Chapter 9
Design Optimization of Parallel Robotic Machines

9.1 Preamble

Optimization plays an important role in engineering design problems; it deals with problems of minimizing or maximizing a function with several variables. The purpose of optimization design is aiming at enhancing the performance indices by adjusting the structure parameters such as link length, radii of fixed platform and moving platform, and its distance between the center points of the two platforms. The approach can been called dimensional-synthesis-based performance optimization of parallel manipulator. In the optimum design process, several performance criteria could be involved for a design purpose, such as stiffness, dexterity, accuracy, workspace, etc.

Many researchers have studied on the issue of optimal design of robot manipulators [19, 89, 166, 178]. Zhao et al. [182] exploited the least number method of variables to optimize the leg length of a spatial parallel manipulator for the purpose of obtaining a desired dexterous workspace. Stock and Miller [138] presented a method for multidimensional kinematic optimization of the linear Delta robot architecture's geometry. A utility objective function was formulated incorporating two performance indices, including manipulability and space utilization. Rout and Mittal [131] proposed the experimental approach for the optimization of the dimension of 2-dof R–R planar manipulator. Kucuk and Bingul [89] optimized the workspace of two spherical three links robot manipulators using the local and global performance indices. Mitchell et al. [108] presented kinematic optimization to confirm the smallest configuration that would satisfy the workspace requirements for a lightweight and compact surgical manipulator. Chablat and Angeles [31] investigated on optimum dimensioning of revolute-coupled planar manipulators based on the concept of distance of Jacobian matrix to a given isotropic matrix which was used as a reference model. Boeij et al. [23] proposed numerical integration and sequential quadratic programming method for optimization of a contactless electromagnetic planar 6-dof actuator with manipulator on top of the floating platform. Ceccarelli and Lanni [29] investigated the multiobjective optimization problem of a general 3R manipulator for prescribed workspace limits and numerically using an algebraic formulation. As the primary components of artificial intelligence approach, genetic algorithms and

D. Zhang, *Parallel Robotic Machine Tools*, DOI 10.1007/978-1-4419-1117-9_9,

artificial neural networks play the important roles in various fields of science and technology. In this chapter, the two methods are applied as the optimization criteria for the synthesis of stiffness and other criteria.

The traditional optimization methods can only handle a few geometric variables due to the lack of convergence of the optimization algorithm. However, genetic algorithms have applied the powerful and broadly applicable stochastic search methods and optimization techniques, and they can escape from local optima [69]. Therefore, genetic algorithms have been selected as the best candidate to address the convergence issue and are suitable for performance optimization of the proposed mechanism in the previous several chapters. Neural networks possess the capability of complex function approximation and generalization by simulating the basic functionality of the human nervous system in an attempt to partially capture some of its computational strengths. Since the solution of objective function must be solved before using genetic algorithms, neural networks will be conducted to represent the objective functions of performance indices. On the investigation of multiobjective optimization problem in design process, since it is impossible to maximize or minimize all objective function values if they are conflicting with each other, the trade-off process should be executed. This methodology will pave the way for providing not only the effective guidance, but also a new approach of dimensional synthesis for the optimal design of general parallel mechanisms.

For the mechanisms studied here, the highest global stiffness are desired so as to reach the high rigidity and high precision. This can be achieved either through maximizing the global stiffness or through minimizing the global compliances for a certain parallel mechanism by selecting mechanism's geometric parameters (link length, height, etc.) and behavior parameters (link stiffness). In this chapter, the optimization criteria are first established. An optimization process based on genetic algorithms is applied for the global stiffness of all the proposed spatial parallel/hybrid mechanisms for 6dof to 3dof, and the rationale for using this method together with the determination of parameters and objective function are addressed as well. The detailed analysis of the kinetostatics of the parallel mechanisms conducted in previous chapters will now be used to define and optimize their geometric sizes and properties. Furthermore, for the issue of multiobjective optimization, two cases are investigated where the integration methodology of genetic algorithms and artificial neural networks is implemented to search the optimal architecture and behavior parameters in terms of various optimization objectives including global stiffness, dexterity, and manipulability.

9.2 Optimization Objective and Criteria

In this book, the main consideration for the optimization criteria is to maximize global stiffness (or minimize the global compliances). The global stiffness/compliance used here is the diagonal entry of the Cartesian stiffness/compliance matrix. It represents the pure stiffness/compliance in each direction. Genetic algorithm methods are used to conduct the optimal design of the system in terms of a better system stiffness. The objective functions are established and maximized/minimized in order to find the suitable geometric parameters (coordinates of the attachment points, coordinates of the moving platform, link length, vertex distributions at base and moving platform, platform height, etc.) and behavior parameters (actuator stiffness, actuated link stiffness, kinetostatic model stiffness, etc.) of the mechanisms. Since the objective function is closely related to the topology and geometry of the structure, the general optimization methodology can be described as follows:

- Analyze the requirements including the stiffness, the mechanical interferences, the workspace properties, and the singularities
- Analyze the constraints including geometric size and properties
- Establish a reasonable initial guess of the geometry of the mechanism, then use a numerical optimization to further improve the kinematic properties and ensure the optimum characteristics are obtained. Finally, a program gives a potential solution to allow the verification of other important properties.

9.3 Basic Theory of Evolutionary Algorithms

Introduced in the 1970s by John Holland [69], genetic algorithms are part of the larger class of evolutionary algorithms that also include evolutionary programming [49], evolution strategies [126], and genetic programming [88]. The genetic algorithms (GAs) are powerful and broadly applicable stochastic search and optimization techniques based on the evolutionary principle of natural chromosomes [53]. Specifically, the evolution of chromosomes due to the operation of crossover and mutation and natural selection of chromosomes based on Darwin's survival-of-the-fittest principles are all artificially simulated to constitute a robust search and optimization procedure. The genetic algorithms are the computer simulation of such evolution where the user provides the environment (function) in which the population must evolve.

A comparison between conventional optimization methods and genetic algorithms is now given. The conventional methods are usually limited to convex regular functions while the genetic algorithm is robust, global, and generally more straightforward to apply to all kinds of functions including multimodal, discontinuous, and nondifferentiable functions. Goldberg [53] has summarized the differences as follows:

1. Genetic algorithms work with a coding of the solution set, not the solutions themselves.
2. Genetic algorithms search from a population of solutions, not a single solution.
3. Genetic algorithms use payoff information (fitness function), not derivatives or other auxiliary knowledge.
4. Genetic algorithms use probabilistic transition rules, not deterministic rules.

In recent years, the GAs have been applied to a broad range of real-world problems [25, 26, 42, 43, 51, 54, 107, 110, 157] such as ecosystem modeling, combinatorial and parametric optimization, reliability design, vehicle routing and scheduling, machine intelligence, robotic trajectory optimization, neural networks implementations, pattern recognition, analysis of complex systems, and financial prediction.

The basic procedure of genetic algorithms can be described as follows:

1. Create an initial population: The initial population of chromosomes is created randomly.
2. Evaluate all of the individuals (apply some function or formula to the individuals) The fitness is computed in this step. The goal of the fitness function is to numerically encode the performance of the chromosomes.
3. Selection: Select a new population from the old population based on the fitness of the individuals as given by the evaluation function. In this step, the chromosomes with the largest fitness rates are selected while the chromosomes with low fitness rates are removed from the population.
4. Genetic operations (mutation and crossover): If the parents are allowed to mate, a recombination operator is employed to exchange genes between the two parents to produce two children. If they are not allowed to mate, the parents are placed into the next generation unchanged. A mutation simply changes the value for a particular gene.
5. Evaluate these newly created individuals.
6. Repeat steps 3–5 (one generation) until the termination criteria have been satisfied.

Suppose $P(t)$ and $C(t)$ are parents and children in current generation t, then a genetic algorithm is expressed in Fig. 9.1. From Fig. 9.1, one can find that there are only two kinds of operations included in genetic algorithms, i.e., genetic operations (crossover and mutation) and evolution operation (selection).

```
begin
  t = 0;
  initialize P(t);
  evaluate P(t);
  while (unfinished condition) do
    select P ' (t) from P(t);
    reproduce C(t) from P ' (t);
    mutate C(t);
    evaluate  C(t);
    t = t + 1;
  end
end
```

Fig. 9.1 The structure of genetic algorithms

Genetic algorithms have the advantages of robustness and good convergence properties, namely:

- They require no knowledge or gradient information about the optimization problems. They can solve any kind of objective functions and any kind of constraints (i.e., linear or nonlinear) defined on discrete, continuous, or mixed search spaces.
- Discontinuities present on the optimization problems have little effect on the overall optimization performance.
- They are effective at performing global search (in probability) instead of local optima.
- They perform very well for large-scale optimization problems.
- They can be employed for a wide variety of optimization problems.

Genetic algorithms have been shown to solve linear and nonlinear problems by exploring all regions of state space and exponentially exploiting promising areas through mutation, crossover, and selection operations applied to individuals in the population [107].

In the present work, there are many optimization parameters (up to 13 variables, depending on mechanism, make up the optimization problem) and complex matrix computations. Hence, it is very difficult to write out the analytical expressions for each stiffness element. Moreover, with traditional optimization methods, only a few geometric parameters [60] could be handled due to the lack of convergence of the optimization algorithm when used with more complex problems. This arises from the fact that traditional optimization methods use a local search by a convergent stepwise procedure (e.g., gradient, Hessians, linearity, and continuity), which compares the values of the next points and moves to the relative optimal points. Therefore, genetic algorithms are the best candidate for the optimization problems studied here.

In order to use genetic algorithms properly, several parameter settings have to be determined, they are: chromosome representation, selection function, genetic operators, the creation of the population size, mutation rate, crossover rate, and the evaluation function.

1. Chromosome representation:
   This is a basic issue for the GA representation, it is used to describe each individual in the population of interest. In the original algorithm, each individual or chromosome used to be expressed as a sequence of genes from binary digits (0 and 1) [69]. However, it has been shown that more natural representations are more efficient and produce better solutions [107]. Michalewicz [107] has done extensive experimentation comparing real-valued and binary genetic algorithms and shows that the real-valued genetic algorithm is an order of magnitude more efficient in terms of CPU time. He also shows that a real-valued representation moves the problem closer to the problem representation which offers higher precision with more consistent results across replications [107]. It outperformed

binary genetic algorithm and simulated annealing in terms of computational efficiency and solution quality [124]. Hence, real-valued expressions are used in our case to represent each individual or chromosome for function optimization. For the problem studied here, the chromosomes consist of the architecture parameters (coordinates of the attachment points, coordinates of the moving platform, link lengths, vertex distributions at base and moving platform, platform height, etc.) and behavior parameters (actuator stiffness, actuated link stiffness, kinetostatic model stiffness, etc.) of the mechanisms.

2. Selection function:

   This step is a key procedure to produce the successive generations. It determines which of the individuals will survive and continue on to the next generation. A probabilistic selection is performed based on the individual's fitness such that the better individuals have an increased chance of being selected. There are several methods for selection: roulette wheel selection and its extensions, scaling techniques, tournament, elitist models, and ranking methods [53, 107]. In our case, the normalized geometric ranking method [81] is used since it only requires the evaluation function to map the solutions to a partially ordered set and it tends to eliminate chromosomes with extreme values, thus allowing for minimization and negativity. In normalized geometric ranking methods, Joines and Houck [81] define a probability of selection P_i for each individual as

$$P[\text{selecting the } i\text{th individual}] = q'(1-q)^{(r-1)}, \tag{9.1}$$

   where q represents the probability of selecting the best individual; r represents the rank of the individual, where 1 is the best; P is the population size; and $q' = q/(1-(1-q)^P)$.

3. Genetic operators:

   The operators are used to create new children based on the current generation in the population. Basically, there are two types of operators: crossover and mutation. Crossover takes two individuals and produces two new individuals while mutation alters one individual to produce a single new solution.

   In binary representations, the applications of these two types of operators are only binary mutation and simple crossover.

   In real-valued representations, the applications of these two types of operators have been developed by Michalewicz [107], they are uniform mutation, nonuniform mutation, multi-nonuniform mutation, boundary mutation, simple crossover, arithmetic crossover, and heuristic crossover [107].

   Uniform mutation randomly selects one variable and sets it equal to a uniform random number while boundary mutation randomly selects one variable and sets it equal to either its lower or upper bound.

   Nonuniform mutation randomly selects one variable and sets it equal to a nonuniform random number; according to [107], it is defined as follows: If ${s_x}^t = (x_1, x_2, x_3, \ldots, x_m)$ is a chromosome (t is the generation number) and the element x_j was selected for nonuniform mutation, the result is a vector ${s_x}^{t+1} = (x_1, x_2, x_3, \ldots, x'_j, \ldots, x_m)$, where

$$x'_j = x_j + \Delta(t, \mathrm{UB} - x_j), \quad \text{if a random digit is 0,} \tag{9.2}$$
$$x'_j = x_j - \Delta(t, x_j - \mathrm{LB}), \quad \text{if a random digit is 1,} \tag{9.3}$$

where UB and LB are the upper and lower bounds for the variable and $\Delta(t, y)$ is given by

$$\Delta(t, y) = y(1 - r^{\left(1-\frac{t}{G}\right)^b}), \tag{9.4}$$

where r is a uniform random number between (0,1), G represents the maximum number of generations, t is the current generation, and b is a parameter determining the degree of dependency on the generation number.

4. Population size:
   The population size represents the number of individuals or chromosomes in the population. Usually, larger population sizes increase the amount of variation present in the initial population and it requires more fitness evaluations. If the population loses diversity, the population is said to have premature convergence and little exploration is being done. For longer chromosomes and challenging optimization problems, larger population sizes are needed to maintain diversity – higher diversity can also be achieved through higher mutation rates and uniform crossover – and hence better exploration. Usually, the population size is determined by the rule of thumb of 7–8 times the number of the optimization parameters.
5. Mutation rate:
   The mutation rate is defined as the percentage of the total number of genes in the population [51]; it determines the probability that a mutation will occur. Mutation is employed to give new information to the population and also prevents the population from becoming saturated with similar chromosomes (premature convergence). Large mutation rates increase the probability that good schemata will be destroyed, but increase population diversity. The best mutation rate is application dependent but for most applications it is between 0.001 and 0.1.
6. Crossover rate:
   The crossover rate (denoted by p_c) is defined as the ratio of the number of offspring produced in each generation to the population size, P [51]. This ratio controls the expected number $p_c \times P$ of chromosomes to undergo the crossover operation. The best crossover rate is application dependent but for most applications it is between 0.80 and 0.95.
7. Evaluation functions:
   Evaluation functions are subject to the minimal requirement that the function can map the population into a partially ordered set. In the present work, the sum of diagonal elements in stiffness/compliance matrix with relative weight factors for each direction is set as the evaluation function.

