We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Link Mass Optimization Using Genetic Algorithms for Industrial Robot Manipulators

Serdar Kucuk and Zafer Bingul
Kocaeli University
Turkey

1. Introduction

Obtaining optimum energy performance is the primary design concern of any mechanical system, such as ground vehicles, gear trains, high speed electromechanical devices and especially industrial robot manipulators. The optimum energy performance of an industrial robot manipulator based on the minimum energy consumption in its joints is required for developing of optimum control algorithms (Delingette et al., 1992; Garg & Ruengcharungpong, 1992; Hirakawa & Kawamura, 1996; Lui & Wang, 2004). The minimization of individual joint torques produces the optimum energy performance of the robot manipulators. Optimum energy performance can be obtained to optimize link masses of the industrial robot manipulator. Having optimum mass and minimum joint torques are the ways of improving the energy efficiency in robot manipulators. The inverse of inertia matrix can be used locally minimizing the joint torques (Nedungadi & Kazerounian, 1989). This approach is similar to the global kinetic energy minimization.

Several optimization techniques such as genetic algorithms (Painton & Campbell, 1995; Chen & Zalzas, 1997; Choi et al., 1999; Pires & Machado, 1999; Garg & Kumar, 2002; Kucuk & Bingul, 2006; Qudeiri et al., 2007), neural network (Sexton & Gupta, 2000; Tang & Wang, 2002) and minimax algorithms (Pin & Culioli, 1992; Stocco et al., 1998) have been studied in robotics literature. Genetic algorithms (GAs) are superior to other optimization techniques such that genetic algorithms search over the entire population instead of a single point, use objective function instead of derivatives, deals with parameter coding instead of parameters themselves. GA has recently found increasing use in several engineering applications such as machine learning, pattern recognition and robot motion planning. It is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetic. This provides a robust search procedure for solving difficult problems. In this work, GA is applied to optimize the link masses of a three link robot manipulator to obtain minimum energy. Rest of the Chapter is composed of the following sections. In Section II, genetic algorithms are explained in a detailed manner. Dynamic equations and the trajectory generation of robot manipulators are presented in Section III and Section IV, respectively. Problem definition and formulation is described in Section V. In the following Section, the rigid body dynamics of a cylindrical robot manipulator is given as example. Finally, the contribution of this study is presented in Section VII.

2. Genetic algorithms

GAs were introduced by John Holland at University of Michigan in the 1970s. The improvements in computational technology have made them attractive for optimization problems. A genetic algorithm is a non-traditional search method used in computing to find exact or approximate solutions to optimization and search problems based on the evolutionary ideas of natural selection and genetic. The basic concept of GA is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. The obtained optima are an end product containing the best elements of previous generations where the attributes of a stronger individual tend to be carried forward into following generation. The rule is survival of the fittest will. The three basic features of GAs are as follows:

- i. Description of the objective function
- ii. Description and implementation of the genetic representation
- iii. Description and implementation of the genetic operators such as reproduction, crossover and mutation.

If these basic features are chosen properly for optimization applications; the genetic algorithm will work quite well. GA optimization possesses some unique features that separate from the other optimization techniques given as follows:

- i. It requires no knowledge or gradient information about search space.
- ii. It is capable of scanning a vast solution set in a short time.
- iii. It searches over the entire population instead of a single point.
- iv. It allows a number of solutions to be examined in a single design cycle.
- v. It deals with parameter coding instead of parameters themselves.
- vi. Discontinuities in search space have little effect on overall optimization performance.
- vii. It is resistant to becoming trapped in local optima.

These features provide GA to be a robust and useful optimization technique over the other search techniques (Garg & Kumar, 2002). However, there are some disadvantages to use genetic algorithms.

- i. Finding the exact global optimum in search space is not certain.
- ii. Large numbers of fitness function evaluations are required.
- iii. Configuration is not straightforward and problem dependent.

