Chapter 10
Integrated Environment for Design and Analysis of Parallel Robotic Machine

10.1 Preamble

Because of the recent trend toward high-speed machining HSM, there is a demand to develop parallel kinematic machine with high dynamic performance, improved stiffness, and reduced moving mass [2, 11, 93, 148]. However, as researchers at Giddings and Lewis have indicated, full integration of standard automation components, CAD, and a user interface are required before making its parallel kinematic machine readily available for the general market. A virtual environment that can be used for PKM design, analysis, and simulation is urgently demanded. Several efforts have been done on this topic. Pritschow [122] proposed a systematic methodology for the design of different PKM topologies. Merlet [106] developed the software for the optimal design of a specific PKM class – Stewart platform-based mechanisms. Jin and Yang [79, 80] proposed a method for topology synthesis and analysis of parallel manipulators. Huang et al. [75] made some efforts on conceptual design of 3dof translational parallel mechanisms. Nevertheless, there is no complete virtual system existing for PKM design and analyze from the literature.

With the objective of developing a practical methodology and related virtual environment for PKM analysis and design, several activities have been conducted at Integrated Manufacturing Technologies Institute of National Research Council of Canada. PKM is a key component of reconfigurable manufacturing systems in different industrial sectors. It is very important for PKM designers to design and analysis the potential PKM with an integrated virtual environment before fabrication. The virtual environment is used for modeling, simulation, planning, and control of the proposed PKM.

An instance of a virtual parallel machine tool will include models of the machine tool and workpiece mechanics, the cutting process and the control system. The instance of a virtual parallel machine tool will be the reference model for an existing machine tool. A 3D virtual environment is both a visualization tool as well as an interface to the virtual machine tool or the actual machine tool. The software environment is also a design tool for constructing the modular components of a parallel machine tool as well as the integrated system design.

D. Zhang, *Parallel Robotic Machine Tools*, DOI 10.1007/978-1-4419-1117-9_10,

The objectives of the virtual environment are to develop:

1. A software environment for modeling, simulation, control of parallel machine tools
2. Machine tool simulations that can predict the geometry and surface finish of parts
3. Basic modeling and simulation capabilities of using the NRC 3-dof PKM as an example
4. Reconfigurable control systems
5. Systems for real-time inspection of machining operations and path planning in terms of singularity free and workspace verification.

10.2 Case Study

In general, a systematic design methodology for parallel kinematic machine design and analysis consists of two engines: a generator and an evaluator. Some of the functional requirements identified are transformed into structural characteristics. These structural characteristics are incorporated as rules in the generator. The generator defines all possible solutions via a combinational analysis. The remaining functional requirements are incorporated as evaluation criteria in the evaluator to screen out the infeasible solutions. This results in a set of candidate mechanisms. Finally the most promising candidate is chosen for product design. Therefore, the architecture of the virtual environment is illustrated in Fig. 10.1. It consists of several modules from conceptual design (selection of the most promising structure) to embodiment design, from kinematic/dynamic analysis (evaluation criteria) to design optimization, simulation and control. In the following, the key components of the system are described in detail.

For each of kinematic mechanisms, the kinematic chains involved may lead to several possibilities (serial, parallel, or hybrid). A preliminary evaluation of the mobility of a kinematic chain can be found from the Chebychev–Grbler–Kutzbach formula.

$$M = d(n - g - 1) + \sum_{i=1}^{g} f_i, \tag{10.1}$$

where M denotes the mobility or the system DOF, d is the order of the system ($d = 3$ for planar motion, and $d = 6$ for spatial motion), n is the number of the links including the frames, g is the number of joints, and f_i is the number of DOFs for the ith joint.

Kinetostatic analysis [170] is essential for PKMs that are used for metal cutting, which requires large forces. Higher stiffness, equivalently lower compliance, means little deformation, resulting in better surface finish and longer tool life. In this chapter, two global compliance indices are introduced, namely the mean value and the standard deviation of the trace of the generalized compliance matrix. The mean value represents the average compliance of the PKM over the workspace, while the standard deviation indicates the compliance fluctuation relative to the mean value.

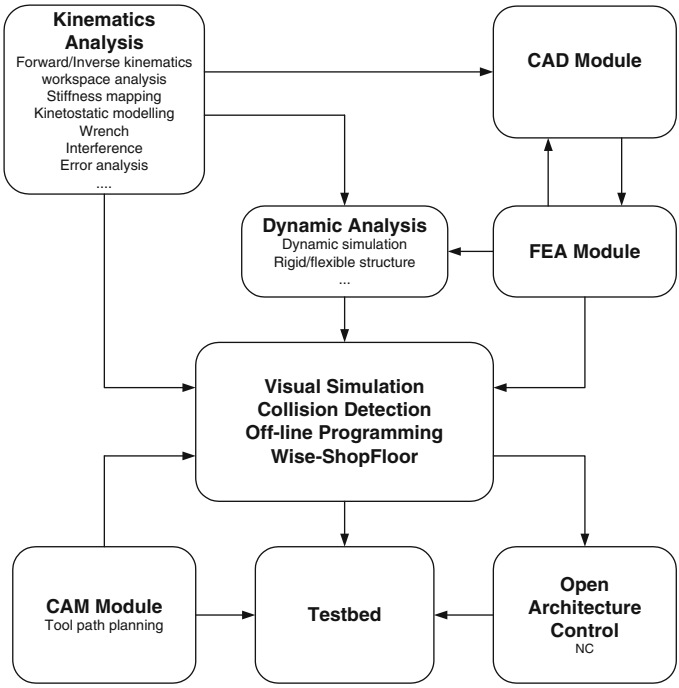


Fig. 10.1 Integrated environment for PKM design and analysis

In this model, it includes forward/inverse kinematics, workspace evaluation, velocity analysis, stiffness modeling, singularity analysis, and kinetostatic performance indices. The model will efficiently support the designer in the choice of a topological class of reconfigurable machine tools (RMT) and in the configuration of the machine belonging to that class.