9.4 Single-Objective Optimization

9.4.1 *Objective of Global Stiffness*

In this research, the stiffness for certain mechanism configurations is expressed by a (6×6) matrix, as discussed before. The diagonal elements of the matrix are the mechanism's pure stiffness in each Cartesian direction. To obtain the optimal stiffness in each direction, one can write an objective function, (9.5), with stiffness element to maximize or write an objective function, (9.6), with compliance elements whose negative is to be maximized, i.e., maximize(val) where

$$\text{val} = \eta_1 K_{11} + \eta_2 K_{22} + \eta_3 K_{33} + \eta_4 K_{44} + \eta_5 K_{55} + \eta_6 K_{66} \tag{9.5}$$

or

$$\text{val} = -(\lambda_1 \kappa_{11} + \lambda_2 \kappa_{22} + \lambda_3 \kappa_{33} + \lambda_4 \kappa_{44} + \lambda_5 \kappa_{55} + \lambda_6 \kappa_{66}), \tag{9.6}$$

where, for $i = 1, \ldots, 6$, K_{ii} represents the diagonal elements of the mechanism's stiffness matrix, κ_{ii} represents the diagonal elements of mechanism's compliance matrix, η_i is the weight factor for each directional stiffness, which characterizes the priority of the stiffness in this direction, and λ_i is the weight factor for each directional compliance, which characterizes the priority of the compliance in this direction.

This would maximize/minimize the SUM of the diagonal elements. Although we could not maximize/minimize each diagonal element individually, we always can optimize each stiffness by distributing the weighting factors. Once the objective function is written, a search domain for each optimization variable (lengths, angles, etc.) should be specified to create an initial population. The limits of the search domain are set by a specified maximum number of generations or population convergence criteria, since the GAs will force much of the entire population to converge to a single solution.

For the optimization of the stiffness, a real-valued method is used combined with the selection, mutation, and crossover operators with their optional parameters used for all these types of parallel mechanism stiffness/compliance function optimization as shown in Table 9.1. The first optional parameter is the number of times to apply the operators for real-valued representation, G_{m} represents the maximum number of generations, and b is a parameter determining the degree of dependency on the generation number, we use 3 in our case [107]. The other optional parameters depend on the operators we are using. Since Matlab requires matrices to have the same length in all rows, many of the parameters are 0 indicating that they are really place holders only. In the following sections, we will describe it in more detail.

Table 9.1 Genetic algorithm parameters used for real-valued stiffness function optimization

Name	Parameters
Uniform mutation	[4 0 0]
Nonuniform mutation	[4 G_m b]
Multi-nonuniform mutation	[6 G_m b]
Boundary mutation	[4 0 0]
Simple crossover	[2 0]
Arithmetic crossover	[2 0]
Heuristic crossover	[2 3]
Normalized geometric selection	0.08

9.4.2 *Spatial Six-Degree-of-Freedom Mechanism with Prismatic Actuators*

The spatial 6-dof mechanism with prismatic actuators is shown in Figs. 5.2 and 5.3. In order to obtain the maximum global stiffness, five architecture and behavior parameters are used as optimization parameters, the vector of optimization variables is

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z, T_\mathrm{p}, T_\mathrm{b}], \tag{9.7}$$

where R_p is the radius of the platform, R_b is the radius of the base, z is the height of the platform, T_p and T_b are the angles to determine the attachment points on the platform and on the base, and their bounds are

$$\begin{gathered} R_\mathrm{p} \in [5, 10]\,\mathrm{cm},\ R_\mathrm{b} \in [12, 22]\,\mathrm{cm}, \\ z \in [45, 56]\,\mathrm{cm}, \\ T_\mathrm{p} \in [18, 26]^\circ,\ T_\mathrm{b} \in [38, 48]^\circ. \end{gathered}$$

In this research work, the objective function of (9.5) is maximized where the following is assumed

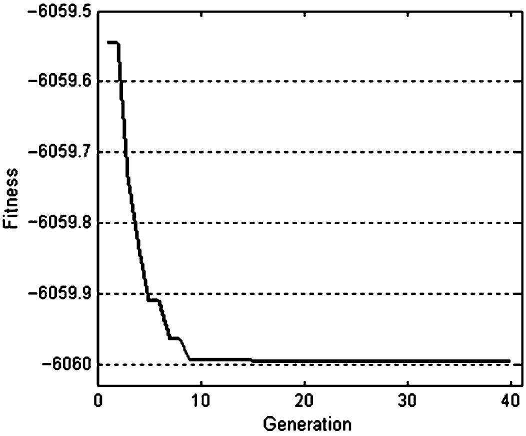
$$\begin{aligned} \eta_i &= 1, \quad i = 1, \ldots, 6, \\ P &= 80, \\ G_\mathrm{max} &= 100, \end{aligned}$$

where P is the population size and G_max is the maximum number of generations.

The genetic algorithm is implemented in Matlab to search for the best solutions. The results are given only for one case with $\phi = 0, \theta = 0$, and $\psi = 0$. Figures 9.2 and 9.3 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. The architectural and behavior parameters found by the GA after 40 generations are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z, T_\mathrm{p}, T_\mathrm{b}] = [10, 12, 56, 18, 48]$$

Fig. 9.2 The evolution of the performance of the 6-dof mechanism with prismatic actuators



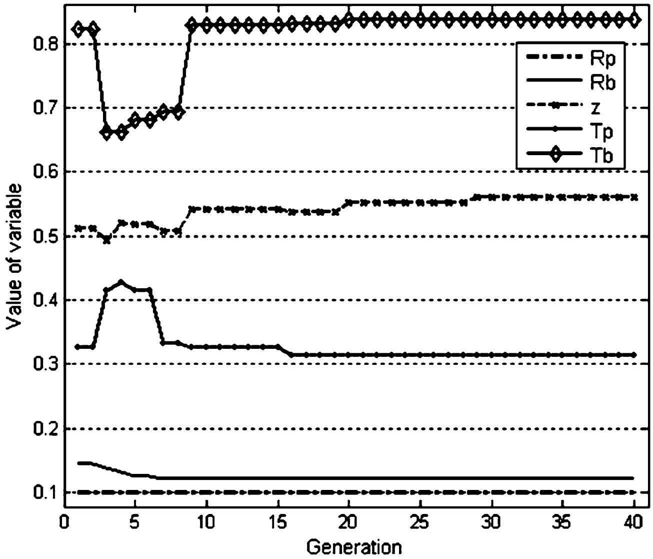


Fig. 9.3 The evolution of the geometrical parameters of the 6-dof mechanism with prismatic actuators

and the stiffness in each direction is

$$\begin{aligned}\mathbf{K} &= [K_x, K_y, K_z, K_{\theta_x}, K_{\theta_y}, K_{\theta_z}]\\ &= [34.1918, 34.1918, 5931.6164, 29.65808182, 29.65808182, 0.68092535].\end{aligned}$$

The sum of the stiffness is 6059.997.

Before optimization, the parameters for this mechanism were given as

$$\mathbf{s}' = [R_\mathrm{p}, R_\mathrm{b}, z, T_\mathrm{p}, T_\mathrm{b}] = [6, 15, 51, 22.34, 42.88]$$

and the stiffness in each direction was

$$\begin{aligned}\mathbf{K}' &= [K'_x, K'_y, K'_z, K'_{\theta_x}, K'_{\theta_y}, K'_{\theta_z}] \\ &= [102.968, 102.968, 5794.06, 10.4293, 10.4293, 0.222188]\end{aligned}$$

and the stiffness sum is 6021.08. Hence, after optimization, the stiffness sum is improved 1.01 times.

9.4.3 *Spatial Six-Degree-of-Freedom Mechanism with Revolute Actuators*

A spatial 6-dof mechanism with revolute actuators is represented in Fig. 6.2. The vertex distribution is the same as in Fig. 5.3. From Fig. 6.3, it is clear that the Cartesian stiffness is a monotonically increasing function of the link stiffness (for all the case with revolute actuators). Nevertheless, there exists a critical link stiffness, which has tiny effects on mechanism's Cartesian stiffness when it is larger than the critical link stiffness, therefore, for all mechanisms with revolute actuators, link stiffness are also included as optimization parameters. Seven optimization parameters are specified in this mechanism for maximizing the mechanism's global stiffness. The vector of optimization variables can be expressed as

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z, T_\mathrm{p}, T_\mathrm{b}, l_1, l_2], \tag{9.8}$$

where R_p is the radius of the platform, R_b is the radius of the base, z is the height of the platform, T_p and T_b are the angles to determine the attachment points on the platform and on the base, l_1 and l_2 are the link lengths, and the bound for each parameter is

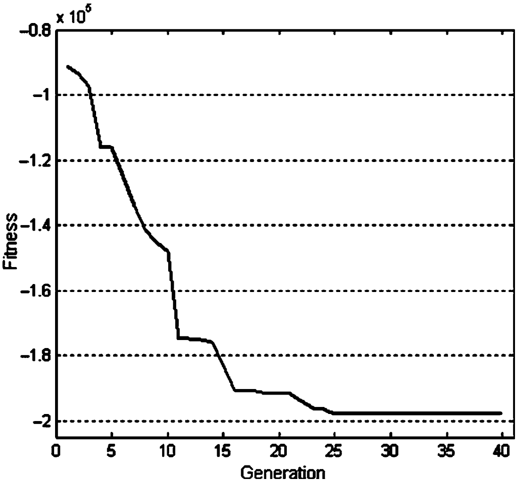
$$\begin{gathered} R_\mathrm{p} \in [5, 7]\,\mathrm{cm},\ R_\mathrm{b} \in [14, 16]\,\mathrm{cm}, \\ z \in [66, 70]\,\mathrm{cm}, \\ T_\mathrm{p} \in [20, 26]^\circ,\ T_\mathrm{b} \in [40, 45]^\circ, \\ l_1 \in [42, 48]\,\mathrm{cm},\ l_2 \in [32, 40]\,\mathrm{cm}. \end{gathered}$$

In this case, the objective function of (9.5) is maximized assuming

$$\begin{aligned} \eta_i &= 1, \quad i = 1, \ldots, 6, \\ P &= 80, \\ G_\mathrm{max} &= 40, \end{aligned}$$

where P is the population size and G_max is the maximum number of generations.

Fig. 9.4 The evolution of the performance of the 6-dof mechanism with revolute actuators



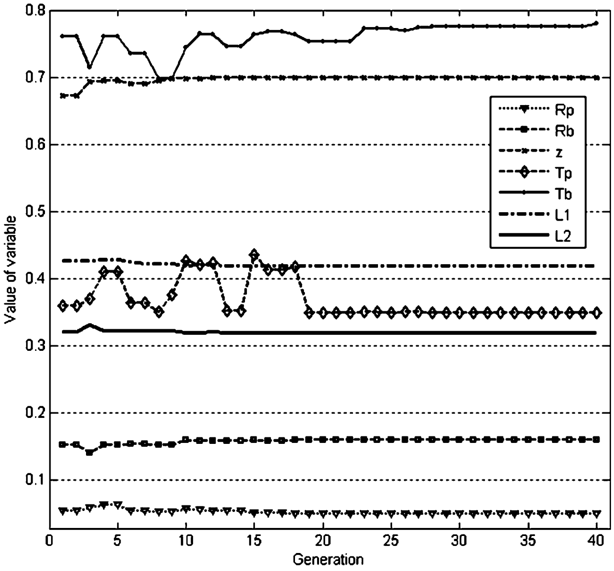


Fig. 9.5 The evolution of the geometrical parameters of the 6-dof mechanism with revolute actuators

A program based on the genetic algorithm is applied to search for the best solutions. The results are given only for one configuration with $\theta = 0, \phi = 0$, and $\psi = 0$. Figures 9.4 and 9.5 show that, after 40 generations, the track of the best individual and the optimal parameters converge to the final best solution. The optimal geometric and behavior parameters obtained by the GA after 40 generations are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z, T_\mathrm{p}, T_\mathrm{b}, l_1, l_2] = [5, 16, 70, 20, 45, 42, 32]$$

and the stiffness in each direction is

$$\begin{aligned}\mathbf{K} &= [K_x, K_y, K_z, K_{\theta_x}, K_{\theta_y}, K_{\theta_z}]\\ &= [18873.14, 18873.14, 159835.85, 199.79, 199.79, 41.04].\end{aligned}$$

The sum of the stiffness is 198022.766.

Initially, the geometric and behavior values were given for this mechanism as

$$\begin{aligned}\mathbf{s}' &= [R_\mathrm{p}, R_\mathrm{b}, z, T_\mathrm{p}, T_\mathrm{b}, l_1, l_2]\\ &= [6, 15, 68, 22.34, 42.883, 46, 36]\end{aligned}$$

and the stiffness in each direction was

$$\begin{aligned}\mathbf{K}' &= [K_x', K_y', K_z', K_{\theta_x}', K_{\theta_y}', K_{\theta_z}']\\ &= [7725, 7725, 21045, 37.8818, 37.8818, 43.0695].\end{aligned}$$

The stiffness sum is 36613.8. Therefore, after optimization, the stiffness sum is improved 5.4 times.

