Mechanics and Mechanical Engineering
Vol. 21, No. 2 (2017) 253–265

Vibration Optimization of a Two–Link Flexible Manipulator with Optimal Input Torques

Milad Pouya
School of Mechanical Engineering
University of Guilan
Rasht, Iran
Miladpouya5@gmail.com

Pooyan Vahidi Pashaki
School of Mechanical Engineering
Islamic Azad University
Tehran, Iran
pouyan.vahidi66@gamil.com

Received (11 January 2017)
Revised (16 January 2017)
Accepted (11 February 2017)

This paper is concerned with the optimal path planning for reduction in residual vibration of two–flexible manipulator. So after presenting the model of a two–link flexible manipulator, the dynamic equations of motion were derived using the assumed modes method. Assuming a desired path for the end effector, the robot was then optimized by considering multiple objective functions. The objective functions should be defined such that in addition to guaranteeing the end effector to travel on the desired path, they can prevent the undesirable extra vibrations of the flexible components. Moreover, in order to assure a complete stop of the robot at the end of the path, the velocity of the end effector at the final point in the path should also reach zero. Securing these two objectives, a time–optimal control may then be applied in order for the robot to travel the path in the minimum duration possible. In all the scenarios, the input motor torques applied to the Two–link are determined as the optimization variables in a given range. The optimization procedures were carried out based on the GA (Genetic Algorithm) and BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithms, and the results are then compared. It is observe that the BFGS algorithm was able to achieve better results compared to GA running a lower number of iterations. Then the final value of the objective function after optimization indicates the decrease in the vibrations of the end effector at the tip of the flexible link.

Keywords: optimization, two–link flexible manipulator, path planning, vibration, Genetic Algorithm (GA), Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm.

1. Introduction

Path planning of a robot between two given states is considered among the important topics in designing industrial robots. Generally, the problem involves using an optimization algorithm in order to find the optimal input torques, by application of which to the system, the end effector of the robot travels the desired path with an acceptable accuracy. Due to the effects of flexibility in flexible robots, achieving an acceptable accuracy has always been a difficulty. Multiple contradictory objective functions such as high stiffness and damping, low–mass, and high accuracy should be considered simultaneously in the optimization process of these robots. Hence, the selection of appropriate objective functions is of great importance in this regard.

Optimization of flexible manipulators is far more difficult than rigid manipulators since multiple objectives such as high stiffness, high damping capacity, low link weight, and high accuracy must be met in order to achieve a high performance. The concept of optimization calls for objective function(s) which may be used as a performance criterion for the design. In the optimization procedure of robots, weight, workspace, supplied energy, etc. may be considered as the objective functions of the design process, while each of which is dependent on several other design variables. The optimization objectives in designing flexible manipulators include increasing stiffness, reducing weight, increasing accuracy coefficient, reducing the end effector deviation from the considered point, and maximizing the operation speed and acceleration. For the accuracy to increase, the deviation of the end effector at high speeds should be reduced. Furthermore, increases in speed and accuracy result in increased efficiency.

Most research on optimization has been conducted using specific algorithms such as genetic and gradient descent algorithms. Hiroyoki and Tetsoji [1] and Kojima [2] used GA to reduce the remaining vibrations in a two–link rigid–flexible manipulator and to optimize the motion trajectory. In [1], the angular velocity of the joints was determined by a third degree polynomial with three parameters and a fitness function with four parameters. In [2], the joint angles were described using a fourth degree polynomial. Using GA, Anderson [3] optimized the system and investigated the effect of different design parameters on the objective function. In the present study, the control error and the energy were first minimized, and then, the effect of different parameters on the objective function was determined. Optimization of Diamond robot is considered for the future work. Lee [4], first, derived the equilibrium equations of the robot using the Euler–Lagrange method, and then investigated the effect of optimization on different parameters such as natural frequency and dynamical stress. Rahman [5] carried out an optimal design of a 3–DOF planar robot with parallel links. He first described the kinematic model of the manipulator and derived the equations of motion as well as the matrices of mass, stiffness, and damping. He then considered two objective functions based on mass and workspace so that the former is minimized and the latter is maximized. Hegde [6] used the Euler–Bernoulli beam theory and the finite element method (FEM) to derive the dynamic equations for the optimization process, for which the intermediate method was utilized. Neto [7] proposed an optimization procedure for finding the optimized design structure of a composite fiber in order to be used in flexible systems. The elastic energy of the flexible bar, which itself is dependent on the layer orientations, was considered