An analysis package (MatLab) is used for comparative study of the characteristics of the RMT (i.e., manipulability, kinematics, stiffness mapping, workspace, kinetostatic, and dynamic analysis) to help the designer in selecting the most promising mechanism for a specific task. This tool will be used also for the development of control algorithms (i.e., inverse kinematic, interpolation, and real-time software collision checking). The kinetostatic model can be used to localize critical components, which mostly influence the global stiffness of the machine and further used for design optimization.

The dynamic model [174] will be used for accurate control and controller evaluation. The model should account for any factors that significantly affect the dynamic behavior of the parallel mechanisms. This includes joint friction, link flexibility, and eigenfrequencies used for the optimization of the model based servo (and for model verification). A general modeling method for the dynamic analysis of the PKM will be developed. In dynamic model, the mass/inertia, gravity of each component (including the links between the fixed base and the moving platform) will be included

in this model. And this model will provide the relationship between the applied force/torque on the tool and the driving force/torque of the actuators as well as the constraint f orce/torque of all joints of the PKM.

Lagrange's formulation is used for dynamic modeling of the 3-dof NRC parallel kinematic machine. First, dynamic equations of the moving platform and the legs are formulated and then are assembled.

In the present work, there are many optimization parameters and complex matrix computations. Hence, it is very difficult to derive the analytical expressions for each stiffness element and workspace volume. Moreover, with traditional optimization methods, only a few geometric parameters can be handled due to the lack of convergence of the optimization algorithm when used with more complex problems [3]. 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. Global optima can be found only if the problem possesses certain convexity properties that essentially guarantee that any local optimum is a global optimum. Classical optimization methods are based on point-to-point rules, and have the danger of falling in local optima, while the genetic algorithms are based on population-to-population rules, which allow them to escape from local optima. For this reason, genetic algorithms are selected as the best candidate for the optimization problems studied here.

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 [4].

To use genetic algorithms properly, several parameter settings have to be determined, and they include chromosome representation, selection function, genetic operators, the creation of the population size, mutation rate, crossover rate, and the evaluation function. They are described in more detail as follows. Among these parameters, the chromosome representation is a basic issue for the GA representation, and it is used to describe each individual in the population of interest. For the reconfigurable parallel kinematic machine tools, the chromosomes consist of the architecture parameters (coordinates of the attachment points, coordinates of the moving platform, vertex distributions at base and moving platform, platform height, etc.) and behavior parameters (actuator stiffness, actuated link stiffness, etc.) of the mechanisms. The roulette wheel approach [4] is applied as a selection function.

A CAD module, used for the implementation from conceptual design phase to embodiment design phase – developed with Unigraphics, will support the designer for further finite element analysis and evaluation of the structure deformation and stress. The CAD module also can be used for simulation to check the interference. Meanwhile, it can be used to generate drawings for fabrication.

Reducing costs and increasing production throughout are two of the major challenges facing manufacturing companies today. Therefore, a CAM module is necessary for manufacturing systems. A CAM system is developed to meet the

challenges with a set of capabilities for NC tool path creation, simulation, and verification. It delivers a single manufacturing solution capable of efficiently machining everything from holes to airfoils. The manufacturing application of Unigraphics allows one to interactively create NC machining programs, generate tool paths, visualize material removal, and post process. The CAM module can generate tool paths for several types of machining, such as planar and cavity milling, sequential milling, turning, surface contouring, drilling, thread milling, post builder, etc.

CAE module is used for finite element analysis (FEA). FEA software (Nastran) is required to do deformation and stress analysis of mechanical components included in the RMT. It can be further used to investigate mechanical properties and integrity of machined surfaces generated in high speed machining. It can be used to predict residual stresses and surface properties, determine the effects of diverse cutting-edge preparations and machining parameters (cutting speed, feed rate, and depth-of-cut) on the residual stress, distribution.

The Wise-ShopFloor is designed to provide users with a web-based and sensor-driven intuitive shop floor environment where real-time monitoring and control are undertaken. It utilizes the latest Java technologies, including Java 3D and Java Servlets, as enabling technologies for system implementation. Instead of camera images (usually large in data size), a physical device of interest (e.g., a milling machine or a robot) can be represented by a scene graph-based Java 3D model in an applet with behavioral control nodes embedded. Once downloaded from an application server, the Java 3D model is rendered by the local CPU and can work on behalf of its remote counterpart showing real behavior for visualization at a client side. It remains alive by connecting with the physical device through low-volume message passing (sensor data and user control commands). The 3D model provides users with increased flexibility for visualization from various perspectives, such as walk-through and fly-around that are not possible by using stationary optical cameras, whereas the largely reduced network traffic makes real-time monitoring, remote control, on-line inspection, and collaborative trouble-shooting practical for users on relatively slow hook-ups (e.g., modem and low-end wireless connections) through a shared Cyber Workspace [150].

By combining virtual reality models with real devices through synchronized real-time data communications, the Wise-ShopFloor allows engineers and shop floor managers to assure normal shop floor operations and enables web-based trouble-shooting – particularly useful when they are off-site.

The generalized stiffness matrix of a PKM relates a wrench including the forces and moments acting on the moving platform to its deformation. It represents how stiff the PKM is in order to withstand the applied forces and moments. By definition, the following relationship holds

$$\mathbf{w} = \mathbf{K}\delta\mathbf{x}, \tag{10.2}$$

where $\mathbf{w}$ is the vector representing the wrench acting on the moving platform, $\delta\mathbf{x}$ is the vector of the linear and angular deformation of the moving platform, and $\mathbf{K}$ is the generalized stiffness matrix. Vectors $\mathbf{w}$ and $\delta\mathbf{x}$ are expressed in the Cartesian coordinates $O - xyz$.