9.4.4 Spatial Five-Degree-of-Freedom Mechanism with Prismatic Actuators

The spatial 5-dof mechanism with prismatic actuators is shown in Figs. 5.6 and 5.7. In order to obtain the maximum global stiffness, the global compliance (since there are infinite terms among the diagonal stiffness elements) is minimized. However, it is clear that the Cartesian stiffness is a monotonically increasing function of the link and actuator stiffness (for all the case with prismatic actuators). Hence, the optimum solution always corresponds to the maximum link or actuator stiffness and these parameters are not included in the optimization variables. Seven parameters are specified as optimization parameters, they are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, l_{61}, l_{62}, z, T_\mathrm{p}, T_\mathrm{b}], \tag{9.9}$$

where R_p is the radius of the platform; R_b is the radius of the base; l_{61} and l_{62} are the link length for the first and second link of the passive leg, respectively; z is the height of the platform; T_p and T_b are the angles to determine the attachment points on the platform and on the base; and their bounds are

$$\begin{gathered}R_\mathrm{p} \in [10, 14]\,\mathrm{cm}, R_\mathrm{b} \in [20, 26]\,\mathrm{cm},\\ l_{61} \in [52, 70]\,\mathrm{cm}, l_{62} \in [52, 70]\,\mathrm{cm},\\ z \in [66, 70]\,\mathrm{cm},\\ T_\mathrm{p} \in [18, 26]^\circ, T_\mathrm{b} \in [38, 48]^\circ.\end{gathered}$$

In this work, the objective function of (9.6) is minimized assuming

$$\begin{aligned} \lambda_i &= 1, \quad i = 1, \ldots, 6, \\ P &= 80, \\ G_{\max} &= 40. \end{aligned}$$

Results are given here only for the case with $\theta_{65} = -\pi$ and $\theta_{66} = 2\pi/3$. Figures 9.6 and 9.7 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. The architectural and behavior parameters found by the GA after 40 generations are

$$\begin{aligned} \mathbf{s} &= [R_\mathrm{p}, R_\mathrm{b}, l_{61}, l_{62}, z, T_\mathrm{p}, T_\mathrm{b}] \\ &= [14, 21.2, 52, 70, 66, 18, 48] \end{aligned}$$

and the compliances in each direction are

$$\begin{aligned} \boldsymbol{\kappa} &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z] \\ &= [0.03687, 0.03113, 0.03646, 0.03962, 0.01657, 2.46 \times 10^{-4}]. \end{aligned}$$

The sum of the compliances is 0.16.

Before optimization, the parameter values of the mechanism were given as

$$\begin{aligned} \mathbf{s}' &= [R_\mathrm{p}, R_\mathrm{b}, l_{61}, l_{62}, z, T_\mathrm{p}, T_\mathrm{b}] \\ &= [12, 22, 68, 68, 68, 22.34, 42.883] \end{aligned}$$

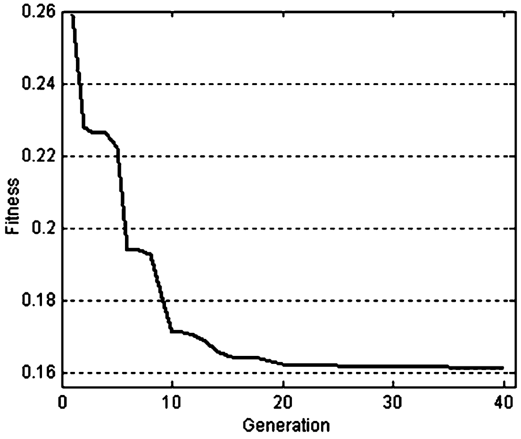


Fig. 9.6 The evolution of the performance of the 5-dof mechanism with prismatic actuators

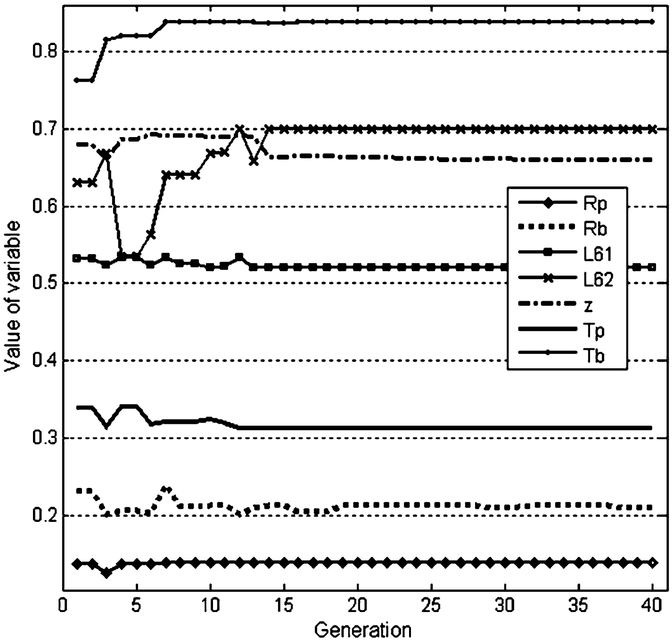


Fig. 9.7 The evolution of the geometrical parameters of the 5-dof mechanism with prismatic actuators

and the compliances in each direction were

$$\begin{aligned}\kappa' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z]\\ &= [0.08627, 0.0981, 0.2588, 0.07342, 0.030325, 2.55 \times 10^{-4}].\end{aligned}$$

The compliance sum is 0.54714. After optimization, the compliance sum is improved 3.4 times.

9.4.5 *Spatial Five-Degree-of-Freedom Mechanism with Revolute Actuators*

The schematic representation of this type mechanism and its vertex distribution is shown in Figs. 6.5 and 6.6, respectively. Twelve architecture and behavior parameters are specified as optimization parameters to minimize the compliances. They can be represented as a vector of **s**

$$\mathbf{s} = [R_p, R_b, l_{61}, l_{62}, l_1, l_2, z, T_p, T_b], \tag{9.10}$$

where R_{p} is the radius of the platform; R_{b} is the radius of the base; l_{61} and l_{62} are the link lengths for the first and second link of the passive leg, respectively; l_1 and l_2 are the link lengths for the first and second link of the each actuated leg, respectively; z is the height of the platform; T_{p} and T_{b} are the angles to determine the attachment points on the platform and on the base; and the bound of each optimization parameter is

$$\begin{gathered}R_{\mathrm{p}} \in [5,7]\,\mathrm{cm},\ R_{\mathrm{b}} \in [14,18]\,\mathrm{cm},\\ l_{61} \in [67,70]\,\mathrm{cm},\ l_{62} \in [67,70]\,\mathrm{cm},\\ l_1 \in [33,35]\,\mathrm{cm},\ l_2 \in [45,47]\,\mathrm{cm},\\ z \in [66,70]\,\mathrm{cm},\ T_{\mathrm{p}} \in [18,30]^\circ,\ T_{\mathrm{b}} \in [38,50]^\circ.\end{gathered}$$

For this mechanism, the objective function of (9.6) is minimized with

$$\begin{aligned}\lambda_i &= 1, \quad i = 1,\dots,6,\\ P &= 80,\\ G_{\max} &= 40.\end{aligned}$$

Results are given here for the case with $\theta_{65} = -\pi$ and $\theta_{66} = 2\pi/3$. Figures 9.8 and 9.9 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. The mechanism's geometric and behavior parameters found by the GA after 40 generations are

$$\begin{aligned}\mathbf{s} &= [R_{\mathrm{p}}, R_{\mathrm{b}}, l_{61}, l_{62}, l_1, l_2, z, T_{\mathrm{p}}, T_{\mathrm{b}}]\\ &= [7, 18, 69.052, 67.179, 33, 45, 70, 18.92, 50.03]\end{aligned}$$

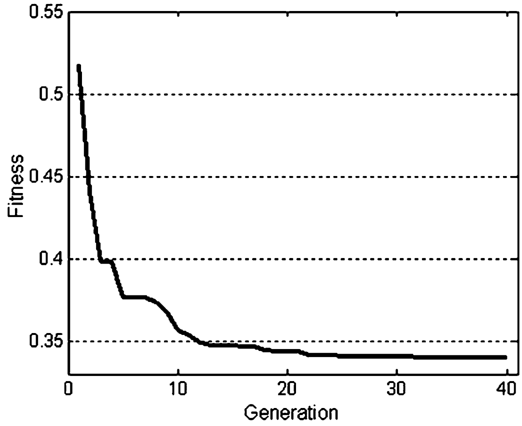


Fig. 9.8 The evolution of the performance of the 5-dof mechanism with revolute actuators

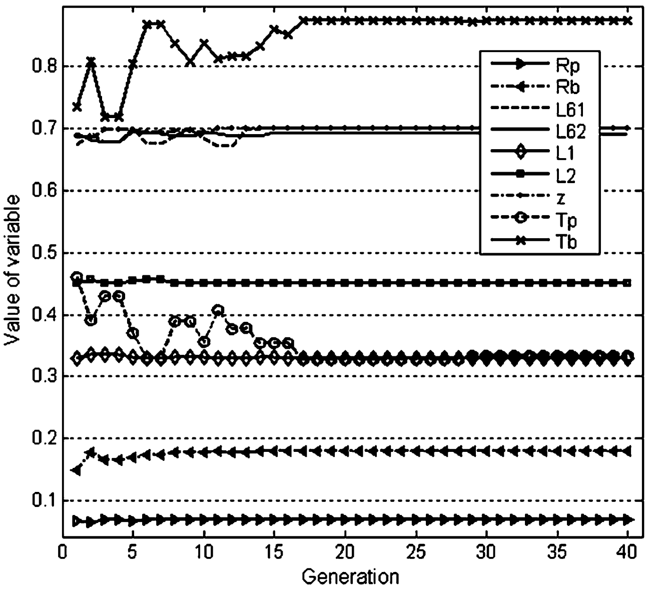


Fig. 9.9 The evolution of the geometrical parameters of the 5-dof mechanism with revolute actuators

and the compliances in each direction are

$$\begin{aligned}\kappa &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z] \\ &= [7.77 \times 10^{-2}, 0.10345, 0.24256, 1.116 \times 10^{-3}, 1.87 \times 10^{-3}, 2.67 \times 10^{-4}].\end{aligned}$$

The sum of the compliances is 0.33977.

The initial guess of the geometric and structure behavior parameters of the mechanism was given as

$$\begin{aligned}\mathrm{s}' &= [R_\mathrm{p}, R_\mathrm{b}, l_{61}, l_{62}, l_1, l_2, z, T_\mathrm{p}, T_\mathrm{b}] \\ &= [6, 15, 68, 68, 34, 46, 68, 22.34, 42.883]\end{aligned}$$

and the compliances in each direction were

$$\begin{aligned}\kappa' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z] \\ &= [0.1244, 0.2327, 0.3732, 0.001, 0.002464, 3.589 \times 10^{-4}].\end{aligned}$$

The compliance sum is 0.734195. Hence after optimization, the compliance sum is improved 2.16 times.

9.4.6 Spatial Four-Degree-of-Freedom Mechanism with Prismatic Actuators

Figure 5.8 shows the spatial 4-dof mechanism with prismatic actuators. For this mechanism, the optimization parameters are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, l_{51}, l_{52}, z, T_\mathrm{p}, T_\mathrm{b}], \tag{9.11}$$

where R_p is the radius of the platform; R_b is the radius of the base; l_{51} and l_{52} are the link lengths for the 1st and 2nd link of the passive leg, respectively; z is the height of the platform; T_p and T_b are the angles to determine the attachment points on the platform and on the base, and their bounds are

$$R_\mathrm{p} \in [10, 14]\,\mathrm{cm},\, R_\mathrm{b} \in [20, 26]\,\mathrm{cm},$$
$$l_{51} \in [52, 70]\,\mathrm{cm},\, l_{52} \in [52, 70]\,\mathrm{cm},$$
$$z \in [66, 70]\,\mathrm{cm},\, T_\mathrm{p} \in [25, 35]^\circ,\, T_b \in [55, 65]^\circ,$$

Again, the compliances are minimized as above.

Results are given here only for one case with $\theta_{55} = -\pi/3$ and $\theta_{56} = 2\pi/3$. Figures 9.10 and 9.11 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. The geometric and behavior parameters found by the GA after 40 generations are

$$\begin{aligned}\mathbf{s} &= [R_\mathrm{p}, R_\mathrm{b}, l_{51}, l_{52}, z, T_\mathrm{p}, T_\mathrm{b}]\\ &= [14, 26, 70, 55, 66, 35, 55]\end{aligned}$$

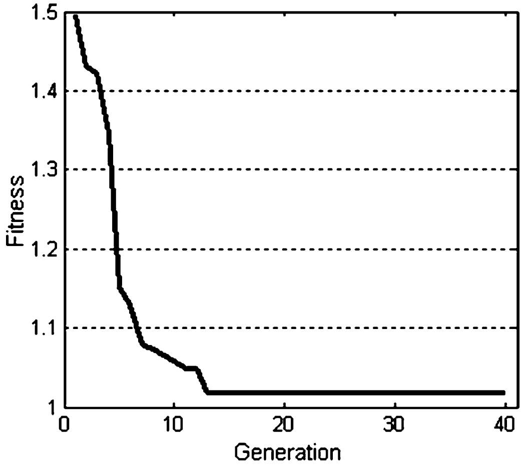


Fig. 9.10 The evolution of the performance of the 4-dof mechanism with prismatic actuators

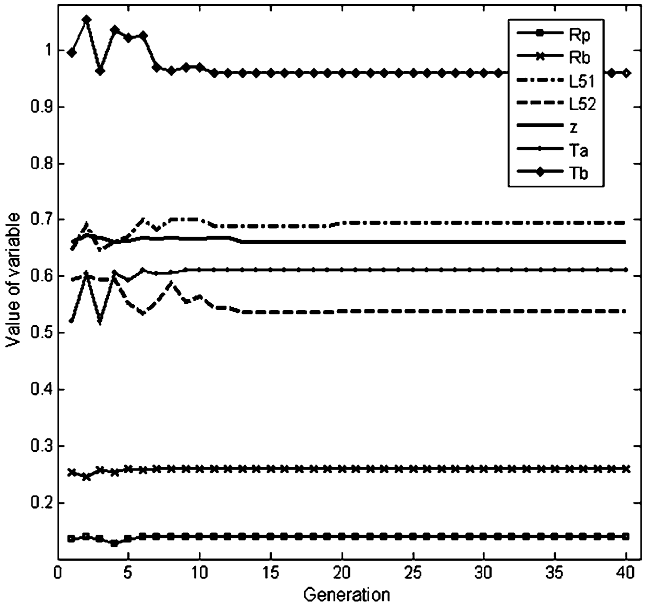


Fig. 9.11 The evolution of the geometrical parameters of the 4-dof mechanism with prismatic actuators

and the compliances in each direction are

$$\begin{aligned}\kappa &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z] \\ &= [0.12, 0.5742, 3.747 \times 10^{-3}, 0.3165, 5.006 \times 10^{-11}, 3.345 \times 10^{-3}].\end{aligned}$$

The sum of the compliances is 1.017897.