The representation or coding of the variables being optimized has a large impact on search performance, as the optimization is performed on this representation of the variables. The two most common representations, binary and real number codings, differ mainly in how the recombination and mutation operators are performed. The most suitable choice of representation is based upon the type of application. In GAs, a set of solutions represented by chromosomes is created randomly. Chromosomes used here are in binary codings. Each zero and one in chromosome corresponds to a gene. A typical chromosome can be given as

10000010

An initial population of random chromosomes is generated at the beginning. The size of the initial population may vary according to the problem difficulties under consideration. A

different solution to the problem is obtained by decoding each chromosome. A small initial with composed of eight chromosomes can be denoted as the form

10110011
11100110
10110010
10001111
11110111
11101111
10110110
10111111

Note that, in practice both the size of the population and the strings are larger then those of mentioned above. Basically, the new population is generated by using the following fundamental genetic evolution processes: reproduction, crossover and mutation.

At the reproduction process, chromosomes are chosen based on natural selections. The chromosomes in the new population are selected according to their fitness values defined with respect to some specific criteria such as roulette wheel selection, rank selection or steady state selection. The fittest chromosomes have a higher probability of reproducing one or more offspring in the next generation in proportion to the value of their fitness.

At the crossover stage, two members of population exchange their genes. Crossover can be implemented in many ways such as having a single crossover point or many crossover points which are chosen randomly. A simple crossover can be implemented as follows. In the first step, the new reproduced members in the mating pool are mated randomly. In the second step, two new members are generated by swapping all characteristics from a randomly selected crossover point. A good value for crossover can be taken as 0.7. A simple crossover structure is shown below. Two chromosomes are selected according to their fitness values. The crossover point in chromosomes is selected randomly. Two chromosomes are given below as an example

10110011
11100110

After crossover process is applied, all of the bits after the crossover point are swapped. Hence, the new chromosomes take the form

101*00110
111*10011

The symbol “*” corresponds the crossover point. At the mutation process, value of a particular gene in a selected chromosome is changed from 1 to 0 or vice versa., The probability of mutation is generally kept very small so these changes are done rarely. In general, the scheme of a genetic algorithm can be summarized as follows.

- i. Create an initial population.
- ii. Check each chromosome to observe how well is at solving the problem and evaluate the fitness of the each chromosome based on the objective function.
- iii. Choose two chromosomes from the current population using a selection method like roulette wheel selection. The chance of being selected is in proportion to the value of their fitness.
- iv. If a probability of crossover is attained, crossover the bits from each chosen chromosome at a randomly chosen point according to the crossover rate.
- v. If a probability of mutation is attained, implement a mutation operation according to the mutation rate.
- vi. Continue until a maximum number of generations have been produced.

3. Dynamic Equations

A variety of approaches have been developed to derive the manipulator dynamics equations (Hollerbach, 1980; Luh et al., 1980; Paul, 1981; Kane and Levinson, 1983; Lee et al., 1983). The most popular among them are Lagrange-Euler (Paul, 1981) and Newton-Euler methods (Luh et al., 1980). Energy based method (LE) is used to derive the manipulator dynamics in this chapter. To obtain the dynamic equations by using the Lagrange-Euler method, one should define the homogeneous transformation matrix for each joint. Using D-H (Denavit & Hartenberg, 1955) parameters, the homogeneous transformation matrix for a single joint is expressed as,

$${}^{i-1}_iT = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where a_{i-1} , α_{i-1} , d_i , θ_i , c_i and s_i are the link length, link twist, link offset, joint angle, $\cos\theta_i$ and $\sin\theta_i$, respectively. In this way, the successive transformations from base toward the end-effector are obtained by multiplying all of the matrices defined for each joint.

The difference between kinetic and potential energy produces Lagrangian function given by

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \quad (2)$$

where q and $\dot{q}$ represent joint position and velocities, respectively. Note that, q_i is the joint angle θ_i for revolute joints, or the distance d_i for prismatic joints. While the potential energy (P) is dependent on position only, the kinetic energy (K) is based on both position and velocity of the manipulator. The total kinetic energy of the manipulator is defined as

$$K(q, \dot{q}) = \frac{1}{2} \sum_{i=1}^n (\dot{v}_i)^T m_i \dot{v}_i + (\omega_i)^T I_i \omega_i \quad (3)$$

where m_i is the mass of link i and I_i denotes the 3x3 inertia tensor for the center of the link i with respect to the base coordinate frame. I_i can be expressed as

$$I_i = {}^0R_i I_m {}^0R_i^T \quad (4)$$

where 0R_i represents the rotation matrix and I_m stands for the inertia tensor of a rigid object about its center of mass. The terms v_i and ω_i refer to the linear and angular velocities of the center of mass of link i with respect to base coordinate frame, respectively.

$$v_i = A_i(q)\dot{q} \text{ and } \omega_i = B_i(q)\dot{q} \quad (5)$$

where $A_i(q)$ and $B_i(q)$ are obtained from the Jacobian matrix, $J_i(q)$. If v_i and ω_i in equations 5 are substituted in equation 3, the total kinetic energy is obtained as follows.