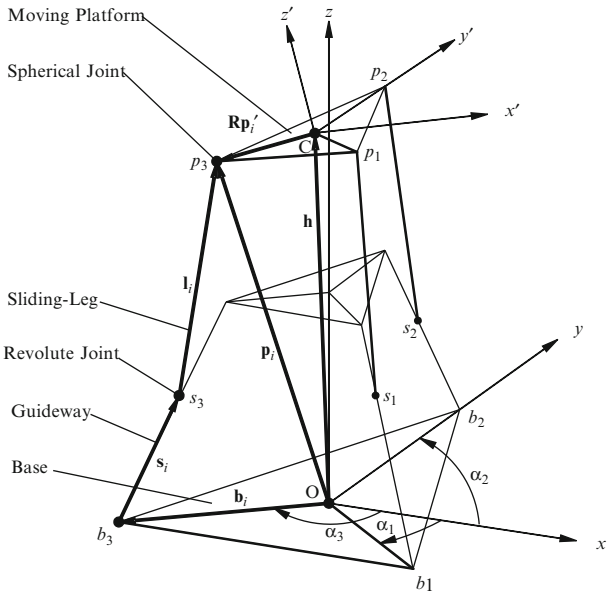


Fig. 10.2 Representation of NRC PKM

Since PKMs are parallel structures, the moving platform stiffness is a combination resulting from all serial chains including actuators. Figure 10.2 shows a schematic illustration of a 3 degree-of-freedom PKM with fixed-length legs that is built at NRC-IMTI. In this type of PKM, the moving platform is driven by sliding the fixed-length legs along the guideways. The advantages of the structure are the following: with this basic structure of parallel mechanism, it can be easily extended to 5-dof by adding two gantry type of guideways to realize the 5-dof machining; meanwhile, with the fixed length legs, one can freely choose the variety of leg forms and materials, and to use linear direct driver to improve the stiffness, and it is lack of heat sources to keep the precision in a high level, the stiffness is stable compare to variable legs.

Three types of compliance contribute to the deformation of the moving platform, namely actuator flexibility, leg bending, and axial deformation. A simple way of deriving the generalized stiffness matrix is to use the force relation and the infinitesimal motion relation as given below

$$\mathbf{w} = \mathbf{J}^{\mathrm{T}}\mathbf{f}, \tag{10.3}$$

$$\delta\mathbf{q} = \mathbf{J}\delta\mathbf{x}, \tag{10.4}$$

where $\mathbf{J}$ is the Jacobian matrix that relates the infinitesimal motion between the subserial chains and the moving platform, $\mathbf{f}$ is the vector representing forces in the sub-serial chains; $\delta\mathbf{q}$ is the vector representing infinitesimal motion of the subserial

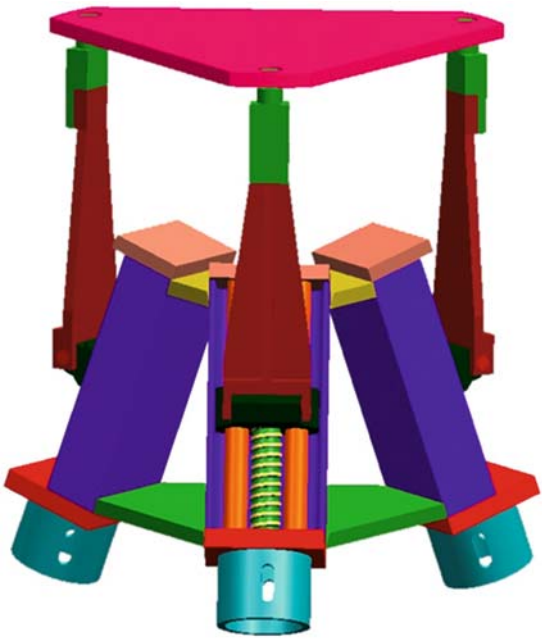


Fig. 10.3 CAD model of the tripod based PKM

chains. The infinitesimal motion of the subserial chains is referred to as the component deformation in the subserial chains. The component deformation would induce the forces, which are called the branch forces in the subserial chains (Fig. 10.3).

Considering the local stiffness in the subserial chains, denoted by $\overline{\mathbf{K}}$, the branch forces induced by the branch deformation can be written as

$$\mathbf{f} = \overline{\mathbf{K}}\delta\mathbf{q}. \tag{10.5}$$

The substitution of (10.4) and (10.5) into (10.3) yields

$$\mathbf{w} = \mathbf{K}\delta\mathbf{x}, \tag{10.6}$$

where the generalized stiffness matrix $\mathbf{K}$ is given as

$$\mathbf{K} = \mathbf{J}^{\mathrm{T}}\overline{\mathbf{K}}\mathbf{J}. \tag{10.7}$$

Equation (10.6) can be rewritten in terms of compliance as

$$\delta\mathbf{x} = \mathbf{C}\mathbf{w}, \tag{10.8}$$

where $\mathbf{C}$ is the generalized compliance matrix and $\mathbf{C} = \mathbf{K}^{-1}$. The generalized compliance matrix represents how much the moving platform would deform under the applied wrench $\mathbf{w}$.