Initially, the parameters for this mechanism were given as

$$\begin{aligned}\mathrm{s}' &= [R_\mathrm{p}, R_\mathrm{b}, l_{51}, l_{52}, z, T_\mathrm{p}, T_\mathrm{b}] \\ &= [12, 22, 68, 68, 68, 30, 60]\end{aligned}$$

and the compliances in each direction were

$$\begin{aligned}\kappa' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z] \\ &= [0.5164, 1.4046, 1.5 \times 10^{-10}, 0.9087, 5.78 \times 10^{-11}, 0.011139].\end{aligned}$$

The compliance sum is 2.84085. Therefore, after optimization the compliance sum has been improved 2.8 times.

9.4.7 Spatial Four-Degree-of-Freedom Mechanism with Revolute Actuators

A spatial 4-dof mechanism with revolute actuators is shown in Fig. 6.9. The parameters are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, l_{51}, l_{52}, L_1, L_2, z, T_\mathrm{p}, T_\mathrm{b}], \tag{9.12}$$

where R_p is the radius of the platform; R_b is the radius of the base; l_{51} and l_{52} are the link lengths for the first and second link of the passive leg, respectively; L_1 and L_2 are the link lengths for the first and second link of the actuated leg, respectively; z is the height of the platform; T_p and T_b are the angles to determine the attachment points on the platform and on the base; and their bounds are

$$\begin{gathered} R_\mathrm{p} \in [5,7]\,\mathrm{cm}, R_\mathrm{b} \in [14,16]\,\mathrm{cm}, \\ l_{51} \in [67,69]\,\mathrm{cm}, l_{52} \in [67,69]\,\mathrm{cm}, \\ L_1 \in [33,35]\,\mathrm{cm}, L_2 \in [45,47]\,\mathrm{cm}, \\ z \in [66,70]\,\mathrm{cm}, \\ T_\mathrm{p} \in [25,35]^\circ, T_\mathrm{b} \in [55,65]^\circ, \end{gathered}$$

Results are given here for one case with $\theta_{55} = -\pi/3$ and $\theta_{56} = 2\pi/3$. Figures 9.12 and 9.13 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. The optimum geometric and behavior parameters for this configuration are

$$\begin{aligned} \mathbf{s} &= [R_\mathrm{p}, R_\mathrm{b}, l_{51}, l_{52}, L_1, L_2, z, T_\mathrm{p}, T_\mathrm{b}] \\ &= [7, 16, 67, 69, 35, 47, 66, 35, 55] \end{aligned}$$

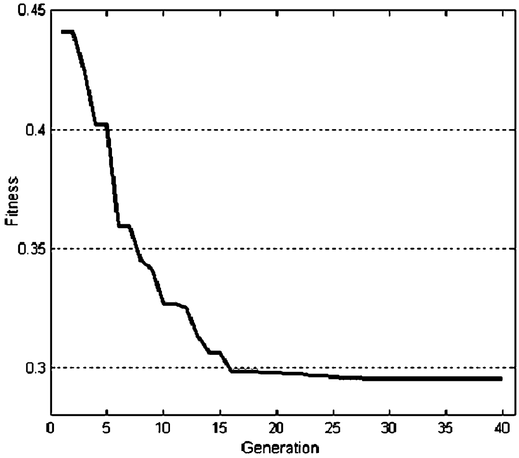


Fig. 9.12 The evolution of the performance of the 4-dof mechanism with revolute actuators

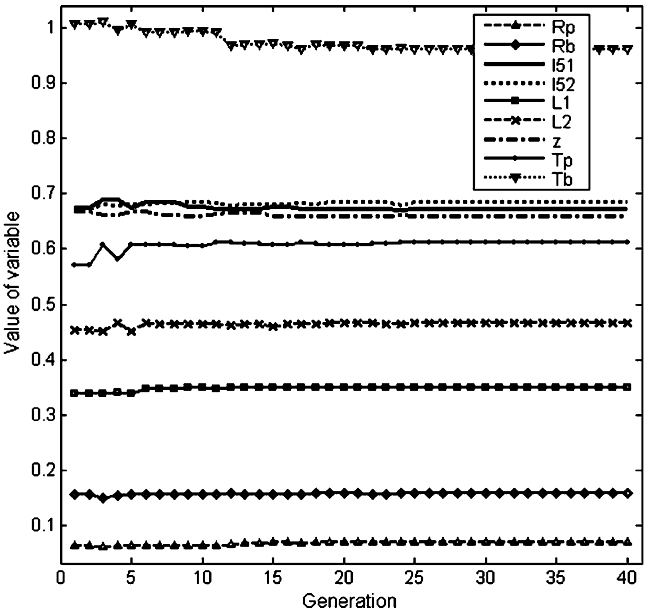


Fig. 9.13 The evolution of the geometrical parameters of the 4-dof mechanism with revolute actuators

and the compliances in each direction are

$$\begin{aligned}\kappa &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z] \\ &= [0.28857, 0.019376, 8.66 \times 10^{-5}, 9.39 \times 10^{-4}, 5.751 \times 10^{-11}, 3.646 \times 10^{-5}].\end{aligned}$$

The sum of the compliances is 0.29518.

The initial guess for this mechanism was

$$\begin{aligned}\mathbf{s}' &= [R_\mathrm{p}, R_\mathrm{b}, l_{51}, l_{52}, L_1, L_2, z, T_\mathrm{p}, T_\mathrm{b}] \\ &= [6, 15, 68, 68, 34, 46, 68, 30, 60]\end{aligned}$$

and the compliances in each direction were

$$\begin{aligned}\kappa' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z] \\ &= [1.2807, 0.0628078, 0, 0.00276278, 0, 0.00003838].\end{aligned}$$

The compliance sum is 1.3463. Hence, after optimization, the total compliance is improved 4.56 times.

9.4.8 Spatial Three-Degree-of-Freedom Mechanism with Prismatic Actuators

The spatial 3-dof mechanism with prismatic actuators is shown in Fig. 5.9. The parameters are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z], \tag{9.13}$$

where R_p is the radius of the platform, R_b is the radius of the base, z is the height of the platform, and their bounds are set as

$$R_\mathrm{p} \in [5, 10]\,\mathrm{cm},\ R_\mathrm{b} \in [12, 14]\,\mathrm{cm},$$
$$z \in [66, 70]\,\mathrm{cm}.$$

Here only the case with $\theta_{45} = \pi/2$ and $\theta_{46} = 0$ is discussed. Figures 9.14 and 9.15 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. After 40 generations, the optimal geometric and behavior parameters found by the GA are

$$\begin{aligned}\mathbf{s} &= [R_\mathrm{p}, R_\mathrm{b}, z]\\ &= [10, 12, 70]\end{aligned}$$

and the compliances in each direction are

$$\begin{aligned}\kappa &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z]\\ &= [6.8355 \times 10^{-2}, 6.8355 \times 10^{-2}, 0, 0, 0, 3.4177 \times 10^{-4}].\end{aligned}$$

The sum of the compliances is 0.137.

The initial geometric and behavior values for this mechanism were given as

$$\begin{aligned}\mathbf{s}' &= [R_\mathrm{p}, R_\mathrm{b}, z]\\ &= [6, 15, 68]\end{aligned}$$

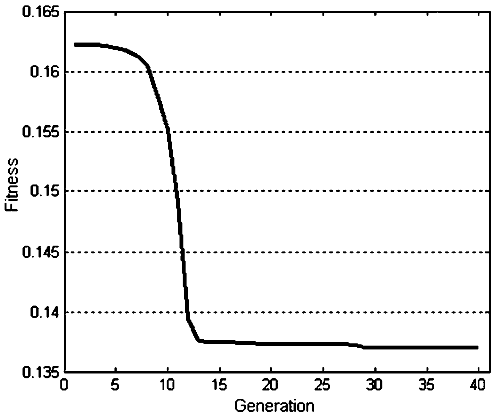


Fig. 9.14 The evolution of the performance of the 3-dof mechanism with prismatic actuators

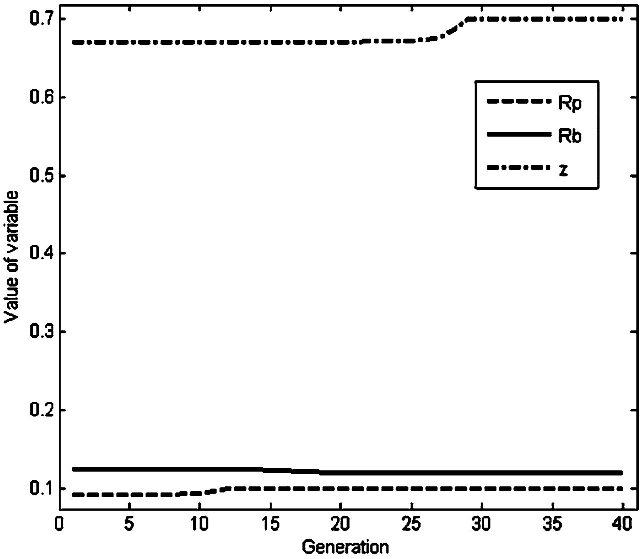


Fig. 9.15 The evolution of the geometrical parameters of the 3-dof mechanism with prismatic actuators

and the compliances in each direction were

$$\begin{aligned}\kappa' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z] \\ &= [0.192, 0.192, 0, 0, 0, 3.4566 \times 10^{-4}].\end{aligned}$$

The compliance sum is 0.3844. Therefore, after optimization, the total compliance is improved 2.81 times which is a minor gain.

9.4.9 Spatial Three-Degree-of-Freedom Mechanism with Revolute Actuators

The spatial 3-dof mechanism with revolute actuators is shown in Fig. 6.10. The parameters are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, l_1, l_2, z], \tag{9.14}$$

where R_p is the radius of the platform, R_b is the radius of the base, l_1 and l_2 are the link length, z is the height of the platform, and their bounds are

$$\begin{gathered} R_\mathrm{p} \in [5, 7]\,\mathrm{cm},\ R_\mathrm{b} \in [14, 16]\,\mathrm{cm}, \\ l_1 \in [33, 35]\,\mathrm{cm},\ l_2 \in [45, 47]\,\mathrm{cm}, \\ z \in [66, 70]\,\mathrm{cm}. \end{gathered}$$

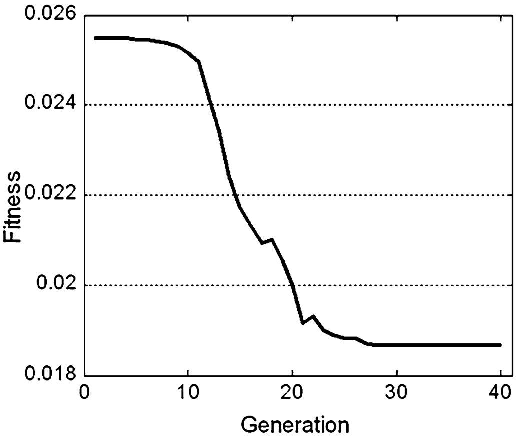


Fig. 9.16 The evolution of the performance of the 3-dof mechanism with revolute actuators

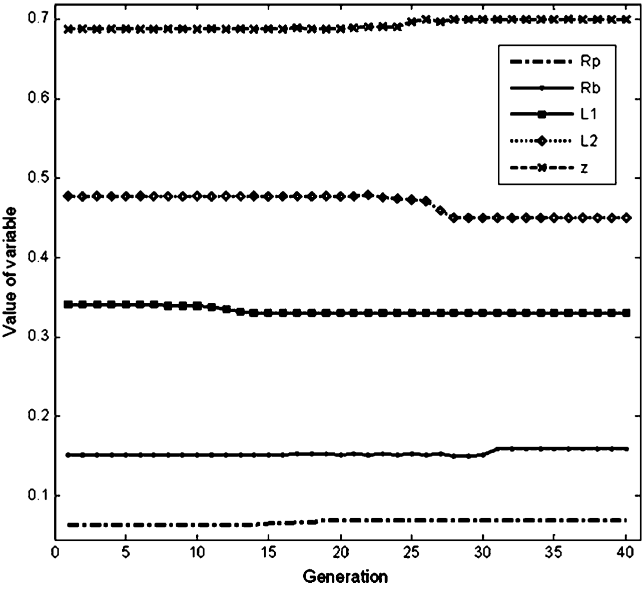


Fig. 9.17 The evolution of the geometrical parameters of the 3-dof mechanism with revolute actuators

Here only one case with $\theta_{45} = \pi/2$ and $\theta_{46} = 0$ is analyzed. Figures 9.16 and 9.17 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. After running the program for 40 generations, the optimal architectural and behavior parameters can be found as

$$\begin{aligned}\mathbf{s} &= [R_\mathrm{p}, R_\mathrm{b}, l_1, l_2, z] \\ &= [7, 16, 33, 45, 70]\end{aligned}$$

and the compliances in each direction are

$$\begin{aligned}\boldsymbol{\kappa} &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z] \\ &= [1.0782 \times 10^{-2}, 1.0782 \times 10^{-2}, 0, 0, 0, 2.64 \times 10^{-5}].\end{aligned}$$

The sum of the compliances is 0.018659.