as the objective function in this procedure. The purpose of this optimization was to reduce the elastic deformation of the plane, hence raising the need for planes with higher stiffness.
In this study, after presenting the model of a two-link flexible manipulator, the dynamic equations of motion were derived using the assumed modes method. Assuming a desired path for the end effector, the robot was then optimized by taking into account multiple objective functions. The objective functions were defined such that in addition to guaranteeing the end effector to travel on the desired path, they could prevent the undesirable extra vibrations of the flexible components. Moreover, in order to assure a complete stop of the robot at the end of the path, the velocity of the end effector at the final point in the path should also reach zero. After securing these two objectives, a time-optimal control may then be applied in order for the robot to travel the path in the minimum duration possible. In all the scenarios, the input motor torques applied to the Two-Link were determined as the optimization variables in a given range such that all the considered objectives are achieved.
As compared to the studies reviewed in the literature on the same subject, the main differences and contributions of the paper lie in the following points: the considered objective function, application of BFGS algorithm, optimization variables in order to identify the most influential optimal parameter.

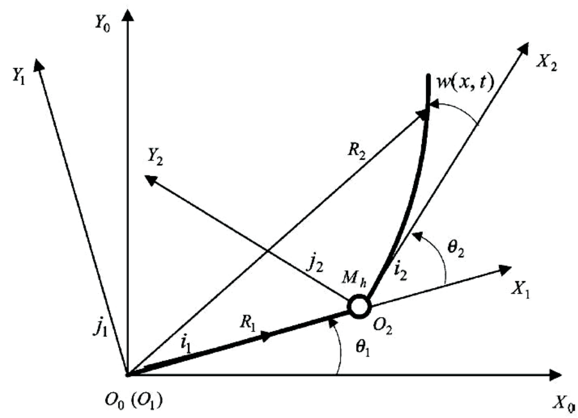

Figure 1 The schematic diagram of a two–link rigid–flexible manipulator

2. Equations of motion for the two–link rigid–flexible manipulator

The considered system in this section includes two members which, as demonstrated in Fig. 1, are connected to each other by a revolute joint and are only capable of planar motions. The first member is considered rigid, while the second is flexible and is modeled as a flexible narrow beam. Longitudinal deformations are neglected in the second member. It is assumed that the second member may freely bend in

the horizontal plane but can resist vertical bending as well as torsion. Hence, the Euler-Bernoulli theory may appropriately be used to describe the bending motions of the flexible member. In addition, the Lagrange equation can be used to derive the dynamic model of the two-link manipulator.
According to Fig. 1, X_0OY_0 is the fixed coordinate system, and X_1OY_1 and X_2OY_2 are the moving coordinate systems attached to the joints corresponding to the rigid and flexible links, respectively. In addition, θ_1 and θ_2 are the rotation angles of each of the links with respect to the X axis of their previous coordinate system, and $w(x,t)$ is the elastic transverse displacement of the flexible member. Since the bending motions of a beam do not impose significant axial vibrations, axial deformations were not included in our study. Two perpendicular pairs of unit vectors (i_1, j_1) and (i_2, j_2) attached to the moving coordinates of the links are shown in Fig. 1. The position vectors of the points on the Two-Link are $\mathbf{R}_1$ and $\mathbf{R}_2$, which are obtained according to the following relations:

$$R_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} r_1 \cos\theta_1 \\ r_1 \sin\theta_1 \end{bmatrix} \tag{1}$$

$$R_{\ 2} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 \cos\theta_1 + r_2 \cos(\theta_1+\theta_2) - w \sin(\theta_1+\theta_2) \\ l_1 \sin\theta_1 + r_2 \sin(\theta_1+\theta_2) + w \cos(\theta_1+\theta_2) \end{bmatrix} \tag{2}$$