When (10.8) is applied to consider the aforementioned three types of compliance, the following three types of the moving platform deformation would be induced

$$\delta \mathbf{x}_{\mathrm{t}} = \mathbf{C}_{\mathrm{t}}\mathbf{w}; \quad \delta \mathbf{x}_{\mathrm{b}} = \mathbf{C}_{\mathrm{b}}\mathbf{w}; \quad \delta \mathbf{x}_{\mathrm{a}} = \mathbf{C}_{\mathrm{a}}\mathbf{w}, \tag{10.9}$$

where subscripts t, b, and a indicate the deformation due to the torsion in the actuators, bending and axial deformation of the legs, respectively. Since these three deformations occur in a serial fashion, the total deformation can be considered as follows (based on the superposition theory in [142]).

$$\delta \mathbf{x} = \delta \mathbf{x}_{\mathrm{t}} + \delta \mathbf{x}_{\mathrm{b}} + \delta \mathbf{x}_{\mathrm{a}}. \tag{10.10}$$

This leads to the following compliance model,

$$\delta \mathbf{x} = \mathbf{C}_{\mathrm{G}}\mathbf{w}, \tag{10.11}$$

where the total generalized compliance matrix $\mathbf{C}_{\mathrm{G}}$ is given as

$$\mathbf{C}_{\mathrm{G}} = \mathbf{C}_{\mathrm{t}} + \mathbf{C}_{\mathrm{b}} + \mathbf{C}_{\mathrm{a}}. \tag{10.12}$$

In (10.12), $\mathbf{C}_{\mathrm{t}} = \mathbf{K}_{\mathrm{t}}^{-1}$, $\mathbf{C}_{\mathrm{b}} = \mathbf{K}_{\mathrm{b}}^{-1}$, and $\mathbf{C}_{\mathrm{a}} = \mathbf{K}_{\mathrm{a}}^{-1}$, and (10.12) can be rewritten as

$$\mathbf{C}_{\mathrm{G}} = \mathbf{K}_{\mathrm{t}}^{-1} + \mathbf{K}_{\mathrm{b}}^{-1} + \mathbf{K}_{\mathrm{a}}^{-1}, \tag{10.13}$$

where

$$\mathbf{K}_{\mathrm{t}} = \mathbf{J}_{\mathrm{t}}^{\mathrm{T}}\overline{\mathbf{K}}_{\mathrm{t}}\mathbf{J}_{\mathrm{t}}; \quad \mathbf{K}_{\mathrm{b}} = \mathbf{J}_{\mathrm{b}}^{\mathrm{T}}\overline{\mathbf{K}}_{\mathrm{b}}\mathbf{J}_{\mathrm{b}}; \quad \mathbf{K}_{\mathrm{a}} = \mathbf{J}_{\mathrm{a}}^{\mathrm{T}}\overline{\mathbf{K}}_{\mathrm{a}}\mathbf{J}_{\mathrm{a}}. \tag{10.14}$$

The total generalized stiffness matrix considering the three types of compliance can be written as

$$\mathbf{K}_{\mathrm{G}} = \mathbf{C}_{\mathrm{G}}^{-1}. \tag{10.15}$$

From (10.13), it can be seen that $\mathbf{C}_{\mathrm{G}}$ is defined by three different Jacobians and local stiffness corresponding to the three types of compliance.

The total generalized compliance matrix as defined in (10.12) does not have the appropriate units due to multiplication of the Jacobian. For this reason, a weighting matrix is applied to $\mathbf{C}_{\mathrm{G}}$ that becomes

$$\mathbf{C}_{\mathbf{W}} = \mathbf{W}\mathbf{C}_{\mathrm{G}}\mathbf{W}, \tag{10.16}$$

where the weighting matrix is defined as

$$\mathbf{W} = \mathrm{diag}(1, 1, 1, L, L, L). \tag{10.17}$$

In (10.16), L is a parameter with length unit. $\mathbf{C}_{\mathbf{W}}$ is a 6×6 matrix with the appropriate compliance units.

As shown in (10.12b), the compliance matrix is determined by the inverse of the stiffness matrix. Considering (10.13) and (10.15), the total generalized compliance matrix can be expressed as

$$\mathbf{C_W} = \mathbf{W}[(\mathbf{J}_t^T)\overline{\mathbf{K}}_t\mathbf{J}_t)^{-1} + (\mathbf{J}_b^T\overline{\mathbf{K}}_b\mathbf{J}_b)^{-1} + (\mathbf{J}_a^T\overline{\mathbf{K}}_a\mathbf{J}_a)^{-1}]\mathbf{W}. \tag{10.18}$$

For the prototype under study, it is an over-constrained kinematic system, and three Jacobains $\mathbf{J}_t$, $\mathbf{J}_b$ and $\mathbf{J}_a$ are 3×6 matrices. For this reason, the generalized inverse is applied and (10.18) is rewritten as

$$\mathbf{C_W} = \mathbf{C_{Wt}} + \mathbf{C_{Wb}} + \mathbf{C_{Wa}}, \tag{10.19}$$

where

$$\mathbf{C_{Wt}} = \mathbf{W}\mathbf{J}_t{}^{+}\overline{\mathbf{K}}_t{}^{-1}(\mathbf{J}_t{}^{+})^{\mathrm{T}}\mathbf{W}, \tag{10.20}$$

$$\mathbf{C_{Wb}} = \mathbf{W}\mathbf{J}_b{}^{+}\overline{\mathbf{K}}_b{}^{-1}(\mathbf{J}_b{}^{+})^{\mathrm{T}}\mathbf{W}, \tag{10.21}$$

$$\mathbf{C_{Wa}} = \mathbf{W}\mathbf{J}_a{}^{+}\overline{\mathbf{K}}_a{}^{-1}(\mathbf{J}_a{}^{+})^{\mathrm{T}}\mathbf{W}. \tag{10.22}$$

In (10.18), the superscript "+" indicates the generalized inverse matrix.