Before optimization, a series of parameters were guessed as

$$\begin{aligned}\mathbf{s}' &= [R_\mathrm{p}, R_\mathrm{b}, l_1, l_2, z] \\ &= [6, 15, 34, 46, 68]\end{aligned}$$

and the compliance in each direction can be computed as

$$\begin{aligned}\boldsymbol{\kappa}' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z] \\ &= [2.12264 \times 10^{-2}, 2.12264 \times 10^{-2}, 0, 0, 0, 3.82 \times 10^{-5}].\end{aligned}$$

The compliance sum is 0.04249. Hence, after optimization, the total compliances is improved 2.28 times.

9.4.10 The Tricept Machine Tool Family

The schematic representation of the Tricept machine tool and the geometry of the joint distribution both on the base and the platform are shown in Figs. 9.18 and 9.19, respectively. The vector of optimization variables is therefore

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z], \tag{9.15}$$

where R_p is the radius of the platform, R_b is the radius of the base, z is the height of the platform, and their bounds are specified based on the dimensions of the Tricept machine tool

$$\begin{gathered}R_\mathrm{p} \in [200, 300]\,\mathrm{mm}, \;\; R_\mathrm{b} \in [400, 600]\,\mathrm{mm}, \\ z \in [900, 1500]\,\mathrm{mm}.\end{gathered}$$

The case with $\theta_{41} = \pi/2$ and $\theta_{42} = 0$ is discussed here. Figures 9.20 and 9.21 show the evolution of the best individual and the optimal parameters for 40 generations, respectively. The optimal architectural and behavior parameters found by the GA after 40 generations are

$$\mathbf{s} = [R_\mathrm{p}, R_\mathrm{b}, z] = [300, 600, 900]$$

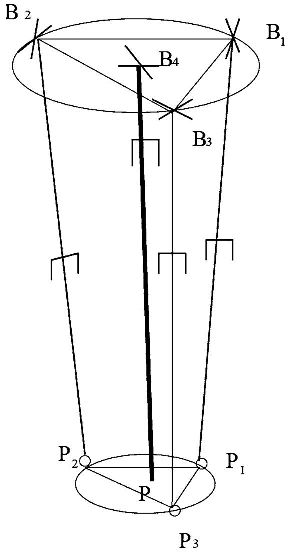


Fig. 9.18 Schematic representation of the Tricept machine tool

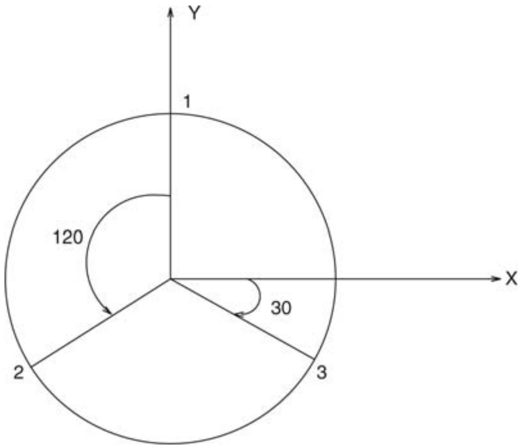


Fig. 9.19 Position of the attachment points on the base and platform

and the compliances in each direction are

$$\begin{aligned}\kappa &= [\kappa_{\theta_x}, \kappa_{\theta_y}, \kappa_{\theta_z}, \kappa_x, \kappa_y, \kappa_z] \\ &= [2.0576 \times 10^{-3}, 2.0576 \times 10^{-3}, 0, 1.667 \times 10^{-3}, 1.667 \times 10^{-3}, 3.703 \times 10^{-4}].\end{aligned}$$

The sum of the compliances is 0.0078189. Before optimization, the dimensions of the Tricept machine tool provided by Neos Robotics AB were

$$\mathbf{s}' = [R_\mathrm{p}, R_\mathrm{b}, z] = [225, 500, 1300]$$

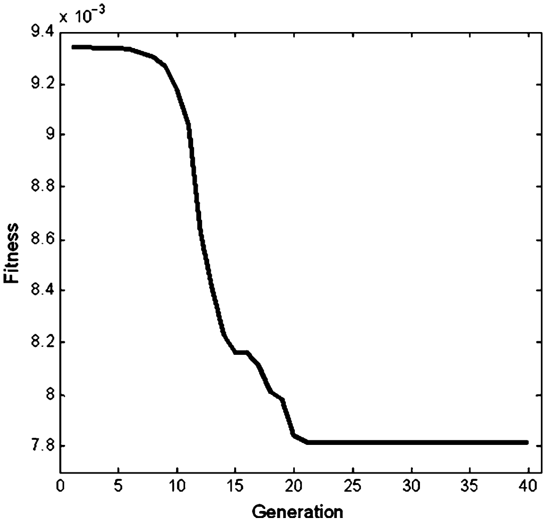


Fig. 9.20 The evolution of the performance of the Tricept machine tool

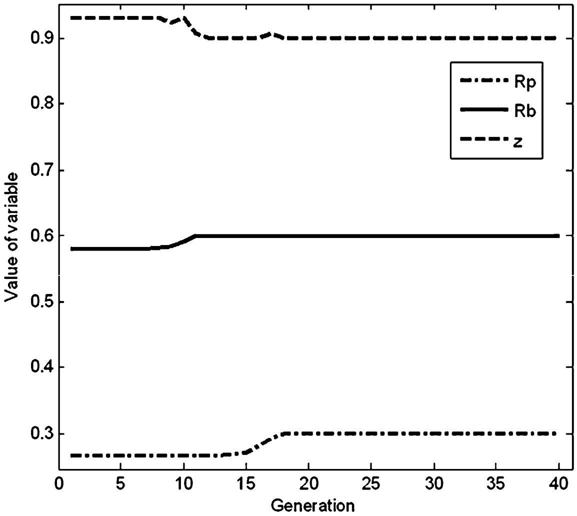


Fig. 9.21 The evolution of the geometrical parameters of the Tricept machine tool

and the compliances in each direction were

$$\begin{aligned}\kappa' &= [\kappa'_{\theta_x}, \kappa'_{\theta_y}, \kappa'_{\theta_z}, \kappa'_x, \kappa'_y, \kappa'_z] \\ &= [2.786 \times 10^{-3}, 2.786 \times 10^{-3}, 0, 4.708 \times 10^{-3}, 4.708 \times 10^{-3}, 3.4825 \times 10^{-4}].\end{aligned}$$

The sum of the compliances is 0.0153369. Hence, after optimization, the sum of the compliances is improved by a factor of 1.96 just by slightly enlarging the radius of the base and the platform.

9.5 Multiobjective Optimization

9.5.1 Case Study 1: Three Degrees of Freedom Parallel Manipulator – Two Translations and One Rotation

9.5.1.1 Structure Description

The new 3-dof parallel manipulator is composed of a base structure, a moving platform, and three legs connecting the base and platform. Among those three legs, two of them are in same plane and consist of identical planar four bar parallelograms as chains connected to the moving platform by revolute joints, while the third leg is one rectangular bar connected to the moving platform by a spatial joint. There is one revolute joint on the top end of each leg, and the revolute joint is linked to the base by an active prismatic joint.

The CAD model of the 3-dof parallel mechanism is shown in Fig. 9.22.

The dof for a closed-loop kinematic chain can be determined using the Chebychev–Grübler–Kutzbach formula [170]:

$$l = d(n - g - 1) + \sum_{i=1}^{g} f_i, \qquad (9.16)$$

where l is the degree of freedom of the kinematic chain; d the degree of freedom of each unconstrained individual body (6 for the spatial case, 3 for the planar case);

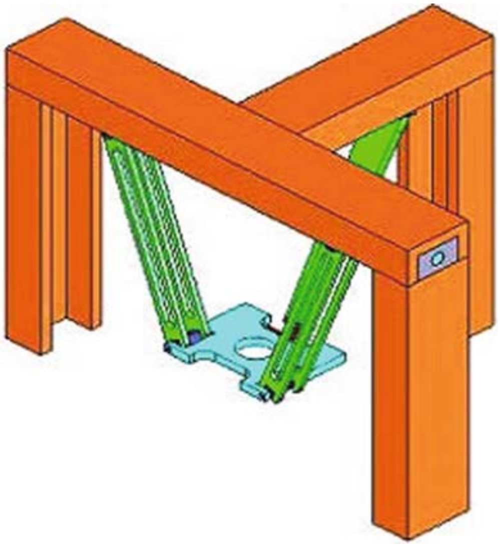


Fig. 9.22 CAD modeling of 3-dof parallel manipulator

n the number of rigid bodies or links in the chain; g the number of joints; and f_i is the number of degrees of freedom allowed by the ith joint.

With eq. (9.16), the degree of freedom of the proposed parallel manipulator is

$$M = 6(8 - 9 - 1) + (5 + 5 + 5) = 3. \tag{9.17}$$

In Fig. 9.22, the parallelograms play the role of improving the kinematics performance and the leg stiffness can be increased largely [21]. In regard to the types of actuated joints, they can be either revolute or prismatic. Since the prismatic joints can easily achieve high accuracy and heavy loads, the majority of the 3-dof parallel mechanism in reality use actuated prismatic joints.

The output link of a planar parallelogram mechanism will remain in a fixed orientation with respect to its input link, and the parallelogram can ensure the desired output, in terms of translation and rotation. The advantages of the proposed parallel manipulator are as follows:

1. The use of the parallelogram structure can greatly increase the stiffness of the legs.
2. Two identical chains offer good symmetry.
3. The joint which connects the third leg and the moving platform gives the rotation about y-axis with respect to reference frame attached to the end-effecter.

A kinematics model of the manipulator is shown in Fig. 9.23. The vertices of the moving platform are p_i $(i = 1, 2, 3)$, and the vertices of the base are b_i $(i = 1, 2, 3)$.

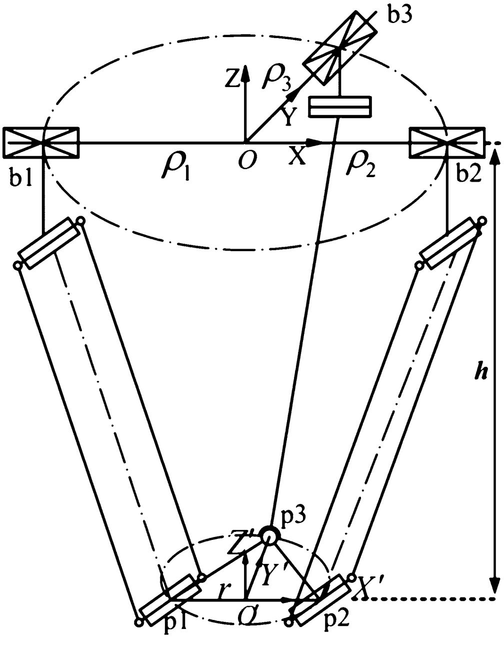


Fig. 9.23 Schematic representation

A global reference system $O : O - xyz$ is located at the point of intersection b_1b_2 and Ob_3. Another reference system, called the moving frame $O' : O' - x'y'z'$, is located at the center of p_1p_2 on the moving platform. The given position and orientation of the end-effecter (the moving platform) is specified by its three independent motions: y, z pure translations and ϕ pure rotation about y-axis. The position is given by the position vectors $(\mathbf{O}')_O$ and the orientation is given by rotation matrix $\mathbf{Q}$ as follows:

$$(\mathbf{O}')_O = (x\ y\ z)^{\mathrm{T}}, \tag{9.18}$$

where $x = 0$ and

$$\mathbf{Q} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}, \tag{9.19}$$

where angle ϕ is the rotation about y-axis. The coordinates of the point p_i in reference system $(\mathbf{O}')$ can be described by the vector $(\mathbf{p}_i)_{O'}$ $(i = 1, 2, 3)$

$$(\mathbf{p}_1)_{O'} = \begin{pmatrix} -r \\ 0 \\ 0 \end{pmatrix}, \ (\mathbf{p}_2)_{O'} = \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix}, \ (\mathbf{p}_3)_{O'} = \begin{pmatrix} 0 \\ r \\ 0 \end{pmatrix}. \tag{9.20}$$

The vectors $(\mathbf{b}_i)_O$ $(i = 1, 2, 3)$ in frame $O : O - xyz$ will be defined as position vectors of joints:

$$(\mathbf{b}_1) = \begin{pmatrix} -\rho_1 \\ 0 \\ 0 \end{pmatrix}, \ (\mathbf{b}_2) = \begin{pmatrix} \rho_2 \\ 0 \\ 0 \end{pmatrix}, \ (\mathbf{b}_3) = \begin{pmatrix} 0 \\ \rho_3 \\ 0 \end{pmatrix}. \tag{9.21}$$

The vector $(p_i)_O$ $(i = 1, 2, 3)$ in frame $O : O - xyz$ can be written as

$$(\mathbf{p}_i)_O = \mathbf{Q}(\mathbf{p}_i)_{O'} + (\mathbf{O}')_O. \tag{9.22}$$

That is

$$(\mathbf{p}_1)_O = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{pmatrix} -r \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ y \\ z \end{pmatrix} = \begin{pmatrix} -r\cos\phi \\ y \\ r\sin\phi + z \end{pmatrix}, \tag{9.23}$$

$$(\mathbf{p}_2)_O = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos\phi \\ y \\ -r\sin\phi + z \end{pmatrix}, \tag{9.24}$$

$$(\mathbf{p}_3)_O = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{pmatrix} 0 \\ r \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ r + y \\ z \end{pmatrix}. \tag{9.25}$$

The inverse kinematics of the manipulator can be solved by applying the following constraint equation:

$$\|\mathbf{p}_i - \mathbf{b}_i\| = L. \tag{9.26}$$

Hence, one can obtain the required actuator inputs:

$$\rho_1 = \sqrt{L^2 - y^2 - (z + r\sin\phi)^2} + r\cos\phi, \tag{9.27}$$

$$\rho_2 = \sqrt{L^2 - y^2 - (z - r\sin\phi)^2} + r\cos\phi, \tag{9.28}$$

$$\rho_3 = \sqrt{L^2 - z^2} + y + r. \tag{9.29}$$

Equations (9.27), (9.28), and (9.29) can be differentiated with respect to time to obtain the velocity equations,

$$(\rho_1 - r\cos\phi)\dot{\rho}_1 + y\dot{y} + (z + r\sin\phi)\dot{z} + r(\rho_1\sin\phi + z\cos\phi)\dot{\phi} = 0, \tag{9.30}$$

$$(\rho_2 - r\cos\phi)\dot{\rho}_2 + y\dot{y} + (z - r\sin\phi)\dot{z} + r(\rho_2\sin\phi - z\cos\phi)\dot{\phi} = 0, \tag{9.31}$$

$$[\rho_3 - (y + r)]\dot{\rho} + [\rho_3 - (y + r)]\dot{y} + z\dot{z} = 0. \tag{9.32}$$