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T \sum_{i=1}^n [A_i(q)^T m_i A_i(q) + B_i(q)^T I_i B_i(q)] \dot{q} \quad (6)$$

The equation 6 can be written in terms of manipulator mass matrix and joint velocities as

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (7)$$

where $M(q)$ denotes mass matrix given by

$$M(q) = \sum_{i=1}^n [A_i(q)^T m_i A_i(q) + B_i(q)^T I_i B_i(q)] \quad (8)$$

The total potential energy is determined as

$$P(q) = - \sum_{i=1}^n m_i g^T h_i(q) \quad (9)$$

where g and $h_i(q)$ denotes gravitational acceleration vector and the center of mass of link i relative to the base coordinate frame, respectively. As a result, the Lagrangian function can be obtained by combining the equations 7 and 9 as follows.

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \sum_{i=1}^n m_i g^T h_i(q) \quad (10)$$

The equations of motion can be derived by substituting the Lagrangian function in equation 10 into the Euler-Lagrange equations

$$\frac{d}{dt} \frac{\partial L(q, \dot{q})}{\partial \dot{q}} - \frac{\partial L(q, \dot{q})}{\partial q} = \tau \quad (11)$$

to create the dynamic equations with the form

$$\sum_{j=1}^n M_{ij}(q)\ddot{q}_j + \sum_{k=1}^n \sum_{j=1}^n C_{kj}^i(q)\dot{q}_k\dot{q}_j + G_i(q) = \tau_i \quad 1 \leq i, j, k \leq n \quad (12)$$

where, τ is the $n \times 1$ generalized torque vector applied at joints, and q , $\dot{q}$ and $\ddot{q}$ are the $n \times 1$ joint position, velocity and acceleration vectors, respectively. $M(q)$ is the $n \times n$ mass matrix, $C(q, \dot{q})$ is an $n \times 1$ vector of centrifugal and Coriolis terms given by

$$C_{kj}^i(q) = \frac{\partial}{\partial q_k} M_{ij}(q) - \frac{1}{2} \frac{\partial}{\partial q_i} M_{kj}(q) \quad (13)$$

$G(q)$ is an $n \times 1$ vector of gravity terms of actual mechanism expressed as

$$G_i(q) = - \sum_{k=1}^3 \sum_{j=1}^n g_k m_j A_{ki}^j(q) \quad (14)$$

The friction term is omitted in equation 12. The detailed information about Lagrangian dynamic formulation can be found in text (Schilling, 1990).

4. Trajectory Generation

In general, smooth motion between initial and final positions is desired for the end-effector of a robot manipulator since jerky motions can cause vibration in the manipulator. Joint and Cartesian trajectories in robot manipulators are two common ways to generate smooth motion. In joint trajectory, initial and final positions of the end-effector are converted into joint angles by using inverse kinematics equations. A time (t) dependent smooth function is computed for each joint. All of the robot joints pass through initial and final points at the same time. Several smooth functions can be obtained from interpolating the joint values. A 5th order polynomial is defined here under boundary conditions of joints (position, velocity, and acceleration) as follows.

$$\begin{aligned} x(0) &= \theta_i & x(t_f) &= \theta_f \\ \dot{x}(0) &= \dot{\theta}_i & \dot{x}(t_f) &= \dot{\theta}_f \\ \ddot{x}(0) &= \ddot{\theta}_i & \ddot{x}(t_f) &= \ddot{\theta}_f \end{aligned}$$

These boundary conditions uniquely specify a particular 5th order polynomial as follows.