The generalized compliance matrix $\mathbf{C}_G$ varies over the PKM workspace. Conventional kinetostatic analysis methods, such as stiffness mapping, would require a large number of graphs to provide an overview of the stiffness variation. An alternative, however, could be based on statistical analysis. This method was proposed to evaluate the generalized mass matrix of PKMs over the workspace. On the basis of this concept, the mean value and the standard deviation of a selected parameter can be used to evaluate the variation over the workspace. Since the trace of the generalized compliance matrix is invariant, it is selected as a parameter for global kinetostatic analysis. The mean value and the standard deviation are defined as

$$\mu = E(\mathrm{tr}(\mathbf{C_W})), \tag{10.23}$$

$$\sigma = SD(\mathrm{tr}(\mathbf{C_W})), \tag{10.24}$$

where $E(\cdot)$ and $SD(\cdot)$ are the mean value and the standard deviation, and tr represents trace operation. The mean value represents the average compliance of the PKM over the workspace, while the standard deviation indicates the compliance fluctuation relative to the mean value. In general, the lower the mean value the lesser the deformation, and the lower the standard deviation the more uniform the compliance distribution over the workspace.

The method presented is generic and can be readily expanded to any kind of PKM or completely new topology of PKM. It is quite efficient for the conceptual design stage to rapidly configure and evaluate several configurations. It can be further used for geometry optimization.

The two global compliance indices introduced in (10.19) and (10.20) are used. In terms of $\mathbf{C_W}$, they are rewritten as

$$\mu_{\mathbf{W}} = E(\mathrm{tr}(\mathbf{C_W})), \tag{10.25}$$

$$\sigma_{\mathbf{W}} = SD(\mathrm{tr}(\mathbf{C_W})). \tag{10.26}$$

With the two indices, analysis can be conducted to consider the effect of change in leg and actuator flexibility. To do so, the following two stiffness ratios are defined

$$\alpha_1 = k_b/k_\mathrm{t}; \quad \alpha_2 = k_a/k_\mathrm{t}, \tag{10.27}$$

where k_t is the actuator's stiffness and it is fixed, k_b is the stiffness induced by compliant link bending, k_a is the stiffness induced by axial deformation, 1 and 2 change from 0.5 to 2.5. If the two ratios are less than 1, it indicates that the actuator is relatively stiffer than the leg. If they are equal to 1, the leg and the actuator are equally stiff. If the ratios are larger than 1, then the leg is relatively stiffer than the actuator. The term α_1 is for the leg bending and the term α_2 for the leg axial deformation. In terms of α_1 and α_2, $\mathbf{C_{Wa}}$ and $\mathbf{C_{Wb}}$ can be rewritten as

$$\mathbf{C_{Wb}} = (1/\alpha_1)\mathbf{W}\mathbf{J_b}^{+}\overline{\mathbf{K}}_\mathrm{t}^{-1}(\mathbf{J_b}^{+})^{\mathrm{T}}\mathbf{W}, \tag{10.28}$$

$$\mathbf{C_{Wa}} = (1/\alpha_2)\mathbf{W}\mathbf{J_a}^{+}\overline{\mathbf{K}}_\mathrm{t}^{-1}(\mathbf{J_a}^{+})^{\mathrm{T}}\mathbf{W}. \tag{10.29}$$

To investigate the effect of change in leg and actuator flexibility on the global kinetostatic behavior of the prototype, the differences of the mean value and standard deviation are used and they are defined for the three types of compliance under consideration as

$$\Delta\mu_{\mathbf{W}\mathrm{t}} = E(\mathrm{tr}(\mathbf{C_W} - \mathbf{C_{Wt}})), \tag{10.30}$$

$$\Delta\mu_{\mathbf{W}\mathrm{b}} = E(\mathrm{tr}(\mathbf{C_W} - \mathbf{C_{Wb}})), \tag{10.31}$$

$$\Delta\mu_{\mathbf{W}\mathrm{a}} = E(\mathrm{tr}(\mathbf{C_W} - \mathbf{C_{W}a})), \tag{10.32}$$

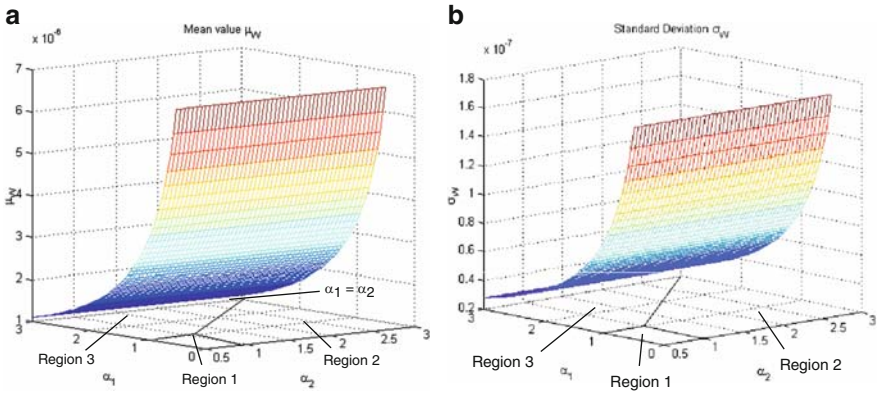
$$\Delta\sigma_{\mathbf{W}\mathrm{t}} = SD(\mathrm{tr}(\mathbf{C_W} - \mathbf{C_{Wt}})), \tag{10.33}$$

$$\Delta\sigma_{\mathbf{W}\mathrm{b}} = SD(\mathrm{tr}(\mathbf{C_W} - \mathbf{C_{Wb}})), \tag{10.34}$$

$$\Delta\sigma_{\mathbf{W}\mathrm{a}} = SD(\mathrm{tr}(\mathbf{C_W} - \mathbf{C_{Wa}})). \tag{10.35}$$

The differences defined in (10.25)–(10.26) indicate the proximity of $\mathbf{C_{Wt}}$, $\mathbf{C_{Wb}}$ and $\mathbf{C_{Wa}}$ to $\mathbf{C_W}$. A smaller value would mean a larger contribution to the total generalized compliance.