Rearranging (9.30), (9.31), and (9.32) we have

$$\mathbf{A}\dot{\boldsymbol{\rho}} = \mathbf{B}\dot{\mathbf{p}}, \tag{9.33}$$

where $\dot{\boldsymbol{\rho}}$ is the vector of input velocities defined as

$$\dot{\boldsymbol{\rho}} = (\dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3)^{\mathrm{T}} \tag{9.34}$$

and $\dot{\mathbf{p}}$ is the vector of output velocities defined as

$$\dot{\mathbf{p}} = (\dot{y}, \dot{z}, \dot{\phi})^{\mathrm{T}} \tag{9.35}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ can be expressed as

$$\mathbf{A} = \begin{bmatrix} r\cos\phi - \rho_1 & 0 & 0 \\ 0 & r\cos\phi - \rho_2 & 0 \\ 0 & 0 & y + r - \rho_3 \end{bmatrix}, \tag{9.36}$$

$$\mathbf{B} = \begin{bmatrix} y & z + r\sin\phi & r(\rho_1\sin\phi + z\cos\phi) \\ y & z - r\sin\phi & r(\rho_2\sin\phi - z\cos\phi) \\ y + r - \rho_3 & z & 0 \end{bmatrix}. \tag{9.37}$$

The Jacobian matrix of the manipulator can be written as

$$\mathbf{J} = \mathbf{A}^{-1}\mathbf{B} \quad \text{or} \quad \mathbf{K} = \mathbf{J}^{-1} = \mathbf{B}^{-1}\mathbf{A}. \tag{9.38}$$

9.5.1.2 Optimization

The purpose of optimization design is to enhance the performance indices by adjusting the structure parameters. We propose the mean value and the standard deviation of the global stiffness as the design indices in this paper. It is noted that the trace of the matrix is an invariant of the matrix, so the distribution of the system stiffness (matrix) is the distribution of the trace. The mean value represents the average stiffness of the parallel robot manipulator over the workspace, while the standard deviation indicates the stiffness variation relative to the mean value. In general, the higher the mean value the less the deformation, and the lower the standard deviation the more uniform the stiffness distribution over the workspace. In this paper, the suitability of these design indices for the system stiffness will be examined by developing their relationship with the stiffness of links and joints. We will further study a design optimization based on the stiffness indices. A multiobjective optimization problem will be defined. Ideally, it may require that the mean value should be a maximum, but the standard deviation is a minimum. However, these two goals could be in conflict, so a trade-off process (e.g., Pareto set theory) will be considered.

The goal of structure parameters design, which is also called dimensional synthesis, is to confirm the best geometric configuration according to objective function and geometric restriction.

Since only a few geometric parameters can be handled due to the lack of convergence, this arises from the fact that traditional optimization methods use a local search by a convergent stepwise procedure, e.g., gradient, Hessians, linearity, and continuity, which compares the values of the next points and moves to the relative optimal points [60]. Global optima can be found only if the problem possesses certain convexity properties which essentially guarantee that any local optimum is a global optimum. In other words, conventional methods are based on a point-to-point rule; it has the danger of falling in local optima. The genetic algorithms are based on the population-to-population rule; it can escape from local optima.

For the implementation of genetic algorithms, one problem is how to model the objective function. It is very difficult and time-consuming exercise especially when the parameters are multifarious and the objective functions are too complex that genetic algorithm cannot work well based on the analytical expression of the performance indices. In artificial neural networks implementation, knowledge is represented as numeric weights, which are used to gather the relationships between data that are difficult to realize analytically. The network parameters can be iteratively adjusted to minimize the sum of the squared approximation errors using a gradient descent method, thereby being utilized to represent the system stiffness for the 3-dof parallel manipulator.

Stiffness is a very important factor in many applications including machine tool design, as it affects the precision of machining. Induced vibration is explicitly linked to machine tool stiffness. For a metal-cutting machine tool, high stiffness allows higher machining speeds and feeds while providing the desired precision, thus reduces vibration (such as chatter). Therefore, to build and study a general stiffness model of parallel mechanisms is very important for machine tool design.

From the viewpoint of mechanics, the stiffness is the measurement of the ability of a body or structure to resist deformation due to the action of external forces. The stiffness of a parallel mechanism at a given point of its workspace can be characterized by its stiffness matrix. This matrix relates the forces and torques applied at the gripper link in Cartesian space to the corresponding linear and angular Cartesian displacements.

The velocity relationship of parallel mechanisms can be written as

$$\dot{\boldsymbol{\theta}} = \mathbf{J}\dot{\mathbf{x}}, \tag{9.39}$$

where $\dot{\boldsymbol{\theta}}$ is the vector of joint rates and $\dot{\mathbf{x}}$ is the vector of Cartesian rates, a six-dimensional twist vector containing the velocity of a point on the platform and its angular velocity. Matrix $\mathbf{J}$ is the Jacobian matrix in (9.23).

From (9.24), one can conclude that

$$\boldsymbol{\delta}\theta = \mathbf{J}\boldsymbol{\delta}\mathbf{x}, \tag{9.40}$$

where $\boldsymbol{\delta}\theta$ and $\boldsymbol{\delta}\mathbf{x}$ represent joint and Cartesian infinitesimal displacements, respectively. Then, one can get the stiffness of this mechanism using the principle of kinematic/static duality. The forces and moments applied at the gripper under static conditions are related to the forces or moments required at the actuators to maintain the equilibrium by the transpose of the Jacobian matrix $\mathbf{J}$. One can then write

$$\mathbf{F} = \mathbf{J}^{\mathrm{T}}\mathbf{f}, \tag{9.41}$$

where $\mathbf{f}$ is the vector of actuator forces or torques and $\mathbf{F}$ is the generalized vector of Cartesian forces and torques at the gripper link. The actuator forces and displacements can be related by Hooke's law, one has

$$\mathbf{f} = \mathbf{K}_{\mathrm{J}}\boldsymbol{\delta}\boldsymbol{\theta}. \tag{9.42}$$

Here $\mathbf{K}_{\mathrm{J}}$ is the joint stiffness matrix of the parallel mechanism, with $\mathbf{K}_{\mathrm{J}} = \mathrm{diag}[k_1, \ldots, k_n]$, where each of the actuators in the parallel mechanism is modeled as an elastic component. k_i is a scalar representing the joint stiffness of each actuator, which is modeled as linear spring. Substituting (9.25) into (9.26), one obtains

$$\mathbf{f} = \mathbf{K}_{\mathrm{J}}\mathbf{J}\boldsymbol{\delta}\mathbf{x}. \tag{9.43}$$

Then, substituting (9.28) into (9.26), yields

$$\mathbf{F} = \mathbf{J}^{\mathrm{T}}\mathbf{K}_{\mathrm{J}}\mathbf{J}\boldsymbol{\delta}\mathbf{x}. \tag{9.44}$$

Hence, $\mathbf{K}_{\mathrm{C}}$, the stiffness matrix of the mechanism in the Cartesian space is then given by the following expression

$$\mathbf{K}_{\mathrm{C}} = \mathbf{J}^{\mathrm{T}}\mathbf{K}_{\mathrm{J}}\mathbf{J}. \tag{9.45}$$

Particularly, in the case for which all the actuators have the same stiffness, i.e., $k = k_1 = k_2 = \cdots = k_n$, then (9.30) will be reduced to

$$\mathbf{K}_{\mathrm{C}} = k\mathbf{J}^{\mathrm{T}}\mathbf{J}. \tag{9.46}$$

Furthermore, the diagonal elements of the stiffness matrix are used as the system stiffness value. These elements represent the pure stiffness in each direction, and they reflect the rigidity of machine tools more clearly and directly. The objective function for mean value and standard deviation of system stiffness can be written as:

$$\mu\text{-stiffness} = \mathbf{E}(\mathrm{tr}(\mathbf{K}_{\mathrm{C}})), \tag{9.47}$$

$$\sigma\text{-stiffness} = \mathbf{D}(\mathrm{tr}(\mathbf{K}_{\mathrm{C}})), \tag{9.48}$$

where $\mathbf{E}(\cdot)$ and $\mathbf{D}(\cdot)$ represent the mean value and the standard deviation, respectively, and tr is the trace of the stiffness matrix $\mathbf{K}_{\mathrm{C}}$.

In order to obtain the optimal system stiffness of the 3-dof parallel manipulator, three geometrical parameters are selected as optimization parameters. The vector of optimization variables is

$$\mathbf{s} = \{L, h, r\}, \tag{9.49}$$

where L is link length, h is the height of the moving platform, r is the radii of fixed platform, and their bounds are

$$L \in [1, 2]\,\mathrm{m},\;\; h \in [0.5, 0.8]\,\mathrm{m},\;\; r \in [0.1, 0.3]\,\mathrm{m} \tag{9.50}$$

The standard back propagation learning algorithm, as the most popular training method for feed-forward neural network, is based on the principle of steepest descent gradient approach to minimize a criterion function representing the instantaneous error between the actual outputs and the predicted outputs.

The criterion function can be expressed as follows:

$$E = \sum_{k=1}^{K} E_k^2 = \frac{1}{2NK}\sum_{k=1}^{K}\left[\sum_{i=1}^{N}(y_{ik} - t_{ik})^2\right], \tag{9.51}$$

where K is the number of output neurons, N is the vector dimension, and y_{ik}, t_{ik} are the predicted outputs and actual outputs of the kth output neuron of the ith input dimension, respectively.

The basic training steps of back propagation neural network are included as follows:

1. Initialize the weights and bias in each layer with small random values to make sure that the weighted inputs of network would not be saturated.
2. Confirm the set of input/output pairs and the network structure. Set some related parameters, i.e., the desired minimal , the maximal iterative times, and the learning speed.

3. Compare the actual output with desired network response and calculate the deviation.
4. Train the updated weights based on criterion function in each epoch.
5. Continue the above two steps until the network satisfy the training requirement.

Figure 9.24 shows the topology of network developed as the objective function to model the analytical solution of mean value of the system stiffness (μ-stiffness). In this case, two hidden layers with sigmoid transfer function are established in which eight neurons exist, respectively. The input vectors are the random arrangement of discretization values from the three structure variables.

Figure 9.25 illustrates the training result using standard back propagation learning algorithm, where the green curve denotes the quadratic sum of output errors with respect to ideal output values. After training for 474 times, target goal error is arrived.

The genetic algorithms can be implemented to search for the best solutions after the trained neural network is ready for the objective function. To avoid the time-consuming iterative operation using traditional technologies, the issue of stiffness optimization can be converted into network optimization. Figure 9.26 shows the evolution process of the best individual (network) based on genetic algorithms. The optimal μ-stiffness value is 2,218.

The evolution of μ-stiffness arises from the optimization of architecture and behavior variables in the implementation process of genetic algorithm as shown in Fig. 9.27. By adjusting the three parameters simultaneously with genetic operators such as selection, crossover, and mutation, the optimal objective is obtained. The final values of three parameters searched by genetic algorithm are

$$\mathbf{s} = \{L, h, r\} = \{1.5718\,\mathrm{m}, 0.78579\,\mathrm{m}, 0.18845\,\mathrm{m}\}. \tag{9.52}$$

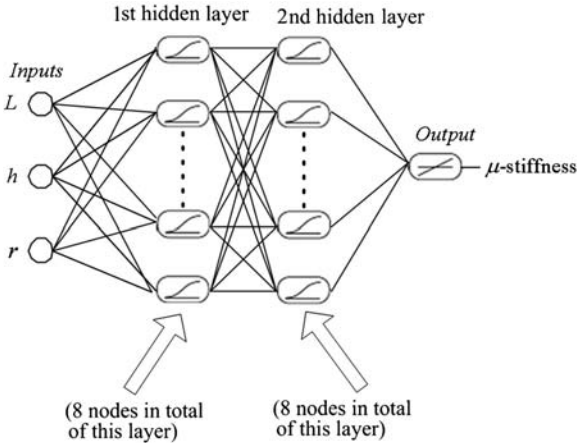

1st hidden layer
2nd hidden layer
Inputs
L
h
r
Output
μ-stiffness
(8 nodes in total of this layer)
(8 nodes in total of this layer)

Fig. 9.24 The topology of feed forward neural network for the solution of μ-stiffness

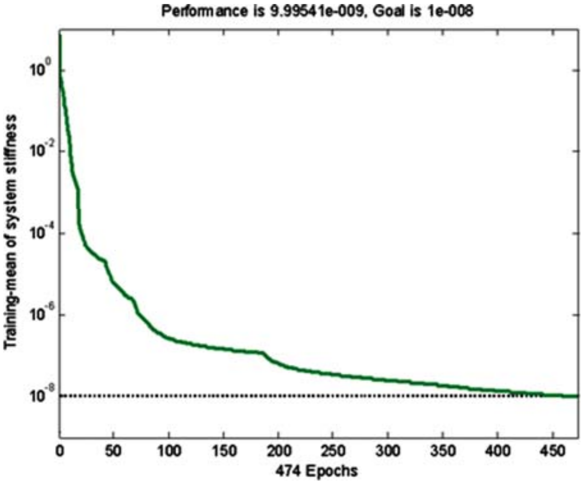
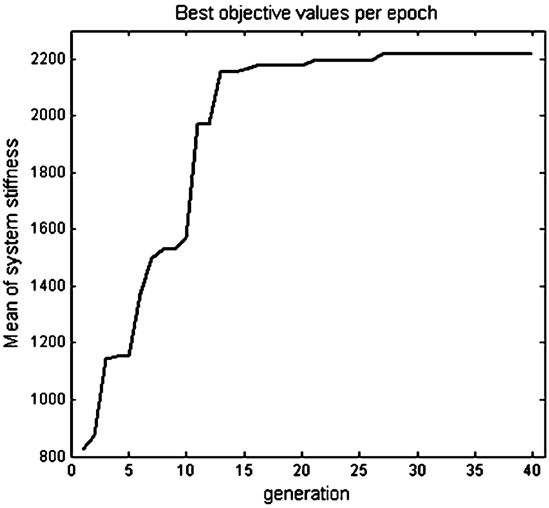