where r_1 and r_2 are the distances of the points on links 1 and 2 to the origin of their corresponding coordinate systems. Moreover, l_1 and l_2 are the lengths of link 1 and 2, respectively. The total kinetic energy of the system is calculated as follows (Eq. (3)):

$$T = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}J_h(\dot{\theta}_1+\dot{\theta}_2)^2 + \frac{1}{2}M_h l_1^2\dot{\theta}_1^2 + \frac{1}{2}\int_0^{l_2} \dot{R}_2^T \dot{R}_2 \rho_{AL} dr_2 \tag{3}$$

where: J_1 is the moment of inertia of the first link around its rotational axis, and J_h and M_h are the moment of inertia and the mass of the driving motor acting on the second link at point O_2, respectively. ρ_{AL} is the mass linear density of the second link, and the elastic potential energy is obtained as:

$$U = \frac{1}{2}\int_0^{l_2} EI(w''(x,t))^2 dx \tag{4}$$

where: EI is the flexural rigidity of the flexible link, and $w''(x,t)$ is the second derivative of the transverse elastic displacement with respect to the variable x. Since the flexible link is considered as a fixed support beam, the following boundary conditions hold true at the two ends of this member:

$$w(0,t)=0 \qquad \frac{\partial}{\partial x}w(0,t)=0 \qquad \frac{\partial^2}{\partial x^2}w(l_2,t)=0 \qquad \frac{\partial^3}{\partial x^3}w(l_2,t)=0 \tag{5}$$

The general form of the equations of motions for the two–link rigid–flexible system is obtained as follows according to Lagrange equations:

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_i}\right] - \left[\frac{\partial L}{\partial \theta_i}\right] = \tau_i - \alpha_i \dot{\theta}_i \qquad (i = 1,\, 2) \tag{6}$$

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{w}_j}\right] - \left[\frac{\partial L}{\partial w_j}\right] = 0 \qquad (j = 1,\, 2) \tag{7}$$

where: L is the Lagrangian function defined as $L = T - U$, τ_i are the generalized torques applied to the system, and $\alpha_i \dot{\theta}_i$ are the damping torques at the i^{th} joint. Substituting the equations of kinetic and potential energies in the above relations, the dynamic equations of motion are concluded [8, 9]. Selecting the n first modes as the assumed-modes for the discretization procedure, the following centralized model is acquired for the system:

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{K}\mathbf{X} = \mathbf{F}(\mathbf{X}, \dot{\mathbf{X}}) + \mathbf{B}\mathbf{u} \tag{8}$$

where: $\mathbf{X} = [\theta_1, \theta_2, w_1, w_2, ..., w_n]^T$ is the vector of generalized coordinates, $\mathbf{M}$ and $\mathbf{K}$ are the mass and stiffness matrices, respectively, the vector $\mathbf{F}$ contains the nonlinear expressions associated with the Coriolis and centripetal forces, and $\mathbf{Bu}$ epresents the inputs to the system.

3. The employed optimization algorithm

3.1. *Broyden–Fletcher–Goldfarb–Shanno algorithm*

As a conventional gradient search method in nonlinear optimization, Hessian matrix is used as the gradient coefficient to update the weights. The algorithms based on this method are known as Newton and quasi–Newton methods, hence sharing similar fundamentals. Similar to conjugate gradient method, these algorithms converge at a high rate. The BFGS method is one of the most well-known and widely used quasi-Newton methods [10]. This algorithm is generally used for optimization of multivariable functions, the basis of which is an approximation of the Hessian matrix. The order of convergence in algorithms based on BFGS is high. Issues such as motion path planning optimization can be addressed using this algorithm. In this method, after making an initial guess $\mathbf{x}^{(0)}$, the gradient vector $\mathbf{c}$ is calculated according to the objective function $f(\mathbf{x})$.

$$\mathbf{c}^{(0)} = \nabla f(\mathbf{x}^{(0)}) \tag{9}$$

If the norm of the gradient is smaller than the suggested convergence value, the program stops iterating, otherwise, iteration continues. Then, similar to all quasi-Newton methods, the Hessian matrix is calculated. The initial value of the Hessian matrix is selected as $\mathbf{H}^{(0)} = \mathbf{I}$, and then the search direction for the k^{th} iteration is determined as:

$$\mathbf{d}^{(k)} = -\mathbf{H}^{-1}\mathbf{c}^{(k)} \tag{10}$$

Then, the optimal step α_k is selected such that $f(\mathbf{x}^k + \alpha_k \mathbf{d}^k)$ is minimized. After correcting the optimal solution according to the following relation (Eq. (11)) :

