268 kg and 1.05 kg for the second and third links, respectively. This led to actuator torques with a quadratic average of 6.0892. However, when full consideration was given to the quadratic torques average with $\mu = 0$, link lengths were reduced to 0.8 m and 1.0482 m with masses of 1.6 kg and 1.0482 kg. The new optimized path with the new values of link lengths and masses led to a lower quadratic average of joint torques *QAT* of 2.1282 at the cost of the task time, which has been increased to 0.949 s. Figure 5 shows the $X - Y$ space of the optimal path at $\mu = 1$. Figure 6 shows torques exerted in the robot joints for both values of μ. It is clear from the figure that the torques are higher with shorter times for $\mu = 1$ and within their allowable ranges than those of $\mu = 0$.

8. Conclusion

In this article the Complex direct search method is adopted to optimize the task time and link lengths and masses of serial robotic manipulators under kinematic and dynamic constraints. A waiting factor is utilized in the objective function to maintain the balance between minimizing task time and minimizing torques required at the robot joints during the motion without violating the imposed constraints. The motion of the robot can be planned freely or along specified paths in joint space with respect to robot joints, or in Cartesian space with respect to robot end effector. Illustrative examples on 3-R planar and spatial serial manipulators with smooth trapezoidal velocity profile and cubic spline trajectories are given to illustrate the effectiveness of the proposed method, which has been shown to have good convergence and easy implementation for such a complicated design problem.

References

Angeles, J., 2007. *Fundamentals of robotic mechanical systems, theory, methods, and algorithms*. 3rd ed. New York: Springer.

Bobrow, J.E., Dubowsky, S., and Gibson, J.S., 1985. Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research*, 4 (3), 3–17.

Box, M.J., 1965. A new method of constrained optimization and a comparison with other methods. *Computer Journal*, 8 (1), 42–52.

Corke, P.I. and Armstrong-Hélouvry, B., 1995. A meta-study of puma 560 dynamics: a critical appraisal of literature data. *Robotica*, 13 (3), 253–258.

Denavit, J. and Hartenberg, R.S., 1955. A kinematic notation for lower-pair mechanisms based on matrices. *Transactions of the ASME Journal of Applied Mechanics*, 22, 215–221.

Gosselin, C. and Angeles, J., 1991. A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design*, 113 (3), 220–226.

Graettinger, T.J. and Krogh, B.H., 1988. The acceleration radius: a global performance measure for robotic manipulators. *IEEE Journal of Robotics and Automation*, 4 (1), 60–69.

Haddad, M., *et al.*, 2007. Trajectory generation. *In*: W. Khalil and E. Dombre, eds. *Robot manipulators, modeling, performance, analysis, and control*. 1st ed. London: ISTE, 189–234.

Hollerbach, J.M., 1983. Dynamic scaling of manipulator trajectories. *In: American Control Conference (ACC 1983)*, 22–24 June, San Francisco, CA, USA. New York: IEEE Press. 752–756.

Kahn, E. and Roth, B., 1971. The near-minimum-time control of open-loop articulated kinematic chains. *ASME Journal of Dynamic Systems, Measurements, and Control*, 39 (3), 164–172.

Khatib, O. and Bowling, A., 1996. Optimization of the inertial and acceleration characteristics of manipulators. In: *1996 IEEE International conference on robotics and automation*, 22–28 April, Minneapolis, MN, USA. New York: IEEE Press. 2883–2889.

Kim, J.O. and Khosla, P.K., 1993. A formulation for task based design of robot manipulators. *In: 1993 IEEE/RSJ international conference on intelligent robots and systems (IROS '93)*, 26–30 July, Yokahoma, Japan. New York: IEEE Press. 2310–2317.

Kim, J.Y., 2006. Task based kinematic design of a two dof manipulator with a parallelogram five-bar link mechanism. *Mechatronics*, 16 (6), 323–329.

Klein, C.A. and Blaho, B.E., 1987. Dexterity measures for the design and control of kinematically redundant manipulators. *International Journal of Robotics Research*, 6 (2), 72–83.

Kolda, T.G., Lewis, R.M., and Torczon, V., 2003. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Review*, 45 (3), 385–482.

Krus, P. and Andersson, J., 2003. Optimizing optimization for design optimization. In: *ASME 2003 international design engineering technical conferences and computers and information in engineering conference (IDETC/CIE2003)*, 2–6 September, Chicago, IL, USA. New York: ASME, 2: 29th Design Automation Conference, Parts A and B, 951–960.

Ma, S., 1996. A balancing technique to stabilize local torque optimization solution of redundant manipulators. *Journal of Robotic Systems*, 13 (3), 177–185.

Park, J.-Y., Chang, P.-H., and Yang, J.-Y., 2003. Task-oriented design of robot kinematics using the grid method. *Advanced Robotics*, 17 (9), 879–907.

Pettersson, M. and Olvander, J., 2007. Adaptive complex method for efficient design optimization. *In: ASME 2007 international design engineering technical conference and computer and information in engineering conference (IDETC/CIE2007)*, 4–7 September, Las Vegas, NV, USA. New York: ASME, 6, 265–272.

Pfeiffer, F. and Johanni, R., 1986. A concept for manipulator trajectory planning. *In: 1986 IEEE international conference on robotics and automation*, 7–10 April, San Francisco, CA, USA. New York: IEEE Press. 1399–1405.

Rao, C.V. and Rawlings, J.B., 2000. Linear programming and model predictive control. *Journal of Process Control*, 10 (2–3), 283–289.

Rao, S.S., 2010. *Engineering optimization theory and practice*. 3rd ed. New Delhi: New Age International Publishers.

Saha, S.K., 2008. *Introduction to robotics*. 1st ed. New Delhi: Tata McGraw-Hill.

Salisbury, J.K. and Craig, J.J., 1982. Articulated hands: force control and kinematic issues. *International Journal of Robotics Research*, 1 (1), 4–17.

Saravanan, R. and Ramabalan, S., 2008. Evolutionary minimum cost trajectory planning for industrial robots. *Journal of Intelligent Robotics Systems*, 52 (1), 45–77.

Shiller, Z. and Dubowsky, S., 1991. On computing the global time-optimal motions of robotic manipulators in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 7 (6), 785–797.

Shin, K. and Mckay, N., 1985. Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, 30 (6), 531–541.

Siciliano, B., *et al.*, 2009. *Robotics modeling, planning and control.* London: Springer.

Stumper, J.F. and Kennel, R., 2011. Computationally efficient trajectory optimization for linear control systems with input and state constraints. *In: American control conference (ACC)*, 29 June–1 July, San Francisco, CA, USA. New York: IEEE Press, 1904–1909.

Tourassis, V.D. and Neuman, C.P., 1985. The inertial characteristics of dynamic robot models. *Mechanism and Machine Theory*, 20 (1), 41–52.

Yoshikawa, T., 1985. Manipulability of robotic mechanisms. *International Journal of Robotics Research*, 4 (2), 3–9.

Yunong, Z., Ge, S.S., and Tong Heng, L., 2004. A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34 (5), 2126–2132.