Fig. 9.25 Network training of the objective function about μ-stiffness

Fig. 9.26 The evolution of μ-stiffness



The topology of neural networks is similar with the above instance for the optimization of variance of system stiffness (σ-stiffness), in which the only difference is that there are 12 neurons exist in each hidden layer. Figure 9.28 illustrates the training result of the objective function about σ-stiffness with the back propagation algorithm, where the solid curve denotes the quadratic sum of output errors with respect to ideal output values. After training for 294 times, target goal error is arrived. The optimization process of σ-stiffness with genetic algorithm is illustrated in Fig. 9.29. After global stochastic search for 40 generations, optimal σ-stiffness value is convergent at 0.47. The evolution of geometrical parameters for σ-stiffness optimization is described in Fig. 9.30. Compared with Fig. 9.27, it can be found that

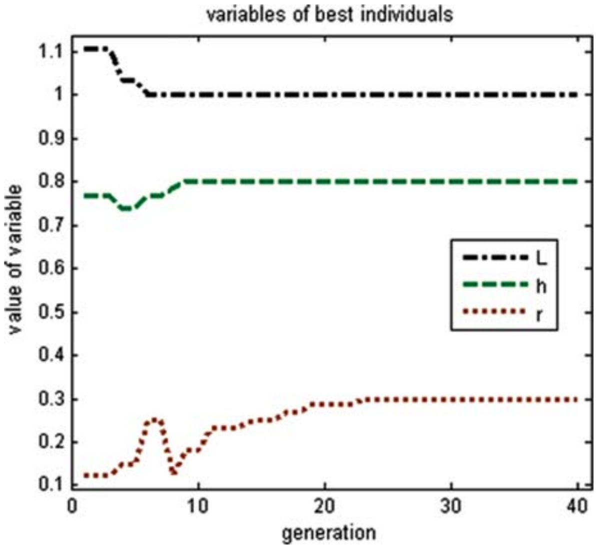


Fig. 9.27 The evolution of geometrical parameters for μ-stiffness optimization

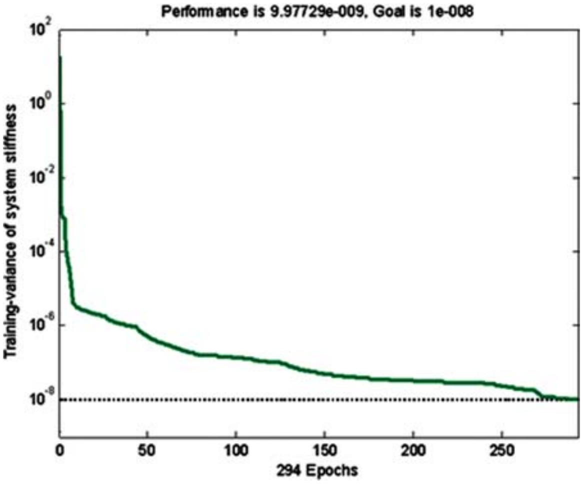


Fig. 9.28 Network training of the objective function about σ-stiffness

the corresponding convergent points of the three parameters in these two figures are not the same, i.e.,

$$\mathbf{s} = \{L, h, r\} = \{1.5718\,\mathrm{m}, 0.78579\,\mathrm{m}, 0.18845\,\mathrm{m}\}. \tag{9.53}$$

In other words, the two objective functions are conflicted with each other. This issue will be addressed in the following section where multiobjective optimization

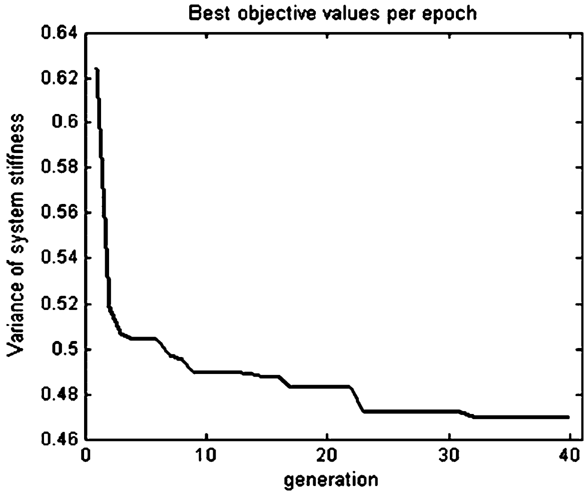


Fig. 9.29 The evolution of σ-stiffness

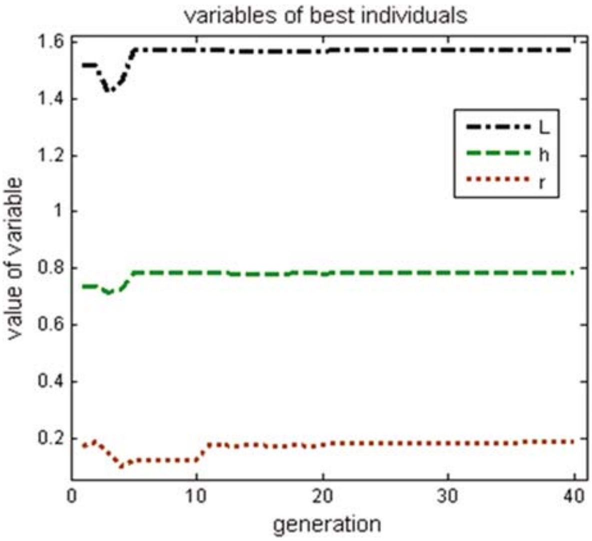


Fig. 9.30 The evolution of geometrical parameters for σ-stiffness optimization

based on Pareto-optimal solution is conducted. Multiobjective optimization problems consist of simultaneously optimizing several objective functions that are quite different from those of single-objective optimization. One single global optimal search is enough for single-objective optimization task. However, in a multiobjective optimization problem, it requires to find all possible tradeoffs among multiple objective functions that usually conflict with each other. The set of Pareto-optimal solutions is generally used for decision maker.

The basic concept of multiobjective optimization is the concept of domination [115]. In the issue of maximizing the k objective functions, decision vector (sets of variable) $x*$ is the Pareto-optimal solution if no other decision vectors satisfy both the following conditions:

$$f_i(x) \geq f_i(x^*), \ \forall i \in \{1, 2, \ldots, k\}, \tag{9.54}$$

$$f_j(x) > f_j(x^*), \ \exists j \in \{1, 2, \ldots, k\}. \tag{9.55}$$

With the same method, if both of the following conditions are true, decision vector x dominates y in the maximization issue, noted by $x > y$. That is:

$$f_i(x) \geq f_i(y), \ \forall i \in \{1, 2, \ldots, k\}, \tag{9.56}$$

$$f_j(x) > f_j(x^*), \ \exists j \in \{1, 2, \ldots, k\}. \tag{9.57}$$

According to the above formulae, Pareto-optimal set can be defined as: if there is no solution in the search space which dominates any member in the set P, then the solutions belonging to the set P constitute a global Pareto-optimal set. The Pareto-optimal set yields an infinite set of solutions, from which the desired solution can be chosen. In most cases, the Pareto-optimal set is on the boundary of the feasible region. Typical application of Pareto-based approach can be found in [64]. Since the implementation of genetic algorithms – including selection, crossover, and mutation operation – focuses on the whole colony which is consisted by all individuals, and generally, Pareto-optimal solutions for multiobjective optimization issues are a multidimensional set. Therefore, genetic algorithms are the effective method to address the Pareto-optimal solutions of multiobjective optimization issues.

Following initial parameters of Pareto-based genetic algorithms are set before implementation:

Number of subpopulation = 3
Number of individuals in each subpopulation = 50, 30, 40
Mutation range = 0.01
Mutation precision = 24
Max generations for algorithm termination = 200

After optimization, the possible optimal solutions in the whole solution space are obtained without combining all the objective functions into a single one by weighting factors. Figure 9.31 shows the Pareto-optimal frontier sets in which the designers can intuitively determine the final solutions depending on their preferences. Hence, the analysis process and cycle time is reduced. From this figure, the trade-off between the objectives of μ-stiffness and σ-stiffness is demonstrated in the distributing trend of these Pareto points for selecting compromisingly. If any other pair of design variables is chosen from the upper/right area of Figure 9.31, its corresponding values will locate an inferior point with respect to the Pareto frontier. Besides, the lower/left side is the inaccessible area of all the possible solution pairs. That is why Pareto solutions are called Pareto-optimal frontier sets. Figure 9.32 illustrates the

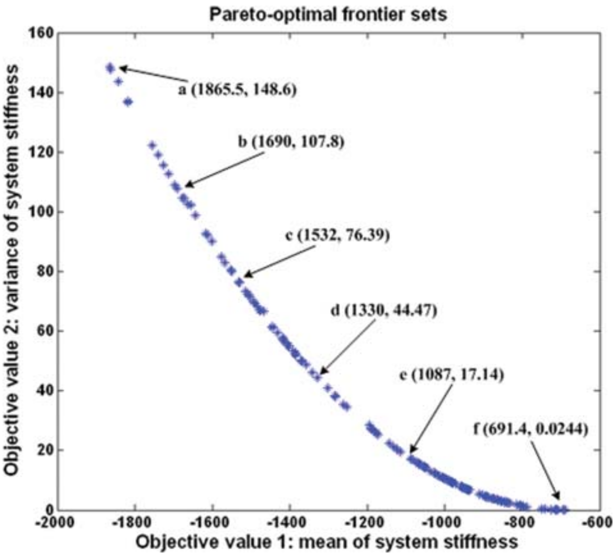


Fig. 9.31 Pareto-optimal frontier sets

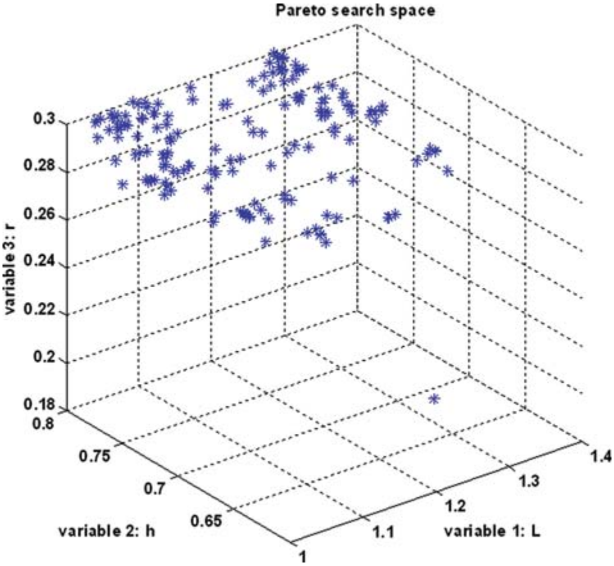


Fig. 9.32 Pareto search space: best individuals at end of optimization

solution distribution in the three-dimensional Pareto search space. It shows that a set of satisfied optimal solutions which provide enough information about alternative solutions for the decision maker with great diversity can be obtained with Pareto-based genetic algorithms. Therefore, the simulation shows the efficiency of the proposed single-objective and multiobjective optimization design methodology of the 3-dof parallel manipulator.

9.5.2 Case Study 2: Tripod Compliant Parallel Micromanipulator

9.5.2.1 Structure Description

As shown Fig. 9.33, the mechanism is treated as two separated components: a parallel mechanism with three 7-dof $S\underline{P}S$ legs and a branched chain with a 3-dof RPR passive leg. Compared to common tripod parallel mechanism with no passive leg, more advantages can be found in this design. First, its motion comes solely from actuation of the prismatic components which facilitates in control and analysis of the motion paths of the mechanism. Second, it provides an adequate working envelope and due to the nature of the deformation incurred by the joints, it has a relatively high duration. Finally, it has a high accuracy but still leaves room for improvement due to the fact that input angles are used as the reliant factor. This also further complicates the motion control slightly and the Jacobian matrix. Since rotation about the z-axis is not required, this leg constrains that motion.

It also minimizes the torque and force on the other components of the mechanism. However, it must sustain induced bending and torsion created by external loads on the moving platform. The additional leg provides further stiffness at the end effecter and increases the overall precision and repeatability.

The repeated use of spherical joints has both advantages and disadvantages associated with it. It can cut down on the cost of the equipment because there are

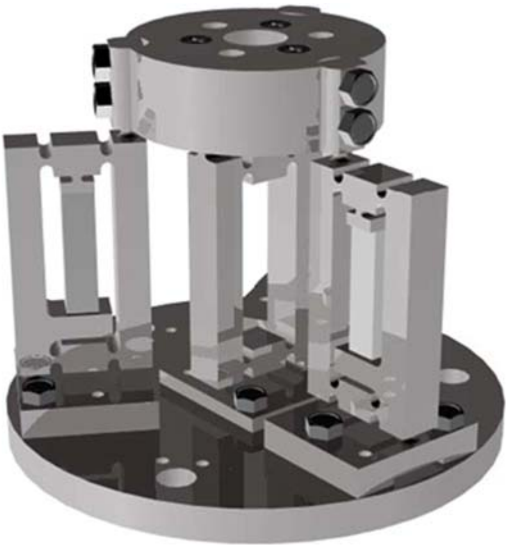


Fig. 9.33 Conceptual design of the parallel compliant micro-manipulator

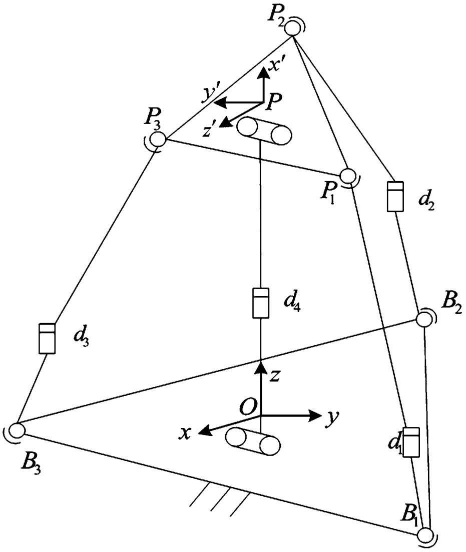


Fig. 9.34 Kinematical structure of a 3S$\underline{P}$S+RPR parallel mechanism

less different types of parts required. This also facilitates the manufacturing and assembly processes. However, the use of spherical joints does limit the motion and working envelope of the mechanism. Because the motion required of the device is so tiny, this drawback can be neglected. In terms of analysis and motion control, it appears to be suitable also. Each of the two mechanisms is solved separately to determine the inverse kinematics.