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \tag{11}$$

And correcting the Hessian matrix based on the proposed relation in [8], the iteration number is set to $k = k + 1$, and we return to the beginning of the algorithm. Generally, the optimization stops based on the norm of the gradient of the objective function and/or the allowed maximum number of iterations so that in case the given tolerance for the norm of the gradient vector is not achieved, the algorithm does not get stuck in the optimization loop.

4. Genetic Algorithm

Genetic algorithm (GA) is an efficient method for searching large, extensive spaces to eventually get directed towards finding one optimal solution, the achievement of which may not be possible during the lifetime of a person if manually searched for [10]. In this method, using a series of coded variables, the design space is transformed into the genetic space. The advantage of working with coded variables lies in the ability of codes to transform a continuous space into a discrete one. GA is significantly different than the traditional optimization algorithms. For instance, GA deals with a population or a set of points at a given time while traditional optimization methods can only be applied to a single specific point. What this feature means is that GA processes a large number of designs at a time. Another interesting feature is concerned with the basics of this method which, in fact, is built upon a guided random search process. Hence, random operators adaptively inspect the search space. Essentially, the three following concepts need to be clarified before using GA:

1. Defining the objective or cost function,
2. Defining and implementing the genetic space,
3. Defining and implementing the genetic operators.

If the above items are defined correctly, we can make sure the algorithm performs well and its performance can be increased by applying some alterations. Fig. 2 demonstrates a schematic of the GA with some specific details which will be explained in later sections.

5. Numerical optimization

In this section, by introducing the objective function and performing the optimization procedure, the optimal torques to the manipulator are determined such that the end effector starts moving from its stationary state, and after traveling the desired path, reduce its speed to zero at the end of the path. When optimal torques are applied, the end effector travels the path with the least vibrations possible. Optimal control is needed to be used in order to achieve these objectives. However, since the application of optimal control to such a nonlinear, complex system is a highly difficult task, usually optimization methods are used instead. The aforementioned objective function is considered as:

$$f = k_1((x - X)^2 + (y - Y))^2 + k_2(\dot{x}_d^2 + \dot{y}_d^2) \tag{12}$$

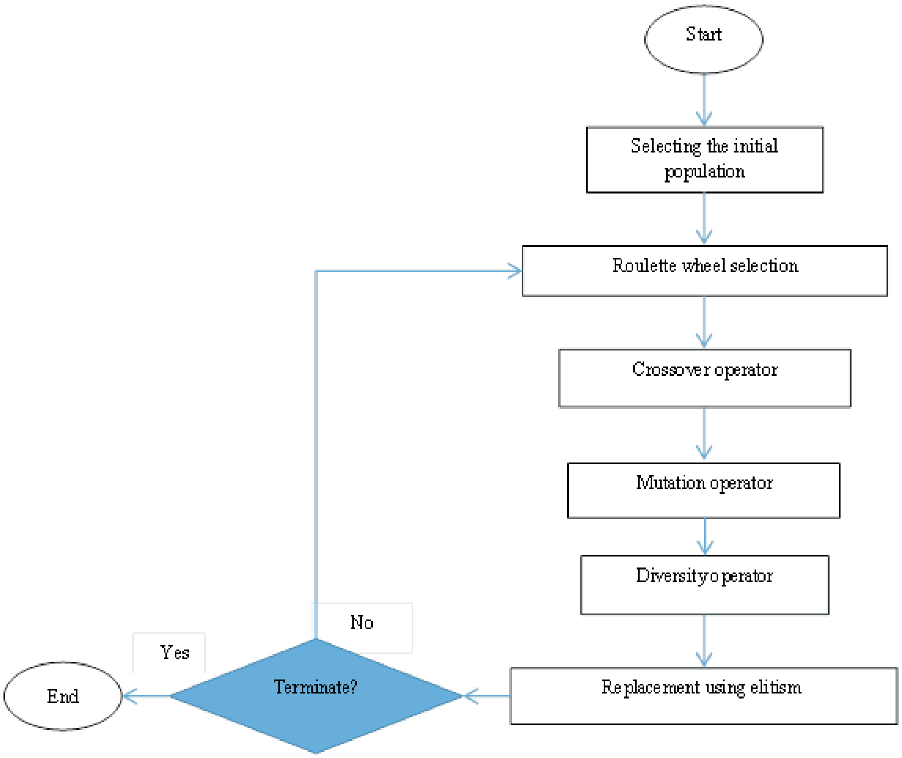

Figure 2 The mechanism of GA

where: x and y are the components of the points of the traveled path in the fixed coordinate system by the end effector attached to the tip of the flexible link and X and Y are the points of the desired path, $\dot{x}_d$ and $\dot{y}_d$ are the velocity components of the end effector at the end of the path, k_1 and k_2 are the weight coefficients of the objective function. The first expression, which is the difference of the points obtained from the execution of the plan and those of the desired path, was considered for correct traveling of the path, and the second expression for reducing the velocity to zero at the end of the path. The specifications of the links and driving motor of the flexible link are given in Tab. 1. The empty cells in the table indicate the parameters which are not present in the equations of motion.

Table 1 The specifications of the links and the driving motor of the flexible link

	Length (m)	Mass (Kg)	Damping Ratio (Nms)	Moment of Inertia (Kgm^2)
Rigid Link	0.61	—	—	0.32
Flexible Link	0.52	0.07	—	—
Motors	—	0.23	0.17	0.02

In the first part of this section, the values of the objective function in the GA and BFGS is investigated and compared. The optimization process is first performed on the rigid manipulator and then on the flexible–rigid manipulator by both algorithms. It should be noted that the coefficients of the objective function for both algorithms were $k_1 = 10000$ and $k_2 = 100$.
In order to run the optimization procedure, the considered total duration is divided into smaller intervals. The manipulator is supposed to travel the path in 0.5 s with 11 steps. The path is considered an inclined straight line and the error is the deviation from the desired path. The lower the error while traveling the path, the more successful the optimization procedure and the less the vibrations of the end effector on the flexible link. The torques at the first and second joints are the inputs to the equations of the motion of the manipulator. After transforming the equations of motion to the state-space form and substituting the initial input torques (the guessed values), by running the direct dynamics and numerically integrating the equations of motion, the position and velocity of the end effector on the second link is calculated at different consecutive moments. The objective function is calculated based on these values. In the main stage, by applying the optimization algorithm, the input torques are changed to minimize the objective function and reduce the vibrations of the manipulator. This procedure continues until the desired accuracy is achieved and the optimization algorithm is terminated. As the GA mechanism shown in Fig. 2 suggests, the initial population in the proposed algorithm is randomly selected using the roulette wheel selection. Two–point crossover occurs with a rate of 0.8 and point mutation with a rate of 0.5. The diversity operator makes modifications to all genes with a probability of 0.8. The selection operator is considered comparative, meaning that between two chromosomes, the one with the better fitness makes it to the next stage. The new population is sorted based on their fitness values and the n chromosomes with the least fitness values are determined, where n is the number of the initial population. Determination of the n chromosomes with the best fitness values results in elitism and increases the convergence rate to the optimal solution. The number of population and iterations were considered 20 and 400, respectively, as given in Tab. 2.

Table 2 The values considered for the parameters of GA

Symbol	Value	Variable name
N_{pop}	20	Initial population
—	Point mutation	Mutation type
P_m	0.5	Mutation rate
f_{m}	Dependent on the objective function (1−best/100)	Mutation intensity
—	Two-point crossover	Crossover type
P_c	0.8	Crossover rate
f_c	$.75 \pm .25 \times$r	Crossover intensity
P_d	0.8	Diversity rate
f_d	f_{m}	Diversity intensity
i_{gen}	400	Number of iterations

The value of the objective function with respect to the number of iterations is given in Fig. 3 for both algorithms. The objective function reaches the maximum allowed iterations (400) in the GA algorithm before achieving the optimal solution while the BFGS achieves the desired tolerance after 19 iterations. It can be concluded that the BFGS algorithm converges to the optimal solution at a higher rate than the GA. The value of the objective function is considerably high for the initial input torques. The slope of changes in the values of the objective function is high but gradually decreases as it converges to the optimal solution. The final value of the objective function after the optimization process indicates a reduction of vibrations at the end effector installed on the flexible link.