Figure 10.4 shows the simulation result considering the full motion range of the moving platform in the vertical direction. For the purpose of examining the two ratios, three regions are divided. Region 1 is for $\alpha_1, \alpha_2 < 1$, corresponding to the case that the leg is more flexible than the actuator. Region 2 is for $\alpha_1, \alpha_2 > 1$, and $\alpha_2 > \alpha_1$, corresponding to the case that the actuator is more flexible than the leg, while for the leg, the bending is larger than the axial deformation. Region 3 is for



Mean Value u_W Standard Deviation σ_W

Fig. 10.4 Simulation results of global compliance mean value u_W and global standard deviation σ_W: Region 1: $\alpha_1, \alpha_2 < 1$, the leg is more flexible than the actuator. Region 2: $\alpha_1, \alpha_2 > 1$, and $\alpha_2 > \alpha_1$, the actuator is more flexible than the leg, while for the leg, the bending is larger than the axial deformation. Region 3: $\alpha_1, \alpha_2 > 1$, and $\alpha_1 > \alpha_2$, the actuator is more flexible than the leg, while for the leg, the axial deformation is larger than the bending

$\alpha_1, \alpha_2 > 1$ and $\alpha_1 > \alpha_2$, corresponding to the case that the actuator is more flexible than the leg, while for the leg, the axial deformation is larger than the bending. As shown in Fig. 10.4, for small α_1, i.e., when the bending is larger, it induces very large compliance at the moving platform. Hence the bending may be considered as a main factor.

The aim of optimization for PKMs is to minimize the global compliance and maximize the workspace volume. Therefore, it is a multiobjective optimization problem. The objective function is given as

$$\text{val} = \max(1/\mu + 1/\sigma + V), \tag{10.36}$$

where μ represents the mean value of the trace of the global compliance matrix of the PKMs; σ is its standard deviation; and V is the workspace volume of the PKMs.

The methods for determination of the workspace can be found in the literature, and the method used here is the inverse kinematics-based method [103].

In the cases being studied, there are many parameters and the complicated matrix computation making it difficult to write out the analytical expressions for each stiffness element. Using traditional optimization methods, only a few geometric parameters [3] can be handled because of 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. Global optima can be found only if the problem possesses certain convexity properties that essentially

guarantee that any local optima is a global optimum. In other words, conventional methods are based on point-to-point rule; it has the danger of falling in local optima.

Genetic algorithms (GAs) are powerful and broadly applicable stochastic search and optimization techniques based on evolutionary principles [5]. The genetic algorithms are based on population-to-population rule; it can escape from local optima. Therefore, genetic algorithms are the suitable for such optimization problems.

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. They are described in more detail as follows:

Chromosome representation. This is a basic issue for the GA representation; it is used to describe each individual in the population of interest. For the problem studied here, the chromosomes consist of the architecture parameters (coordinates of the attachment points, coordinates of the moving platform, vertex distributions at base and moving platform, platform height, etc.) and behavior parameters (actuator stiffness, actuated link stiffness, etc.) of the mechanisms.

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. In the study, the roulette wheel approach is applied.

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.

Population size. The population size represents the number of individuals or chromosomes in the population.

Mutation rate. The mutation rate is defined as the percentage of the total number of genes in the population; it determines the probability that a mutation will occur. The best mutation rate is application dependent but for most applications is between 0.001 and 0.1. In the case studied, mutation rate is 0.1.

Crossover rate. The best crossover rate is application dependent but for most applications it is between 0.80 and 0.95. For the case studied, crossover rate is 0.85.

Evaluation functions. Evaluation functions are subject to the minimal requirement that the function can map the population into a partially ordered set.

Simulations are carried out on the 3-dof PKM prototype built at the Integrated Manufacturing Technologies Institute of the National Research Council of Canada as shown in Fig. 10.2. The base platform is a triangular plate with a side length of 245.5 mm and the moving platform is another triangular plate with a side length of 139.7 mm. The guideway length is 95.25 mm and the sliding leg length is 215.9 mm. The guideway angle relative to the vertical direction is 20°. The three stiffness values of the prototype are $k_{\mathrm{t}} = 1.26\mathrm{e}^{10}\mathrm{N/m}$, $k_{\mathrm{b}} = 3.13\mathrm{e}^{10}\mathrm{N/m}$, $k_{\mathrm{a}} = 1.95\mathrm{e}^{7}\mathrm{N/m}$, and they are the same for the three subserial chains.

For the problem studied here, the chromosomes consist of the architecture parameters including coordinates of the attachment points, coordinates of the moving platform, link length, vertex distributions at base and moving platform, platform height, etc. Hence, the parameters selected for optimization are the following: $R_{\mathrm{p}}, R_{\mathrm{m}}, h_{\mathrm{m}}, \gamma$, where R_{p} is the radius of the moving platform; R_{m} is the radius of the middle plate; h_{m} is the height of the middle plate with respect to the base plate; γ is the rotation angle of the middle plate with respect to Cartesian Z-axis. And their bounds are

$$R_{\mathrm{p}} \in [60.96, 128.9] \text{ mm}, \quad R_{\mathrm{m}} \in [128.9, 304.8] \text{ mm},$$
$$h_{\mathrm{m}} \in [243.84, 365.76] \text{ mm}, \quad \gamma \in [-\pi/3, 0] \text{ rad},$$

Some other parameters are set as

$$P = 40,$$
$$G_{\max} = 100,$$

where P is the population and $G_{\max}$ the maximum number of generations.