There are some important assumptions that must be noted before progressing with the inverse kinematic modeling. It is assumed that each leg is driven by one actuator which drives the prismatic joint. It is supposed that the centers of the joints which form a triangle on both the base and moving platform are located on circles. The centers of these circles serve as the origins for both the fixed reference frame, denoted by $O\,\{x, y, z\}$ in Fig. 9.34 and a moving coordinate frame, denoted by $P\,\{x', y', z'\}$. The points of attachment of the revolute joints at the base are expressed by B_i and of the spherical joints at the moving platform by P_i where $i = 1$, 2, 3. Points B_1, B_2, and B_3 lie on the x–y plane. Similarly, points P_1, P_2, and P_3 lie on the y'–z' plane. Furthermore, each platform is supposed to be an equilateral triangle.

9.5.2.2 Performance Indices Optimization

The goal of structure parameters design, which is also called dimensional synthesis, is to confirm the best geometric configuration according to objective function and geometric restriction. To make sure that the parallel manipulator will possess well performance such as high system stiffness and dexterity, dimensional synthesis for optimization is one of the significant steps in the design process of parallel manipulators. Both the single-objective optimizing and multiobjective optimizing issues will be investigated in this section to demonstrate the validity of synthesis of radial basis function network (RBFN) and genetic algorithm for this case.

As one type of feed-forward neural networks that is different from common networks such as back propagation networks, RBFN has a special structure consisting of two layers: a nonlinear hidden layer and a linear output layer. Each of the units in hidden layer applies a fixed-feature detector which uses a specified kernel function (i.e., Gaussian, thin plate spline, or multiquadratic) to detect and respond to localized portions of the input vector space. The network output is a weighted linear summation of the output of the hidden neurons [52, 141]. One advantage of radial basis networks over BPNN is that the localized nature of the hidden layer response makes the networks less susceptible to weight loss. The RBFN is a universal function approximation approach that demonstrates more robustness and flexibility than traditional regression approaches such as polynomial fits. The RBFN works by choosing not just a single nonlinear function, but a weighted sum of a set of nonlinear functions (Fig. 9.35).

The kernel functions in the hidden layer produce a localized response to the input by using the distance between the input vector and the center associated with the hidden unit as the variable. Suppose the input sample $\mathbf{X} \in \mathbf{R}^n$, the corresponding output of RBFN is:

$$\phi_j(x) = k_j(\|\mathbf{X} - \mathbf{C}_j\|_2, \sigma), \tag{9.58}$$

where $\mathbf{C}_j$ is the center associated with the hidden node j and σ is the controlling coefficient of kernel function for hidden node j, which represent a measure of the spread of data. $\|\mathbf{X} - \mathbf{C}_j\|$ is a norm of $\mathbf{X} \to \mathbf{C}_j$ that is usually Euclidean, which denotes the distance between the input vector $\mathbf{X}$ and $\mathbf{C}_j$. k_j is a kernel function with radial symmetry, which achieves the unique maximum at the point of $\mathbf{C}_j$. Generally, Gaussian function is selected as the kernel function, namely,

$$k_j(\|\mathbf{X} - \mathbf{C}_j\|_2, \sigma) = \exp\left(-\frac{\|\mathbf{X} - \mathbf{C}_j\|_2^2}{2\sigma^2}\right). \tag{9.59}$$

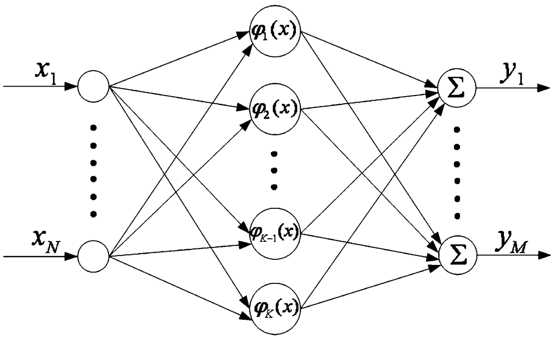


Fig. 9.35 The topology of RBFN

The response of each output node is calculated by a linear function of its input (including the bias), that is the output of hidden layer. Suppose that the number of hidden neurons and output neurons is K and M, respectively, the output value y_m of the mth output neuron for the input variable X can be represented by the following equation:

$$y_m = \sum_{i=1}^{K} w_{mi} k_j (\left\| \mathbf{X} - \mathbf{C}_j \right\|, \sigma), \tag{9.60}$$

where w_{mi}, which is adjusted to minimize the mean square error of the net output, is the weight between the mth output neuron and the ith hidden neuron.

Most of the training algorithms for RBFN have been divided into two stages. First, using unsupervised learning algorithm, the centers for hidden layer nodes can be determined. After the centers are fixed, the widths are determined in a way that reflects the distribution of the centers and input patterns. The pseudoinverse learning algorithm yields improved performance at a fraction of the computational and structural complexity of existing gradient descent algorithms for net weights training.

According to (9.60), the expression of error cost function $E(W)$ is as:

$$\mathbf{E(W)} = \frac{1}{2} \left\| \mathbf{T} - \mathbf{Y} \right\|_F^2, \tag{9.61}$$

where T is the net target output and $\|\cdot\|_F$ represents the F-norm of the given matrix.

$$\mathbf{E(W)} = \frac{1}{2} \left\| \mathbf{T} - \mathbf{HW} \right\|_F^2 = \frac{1}{2} \sum_{j=1}^{M} \sum_{i=1}^{L} \left(t_{ij} - \sum_{k=1}^{m} h_{ik} w_{kj} \right)^2, \tag{9.62}$$

where H denotes the output matrix of hidden layer. The partial derivative of $E(W)$ can be calculated as:

$$\frac{\partial \mathbf{E(W)}}{\partial \mathbf{W}} = \left(\frac{\partial \mathbf{E}}{\partial w_{uv}} \right)_{M \times M} \tag{9.63}$$

$$\frac{\partial \mathbf{J}}{\partial w_{uv}} = -\sum_{j=1}^{L} h_{ui}^{\mathrm{T}} \left(e_{iv} - \sum_{i=1}^{M} h_{ik} w_{kj} \right). \tag{9.64}$$

Thus following equation can be deduced:

$$\left(\frac{\partial \mathbf{E}}{\partial w_{uv}} \right)_{M \times M} = \mathbf{H}^{\mathrm{T}} (\mathbf{T} - \mathbf{HW}). \tag{9.65}$$

To achieve zero error of the net output, it has

$$\mathbf{H}^{\mathrm{T}} \mathbf{HW} = \mathbf{H}^{\mathrm{T}} \mathbf{E}. \tag{9.66}$$

Then the optimal solution of weights W^* can be obtained as

$$\mathbf{W}^* = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{E} = \mathbf{H}^{+}\mathbf{E}, \tag{9.67}$$

where H^+ is the Moore–Penrose pseudoinverse of hidden output H.

Since only a few geometric parameters can be handled due to the lack of convergence, this arises from the fact that traditional optimization methods use a local search by a convergent stepwise procedure, e.g., gradient, Hessians, linearity, and continuity, which compares the values of the next points and moves to the relative optimal points [60]. Global optima can be found only if the problem possesses certain convexity properties which essentially guarantee that any local optimum is a global optimum. In other words, conventional methods are based on a point-to-point rule; it has the danger of falling in local optima.

The genetic algorithms are based on the population-to-population rule; it can escape from local optima. Genetic algorithms have the advantages of robustness and good convergence properties, i.e.,

1. They require no knowledge or gradient information about the optimization problems; only the objective function and corresponding fitness levels influence the directions of search.
2. Discontinuities present on the optimization problems have little effect on the overall optimization performance.
3. They are generally more straightforward to introduce, since no restrictions for the definition of the objective function exist.
4. They use probabilistic transition rules, not deterministic ones.
5. They perform well for large-scale optimization problems.

Genetic algorithms have been shown to solve linear and nonlinear problems by exploring all regions of state space and exponentially exploiting promising areas through mutation, crossover, and selection operations applied to individuals in the population. Therefore, genetic algorithms are suitable for the optimization problems studied here.

Although a single-population genetic algorithm is powerful and performs well on a wide variety of problems. However, better results can be obtained by introducing multiple subpopulations. Figure 9.36 shows the optimization rationale of the extended multipopulation genetic algorithm adopted in this research.

Multiobjective optimization problems consist of simultaneously optimizing several objective functions that are quite different from those of single-objective optimization. One single global optimal search is enough for single-objective optimization task. However, in a multiobjective optimization problem, it is required to find all possible tradeoffs among multiple objective functions that are usually conflicting with each other. The set of Pareto-optimal solutions is generally used for decision maker.

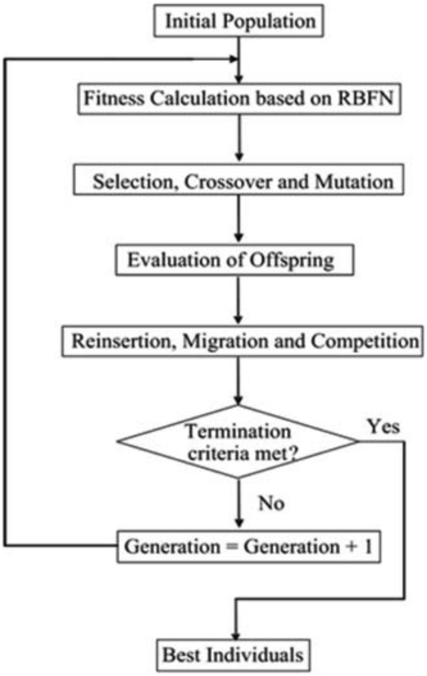

Initial Population
Fitness Calculation based on RBFN
Selection, Crossover and Mutation
Evaluation of Offspring
Reinsertion, Migration and Competition
Termination criteria met?
Yes
No
Generation = Generation + 1
Best Individuals

Fig. 9.36 Schematic representation of the optimization rationale based on genetic algorithms

Following initial parameters of Pareto-based genetic algorithms are set before implementation:

Number of subpopulation = 5
Number of individuals in each subpopulation = 110, 90, 90, 100, 110
Mutation range = 0.01
Mutation precision = 24
Max generations for algorithm termination = 80

Global stiffness (compliance), dexterity, and manipulability are considered together for the simultaneous optimization. After implementation, the possible optimal solutions in the whole solution space are obtained without combining all the objective functions into a single-objective function by weighting factors. Figure 9.37 shows the Pareto-optimal frontier sets in which the designers can intuitionistically determine the final solutions depending on their preferences. Hence, the analysis process and cycle time is reduced in large scale. From this picture, trade-off between the objectives of system stiffness, dexterity, and manipulability is demonstrated in the distributing trend of these Pareto points for selecting compromisingly. It shows that a set of satisfied optimal solutions which provide enough information about alternative solutions for the decision maker with great diversity can be obtained with Pareto-based genetic algorithms. Therefore, the simulation shows the efficiency of the proposed single-/multiobjective optimization methodology of the 3-dof parallel manipulator.

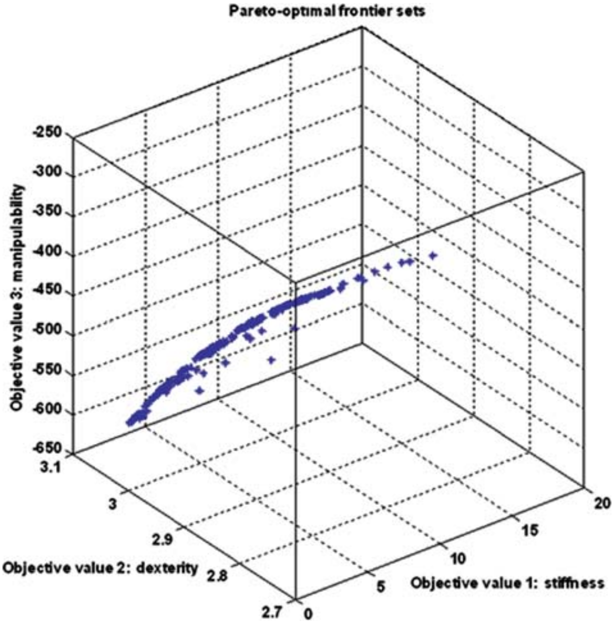


Fig. 9.37 Pareto-optimal solutions and Pareto frontier in the solution space

9.6 Conclusions

The kinetostatic model with its underlying design principles has been made more explicit through the implementation of optimization based on genetic algorithms in this chapter. A very remarkable implementation is the optimization of the Tricept machine tool family. After slightly adjusting the radius of the platform and the base, the total global stiffness can be improved 1.96 times. For the other mechanisms, the global stiffness are all obviously improved (normally 1.01–5.4 times). The kinetostatic model analyzed and obtained in previous chapters is employed for optimal structure design. From the results which have been achieved, it can be seen that the kinetostatic model can be applied for flexible mechanism analysis and global stiffness analysis and it can be further used as an optimization tool for parallel mechanisms. Moreover, the versatility of genetic algorithm compared to the conventional optimization methods is shown in this chapter; it is quite appropriate for dealing with multiparameters problem.