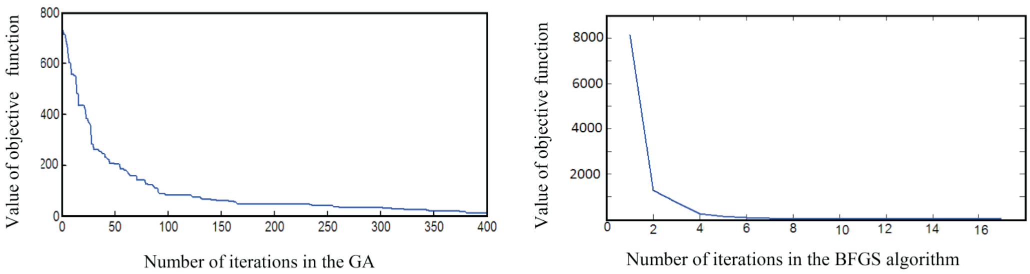

Figure 3 The value of the objective function with respect to the number of iterations in the BFGS and genetic algorithms

The path traveled by the end effector after optimization is shown in Fig. 4 for both algorithms. It is obviously observed that the vibrations of the end effector after optimization using BFGS algorithm are fewer than GA, and hence, the manipulator travels the path with fewer errors. In the BFGS algorithm, the manipulator tracks the desired path roughly adjacent to it. Regarding the GA, it should be noted that the traveled path may be improved by increasing the number of iterations, but the convergence rate would significantly be slow. The desired path and the travel duration for both algorithms were similar.

The torques applied by the first and second joints before and after optimization using the two algorithms are shown in the diagrams of Figs. 5 and 6. As shown, many changes were made to the initial torques in order to achieve the optimal solution. These changes are due to the random selection of the initial torques. The closer the initial guesses to the optimal values, the faster the convergence rate in achieving the optimal solution. Regarding the initial torques, it should be noted that their values in both algorithms were selected such that the convergence rate increased according to their corresponding algorithms.

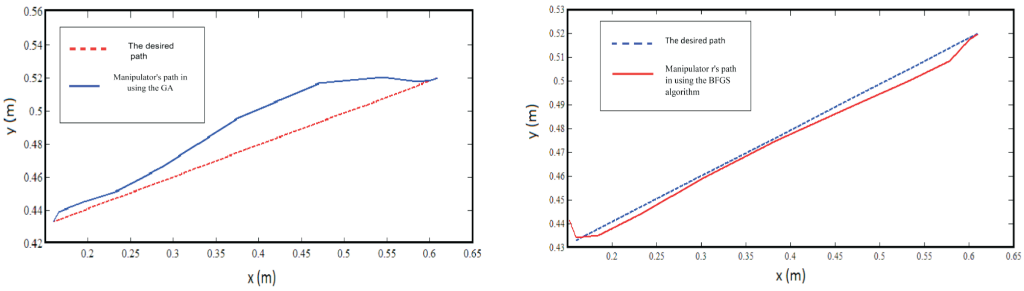

Figure 4 Comparison between the traveled path by the manipulator and the desired path using the BFGS and the genetic algorithms