$$x(t) = s_0 + s_1 t + s_2 t^2 + s_3 t^3 + s_4 t^4 + s_5 t^5 \quad (15)$$

The desired velocity and acceleration calculated, respectively

$$\dot{x}(t) = s_1 + 2s_2 t + 3s_3 t^2 + 4s_4 t^3 + 5s_5 t^4 \quad (16)$$

and

$$\ddot{x}(t) = 2s_2 + 6s_3 t + 12s_4 t^2 + 20s_5 t^3 \quad (17)$$

where $s_0, s_1, \dots, s_5$ are the coefficients of the polynomial and given by

$$s_0 = \theta_i \tag{18}$$

$$s_1 = \dot{\theta}_i \tag{19}$$

$$s_2 = \frac{\ddot{\theta}_i}{2} \tag{20}$$

$$s_3 = \frac{20(\theta_f - \theta_i) - (8\dot{\theta}_f + 12\dot{\theta}_i)t_f + (\ddot{\theta}_f - 3\ddot{\theta}_i)t_f^2}{2t_f^3} \tag{21}$$

$$s_4 = \frac{30(\theta_i - \theta_f) + (14\dot{\theta}_f + 16\dot{\theta}_i)t_f + (3\ddot{\theta}_i - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \tag{22}$$

and

$$s_5 = \frac{12(\theta_f - \theta_i) - 6(\dot{\theta}_f + \dot{\theta}_i)t_f - (\ddot{\theta}_i - \ddot{\theta}_f)t_f^2}{2t_f^5} \tag{23}$$

where $\theta_i, \theta_f, \dot{\theta}_i, \dot{\theta}_f, \ddot{\theta}_i, \ddot{\theta}_f$ denote initial and final position, velocity and acceleration, respectively. t_f is the time at final position. Fig. 1 illustrates the position velocity and acceleration profiles for a single link robot manipulator. Note that the manipulator is assumed to be motionless at initial and final positions.

Figure 1. Position, velocity and acceleration profiles for a single link robot manipulator

5. Problem Formulation

If the manipulator moves freely in the workspace, the dynamic equation of motion for n DOF robot manipulator is given by the equation 12. This equation can be written as a simple matrix form as

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \quad (24)$$

Note that, the inertia matrix $M(q)$ is a symmetric positive definite matrix. The local optimization of the joint torque weighted by inertia results in resolutions with global characteristics. That is the solution to the local minimization problem (Lui & Wang, 2004). The formulation of the problem is

$$\text{Minimize} \quad \tau^T M^{-1} \tau \quad (m_1, m_2, m_3)$$

Subjected to

$$J(q)\ddot{q} + \dot{J}(q)\dot{q} - \ddot{r} = 0 \quad (25)$$

where $f(q)$ is the position vector of the end-effector. $J(q)$ is the Jacobian matrix and defined as

$$J(q) = \frac{\partial f(q)}{\partial q} \quad (26)$$

The total kinetic energy is given by

$$\frac{1}{2} \int_{t_0}^{t_f} \dot{q}^T M \dot{q} \quad (27)$$

The objective function in equation 25 also minimizes this kinetic energy.

6. Example

A three-link robot manipulator including two revolute joints and a prismatic joint is chosen as an example to examine the optimum energy performance. The transformation matrices are obtained using well known D-H method (Denavit & Hartenberg, 1955). For simplification, each link of the robot manipulator is modelled as a homogeneous cylindrical or prismatic beam of mass, m which is located at the centre of each link. The kinematics and dynamic representations for the three-link robot manipulator are shown in Fig. 2.

Figure 2. Kinematics and dynamic representations for the three-link robot manipulator

The kinematics and dynamic link parameters for the three-link robot are given in Table 1.

i	θ_i	α_{i-1}	a_{i-1}	d_i	q_i	m_i	L_{C_i}	I_{m_i}
1	θ_1	0	0	h_1	θ_1	m_1	$\frac{l_1}{2}$	$I_{m_1} = \text{diag}(I_{xx_1}, I_{yy_1}, I_{zz_1})$
2	θ_2	0	l_1	0	θ_2	m_2	$\frac{l_2}{2}$	$I_{m_2} = \text{diag}(I_{xx_2}, I_{yy_2}, I_{zz_2})$
3	0	0	l_2	d_3	d_3	m_3	$\frac{l_3}{2}$	$I_{m_3} = \text{diag}(I_{xx_3}, I_{yy_3}, I_{zz_3})$