One can rewrite the objective function (27) as

$$\mathrm{Val}(i) = W_{\mu}/\mu + W_{\sigma}/\sigma + W_r R_{\max} + W_z Z_{\mathrm{range}} \tag{10.37}$$

with $i = 1, 2, 3 \ldots 40$; $R_{\max}$ is the maximum radius of the workspace; Z_{range} is the range of movement in Cartesian Z-axis direction; and W_x is the weight factor for each entry. In this case, $W_{\mu} = 1, W_{\sigma} = 1, W_r = 0.1; W_z = 0.05$.

The objective functions are established and maximized 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, and it is used to increase working volume to a certain value and to minimize the mean value and standard deviation of the global compliance matrix.

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, since the GAs will force much of the entire population to converge to a single solution.

It is very difficult to optimize both global stiffness and workspace to their maximum values simultaneously, as larger workspace always leads smaller stiffness, and vice versa [159]. However, one can solve the problem by determining which item between workspace and stiffness is the dominant one for design and application, and maximize the dominant one while set the other one as a constant (but set as larger than the original). In this research, we set the workspace to a certain

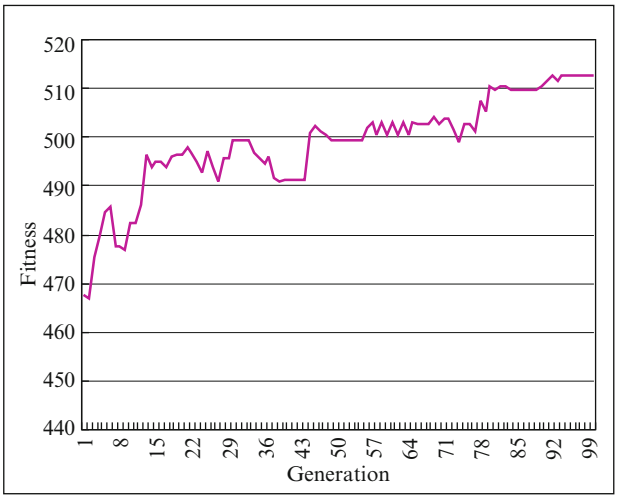


Fig. 10.5 The evolution of the performance of the NRC PKM

value, i.e., the radius of workspace is 304.8 mm, and then maximize the global stiffness. The algorithm converged at the 95th generation (Fig. 10.5). The optimized structure parameters are: $[R_p, R_m, h_m, \gamma] = [151, 259.8, 280.5, -0.1762]$, and the increased radius of the workspace is 304.8 mm; the range of movement along Z-axis is 304.8 mm; the sum compliance of the structure is 0.1568 mm/N.

The proposed methodology is implemented to design and optimization of the reconfigurable PKMs built at NRC-IMTI. The chromosomes consist of the architecture parameters including coordinates of the attachments at base and moving platforms, link length, platform height, coordinates of the moving platform. After optimization, the global stiffness is improved by a factor of 1.5, and workspace is increased 12%. A detailed example of industrial application is presented and analyzed in [170].

Java 3D is designed to be a mid to high-level fourth-generation 3D API [17]. What sets a fourth-generation API apart from its predecessors is the use of scene-graph architecture for organizing graphical objects in the virtual 3D world. Unlike the display lists used by the third-generation APIs (such as VRML, OpenInventor, and OpenGL), scene graphs can isolate rending details from users while offering opportunities for more flexible and efficient rendering. Enabled by the scene-graph architecture, Java 3D provides an abstract, interactive imaging model for behavior and control of 3D objects. Because Java 3D is part of the Java pantheon, it assures users ready access to a wide array of applications and network support functionality [136]. Java 3D differs from other scene graph-based systems in that scene graphs may not contain cycles. Thus, a Java 3D scene graph is a directed acyclic graph. The individual connections between Java 3D nodes are always a direct relationship: parent to child. Figure 10.6 illustrates a scene graph architecture of Java 3D for the NRC PKM. This test bed is a gantry system, which consists of an x-table and a 3-dof

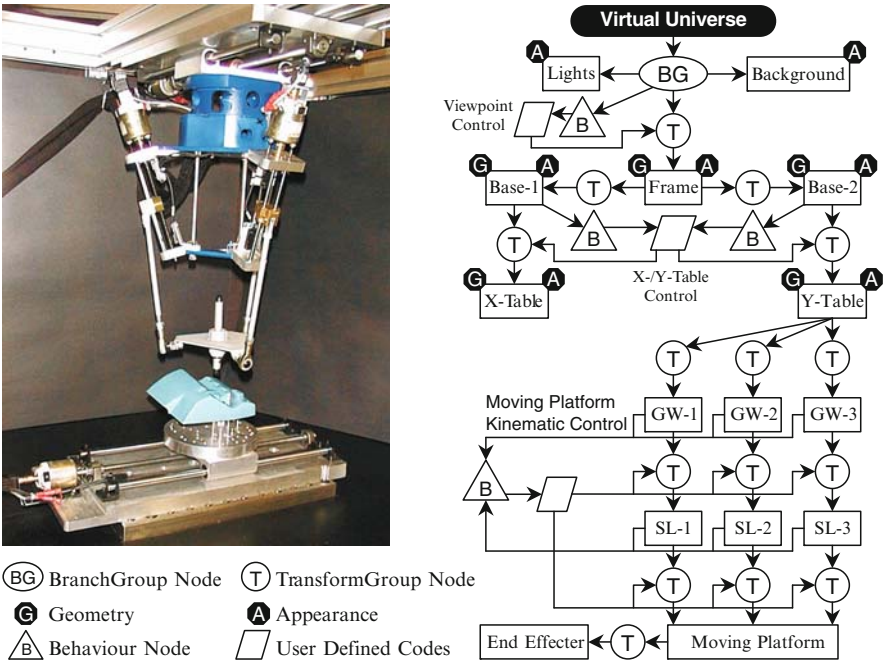


Fig. 10.6 Java 3D scene graph architecture for NRC PKM

PKM unit mounted on a y-table. The end effecter on the moving platform is driven by three sliding-legs that can move along three guide-ways, respectively.