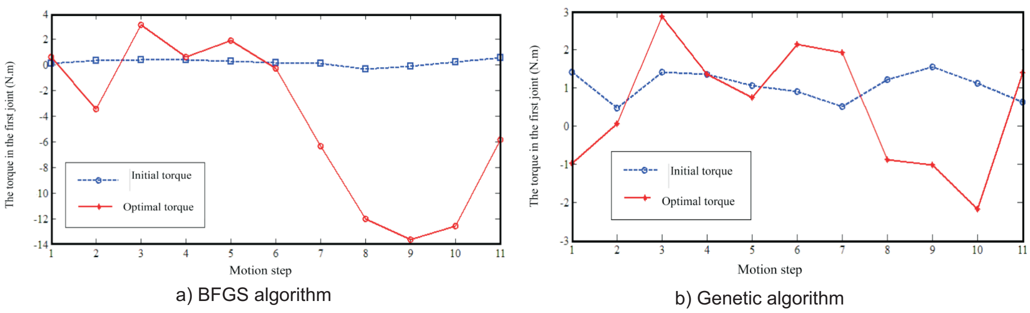

Figure 5 The torque in the first joint before and after optimization

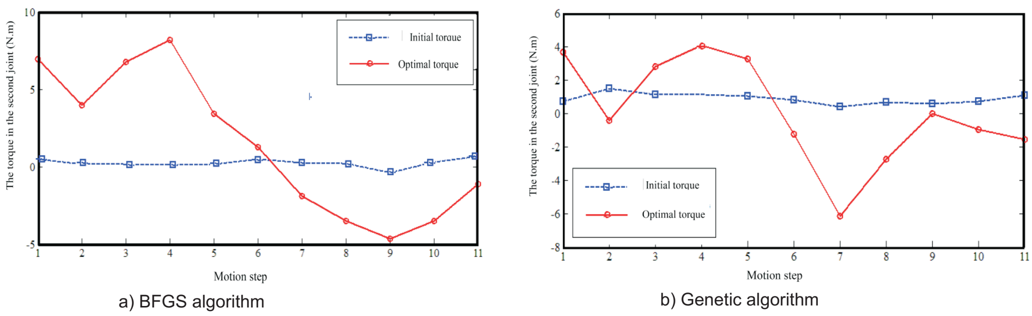

Figure 6 The torque in the second joint before and after optimization

In order to illustrate the effect of optimization on the velocities at the beginning and end of the manipulator path, the zero-velocity condition at these two points should be met. Table 3 demonstrates these changes before and after optimization, and as shown, the velocity roughly decreases to zero.

Table 3 The effect of the optimization in velocities at the beginning and end of the manipulator path

Velocity in the beginning of the path (m/s)	Velocity in the end of the path (m/s)	
0.5235	-0.8433	Before optimization
0.0446	0.0626	After optimization

6. Effect of initial population in the genetic algorithm

Since the objective function in the genetic algorithm is dependent on many parameters, efforts are focused on employing the best parameters possible. Obviously, larger sizes for the initial population results in better results in the end, however, on the other hand, it also causes longer durations to achieve the desired solution. Tab. 4 demonstrates the advantages and weaknesses of selecting large population sizes.

Table 4 The effect of initial population in optimization with genetic algorithm

Size of the initial population N_{pop}	Number of iterations i_{gen}	Algorithm runtime [s]	Value of the optimal function
10	400	669.64	17.22
20	400	1253.06	11.49
30	400	1875.32	9.23
40	400	2689.12	7.45

6.1. *Effect of number of iterations in the genetic algorithm on the optimization of two–link rigid–flexible manipulator*

The number of iterations is another parameter that highly affects the final solution. As the number of iterations increases, better solutions are achieved, however, on the other hand, longer runtimes are required for the calculations. In Table 5, the algorithm was run for different numbers of iterations.

Table 5 The effect of the number of iterations in optimization using genetic algorithm

Number of iterations i_{gen}	Number of initial population N_{pop}	Algorithm runtime [s]	Value of optimal function
300	10	394.3	22.19
400	10	669.64	17.22
500	10	849.28	14.75
600	10	1004.36	10.59
700	10	1246.49	8.37

7. Performance improvement of the genetic algorithm for a rigid–flexible manipulator

Efforts were made to increase the convergence rate of the employed GA as well as its accuracy. The number of steps for traveling the desired path is one of the factors affecting the convergence rate of the algorithm. For instance, assuming 400 iterations for the algorithm, if the number of steps is decreased from 11 to 6, the algorithm runtime is reduced by half while the value of the objective function is doubled.

Increasing the mutation rate in low values of the optimal function is another effort to increase the converging rate in the GA. Although in the beginning of the optimization process, usually the values of the optimal function quickly decrease, this value rarely decreases as we approach the end of the process. Hence, we may multiply the mutation rate by a factor to increase the chromosome mutations for lower values of the optimal function.