Table 1. The kinematics and dynamic link parameters for the three-link robot manipulator

where m_i and L_{C_i} stand for link masses and the centers of mass of links, respectively. The matrix I_m includes only the diagonal elements I_{xx} , I_{yy} and I_{zz} . They are called the principal moment of inertia about x, y and z axes. Since the mass distribution of the rigid object is symmetric relative to the body attached coordinate frame, the cross products of inertia are taken as zero. The transformation matrices for the robot manipulator illustrated in Fig. 2 are obtained using the parameters in Table 1 as follows.

$${}^0_iT = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(28)

$${}^1_2T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(29)

and

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

The inertia matrix for the robot manipulator can be derived in the form

$$M = \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{12} & M_{22} & 0 \\ 0 & 0 & m_3 \end{bmatrix} \quad (31)$$

where

$$M_{11} = \frac{1}{4}m_1l_1^2 + I_{zz_1} + m_2\left(\frac{1}{4}l_2^2 + l_1l_2 \cos\theta_2 + l_1^2\right) + I_{zz_2} + m_3(l_2^2 + 2l_1l_2 \cos\theta_2 + l_1^2) + I_{zz_3}, \quad (32)$$

$$M_{12} = m_2\left(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos\theta_2\right) + I_{zz_2} + m_3(l_2^2 + l_1l_2 \cos\theta_2) + I_{zz_3} \quad (33)$$

and

$$M_{22} = \frac{1}{4}m_2l_2^2 + I_{zz_2} + l_2^2m_3 + I_{zz_3} \quad (34)$$

Generalized torque vector applied at joints are

$$\tau = [\tau_1 \ \tau_2 \ \tau_3]^T \quad (35)$$

where

$$\begin{aligned} \tau_1 = & \left[\frac{1}{4}m_1l_1^2 + I_{zz_1} + m_2\left(\frac{1}{4}l_2^2 + l_1l_2 \cos\theta_2 + l_1^2\right) + I_{zz_2} + m_3(l_2^2 + 2l_1l_2 \cos\theta_2 + l_1^2) + I_{zz_3} \right] \ddot{\theta}_1 \\ & + \left[m_2\left(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos\theta_2\right) + I_{zz_2} + m_3(l_2^2 + l_1l_2 \cos\theta_2) + I_{zz_3} \right] \ddot{\theta}_2 \\ & - l_1l_2 \sin\theta_2(m_2 + 2m_3)\dot{\theta}_1\dot{\theta}_2 - l_1l_2 \sin\theta_2\left(\frac{1}{2}m_2 + m_3\right)\dot{\theta}_2^2, \\ \tau_2 = & \left[m_2\left(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos\theta_2\right) + I_{zz_2} + m_3(l_2^2 + l_1l_2 \cos\theta_2) + I_{zz_3} \right] \ddot{\theta}_1 \end{aligned} \quad (36)$$

$$+ \left[\frac{1}{4} m_2 l_2^2 + I_{zz_2} + l_2^2 m_3 + I_{zz_3} \right] \ddot{\theta}_2 + \left[\frac{1}{2} l_1 l_2 m_2 \sin \theta_2 + l_1 l_2 m_3 \sin \theta_2 \right] \dot{\theta}_1^2 \quad (37)$$

$$\tau_3 = m_3 \ddot{d}_3 + g m_3 \quad (38)$$

As a result, the objective function is obtained as follows.

$$\begin{aligned} \tau^T M^{-1} \tau = & \left(-\tau_1 \frac{m_3 l_2^2 + I_{zz_3}}{m_3 l_1^2 (m_3 l_2^2 \cos^2 \theta_2 - l_2^2 m_3 - I_{zz_3})} + \tau_2 \frac{m_3 l_2^2 + m_3 l_1 l_2 \cos \theta_2 + I_{zz_3}}{m_3 l_1^2 (m_3 l_2^2 \cos^2 \theta_2 - l_2^2 m_3 - I_{zz_3})} \right) \tau_1 \\ & + \left(\tau_1 \frac{m_3 l_2^2 + m_3 l_1 l_2 \cos \theta_2 + I_{zz_3}}{m_3 l_1^2 (m_3 l_2^2 \cos^2 \theta_2 - l_2^2 m_3 - I_{zz_3})} - \tau_2 \frac{m_3 l_2^2 + 2 m_3 l_1 l_2 \cos \theta_2 + m_3 l_1^2 + I_{zz_3}}{m_3 l_1^2 (m_3 l_2^2 \cos^2 \theta_2 - l_2^2 m_3 - I_{zz_3})} \right) \tau_2 \\ & + \left(\frac{1}{m_3} \tau_3 \right) \tau_3 \end{aligned} \quad (39)$$