As shown in Fig. 10.6, the scene graph contains a complete description of the entire scene with a virtual universe as its root. This includes the geometry data, the attribute information, and the viewing information needed to render the scene from a particular point of view. All Java 3D scene graphs must connect to a *Virtual Universe* object to be displayed. The *Virtual Universe* object provides grounding for the entire scene. A scene graph itself, however, starts with *BranchGroup* (BG) nodes. A *BranchGroup* node serves as the root of a branch graph of the scene graph. The *TransformGroup* nodes inside of a branch graph specify the position, the orientation, and the scale of the geometric objects in the virtual universe. Each geometric object consists of a *Geometry* object, an *Appearance* object, or both. The *Geometry* object describes the geometric shape of a 3D object. The *Appearance* object describes the appearance of the geometry (color, texture, material reflection characteristics, etc.). The behavior of the 3-dof PKM model is controlled by *Behavior* nodes, which contain user-defined control codes and state variables. Sensor data processing can be embedded into the codes for remote monitoring. Once applied to a *TransformGroup* node, the so-defined behavior control affects all the descending nodes. In this example, the movable objects (X-Table, Y-Table, and Moving Platform) are controlled by using three control nodes, for on-line monitoring/control and off-line simulation. As the Java 3D model is connected with its physical counterpart through the

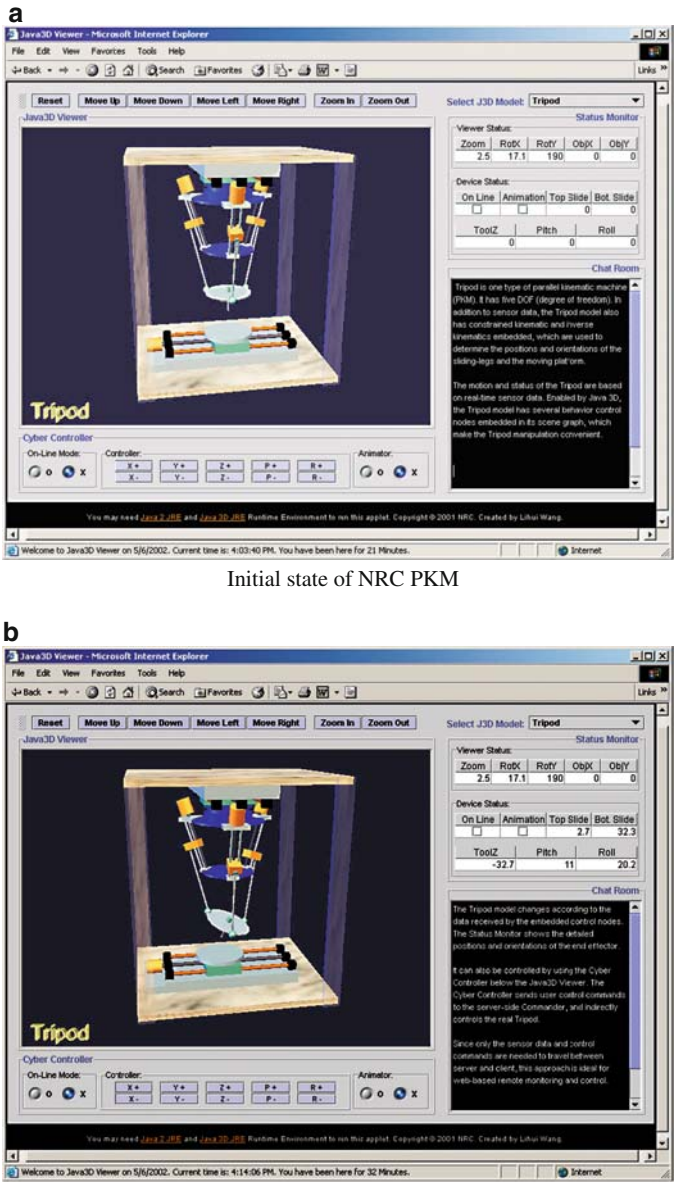


Fig. 10.7 Web-based remote monitoring and control

control nodes by low-volume message passing (real-time sensor signals and control commands, etc.), it becomes possible to remotely manipulate the real NRC PKM through its Java 3D model.

Web-based remote device monitoring and control are conducted by using the *StatusMonitor* and *CyberController*, which communicate indirectly with the device controller through an application server. In the case of PKM monitoring and control, they are further facilitated by the kinematic models, to reduce the amount of data traveling between web browsers and the PKM controller. The required position z_c and orientations θ_x, θ_y of the moving platform are converted into the joint coordinates $s_i (i = 1, 2, 3)$ by the inverse kinematics for both Java 3D model rendering at client-side and device control at server-side. The three sliding-legs of the PKM are driven by three 24V DC servomotors combined with three lead screws. Each actuator has a digital encoder ($1.25\mu m$/count) for position feedback. The position data $s_i (i = 1, 2, 3)$ of the sliding-legs are multicast to the registered clients for remote monitoring, while only one user at one time is authorized to conduct remote control. A sampling rate of 1 kHz is used for the case study. Figure 10.7 shows how the PKM is manipulated from one state to another within the proposed *Wise-ShopFloor* framework. The ToolZ (z_c), Pitch (θ_x), and Roll (θ_y) are the three independent variables that define the position and orientations of the moving platform of the PKM.

10.3 Conclusions

An integrated virtual environment for PKM design, analyze, validation, path planning, and remote control is proposed in the article, it can be used in the early stage for conceptual design of PKM and embodiment design stage with the CAD model and simulation. An example is implemented under the system. It is shown that the system is very efficient and generic for most of the PKM design and analyze.