As it was mentioned in the previous sections, the BFGS algorithm outperforms the GA. However, the advantage of GA over BFGS algorithm is that its different parameters can freely be adjusted by the users. Hence, the performance of GA can be further improved than that of the BFGS algorithm by changing its parameters or presenting new operators. To this end, in order to improve the convergence rate and decrease the value of the objective function in GA, a novel heuristic algorithm is presented and integrated with the algorithm, the details of which are given in the following section.

8. Conclusion

In this study, after presenting a model of a two-link flexible manipulator, the dynamic equations of motion were derived using the assumed modes method. Then, considering the desired path for the end effector of the manipulator, the manipulator was optimized by utilizing a multivariable objective function. The objective functions were selected such that in addition to guaranteeing the end effector to travel on the desired path, they can prevent the undesirable extra vibrations of the flexible components. Moreover, in order to assure a complete stop of the robot at the end of the path, the velocity of the end effector at the final point in the path should also reach zero. In order to validate the results, the optimization process was carried out using the BFGS algorithm and the genetic algorithm. In all the scenarios, the input motor torques applied to the Two-Link are determined as the optimization variables in a given range such that all the considered objectives are achieved. According to the results, it was observed that the BFGS algorithm was faster than GA in converging to the optimal solution. The slope of changes in the values of the objective function is high but gradually decreases as it converges to the optimal solution. The BFGS algorithm was able to achieve better results compared to GA running a lower number of iterations. Since the final value of the objective function after optimization indicates the decrease in the vibrations of the end effector at the tip of the flexible link, the efficiency of optimization results in the reduction of vibration.

References

[1] **Kojima, H. and Kibe, T.**: Residual Vibration Reduction Control of a Two–Link Flexible Robot Arm Using Optimal Trajectory Planning based on Genetic Algorithm, *Journal of the Robotics Society of Japan*, 19, 905–12, **2001**.

[2] **Kojima, H. and Kibe, T.**: Optimal trajectory planning of a two-link flexible robot arm based on genetic algorithm for residual vibration reduction, *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems Expanding the Societal Role of Robotics in the the Next Millennium*, (Cat No01CH37180), 4, 2276–81,**2001**.

[3] **Andersson, J.**: Sensitivity analysis in Pareto optimal design, *Space*, 2, 2, **2002**.

[4] **Li, H., Yang, Z. and Huang, T.**: Dynamics and elasto-dynamics optimization of a 2–DOF planar parallel pick–and–place robot with flexible links, *Structural and Multidisciplinary Optimization*, 38, 195–204, **2009**.

[5] **Caro, S., Chablat, D., Ur–Rehman, R. and Wenger, P.**: Multiobjective Design Optimization of 3–PRR Planar Parallel Manipulators, In: *Bernard, A. editor*, Global Product Development: Proceedings of the 20th CIRP Design Conference, Ecole Centrale de Nantes, Nantes, France, 19th-21st April 2010, *Springer*, 373–83, **2011**.

[6] **Hegde, G. S., Vinod, M. S. and Shankar, A.**: Optimum dynamic design of flexible robotic manipulator, *International Journal of Mechanics and Materials in Design*, 5, 315–25, **2009**.

[7] **Neto, M. A., Ambrósio, J. and Leal, R. P.**: Sensitivity analysis of flexible multibody systems using composite materials components, *International Journal for Numerical Methods in Engineering*, 77, 386–413, **2009**.

[8] **Zhang, X., Xu, W. and Nair, S.S.**: Comparison of some modeling and control issues for a flexible two link manipulator, *ISA Transactions*, 43, 509–25, **2004**.

[9] **Dubay, R., Hassan, M., Li, C. and Charest, M.**: Finite element based model predictive control for active vibration suppression of a one-link flexible manipulator, *ISA Transactions*, 53, 1609–19, **2014**.

[10] **Arora, J. S.**: Introduction to Design Optimization, Introduction to Optimum Design, *Academic Press*, Boston, 1–15, **2012**.

[11] **Gennert, M. A. and Yuille, A. L.**: Determining the optimal weights in multiple objective function optimization, *ICCV*, 87–9, **1988**.