6.1 Optimization with Genetic Algorithms

The displacement of the end-effector with respect to the time is illustrated in Fig. 3. The end-effector starts from the initial position $p_{xi}=44$, $p_{yi}=0$, $p_{zi}=18$ at $t=0$ second, and reach the final position $p_{xf}=13.2721$, $p_{yf}=12.7279$, $p_{zf}=2$ at $t=3$ seconds in Cartesian space.

When the end-effector perform a motion from the initial position $p_{xi}=44$, $p_{yi}=0$, $p_{zi}=18$ to the final position $p_{xf}=13.2721$, $p_{yf}=12.7279$, $p_{zf}=2$ in 3 seconds in Cartesian space, each joint has position velocity and acceleration profiles obtained from the 5th order polynomial as shown in Fig. 4.

Ten samples obtained from Fig. 4 at time intervals between 0 and 3 seconds for the optimization problem are given in Table 2. Pos.-i, Vel.-i and Acc.-i represent position, velocity and acceleration of i^{th} joint of the robot manipulator, respectively ($1 \leq i \leq 3$). The letter "S" represents sample.

Figure 3. The displacement of the end-effector with respect to the time

Figure 4. The position velocity and acceleration profiles for first, second and third joints

S	Pos.-1	Vel.-1	Acc.-1	Pos.-2	Vel.-2	Acc.-2	Pos.-3	Vel.-2	Acc.-3
1	0.126	3.737	72.177	0.047	1.401	27.066	0.005	0.166	3.207
2	6.917	48.071	203.37	2.594	18.02	76.266	0.307	2.136	9.039
3	31.220	115.20	228.97	11.707	43.20	85.866	1.387	5.120	10.176
4	75.555	177.77	177.77	28.333	66.66	66.666	3.358	7.901	7.901
5	135.53	217.07	78.577	50.823	81.40	29.466	6.023	9.647	3.492
6	202.43	223.00	-39.822	75.912	83.62	-14.933	8.997	9.911	-1.769
7	265.82	194.13	-48.622	99.684	72.80	-55.733	11.814	8.628	-6.605
8	316.12	137.67	-19.022	118.54	51.62	-82.133	14.050	6.118	-9.734
9	347.22	69.444	-22.222	130.20	26.04	-83.333	15.432	3.086	-9.876
10	359.03	13.937	-29.422	134.63	5.226	-48.533	15.957	0.619	-5.752

Table 2. Position, velocity and acceleration samples of first, second and third joints

In robot design optimization problem, the link masses m_1 , m_2 and m_3 are limited to an upper bound of 10 and to a lower bound of 0. The objective function does not only depend on the design variables but also the joint variables (q_1 , q_2 , q_3) which have lower and upper bounds ($0 < q_1 < 360$, $0 < q_2 < 135$ and $0 < q_3 < 16$), on the joint velocities ($\dot{q}_1$, $\dot{q}_2$, $\dot{q}_3$) and joint accelerations ($\ddot{q}_1$, $\ddot{q}_2$, $\ddot{q}_3$). The initial and final velocities of each joint are defined as zero. In order to optimize link masses, the objective function should be as small as possible at all working positions, velocities and accelerations. The following relationship was adapted to specify the corresponding fitness function.

$$\tau^T M^{-1} \tau = \tau^T M^{-1} \tau(1) + \tau^T M^{-1} \tau(2) + \dots + \tau^T M^{-1} \tau(9) + \tau^T M^{-1} \tau(10)$$

(40)

In the GA solution approach, the influences of different population sizes and mutation rates were examined to find the best GA parameters for the mass optimization problem where the minimizing total kinetics energy. The GA parameters, population sizes and mutation rates were changed between 20-60 and 0.005-0.1 where the number of iterations was taken 50 and 100, respectively. The GA parameters used in this study were summarized as follows.

Population size	:20, 40, 60
Mutation rate	:0.005, 0.01, 0.1
Number of iteration	:50, 100

All results related these GA parameters are given in Table 3. As can be seen from Table 3, population sizes of 20, mutation rate of 0.01 and iteration number of 50 produce minimum kinetic energy of $0.892 \cdot 10^{-14}$ where the optimum link masses are $m_1=9.257$, $m_2=1.642$ and $m_3=4.222$.

Population size	20	20	20	20	20	20	40	40	40
Mutation rate	0.005	0.01	0.1	0.005	0.01	0.1	0.005	0.01	0.1
Number of iteration	50	50	50	100	100	100	50	50	50
Fitness Value (10 ⁻¹⁴)	1.255	0.892	5.27	1.619	3.389	554.1	1.850	8.517	6.520
m ₁	0.977	9.257	6.559	3.421	3.167	0.146	7.507	4.271	9.462
m ₂	1.915	1.642	0.508	1.661	0.508	0.039	0.185	0.254	0.156
m ₃	1.877	4.222	0.205	0.889	1.446	0.009	2.717	0.332	0.664

Population size	40	40	40	60	60	60	60	60	60
Mutation rate	0.005	0.01	0.1	0.005	0.01	0.1	0.005	0.01	0.1
Number of iteration	100	100	100	50	50	50	100	100	100
Fitness Value (10 ⁻¹⁴)	1.339	2.994	13.67	4.191	1.876	12.35	2.456	4.362	19.42
m ₁	3.040	3.773	9.677	5.347	3.949	0.078	6.432	6.041	1.349
m ₂	0.048	0.889	0.117	0.449	1.397	0.078	0.713	0.381	0.058
m ₃	3.958	0.351	0.263	0.772	0.498	0.351	1.388	0.811	0.215

Table 3. The GA parameters, fitness function and optimized link masses

7. Conclusion

In this chapter, the link masses of the robot manipulators are optimized using GAs to obtain the best energy performance. First of all, fundamental knowledge about genetic algorithms, dynamic equations of robot manipulators and trajectory generation were presented in detail. Second of all, GA was applied to find the optimum link masses for the three-link serial robot manipulator. Finally, the influences of different population sizes and mutation rates were searched to achieve the best GA parameters for the mass optimization. The optimum link masses obtained at minimum torque requirements show that the GA optimization technique is very consistent in robotic design applications. Mathematically simple and easy coding features of GA also provide convenient solution approach in robotic optimization problems.

8. References

Denavit, J.; Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices, *Journal of Applied Mechanics*, pp. 215-221, vol. 1, 1955

Hollerbach, J. M. (1980). A Recursive Lagrangian Formulation of Manipulator Dynamics and A Comparative Study of Dynamic Formulation Complexity, *IEEE Trans. System., Man, Cybernetics.*, pp. 730-736, vol. SMG-10, no. 11, 1980

- Luh, J. Y. S.; Walker, M. W. & Paul, R. P. C. (1980). On-Line Computational Scheme for Mechanical Manipulators, *Trans. ASME, J. Dynamic Systems, Measurement and Control*, vol. 102, no. 2, pp. 69-76, 1980
- Paul, R. P. (1981). Robot Manipulators: Mathematics, Programming, and Control, MIT Press, Cambridge Mass., 1981
- Kane, T. R.; Levinson, D. A. (1983). The Use of Kane's Dynamic Equations in Robotics, *Int. J. of Robotics Research*, vol. 2, no. 3, pp. 3-20, 1983
- Lee, C. S. G.; Lee, B. H. & Nigam, N. (1983). Development of the Generalized D'Alembert Equations of Motion for Robot Manipulators, *Proc. Of 22nd Conf. on Decision and Control*, pp. 1205-1210, 1983, San Antonio, TX
- Nedungadi, A.; Kazerounian, K. (1989). A local solution with global characteristics for joint torque optimization of redundant manipulator, *Journal of Robotic Systems*, vol. 6, 1989
- Schilling, R. (1990). Fundamentals of Robotics Analysis and Control, Prentice Hall, 1990, New Jersey
- Delingette, H.; Herbert, M. & Ikeuchi, K. (1992). Trajectory generation with curvature constraint based on energy minimization, *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pp. 206-211, Japan, 1992, Osaka
- Garg, D.; Ruengcharungpong, C. (1992). Force balance and energy optimization in cooperating manipulation, *Proceedings of the 23rd Annual Pittsburgh Modeling and Simulation Conference*, pp. 2017-2024, vol. 23, Pt. 4, USA, 1992, Pittsburgh
- Pin, F.; Culioli, J. (1992). Optimal positioning of combined mobile platform-manipulator systems for material handling tasks, *Journal of Intelligent Robotic System, Theory and Application* 6, pp. 165-182, 1992
- Painton, L.; Campbell, J. (1995). Genetic algorithms in optimization of system reliability, *IEEE Transactions on Reliability*, pp. 172-178, 44 (2), 1995
- Hirakawa, A.; Kawamura, A. (1996). Proposal of trajectory generation for redundant manipulators using variational approach applied to minimization of consumed electrical energy, *Proceedings of the Fourth International Workshop on Advanced Motion Control*, Part 2, pp. 687-692, , Japan, 1996, Tsukuba
- Chen, M.; Zalzas, A. (1997). A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration, *Journal of Robotic Systems*, pp. 529-544, vol. 14, 1997
- Stocco, L.; Salcudean, S. E. & Sassani, F. (1998). Fast constraint global minimax optimization of robot parameters, *Robotica*, pp. 595-605, vol. 16, 1998
- Choi, S.; Choi, Y. & Kim, J. (1999). Optimal working trajectory generation for a biped robot using genetic algorithm, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1456-1461, vol. 3, 1999
- Pires, E.; Machado, J. (1999). Trajectory planner for manipulators using genetic algorithms, *Proceeding of the IEEE Int. Symposium on Assembly and Task Planning*, pp. 163-168, 1999
- Sexton, R.; Gupta, J. (2000). Comparative evaluation of genetic algorithm and back propagation for training neural networks, *Information Sciences*, pp. 45-59, 2000
- Garg, D.; Kumar, M. (2002). Optimization techniques applied to multiple manipulators for path planning and torque minimization, *Journal for Engineering Applications of Artificial Intelligence*, pp. 241-252, vol. 15, 2002

- Tang, W.; Wang, J. (2002). Two recurrent neural networks for local joint torque optimization of kinematically redundant manipulators, *IEEE Trans. Syst. Man Cyber*, Vol. 32(5), 2002
- Lui, S.; Wang, J. (2004). A dual neural network for bi-criteria torque optimization of redundant robot manipulators, *Lecture Notes in Computer Science*, vol. 3316, pp. 1142-1147, 2004
- Kucuk S.; Bingul Z. (2006). Link mass optimization of serial robot manipulators using genetic algorithms, *Lecture Notes in Computer Science, Knowledge-Based Intelligent Information and Engineering Systems*, vol. 4251, pp. 138-144, 2006
- Qudeiri, J. A.; Yamamoto, H. & Ramli, R. (2007). Optimization of operation sequence in CNC machine tools using genetic algorithm, *Journal of Advance Mechanical Design, Systems, and Manufacturing*, pp. 272-282, vol. 1, no. 2, 2007

IntechOpen

Robot Manipulators

Edited by Marco Ceccarelli

ISBN 978-953-7619-06-0

Hard cover, 546 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

In this book we have grouped contributions in 28 chapters from several authors all around the world on the several aspects and challenges of research and applications of robots with the aim to show the recent advances and problems that still need to be considered for future improvements of robot success in worldwide frames. Each chapter addresses a specific area of modeling, design, and application of robots but with an eye to give an integrated view of what make a robot a unique modern system for many different uses and future potential applications. Main attention has been focused on design issues as thought challenging for improving capabilities and further possibilities of robots for new and old applications, as seen from today technologies and research programs. Thus, great attention has been addressed to control aspects that are strongly evolving also as function of the improvements in robot modeling, sensors, servo-power systems, and informatics. But even other aspects are considered as of fundamental challenge both in design and use of robots with improved performance and capabilities, like for example kinematic design, dynamics, vision integration.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Serdar Kucuk and Zafer Bingul (2008). Link Mass Optimization Using Genetic Algorithms for Industrial Robot Manipulators, Robot Manipulators, Marco Ceccarelli (Ed.), ISBN: 978-953-7619-06-0, InTech, Available from: http://www.intechopen.com/books/robot_manipulators/link_mass_optimization_using_genetic_algorithms_for_industrial_robot_manipulators

INTech
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen