NISTIR 6836

Tolerance Synthesis Scheme

U. Roy
N. Pramanik
H. Wang
R. Sudarsan
R. D. Sriram
K. W. Lyons

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 6836

Tolerance Synthesis Scheme

U. Roy
N. Pramanik
H. Wang
R. Sudarsan
R. D. Sriram
K. W. Lyons

June 03

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary
TECHNOLOGY ADMINISTRATION
Phillip J. Bond, Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Arden L. Bement, Jr., Director

Tolerance Synthesis Scheme

U. Roy, N. Pramanik and H. Wang
Knowledge Based Engineering Laboratory
Dept. of Mechanical , Aerospace, Manufacturing Engineering,
Syracuse University Syracuse, NY 13244-1240
Email: {uroy, pramanik, hawang}@ecs.syr.edu

R. Sudarsan[1], R. D. Sriram and K. W. Lyons
Design and Process Group
Manufacturing Systems Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
[1] George Washington University
Email: {sudarsan, sriram, klyons}@cme.nist.gov

Abstract

The objective of this report is to identify representations and issues for the generic assembly information model and its use in a proactive tolerance synthesis scheme. It is proposed to use the small displacement torsors (the screw parameters) as a mathematical tool for analyzing the causal relationship between variations in parts' shapes and sizes (due to the small allowances permitted by tolerance specifications), and their effects on relative positioning on adjacent parts. For a given assembly, the allowable geometric variations of assembled parts, the assembly specification and the desired functional requirements have been characterized by finite sets of equality and inequality constraints. An optimization process is then formulated to study the tolerance synthesis and analysis problems in the assembly.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

This report gives details of work on tolerance synthesis subsequent to our earlier work [ROY01]. For tolerance synthesis and analysis, we principally need a detailed description of the "kinematic functions" of the assembly, by which we mean those functions defined essentially by the location, size and shape (form) of associated mating features. These kinematic functional specifications are not directly provided by the customer or by early specifications of the desired product / assembly function. They are slowly evolved with the assembly as the design takes concrete shape and size in the later phases of the conceptual design. Tolerance synthesis and analysis needs an exhaustive functional analysis mechanism to make sure that the identified functional requirements between the mating components of the assembly are met and are suitably described in the form of critical toleranced dimensions / size / sizes / forms or in the form of toleranced gaps. The tolerance representation procedure uses the small deviation torsor scheme [BAL00] to represent the variations associated with each feature of a part in the assembly. The tolerance representation and synthesis scheme has been elaborated in the following Section 2. Section 3 discusses the formulation of cost functions while Section 4 presents a detailed discussion on the several constraint generation procedures for establishing the optimization process as described in section 5. Section 6 briefly describes the implementation issues with an example.

2. Tolerance Synthesis Scheme

Based on the considerations discussed in earlier works, we propose the following steps for the tolerance synthesis scheme:

1. For each component of the artifact, develop the kinematic Degrees Of Freedom (DOF) requirement between the mating features and decide the intrinsic displacement and deviation of the surface from the nominal surface as torsor components.

2. For each pair of connected features, establish the gap between the features as a gap torsor taking into account the indeterminate and/or blocked degrees of freedoms.

3. Generate assembleability constraints by considering all chains of connected components as a kinematic link (either closed loop or path between two datum features) and aggregate the torsors along the path with reference to a global frame of reference. Each loop would generate a maximum possible of six equations binding the deviation parameters associated with all the related surfaces. These sets of constraints would form a deviation for the possible geometric variations of the connected components.

4. Generate functional requirement constraints based on the specified functions/objectives to be fulfilled by the artifact. This will restrict the range of values of the deviations within which a component must remain to fulfill specific functional requirements.

5. Generate manufacturing / production cost models for each component (based on a suitable knowledgebase for manufacturing/ production costs) for different machining operations, and

formulate a minimization scheme by considering the sum of costs for machining all the parts as the cost function subject to the constraints defined in steps 3 and 4.

6. Use standard mathematical tools to solve the above optimization scheme to find the deviation parameters for each component. Convert the deviation parameters associated with each surface into appropriate tolerance zones through a tolerance mapping scheme.

Let us take an example to elaborate the steps described above for the tolerance synthesis process. The example has been taken from the design evolution solution in our earlier work [ROY01] where we illustrated the design evolution process by taking an example of designing a device for transmitting torque with changing rotational speed. We used a sample artifact library and arrived at a few tentative design solutions that matched the functional requirements. We take one of the solutions from the above design example and assume the tentative sizing of the components.

The gear-subassembly consists of three parts: a spur gear, a shaft with a keyhole and a key. The geometric models of these three parts, including their face-level mating relationships and behavioral models have been shown in Figure 1, Figure 2.

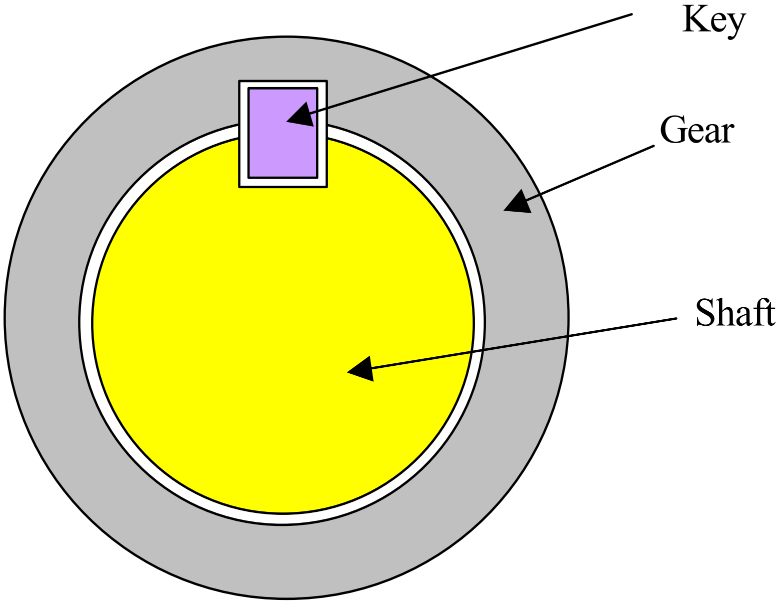


Figure 1: A spur gear sub-assembly

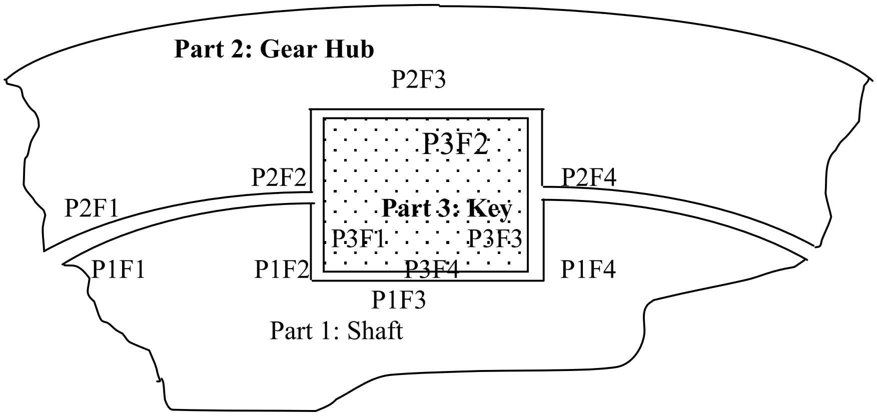


Figure 2: Mating surfaces of the gear sub-assembly

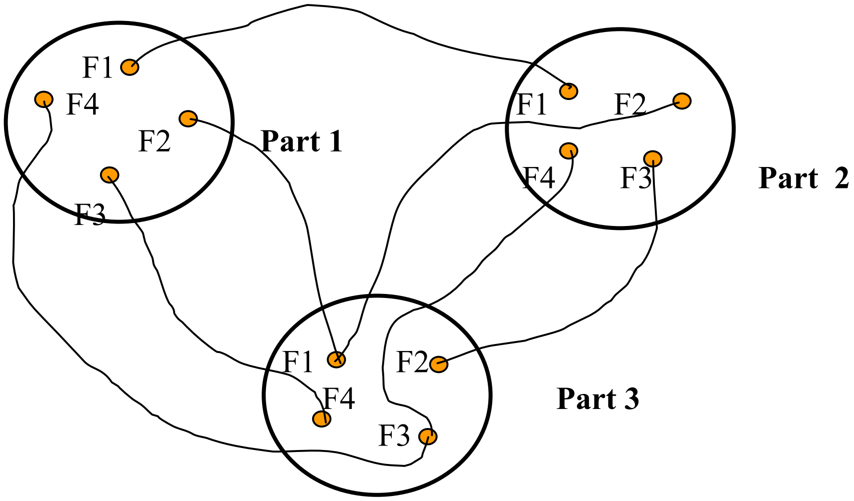


Figure 3: Assembly Graph of the gear sub-assembly

After the torsors for each of the mating surfaces have been defined, we consider the assembly graph of the artifact (Figure 3) and enumerate closed loops of independent paths for forming equations that represent various possible geometric configuration hulls for the artifact. Thus, for example, a typical closed loop from the Figure 3 would be represented by: T[F1/P1] → Gap[F1/P1,F1/P2] → T[F1/P2,F4/P2] → Gap[F4/P2,F3/P3] → T[F3/P3] → Gap[F3/P3,F4/P1] → T[F4/P1] → T[F1,P1]. (Please see appendix –A for details). The aggregation of these torsors (transforming into global reference frame, as needed) should then be equated to a null torsor. This will give six equations connecting the components of the torsors associated with the loop. The gap between the mating features will be used as constraints to satisfy specific design functional requirements as well as kinematic requirements.

For computations in each loop, the local variations are to be converted to a global coordinate system by transformation of the corresponding torsors in the local frame. Thus, in the case of this example, in, we can take the center of the inner shaft as the reference origin and express all the components in terms of vectors emanating from that point.

While traversing these loops, the gap between the two mating features would play a vital role in deciding the type of interface between the two mating features. The gap torsor is required to model the kinematic/ functional specifications specific to the given mating conditions. However, we also need to assess the possibility of any interference between the mating features by applying the torsor on some specific points on the two mating features. This would require studying the torsor equations applied on those strategic points. These strategic points could be some inspection points to be used in future inspection of the features. We have kept points of kinematic interest in the artifact class definition so that we can create those points as required on specific surfaces/features. As an example, in case of a rectangular feature, we can select the four furthest points (corner points) as the desired points. For circular features, we can keep eight points uniformly placed on the circumference (Figure 4). It may be mentioned that additional points as desired may be introduced at any stage to improve accuracy; however, the number of equations and complexity of the problem would increase with each additional point.

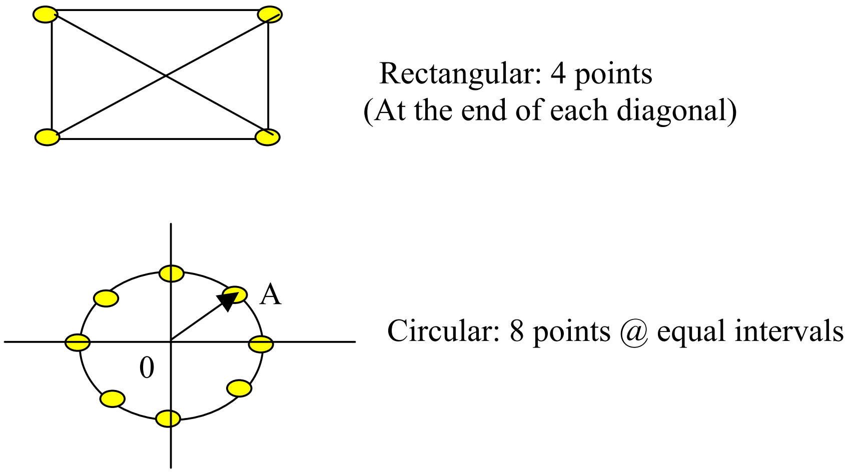

Rectangular: 4 points
(At the end of each diagonal)
A
0
Circular: 8 points @ equal intervals

Figure 4: Points of interest on features.

After all the loops have been considered and corresponding equations have been formed, we will get a system of equations that will define the possible geometric variations of the artifact configuration. Depending on the type of mating and functional requirements for each mating pairs of surfaces, these sets of equations may be under or over constrained. In other words, there may be too few equations connecting too many variables or vice-versa. These will constitute the constraints of the optimization process for tolerance synthesis.

The proposed optimization scheme could be now summarized as below:

Minimize: *Cost of manufacturing*

Subject to: *Constraints associated with the geometry of the product as imposed by the kinematic requirements between mating features and the constraints associated with functional requirements.*

In the following sections, we will detail:

i) Model for cost of manufacturing
ii) Constraints related to Performance/Functional requirements
iii) Constraints related to Assembleability/Kinematic Mating
iv) Constraints related to mapping of tolerance to feature deviation parameters, and
v) Optimization process, and other implementation issues

3. Generation of Constraints

We have considered two types of constraints associated with the tolerance synthesis process. Constraints arising to meet the functional requirements and constraints associated with the kinematic mating requirements / assembleability. Since both the constraints play a crucial role in the tolerance synthesis process, we have elaborated each process separately in the following sub-sections.

3.1 Constraints Related to Performance/ Functional Requirements

Functional requirements are design requirements specified explicitly as well as implicitly by the user and designers to meet a certain desired performance from the artifact/assembly. In terms of low-level tolerance and deviation parameters the functional requirements would translate into constraint i.e. functions establishing some relationship between parameters. These relations could be of two types: equality and inequality.

The functional requirements can be thought of in two different ways: as a goal satisfaction to minimize a penalty cost for deviating from the design specification or to convert the functional requirements as a set of constraints on the tolerance zone. Since the functional requirements are developed based on customer's specification and designer's technical requirements, it is desirable to satisfy the product specification / functional requirements as necessary constraints rather than a goal satisfaction process. Thus, we have decided to treat the functional requirements as constraints on the geometric variations of the mating features of the components and considered manufacturing cost as a minimization goal to arrive at an optimal tolerance design. We need to develop a generic process to transform the functional requirements associated with a design into corresponding set of constraints. The functional requirements mainly translate into algebraic inequalities representing constraints of the form:

$$G(x) \leq 0,\ x \in R^n. \qquad (3.1\text{-}1)$$

Functional requirements specified between two mating features could be represented by using the gap torsor parameters associated with each mating. A Functional requirement can be thought of in two different ways: either as a goal satisfaction to minimize a penalty cost for deviating from the design specification or as a set of constraints on the tolerance zone.

We are working to evolve a generic approach to transform the functional requirements associated with a design into a corresponding set of constraints. We have already introduced some treatment of functional requirements and constraints in our earlier work [ROY01] and we need to extend further the work done to transform the constraint equations as functions of the deviation parameters.

3.2 Constraints related to Assembleability / Kinematic Mating

The constraints arising from the kinematic requirements between the mating surfaces to meet the requirement of transmission and/or blocking the desired flow of energy / force / fluent / information through the mating area could be treated as kinematic constraints. The torsor scheme for the modeling of the small geometric variations of connecting features / faces of an artifact has been adopted for the present study and the scheme has been elaborated in our earlier report [ROY00b].

These kinematic assembleability constraints are established by considering the sum of the torsor components along each independent paths/loops that exist between datum surfaces. The torsor summation is carried out using the torsor transformation rules by transferring the effect to a global point on the datum surface. In our assumed model with small displacements / deviations, the assembleability constraints become linear and can be put into the compact form: $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A} = [\boldsymbol{a_{ij}}]$ an $m \times n$ matrix, $\boldsymbol{a_{ij}} \in \boldsymbol{R}$, $\boldsymbol{x} \in \boldsymbol{R^n}$, $\boldsymbol{b} \in \boldsymbol{R^m}$, $\boldsymbol{m}$ is the number of constraints and in general $\boldsymbol{m}$ is less than number of deviation parameters $\boldsymbol{n}$. We would need to work further to establish an automated process to generates the $\boldsymbol{A}$ matrix from the specified assembly configuration.

3.2.1 Derivation of the A *matrix*

Since we will the need torsor transformations in the derivation of the *A* matrix, the torsor transformations are presented below (Figure 5).

Transformation of Torsors

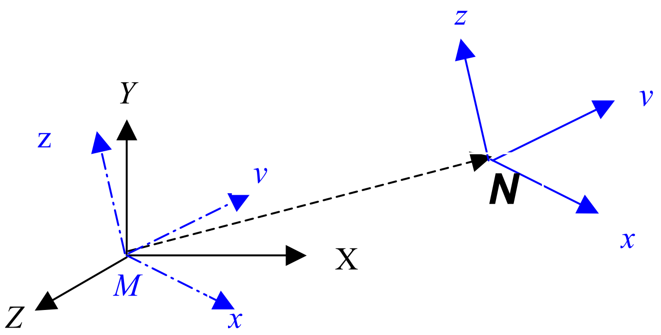


Figure 5: Transformation of Torsors – Local to Global

A small deviation torsor (SDT) $\boldsymbol{T_N}$ defined at a point $\boldsymbol{N}$ in the local coordinate system *(x, y, z)* with six components $(\theta_x,\ \theta_y,\ \theta_z,\ \delta_x,\ \delta_y\ ,\ \delta_z)$ (where θ denotes rotational and δ denotes linear components of movements) transforms to $\boldsymbol{(T_N)_{ML}}$ (where *L* denotes local coordinate system) in the same local coordinate system shifted to a new point **M** is be given by: $\boldsymbol{(T_N)_{ML}} = (\theta_x,\ \theta_y,\ \theta_z,\ \ \delta_x + (\theta_y{*}\Delta z\ \text{-}\ \theta_z\ {*}\Delta z),\ \delta_y + (\theta_z\ {*}\Delta x - \theta_x\ {*}\Delta z),\ \delta_z + (\theta_x\ {*}\Delta y - \theta_y\ {*}\Delta x))$. This could be written in compact notation as: $\boldsymbol{(T_N)_{ML} = DT_N}$ where $\boldsymbol{D}$ is the 6x6 transformation matrix given by

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\Delta z & \Delta y & 1 & 0 & 0 \\ \Delta z & 0 & -\Delta x & 0 & 1 & 0 \\ -\Delta y & \Delta x & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (3.2.1\text{-}1)$$

where $\Delta = (\Delta x,\ \Delta y,\ \Delta z)^T = (X_N\text{-}X_M,\ Y_N\text{-}Y_M,\ Z_N\text{-}Z_M)^T$ is the vector from point *M* to *N* in global coordinate system *(X, Y, Z).*

After the above transformation, the components of torsor $\boldsymbol{(T_N)_{ML}}$ are still in the local coordinate system *(x,y,z).* Let the local coordinate system *(x,y,z)* be defined by the three unit vectors $x = (k_{xX},\ k_{xY},\ k_{xZ}),\ \ y = (k_{yX},\ k_{yY}, k_{yZ})$ and $z = (k_{zX},\ k_{zY},\ k_{zZ})$ in the global coordinate system, in other words, the local coordinate system could be represented by the [*K*] matrix given by:

$$[K] = \begin{bmatrix} k_{xX} & k_{yX} & k_{zX} \\ k_{xY} & k_{yY} & k_{zY} \\ k_{xZ} & k_{yZ} & k_{zZ} \end{bmatrix}. \qquad (3.2.1\text{-}2)$$

The components of $\boldsymbol{(T_N)_{ML}}$ could then be converted to the global system by the following transformation (applicable for both the linear movement and the small rotations in the sense of SDT) :

$$\theta_X = \theta_x * k_{xX} + \theta_y * k_{yX} + \ \theta_z * k_{zX,,} \qquad (3.2.1\text{-}3)$$

$$\theta_Y = \theta_x * k_{xY} + \theta_y * k_{yY} + \ \theta_z{*}\ k_{zY}\ , \qquad (3.2.1\text{-}4)$$

$$\theta_Z = \theta_x * k_{xZ} + \theta_y * k_{yZ} + \ \theta_z{*}\ k_{zZ\ ,} \qquad (3.2.1\text{-}5)$$

$$\theta_G = [\theta_X\ \ \theta_Y\ \ \theta_Z]^T = [K][\ \theta_x\ \ \theta_y\ \ \theta_z]^T,$$

$$\theta_G = [K]\ \ \theta_{L.} \qquad (3.2.1\text{-}6)$$

where the subscripts $\boldsymbol{_G}$ and $\boldsymbol{_L}$ refer to global and local coordinate systems respectively.

$$\delta_X = [\delta_x + (\theta_y\ {*}\Delta z\ \text{-}\ \theta_z{*}\Delta y)] * k_{xX} + [\delta_y + (\theta_z{*}\Delta x - \theta_x{*}\Delta z)] * k_{yX} + [\delta_z + (\theta_x{*}\Delta y - \theta_y\ {*}\Delta x)] * k_{zX\ ,} \quad (3.2.1\text{-}7)$$

$$\delta_Y = [\delta_x + (\theta_y\ {*}\Delta z\ \text{-}\ \theta_z{*}\Delta y)] * k_{xY} + [\delta_y + (\theta_z{*}\Delta x - \theta_x{*}\Delta z)] * k_{yY} + [\delta_z + (\theta_x{*}\Delta y - \theta_y{*}\Delta x)] * k_{zY,} \quad (3.2.1\text{-}8)$$

$$\delta_Z = [\delta_x + (\theta_y\ {*}\Delta z\ \text{-}\ \theta_z{*}\Delta y)] * k_{xZ} + [\delta_y + (\theta_z{*}\Delta x - \theta_x{*}\Delta z)] * k_{yZ} + [\delta_z + (\theta_x{*}\Delta y - \theta_y{*}\Delta x)] * k_{zZ,} \quad (3.2.1\text{-}9)$$

$$\boldsymbol{\delta_G} = [\delta_X\ \ \delta_Y\ \ \delta_Z]^T = [\boldsymbol{K}]\ (\ [\delta_x\ \ \delta_y\ \ \delta_z]^T\ +\ [\Delta x\ \ \Delta y\ \ \Delta z]^T \times [\theta_z\ \ \theta_y\ \ \theta_z]^T). \qquad (3.2.1\text{-}10)$$

where × is the vector cross product.

$\boldsymbol{\delta_G} = [\boldsymbol{K}]\ (\delta_L + \Delta \times \theta_L)$ *where* $(\delta_L = [\delta_x\ \delta_y\ \delta_z]^T,\ \Delta = (\Delta x, \Delta y, \Delta z)^T,\ \theta_L = [\theta_x\ \theta_y\ \theta_z]^T)$.

Combining the two transformations, we can write the result as: (where subscript G refers to global)
$\boldsymbol{(T_N)_{MG} = (\theta_G, \delta_G)}$

$$(T_N)_{MG} = ([\boldsymbol{K}][\theta_x\ \theta_y\ \theta_z]^T,\ [\boldsymbol{K}]\ ([\delta_x\ \delta_y\ \delta_z]^T + [\Delta x\ \Delta y\ \Delta z]^T \times [\theta_x\ \theta_y\ \theta_z]^T). \qquad (3.2.1\text{-}11)$$

The above relation could be written as

$$\boldsymbol{(T_N)_{MG} = K2\ (T_N)_{ML}}, \qquad (3.2.1\text{-}12)$$

where ***K2*** is the 6 x 6 transformation matrix generated from (3.2.1-10).

We will have to use above results for transforming each torsor (defined in the local coordinate system) in a loop to the global reference point before they can be combined. For each loop/cycle in the assembly graph (parts connected through mating of features), we would have (at most) six equations by summing all the deviations along a cycle/loop to zero (or a pre-assigned quantity). In this derivation, we will use following notation. Without any loss of generality and without any ambiguity we will assume that each loop/path could be enumerated as a linked list ***L*** defined by:

$$\boldsymbol{L = \{\ (p, f),\ g,\ [(f, p, f), g]\ldots, (f, p)\}}, \qquad (3.2.1\text{-}13)$$

where ***p*** = part, ***f*** = feature, ***g*** = gap between mating features. Symbolically this can be represented as shown in Figure 6.
Items within [ ]… could be repeated zero or more times as required to traverse a loop/path, *(p, f)* is the starting point (from_part, from_feature), (f, p) the end point (to_feature, to_part), *[(f, p, f), g]* are intermediate connections (to_feature, part, from_feature), gap along the path to the next part.

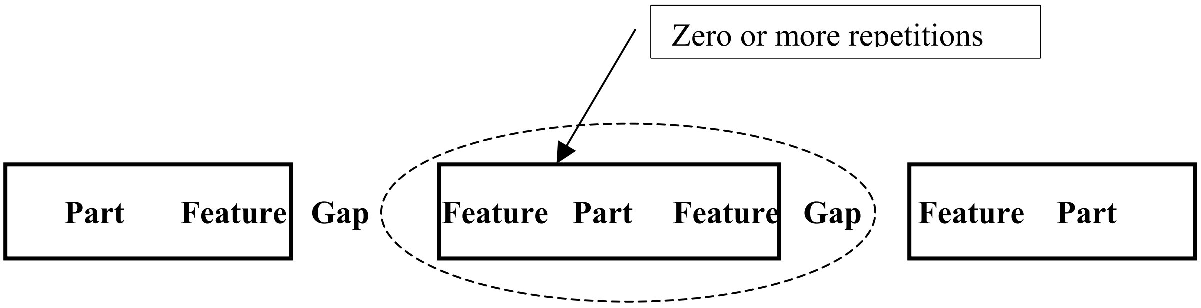


Figure 6: Loops/Paths in an assembly

In case of a closed loop, the last part would be the same as the first part. Both the starting and the final parts for open loops should be terminals (datums). In the case of closed loops, the summation of the torsors would be null (0), whereas for an open path, the summation would be equal to a torsor (possibly non-null) that has to be established from known functional specifications between the two terminal parts. We will designate this quantity as ***b***. We will use ***T*** to indicate torsor with index ***p, f, g*** for parts, features and gaps respectively.

P_j = Part number ***j***, ***j =1 to n***, ***n*** = number of parts in the loop,

F_k = Feature ***k*** of part P_j , ***k = 1 to m***, ***m*** = number of features of part P_j,

T_p = 6-component displacement torsor of the part ***P*****,** defined at a suitable point ***O*** inside the part in a local coordinate system,

T_f = 6-component deviation torsor of the feature ***k*** (of some part P_j)**,** defined at a suitable point ***N*** on the feature in a local coordinate system with *x*-axis along the outward (away from the material) normal and *y* and *z* are the other two axes mutually perpendicular to the *x*-axis,

T_g = 6-component gap torsor defined at a mating between two features of two parts.

With the above notation, we can write the desired equations for each loop, with implied local to global transformations applied to each torsor as viewed from any convenient point (typically, at the center of the first part where the loop begins), as:

$$(T_p + T_f) + T_g\ [+ (T_f + T_p + T_f + T_g)]\ldots + (T_f + T_p) = b. \qquad (3.2.1\text{-}14)$$

Summing over the entire loop, we can write,

$$\sum_{j \in EntireLoop} K_j * (\delta_j + \Delta x_j \times \theta_j) = b_\delta, \qquad (3.2.1\text{-}15)$$

$$\sum_{j \in EntireLoop} K_j * \theta_j = b_\theta, \qquad (3.2.1\text{-}16)$$

where,

K_j is the local to global transformation matrix (eqn. (3.2.1-2)) for feature j,
$\delta_j = [\delta_x \ \ \delta_y \ \ \delta_z]^T{}_j$ and $\theta_j = [\theta_z \ \ \theta_y \ \ \theta_z]^T{}_j$ are the two vectors formed from the linear and rotational components respectively of the deviation torsor of the feature ***j.***

The scheme for the representations and transformations of the torsors have been elaborated in Appendix – A with an example of an artifact with three blocks.

Since the proposed scheme uses six small displacement parameters to represent the possible geometric deviations from the design / nominal surface, we need a mechanism to express standard tolerances in terms of these torsor parameters. We consider some engineering features/surfaces and develop the tolerance to deviation mappings in the following section.

3.3 Constraints related to mapping of tolerance to deviation parameters

The category of relations arising from the mapping of the deviation parameters of the features to corresponding tolerance values is of the generic non-linear inequality ("less than or equal to" type) and takes the form: $\boldsymbol{F_{DP}^{k}(x) \leq F_{Tol}^{k}(t)}$, where $\boldsymbol{F_{DP}}$ is some function of the deviation parameters $x \in \boldsymbol{R^n}$, $\boldsymbol{F_{Tol}}$ is some function of the tolerance parameters $\boldsymbol{t}$, and $\boldsymbol{k \in N^+}$ is an index. Exact forms of $\boldsymbol{F_{DP}^{k}(x)}$ and $\boldsymbol{F_{Tol}^{k}(t)}$, depend on the nature of the feature/surface. For a planar surface, the above relationship becomes a set of linear constraints. The interesting feature in the above relationship is the separable nature of the equation in terms of the variables $\boldsymbol{t}$ and $\boldsymbol{x}$. We would like to explore the possibility of taking advantage of the separable nature of the mapping to define the deviation hull in a semi-analytical way, if possible. The linear nature of the equations, however, does not exist when we consider other non-planar features. So far, we have been successful in establishing the required relationships for the features: rectangular planar, circular planar, cylindrical, and spherical. Following sections give the detailed results.

3.3.1 Representation of Geometric Deviations of Features of a Part

The geometric variations of the features are defined in terms of deviations of their independent degrees of freedom. In this approach, at the most six independent parameters would be used to represent a six-component vector corresponding to the possible six degrees of freedom. In actual case, depending on the type of surface (planar, cylindrical, etc) some of the degrees of freedom are 'free' (variations along which keeps the surface invariant) and thus, in general, less than six deviation parameters are required. For example, for a planar surface, rotation along the outward normal to the surface and movements along the two axes on the surface keep the feature invariant and hence a planar surface requires three deviation parameters (axial movement, two rotations along the two transverse axes).

In order to make the process of converting the functional requirements as well as kinematic requirements into a set of constraints in a generic manner, we need a procedure to represent the variations associated with each part in terms of a set of generic parameters of the mating features/surfaces. Assuming small displacements/deviations of these features due to manufacturing inaccuracies/other defects, the actual surface of a feature deviates from the theoretical (nominal) surface slightly. Such variations could be systematically treated by considering variations of six small parameters defined at a point on the surface. The six parameters are: three linear translations and three rotations along the three axes of the local orthogonal coordinate system corresponding to the six degrees of freedom. These six small displacement parameters have been called screw parameters or torsors.

Since the proposed method uses six small displacement variation parameters to represent the possible geometric deviations from the design / nominal surface, we need a mechanism to express geometric variations in terms of these torsor parameters. It has been mentioned above that not all the six parameters for each surface may be uniquely defined and we may not need to know all the six parameters in specific cases. We plan to consider each type of engineering surfaces and formulate methods to represent the tolerance values as functions of the parameters. This task require a mapping of the deviation parameters to geometric tolerances. So far, we have completed mappings for Planar, Cylindrical, and Spherical features. Conical, Toroidal, and Helical cases will be taken as future work.

In order to give a meaningful interpretation to the assemblability of an artifact in terms of tolerance specifications as per industry standard codes and practices like ASME Y14.5.1M-1994, ISO, we need to consider relationship between the tolerance zones (or virtual condition boundaries) and the deviation parameters of a feature. By deviation parameters, we mean the representation of the possible deviation of a feature by using (at most) six independent parameters corresponding to the six degrees of freedom of a feature. In general, the deviation parameters to tolerance mappings are not one-to-one and can only be represented as inequalities involving functions of the tolerance parameters and the deviation parameters. Mappings developed from these are, in general, a set of non-linear inequalities. However, for some specific cases, like the planar surface, the inequalities are linear, and represent a diamond shaped zone by the intersection of inequalities, each of which are half-spaces in the D x T (D: Deviation, T: Tolerance) space [BAL98].

There are four types of tolerances used in the industry: size tolerance, form tolerance, orientation tolerance, and location/position tolerance. These tolerances are sometimes specified with material conditions as modifiers. There are three such material conditions: MMC (Maximum Material Condition), LMC (Least Material Condition) and RFS (Regardless of Feature's Size). Since we assume that the shape of a toleranced feature remains similar to the nominal feature (for example, a plane remains a plane, a cylinder remains a cylinder, etc.), the variation of position and orientation due to form tolerance is very small [Gil91] and in this study, we consider only the size tolerance, orientation tolerance, and positional tolerance.

A geometric tolerance applied to a feature of size and modified to MMC will establish a *virtual condition boundary (VCB)* [4] outside of the material space adjacent to the feature. The feature shall not cross this VCB. Likewise, a geometric tolerance applied to a feature of size and modified to LMC establishes a VCB inside the material and the feature shall not cross the VCB. This translates to the following relationships [DRA99]:

Modified to MMC:
For an internal feature of size:
MMC virtual condition = MMC size limit – geometric tolerance; (3.3.1-1)
For an external feature of size:
MMC virtual condition = MMC size limit + geometric tolerance; (3.3.1-2)

Modified to LMC:
For an internal feature of size:
LMC virtual condition = LMC size limit + geometric tolerance; (3.3.1-3)
For an external feature of size:
LMC virtual condition = LMC size limit – geometric tolerance. (3.3.1-4)

If a geometric tolerance applied to a feature of size is neither modified to MMC nor LMC, by definition of standards such as ASME Y14.5, it is modified to RFS. In this case, instead of VCB, the tolerance specification will generate a tolerance zone into which the derived element will not interfere.
A VCB or tolerance zone represents the basic intention of the designer that each point on the toleranced feature/derived element should remain within (or outside) this boundary/zone. Hence, the VCB or tolerance zone would be ideally suitable for representing the relational limits on the deviation parameters associated with the deviation of the feature from its nominal position. Since the deviation parameters for a feature could be used to define the deviation of all points on a feature (by employing

[4] **Virtual condition boundary**: A constant boundary generated by the collective effects of a size feature's specified MMC or LMC material condition and the geometric tolerance for that material condition.

one or more independent parameters), the tolerance specifications could be thought of a set of limits for the deviation parameters and vice-versa. These relationships could be used for tolerance synthesis as well as for tolerance analysis including checking the assemblability in the worst-case scenario.

In the following sections of this paper, we consider the mapping for size tolerance, positional tolerance, and orientation tolerance for four types of features: planar, cylindrical, spherical, and conical. For the planar case, there is no material condition as it is a non-size feature. For the cylindrical feature we have two cases: MMC and RFS and for the spherical feature we have MMC and for conical feature we have the size tolerance only.

We use the following basic steps to convert a tolerance specification into a set of inequalities in the deviation parameters:

a. Generate the intrinsic torsor and the deviation torsor for the feature by eliminating the deviation parameters that are invariant for the feature to reduce the degrees of freedom. This reduces the number of deviation parameters needed to represent the variation of the feature. For example, for a cylindrical feature, axial movement and axial rotation are two invariants. Hence, we do not need these two parameters.

b. Generate a VCB (or a tolerance zone) based on the tolerance specification. In this step, we compute the size of the VCB or the tolerance zone for restricting the variation of the feature or derived element respectively.

c. In case of VCB, take an arbitrary point on its nominal surface of the feature in a parametric form and transform it to a new position by applying the effect of the deviation torsors. For example, for a cylindrical feature a point *P* on its nominal surface is represented by two parameters (θ, z); $P = (r\cos\theta\ ,\ r\sin\theta, z)$, where $\theta \in (0, 2\pi)$ and $z \in (0, L)$, *L*=Length of the cylinder. In the case of tolerance zones, transform the derived element, such as center plane, center axis etc., to a new position by applying the effect of the deviation torsor.

d. Eliminate the free parameters, (such as the (θ, z) mentioned in step c above), by applying the condition that the extreme points of the transformed position should remain within the VCB or that the derived element should remain within the tolerance zone. This step generates a set of inequalities connecting the deviation parameters with the tolerance specification.

The procedure for applying the above four steps will be clear in subsequent sections where we generate the mappings for the basic features under various material conditions

The symbols used in the following sections, unless otherwise specified, are defined as below:

TU = Upper limit, size tolerance,
TL = Lower limit, size tolerance,
T_P = Positional tolerance,
T_V = Perpendicularity tolerance,
$(\Delta\theta_x, \Delta\theta_y, \Delta\theta_z, \Delta x, \Delta y, \Delta z)$ = six components of the deviation torsor.

3.3.2 Planar Features

For each surface/feature, a local coordinate system (LCS) is defined and the deviation parameters are defined in that LCS. For a planar surface, the LCS is: the z-axis is the outward normal (emanating from the material side of the feature) and (x, y) are local orthogonal coordinates on the plane, so that the equation for the nominal surface (plane) is given by $z = 0$. For this plane the deviation parameters of the small deviation torsor (SDT) are: $(\Delta\theta_x, \Delta\theta_y, 0, 0, 0, \Delta z)^T$.

Size Tolerance 1: Size tolerance with respect to a datum.

Case 1: Rectangular *(2a x 2b)* planar surface. (Figure 7)

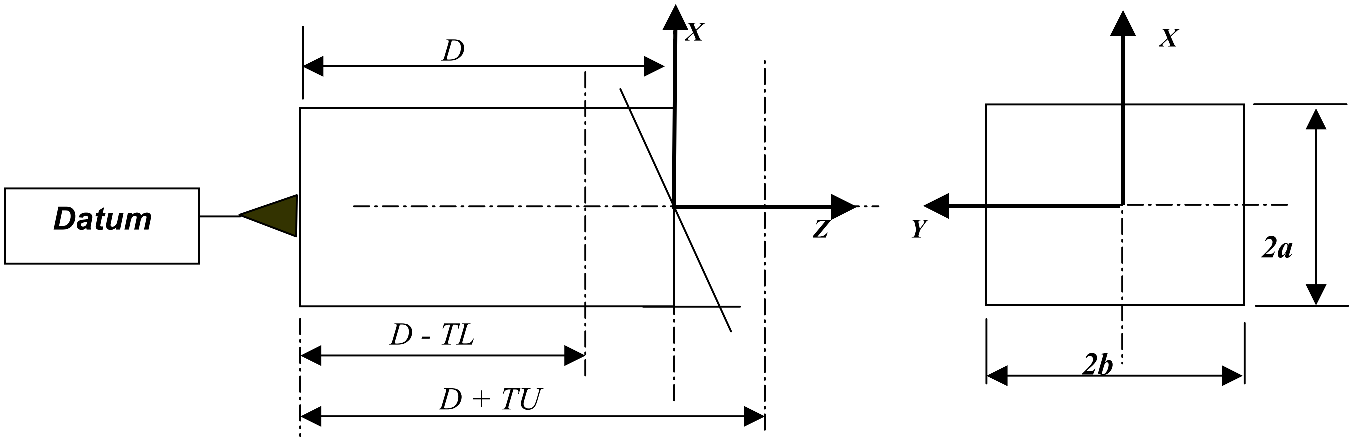


Figure 7: Planar feature (rectangular) with size tolerance

For the above feature, deviation parameters are $(\Delta\theta_x, \Delta\theta_y, 0, 0, 0, \Delta z)$, and the tolerance parameters are (TU, TL) and the SDT is: $D_d = \{\Delta\theta \ \ \Delta d\}^T = \{\Delta\theta_x \ \ \Delta\theta_y \ \ 0 \ 0 \ 0 \ \Delta z\}^T$ (3.3.2-1)

Since this is a rectangular planar feature, the four corners with nominal coordinates $(\pm a, \pm b, 0)$ will have the maximum deviation due to the effect if D_d. Transforming D_d on to these four points we get, $D' = D_d + ((\pm a, \pm b, 0) \times (\Delta\theta_x \ \ \Delta\theta_y \ \ 0)^T$ which gives deviations for the four points as:

$$D_1 = \{\Delta\theta_x \ \ \Delta\theta_y \ \ 0 \ \ 0 \ \ 0 \ \ \Delta z + a*\Delta\theta_y - b*\Delta\theta_x\}^T,$$
$$D_2 = \{\Delta\theta_x \ \ \Delta\theta_y \ \ 0 \ \ 0 \ \ 0 \ \ \Delta z + a*\Delta\theta_y + b*\Delta\theta_x\}^T,$$
$$D_3 = \{\Delta\theta_x \ \ \Delta\theta_y \ \ 0 \ \ 0 \ \ 0 \ \ \Delta z - a*\Delta\theta_y + b*\Delta\theta_x\}^T,$$
$$D_4 = \{\Delta\theta_x \ \ \Delta\theta_y \ \ 0 \ \ 0 \ \ 0 \ \ \Delta z - a*\Delta\theta_y - b*\Delta\theta_x\}^T.$$

Comparing the z-component with the specified tolerance value (TU, TU), we get following four inequalities for the desired mapping:

$$-TL \leq \Delta z + a*\Delta\theta_y - b*\Delta\theta_x \leq TU, \qquad (3.3.2\text{-}2)$$
$$-TL \leq \Delta z + a*\Delta\theta_y + b*\Delta\theta_x \leq TU, \qquad (3.3.2\text{-}3)$$
$$-TL \leq \Delta z - a*\Delta\theta_y + b*\Delta\theta_x \leq TU, \text{ and} \qquad (3.3.2\text{-}4)$$
$$-TL \leq \Delta z - a*\Delta\theta_y - b*\Delta\theta_x \leq TU. \qquad (3.3.2\text{-}5)$$

Case 2: Circular (radius *r*) planar surface. (Figure 8)

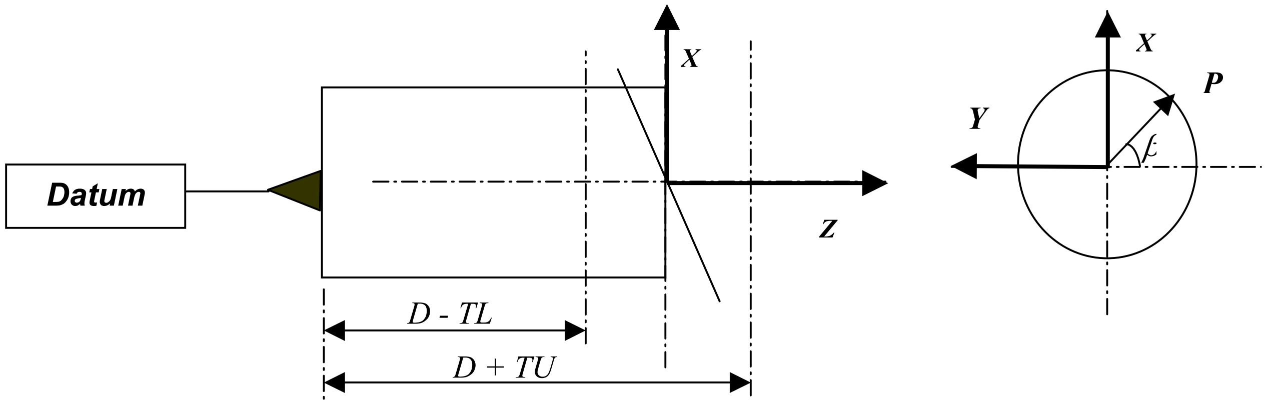


Figure 8: Planar feature (circle) with size tolerance

Deviation parameters = $(0, 0, \Delta z, \Delta\theta_x, \Delta\theta_y, 0)$, tolerance parameters = (TU, TL).
For arbitrary point *P* on the circumference, at an angle β with the *y*-axis, we have,

$$\delta_z = \Delta z - R\Delta\theta_y Sin(\beta) + R\Delta\theta_x Cos(\beta),$$

where δ_z is the pure position variation of a point in the *Z* direction, where there is a size tolerance control. The tolerance zone in this case is made up of two planes,
$z = +TU$ *and* $z = -TL$.

We need to ensure that δ_z is within the above boundary $\forall\beta \in (0, 2\pi)$
This can be done by treating the δ_z as a function of β and then finding the extreme points by putting $\partial(\delta_z)/\partial\beta = 0$. This gives,

$$\delta_{z\ max/min} = \Delta z \pm R\sqrt{(\Delta\theta_x^2 + \Delta\theta_y^2)}.$$

This leads to the following mapping (constraints):

$$(\Delta\theta_x^2 + \Delta\theta_y^2) \leq (TU + TL)^2/(4R^2).$$

We can also write in the parametric form as

$$TL \leq \Delta z + R\Delta\theta_x Sin(\beta) + R\Delta\theta_y Cos(\beta) \leq TU, \quad \forall\beta \in (0, 2\pi).$$

The last inequality, using the value of $\delta_{z\ max/min}$ could be written, independent of β, as:

$$TL \le (\Delta z - R\sqrt{(\Delta\theta_x^2 + \Delta\theta_y^2)}) \;\; and \;\; (\Delta z + R\sqrt{(\Delta\theta_x^2 + \Delta\theta_y^2)}) \le TU.$$

3.3.3 Cylindrical Feature - Modified to MMC

Let us assume a cylindrical feature with a given tolerance specification as shown in Figure 9. A local coordinate system (LCS), as shown in Figure 10, is defined for cylindrical surface and the deviation parameters are defined in that LCS. For a cylindrical surface, the LCS is: *z*-axis along the axis of the cylinder and *(x, y)* are local orthogonal co-ordinates on the middle of the axis (Figure 9, Figure 10). For this surface, the deviation parameters of the SDT are: $(\Delta x, \Delta y, 0, \Delta\theta_x, \Delta\theta_y, 0)$ and dr. Based on this notation, constraints could be derived connecting these parameters with the specified tolerance values as detailed below.

Deviation parameters = $(\Delta x, \Delta y, 0, \Delta\theta_x, \Delta\theta_y, 0)$, tolerance parameters = (TU, TL, T_P), SDT is:

$D_d = \{\Delta\theta \;\; \Delta d\} = \{\Delta\theta_x \;\; \Delta\theta_y \;\; 0 \;\; \Delta x \;\; \Delta y \;\; 0\}^T$,
$D_i = \{\Delta\theta \;\; \Delta d\} = \{0 \;\; 0 \;\; 0 \;\; dr{*}cos\theta \;\; dr{*}sin\theta \;\; 0\}^T$,
where D_d is *displacement torsor*[5] and D_i is *intrinsic torsor*[6].

Let us assume that there is a point $P\ (r\cos\theta,\ r\sin\theta, z)$ (Figure 9) on the nominal cylindrical surface. After applying the transformation due to the effect of the two small displacement torsors, the new position of transformed point P' can be calculated as follows.

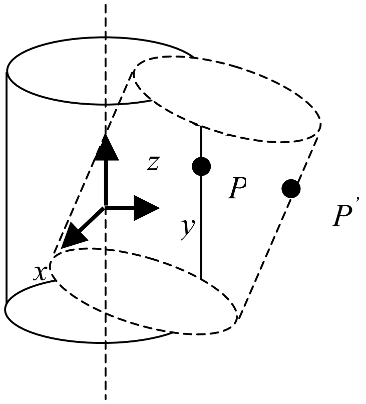


Figure 9: Cylindrical Feature with LCS

[5] Displacement torsor: torsor for the deviation of the feature from the nominal position.
[6] Intrinsic torsor: torsor to represent the intrinsic variation of size of the feature.

$$\begin{bmatrix} 1 & 0 & \Delta\theta_y & \Delta x \\ 0 & 1 & -\Delta\theta_x & \Delta y \\ -\Delta\theta_y & \Delta\theta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (r+dr)\cos\theta \\ (r+dr)\sin\theta \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (r+dr)\cos\theta + \Delta\theta_y z + \Delta x \\ (r+dr)\sin\theta - \Delta\theta_x z + \Delta y \\ z - \Delta\theta_y (r+dr)\cos\theta + \Delta\theta_x (r+dr)\sin\theta \\ 1 \end{bmatrix}.$$

The 4 by 4 matrix on left hand side of the equation is the transformation matrix for point *P* following the assumed transformation sequence $\Delta x \rightarrow \Delta y \rightarrow \Delta\theta_x \rightarrow \Delta\theta_y$ by displacement torsor. The 4 by 1 matrix (vector) on the left hand side of the equation represent the changed position of point *P* by intrinsic torsor.

So, the new position P' (Figure 9) is given by
$((r+dr)\cos\theta+\Delta\theta_y\ z+\Delta x,(r+dr)\sin\theta-\Delta\theta_x\ z+\Delta y,z-\Delta\theta_y\ (r+dr)\cos\theta+\Delta\theta_x(r+dr)\sin\theta)$.

The VCB of the cylindrical surface is constructed by using the rules (3.3.1-1, 3.3.1-2,3.3.1-3,3.3.1-4) and has the following properties:

- A perfect shape as that of the nominal cylindrical feature,
- A size (diameter) of max size plus geometric tolerance, because its external feature and the positional tolerance is modified to MMC,
- A perfect orientation as that of the nominal cylindrical feature (vertical to C),
- A perfect location as that of the nominal cylindrical feature.

Hence, the constraint equation will be:

$$((r+dr)\cos\theta + \Delta\theta_y z + \Delta x)^2 + ((r+dr)\sin\theta - \Delta\theta_x z + \Delta y)^2 \le (2\mathrm{r} + \mathrm{TU} + \mathrm{Tp})^2 / 4\,, \qquad (3.3.3\text{-}1)$$

where $\theta \in (0, 2\pi)$. The left hand side is a function of two independent parameters θ and $z \in (0, L)$. The inequality should be valid $\forall\theta \in (0, 2\pi)$ and $\forall z \in (0, L)$. We thus need to eliminate θ and z from the LHS of (3.3.3-1) by finding the maximum of the above equation. LHS of equation (3.3.3-1) could be written as:

$$f(\theta,z) = (a\cos\theta + bz + c)^2 + (a\sin\theta + dz + e)^2, \qquad (3.3.3-2)$$

where $a = r + dr,\quad b = \Delta\theta_y\,,\quad c = \Delta x\,,\quad d = -\Delta\theta_x\,,\quad e = \Delta y.$

By equating the partial derivatives of f to zero,

$\frac{\partial f}{\partial\theta} = 0,\ \ \textit{and}\ \ \frac{\partial f}{\partial z} = 0,\quad \textit{we}$ get,

$$f_{\max} = \max\left\{ \frac{\left(\frac{\pm 1}{\sqrt{d^2+b^2}}(ad^2+ab^2) - be + cd \right)^2}{d^2+b^2}, 0 \right\},\quad f_{\min} = \min\left\{ \frac{\left(\frac{\pm 1}{\sqrt{d^2+b^2}}(ad^2+ab^2) - be + cd \right)^2}{d^2+b^2}, 0 \right\}.$$

$$(3.3.3-3)$$

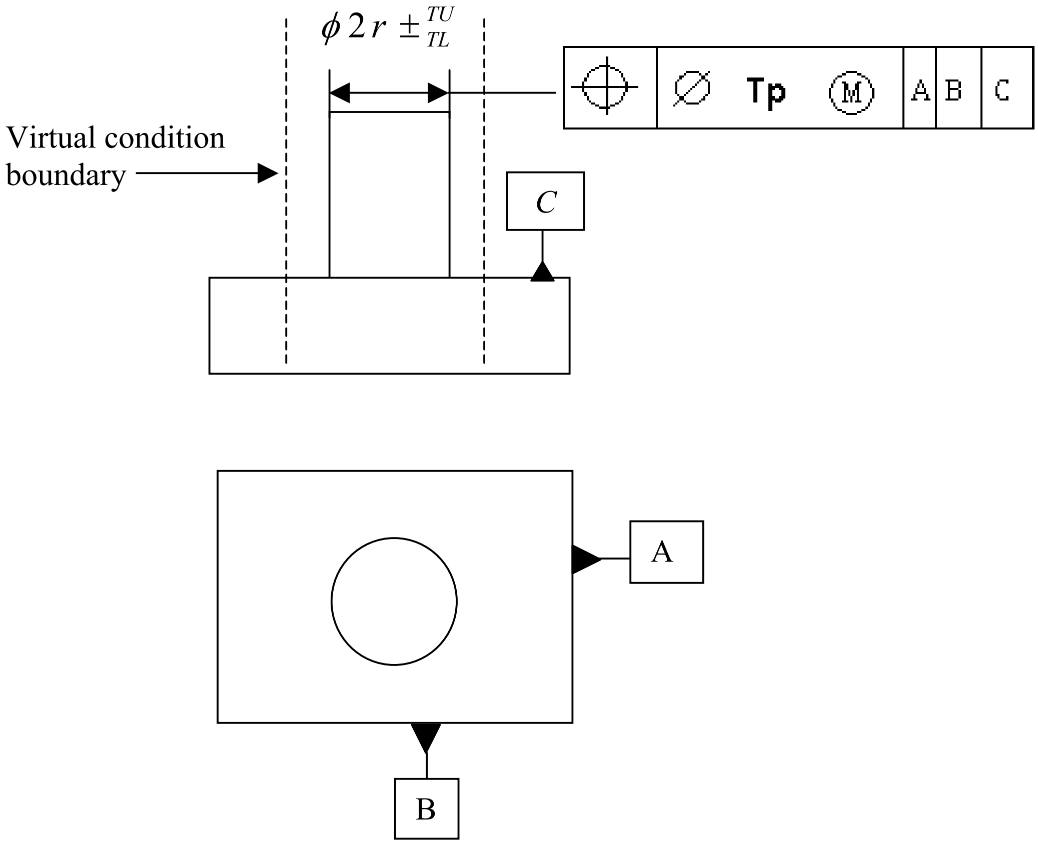


Figure 10: Cylindrical Feature with Tolerance Modified to MMC

The final set of constraints is:

$$f_{\max} \le \left(\frac{2r+TU+Tp}{2}\right)^2,$$

$$-TL \le \quad dr \le TU.$$

When the toleranced feature is internal, as for example, for a hole, the constraints are:

$$f_{\min} \ge \left(\frac{2r+TU-Tp}{2}\right)^2,$$

$$-TL \le \quad dr \le TU.$$

If the external feature is modified to LMC, the constraint will be:

$$f_{\min} \geq \left(\frac{2r - TL - Tp}{2} \right)^{2,}$$

$$-TL \leq \quad dr \leq TU \,.$$

If the internal feature is modified to LMC, the constraint will be:

$$f_{\max} \leq \left(\frac{2r - TL + Tp}{2} \right)^{2},$$

$$-TL \leq \quad dr \leq TU \,.$$

3.3.4 Cylindrical Feature Modified to RFS

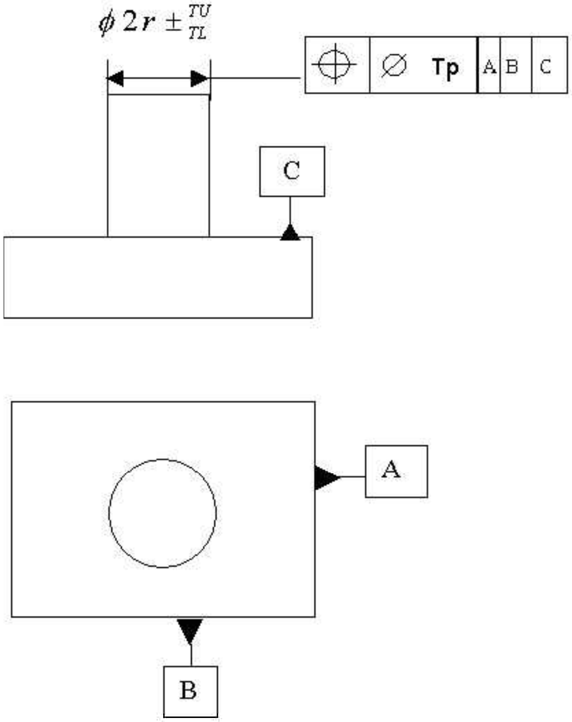


Figure 11: Cylindrical Feature with Tolerance Modified to RFS

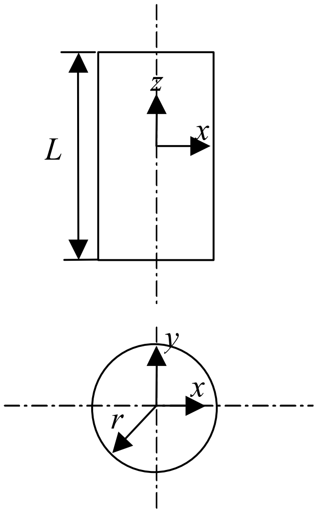


Figure 12: Cylindrical tolerance zone

Figure 12 shows a cylindrical tolerance zone (the cylindrical feature's center axis) that could be due to positional tolerance, concentricity tolerance and/or run-out tolerance for the cylindrical surface. A LCS (local coordinate system) is defined as shown in Figure 6. The displacement torsor and the intrinsic torsor are:
$D_d = \{\Delta\theta \ \ \Delta d\} = \{\Delta\theta_x \ \ \Delta\theta_y \ \ 0 \ \ \Delta x \ \ \Delta y \ \ 0\}^T$,
$D_i = \{\Delta\theta \ \ \Delta d\} = \{0 \ \ 0 \ \ 0 \ \ dr{*}cos\theta \ \ dr{*}sin\theta \ \ 0\}^T$.

The basic difference between this case and the one in Figure 10 is that D_d is independent from D_i. It means that the size of the tolerance zone, *r,* will not change due to the variation of *dr*, which is the departure from nominal radius. Because it is the derived element, axis of the feature of size, that is toleranced but not the surface of the feature of size, different methods are used to generate the constraints on mapping deviation parameters to the tolerance zone. Since D_d and D_i are independent, we need to treat them separately. For D_d we have 4 deviation parameters: $\Delta\theta_x$, $\Delta\theta_y$, Δx, and Δy. For ease of computation, let us assume that the deviations of the axis takes place in the sequence of $\Delta x \rightarrow \Delta y \rightarrow \Delta\theta_x \rightarrow \Delta\theta_y$. We think this assumption will not affect the generality because it does not matter what sequence we choose. (In fact, the matrix multiplication described earlier is based on the same assumption.)
The constraints on Δx and Δy are:

$$\begin{aligned} &-r \le \Delta x \le r \\ &-\sqrt{r^2 - \Delta x^2} \le \Delta y \le \sqrt{r^2 - \Delta x^2} \end{aligned} \qquad (3.3.4\text{-}1)$$

where *r* is the radius of tolerance zone.

Figure 13 shows the movement of the center axis by Δx and Δy followed by a rotation of the center axis by $\Delta\theta_x$ until the end of it touches tolerance zone boundary.
We determine the constraint on $\Delta\theta_x$, given those of Δx, Δy, as follows, following Figure 13:

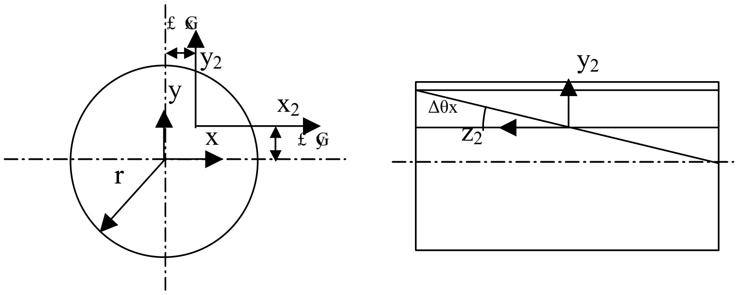


Figure 13: Derivation of constraint on Dqx

$$-\frac{\sqrt{r^2-\Delta x^2}-|\Delta y|}{L/2} \le \Delta\theta_x \le \frac{\sqrt{r^2-\Delta x^2}-|\Delta y|}{L/2}. \tag{3.3.4-2}$$

Figure 14 shows the movement of the center axis by *Δx* and *Δy* followed by a rotation of the center axis by *Δθ*$_x$, then rotate the center axis by *Δθ*$_y$ until the end of it touches tolerance zone boundary.

From Figure 14, we get,

$$-\frac{\sqrt{r^2-(\Delta\theta_x\cdot L/2+|\Delta y|)^2}-|\Delta x|}{(L/2)\cos\Delta\theta_x} \le \Delta\theta_y \le \frac{\sqrt{r^2-(\Delta\theta_x\cdot L/2+|\Delta y|)^2}-|\Delta x|}{(L/2)\cos\Delta\theta_x}.$$

Because *Δθ*$_x$ is very small, *cos(Δθ*$_x$*) ≈1*, so we have,

$$-\frac{\sqrt{r^2-(\Delta\theta_x\cdot L/2+|\Delta y|)^2}-|\Delta x|}{(L/2)} \le \Delta\theta_y \le \frac{\sqrt{r^2-(\Delta\theta_x\cdot L/2+|\Delta y|)^2}-|\Delta x|}{(L/2)}. \tag{3.3.4-3}$$

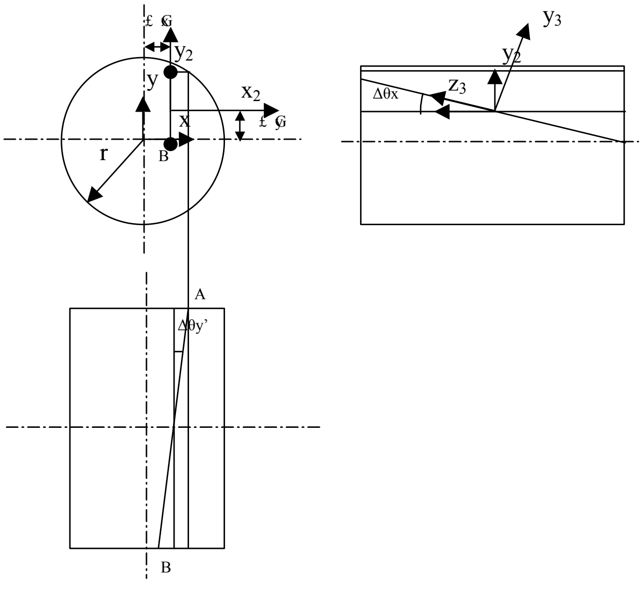


Figure 14: Derivation of constraint on Dqy

Inequalities (3.3.4-1,3.3.4-2,3.3.4-3) are the mapping results for the positional tolerance modified by RFS specified on a cylindrical feature of size.

3.3.5 Spherical Surface

A local coordinate system (LCS) is defined for a spherical surface and the deviation parameters are defined in that LCS. For a spherical surface, the LCS is: the z-axis is along a radius of the sphere, and (x, y) are local orthogonal co-ordinates at the center of the sphere (Figure 15, Figure 16). For this surface the deviation parameters of the SDT are: $(\Delta x, \Delta y, \Delta z, 0, 0, 0)$ and dr. Based on this notation, constraints could be derived connecting these parameters with the specified tolerance values as detailed below.

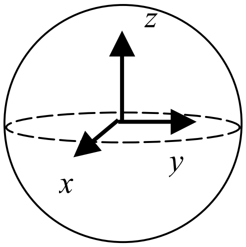


Figure 15: Spherical feature with LCS

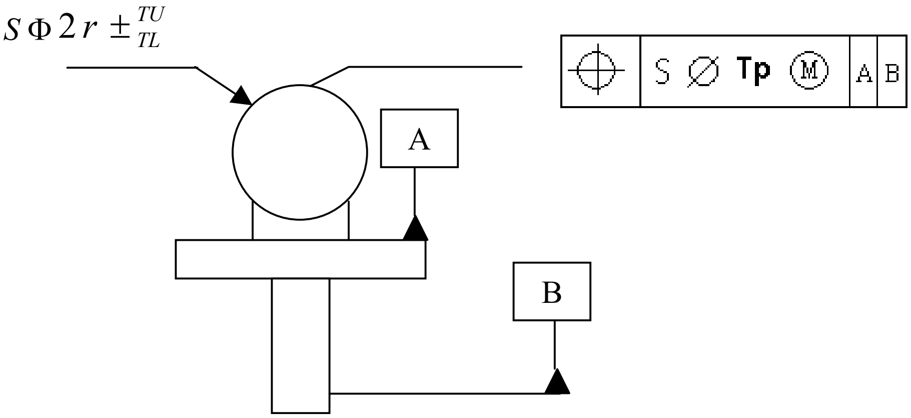


Figure 16: Spherical feature with tolerance specification

The VCB of the spherical surface is constructed with the following properties:

- A perfect shape as that of the nominal spherical feature,
- A size (diameter) of $2r+TU+T_p$, because it is an external feature and the positional tolerance is modified to MMC,
- A perfect location as that of the nominal cylindrical feature.

By applying the same procedures as was followed for the cylindrical case, we have:

$D_d = \{\Delta\theta \;\; \Delta d\} = \{0 \;\; 0 \;\; 0 \;\; \Delta x \;\; \Delta y \;\; \Delta z\}^T,$
$D_i = \{\Delta\theta \;\; \Delta d\} = \{0 \;\; 0 \;\; 0 \;\; dr*sin\phi*cos\ \psi \;\;\; dr*sin\phi*sin\psi \;\;\; dr*cos\phi\}^T,$ and

$$\begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (r+dr)\sin\phi\cos\psi \\ (r+dr)\sin\phi\sin\psi \\ (r+dr)\cos\psi \\ 1 \end{bmatrix} = \begin{bmatrix} \Delta x + (r+dr)\sin\phi\cos\psi \\ \Delta y + (r+dr)\sin\phi\sin\psi \\ \Delta z + (r+dr)\cos\psi \\ 1 \end{bmatrix}.$$

The 4 by 4 matrix on the left hand side of the above equation is the transformation matrix for any point on the spherical surface due to displacement torsor assuming the transformation sequence $\Delta x \rightarrow \Delta y \rightarrow \Delta z$. The 4 by 1 matrix (vector) on the left hand side of the equation represents the changed position of the point due to the intrinsic torsor.

The constraint equation is given by:

$$(\Delta x + (r+dr)\sin\phi\cos\psi)^2 + (\Delta y + (r+dr)\sin\phi\sin\psi)^2 + (\Delta z + (r+dr)\cos\psi)^2 \le (2r + TU + TP)^2 / 4$$

and $\quad dr \ge -TL$.

We need the above inequality to be valid $\forall\Phi \in (0,\ 2\pi)$ and $\forall\Psi \in (0,\ 2\pi)$. So we eliminate parameters Φ and Ψ by maximizing the following function.

$$f(\phi,\psi) = a\sin^2\phi + a\cos^2\psi + 2\Delta x\sin\phi\cos\psi + 2\Delta y\sin\phi\sin\psi + 2\Delta z\cos\psi\,,$$

where $a = r + dr$

$$\partial f / \partial\phi = 0 \quad \Rightarrow \quad 2\cos\phi(a\sin\phi + \Delta x\cos\psi + \Delta y\sin\psi) = 0\,,$$
$$\partial f / \partial\psi = 0 \quad \Rightarrow \quad -2a\cos\psi\sin\psi - 2\Delta x\sin\phi\sin\psi + 2\Delta y\sin\phi\cos\psi - 2\Delta z\sin\psi = 0.$$

Here, we have two sets of solutions (a symbolic software package was used to get the solutions in symbolic form):

1) $\phi = (2n+1)\dfrac{\pi}{2}, \quad \psi = -tg^{-1}\dfrac{a\beta - \Delta y}{\Delta x + \Delta z},$

where $\quad \beta^2 = (\,z^4a^2 - 2z^3a\Delta y + (\Delta z^2 + \Delta y^2 + 2\Delta x\Delta z - a^2 + \Delta x^2)z^2 + 2az\Delta y - \Delta y^2)$

$$\phi = -\sin^{-1}\left(\frac{\Delta x(-\gamma^2 a^2 + \gamma^2 \Delta x^2 + \gamma^2 \Delta y^2 - \gamma \Delta za - \Delta y^2)}{(-\gamma a^2 + \gamma \Delta x^2 - \gamma \Delta y^2 - \Delta za)a}\right), \psi = tg^{-1}\left(\frac{\Delta y \Delta x(2\gamma^2 - 1)}{-\gamma^2 a^2 + \gamma^2 \Delta x^2 - \gamma^2 \Delta y^2 - \gamma \Delta za}\right),$$

2) where $\gamma^2 = ((a^4 - 2a^2\Delta x^2 + 2a^2\Delta y^2 + \Delta x^4 + \Delta y^4 + 2\Delta y^2\Delta x^2)z^4 + (2\Delta za^3 - 2a\Delta z\Delta x^2 + 2a\Delta z\Delta y^2)z^3 + (-a^4 - 2\Delta y^2\Delta x^2 - 2a^2\Delta y^2 - \Delta y^4 - \Delta x^4 + 2a^2\Delta x^2 + \Delta z^2 a^2)z^2 + (2a\Delta z\Delta x^2 - 2\Delta za^3 - 2a\Delta z\Delta y^2)z - \Delta z^2 a^2 + \Delta y^2\Delta x^2)$.

The final set of constraints is:

$$f_{\max} \le \left(\frac{2r + TU + Tp}{2}\right)^2,$$

$$-TL \le \quad dr \le TU.$$

When the toleranced feature is internal, for example a hole, the final constraint will be:

$$f_{\min} \ge \left(\frac{2r + TU - Tp}{2}\right)^2,$$

$$-TL \le \quad dr \le TU.$$

If the external feature is modified to LMC, the constraint will be:

$$f_{\min} \ge \left(\frac{2r - TL - Tp}{2}\right)^2,$$

$$-TL \le \quad dr \le TU.$$

If the internal feature is modified to LMC, the constraint will be:

$$f_{\max} \le \left(\frac{2r - TL + Tp}{2}\right)^2,$$

$$-TL \le \quad dr \le TU.$$

3.3.6 Conical Feature

A local coordinate system (LCS) is defined for a conical surface and the deviation parameters are defined in that LCS. For a conical surface, the LCS is: the z-axis is along the cone axis and *(x, y)* are local orthogonal co-ordinates on the bottom plane of the cone (Figure 17). For this surface the deviation parameters of the SDT are: $(\Delta x, \Delta y, \Delta z, \Delta\theta_x, \Delta\theta_y, 0)$ and *dr*. Based on this notation, constraints could be derived connecting these parameters with the specified tolerance values as detailed below.

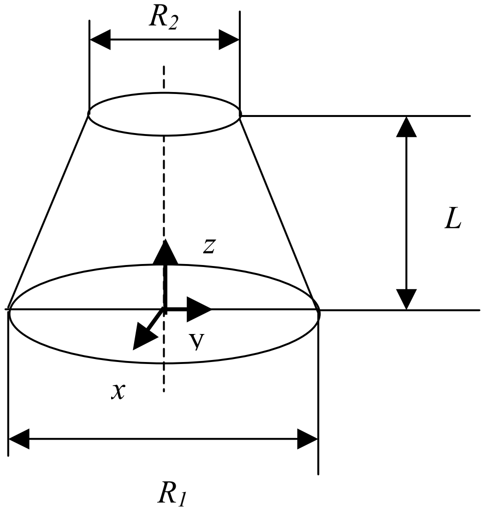


Figure 17: Conical feature with LCS

$$D_d = \begin{Bmatrix} \Delta\theta_x & \Delta x \\ \Delta\theta_y & \Delta y \\ 0 & \Delta z \end{Bmatrix} \qquad D_I = \begin{Bmatrix} 0 & dr\cos\theta \\ 0 & dr\sin\theta \\ 0 & 0 \end{Bmatrix}, \tag{3.3.6-1}$$

A point P on the nominal surface $\left(r\cos\theta, r\sin\theta, \frac{R_1 - r}{R_1 - R_2} L \right)$, will transform to:

$$\begin{bmatrix} 1 & 0 & \Delta\theta_y & \Delta x \\ 0 & 1 & -\Delta\theta_x & \Delta y \\ -\Delta\theta_y & \Delta\theta_x & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (r+dr)\cos\theta \\ (r+dr)\sin\theta \\ \frac{R_1 - r}{R_1 - R_2} L \\ 1 \end{bmatrix} = \begin{bmatrix} (r+dr)\cos\theta + \Delta\theta_y \frac{R_1 - r}{R_1 - R_2} L + \Delta x \\ (r+dr)\sin\theta - \Delta\theta_x \frac{R_1 - r}{R_1 - R_2} L + \Delta y \\ \frac{R_1 - r}{R_1 - R_2} L - \Delta\theta_y (r+dr)\cos\theta + \Delta\theta_x (r+dr)\sin\theta + \Delta z \\ 1 \end{bmatrix} \tag{3.3.6-2}$$

For the tolerance specification of the following feature, the constraints will be:

$$r - TL \le \sqrt{x'^2 + y'^2} + \frac{R_1 - R_2}{L} z' \le r + TU,$$

where

$$\begin{aligned} x' &= (r + dr)\cos\theta + \Delta\theta_y \frac{R_1 - r}{R_1 - R_2} L + \Delta x, \\ y' &= (r + dr)\sin\theta - \Delta\theta_x \frac{R_1 - r}{R_1 - R_2} L + \Delta y, \\ z' &= \frac{R_1 - r}{R_1 - R_2} L - \Delta\theta_y (r + dr)\cos\theta + \Delta\theta_x (r + dr)\sin\theta + \Delta z. \end{aligned} \quad (3.3.6\text{-}3)$$

We need to find the Max and Min of $f(r,\theta) = \sqrt{x'^2 + y'^2} + \frac{R_1 - R_2}{L} z'$, to eliminate *r* and *θ*. However, we cannot yet solve the two non-linear equations formed by equating the partial derivatives of the function *f* with respect to *r* and *t* to zero in compact (symbolic) form. So in order to use the results for the conical surfaces in the tolerance synthesis/assemblability analysis, we have to explore further to get a solution or else develop/adopt methods to treat the equation keeping the two free parameters as extra variables.

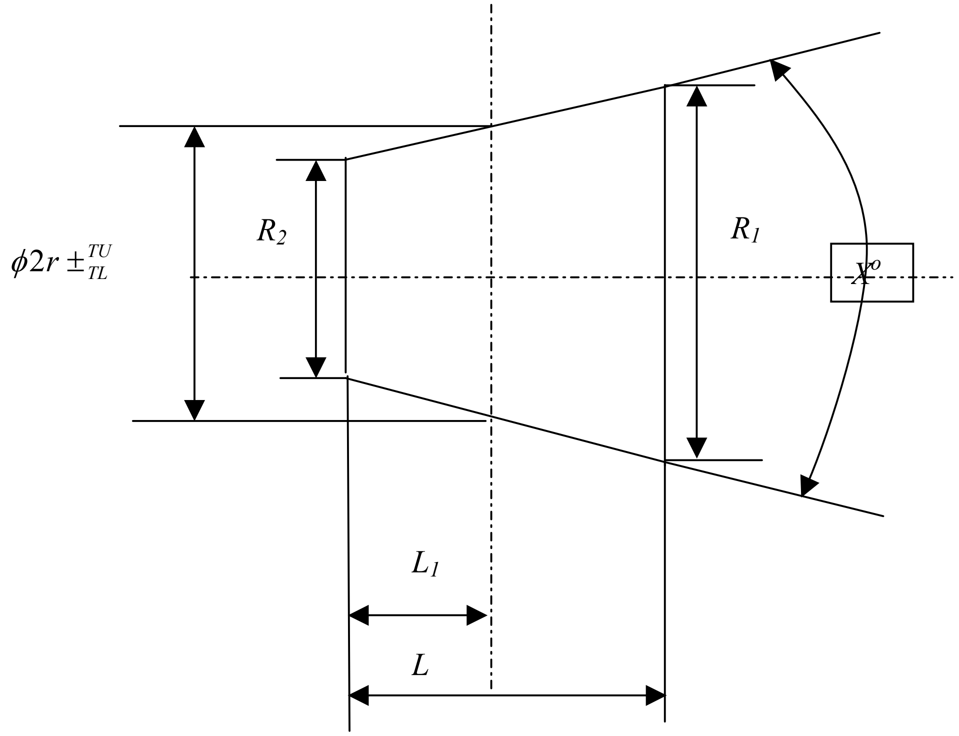


Figure 18: Conical feature with tolerance specification

3.3.7 Special cases

3.3.7.1 Composite tolerance specification

Sometimes we have more than one tolerance specified on one feature of size, as shown in Figure 19. In this case, more constraints need to be generated. A designer has two intents when specifying combined tolerances: 1) both tolerances must be satisfied at the same time, and 2) some DOFs of the feature of size will be constrained specifically by the additional tolerance. In the following case, the perpendicular tolerance gives more constraint on the orientation of the feature of size.

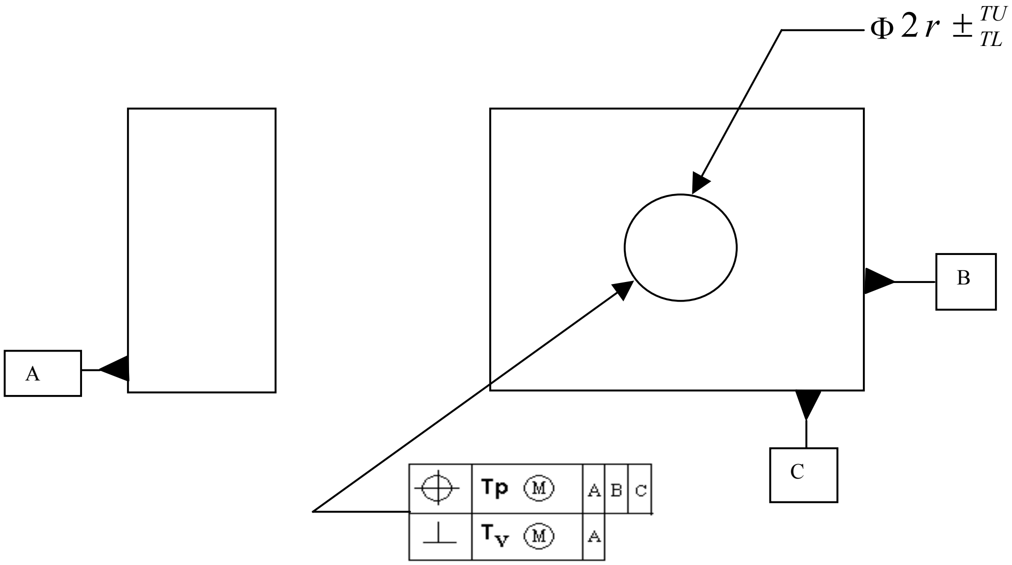


Figure 19: Composite tolerance specification

The ASME Y14.5 standard requires that all the tolerances specified to one feature of size must be satisfied independently. This means that we need to treat every tolerance seprately. For the above example, the constraints for positional tolerance are same as those in Section 3.3.3; the following is the derivation of the constraints for perpendicular tolerance.

The function of the perpendicular tolerance here is to give more constraints on the orientation of the feature. Because perpendicular tolerance does not fix the position of virtual condition boundary, and the tolerance only controls orientation, the position of the transformed point due to the perpendicular tolerance can be can be calculated as follows:

$$\begin{bmatrix} 1 & 0 & \Delta\theta_y & 0 \\ 0 & 1 & -\Delta\theta_x & 0 \\ -\Delta\theta_y & \Delta\theta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (r+dr)\cos\theta \\ (r+dr)\sin\theta \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (r+dr)\cos\theta + \Delta\theta_y z \\ (r+dr)\sin\theta - \Delta\theta_x z \\ z - \Delta\theta_y (r+dr)\cos\theta + \Delta\theta_x (r+dr)\sin\theta \\ 1 \end{bmatrix}.$$

Therefore, the constraint equation will be:

$$((r+dr)\cos\theta + \Delta\theta_y z)^2 + ((r+dr)\sin\theta - \Delta\theta_x z)^2 \le (2r + TU + T_V)^2 / 4, \qquad (3.3.7.1\text{-}1)$$

with the constraint equation by positional tolerance:

$$((r+dr)\cos\theta + \Delta\theta_y z + \Delta x)^2 + ((r+dr)\sin\theta - \Delta\theta_x z + \Delta y)^2 \le (2\mathrm{r} + \mathrm{TU} + \mathrm{Tp})^2 / 4\,. \qquad (3.3.7.1\text{-}2)$$

Equation (3.3.7.1-1) and (3.3.7.1-2) are the constraints due to the composite tolerance specification in the above example. Using the same procedure as in section 3.3.3 to eliminate θ and z, we have thefollowing constraints on deviations for the perpendicular tolerance:

$f'(\theta, z) = (a\cos\theta + bz)^2 + (a\sin\theta + dz)^2,$

where $a = r + dr,\quad b = \Delta\theta_y\,,\quad d = -\Delta\theta_x.$

By equating the partial derivative s of f′ to zero,

$\frac{\partial f'}{\partial\theta} = 0,$ and $\frac{\partial f'}{\partial z} = 0,$ we get,

$$f'_{\max} = \max\left\{\frac{\left(\frac{\pm 1}{\sqrt{d^2+b^2}}(ad^2+ab^2)\right)^2}{d^2+b^2}, 0\right\},$$

$$f'_{\min} = \min\left\{\frac{\left(\frac{\pm 1}{\sqrt{d^2+b^2}}(ad^2+ab^2)\right)^2}{d^2+b^2}, 0\right\}.$$

From the last section for a positional tolerance of similar specification, we have:

$f(\theta, z) = (a\cos\theta + bz + c)^2 + (a\sin\theta + dz + e)^2,$

where $a = r + dr,\quad b = \Delta\theta_y\,,\quad c = \Delta x,\quad d = -\Delta\theta_x,\quad e = \Delta y.$

By equating the partial derivative s of f to zero,

$\frac{\partial f}{\partial\theta} = 0,$ *and* $\frac{\partial f}{\partial z} = 0,$ *we* get,

$$f_{\max} = \max\left\{\frac{(\frac{\pm 1}{\sqrt{d^2+b^2}}(ad^2+ab^2) - be + cd)^2}{d^2+b^2}, 0\right\},\quad f_{\min} = \min\left\{\frac{(\frac{\pm 1}{\sqrt{d^2+b^2}}(ad^2+ab^2) - be + cd)^2}{d^2+b^2}, 0\right\}.$$

The final constraints due to perpendicular tolerance and positional tolerance are:

$$\begin{aligned} f_{\max} &\le \left(\frac{2r + TU + Tp}{2}\right)^2, \\ f'_{\max} &\le \left(\frac{2r + TU + Tv}{2}\right)^2, \\ -TL \le\ & dr \le TU\,. \end{aligned} \qquad (3.3.7.1\text{-}3)$$

When the toleranced feature is internal, as for example, for a hole, the constraints are:

$$\begin{aligned} &f_{\min} \geq \left(\frac{2r + TU - Tp}{2}\right)^2, \\ &f'_{\min} \geq \left(\frac{2r + TU - Tv}{2}\right)^2, \\ &-TL \leq \quad dr \leq TU\,. \end{aligned} \qquad (3.3.7.1\text{-}4)$$

If the external feature is modified to LMC, the constraints will be:

$$\begin{aligned} &f_{\min} \geq \left(\frac{2r - TL - Tp}{2}\right)^2, \\ &f'_{\min} \geq \left(\frac{2r - TL - Tv}{2}\right)^2, \\ &-TL \leq \quad dr \leq TU\,. \end{aligned} \qquad (3.3.7.1\text{-}5)$$

If the internal feature is modified to LMC, the constraints will be:

$$\begin{aligned} &f_{\max} \leq \left(\frac{2r - TL + Tp}{2}\right)^2, \\ &f'_{\max} \leq \left(\frac{2r - TL + Tv}{2}\right)^2, \\ &-TL \leq \quad dr \leq TU\,. \end{aligned} \qquad (3.3.7.1\text{-}6)$$

3.3.7.2 MMC Specification on Datums

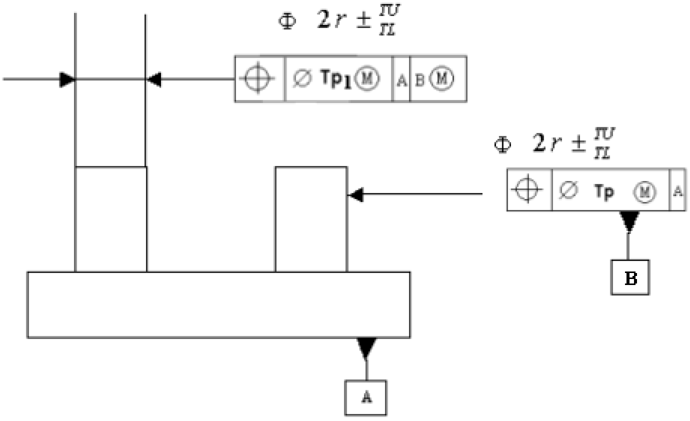


Figure 20: MMC Specified to datum

When accurate relative position of two parts is not required, the designer may specify MMC on the datum to allow more relative position variation between a datum feature of size and the toleranced feature of size, if the size of the datum feature of size departs from its MMC. This specification will not increase the size of the VCB but will allow its center axis to move if the datum feature of size departs from its MMC.

In contrast to the composite tolerance specification described in previous section, the position of the VCB of the left hand side cylindrical feature is dependent on the real size of the cylindrical feature on the right hand side. Because this specification means that both cylindrical features will mate features on the other part and the clearance due to the departure of size of datum feature of size from its MMC is allowed to be used to adjust the position of left side cylindrical feature, this problem can be modeled as if we have the positional tolerance and perpendicular tolerance specified on the left-hand feature.

The positional tolerance will be $T_{p1}+(TU+T_p\text{-}dr)$ [6] and the perpendicular tolerance is T_{p1}. Following the procedures as described in Section 3.2, we can develop the constraints on deviations of the left hand side cylindrical feature by substituting $T_p= T_{p1}+(TU+T_p\text{-}dr)$ and $T_v=T_{p1}$:

$$f_{\max} \le \left(\frac{2r + TU + \mathrm{Tp1} + (\mathrm{TU} + \mathrm{Tp} \text{ - } \mathrm{dr})}{2}\right)^2$$

$$f'_{\max} \le \left(\frac{2r + TU + Tp1}{2}\right)^2,$$

$$-TL \le \quad dr \le TU\,.$$

Example of Some Mappings

Case 1: Planar Feature

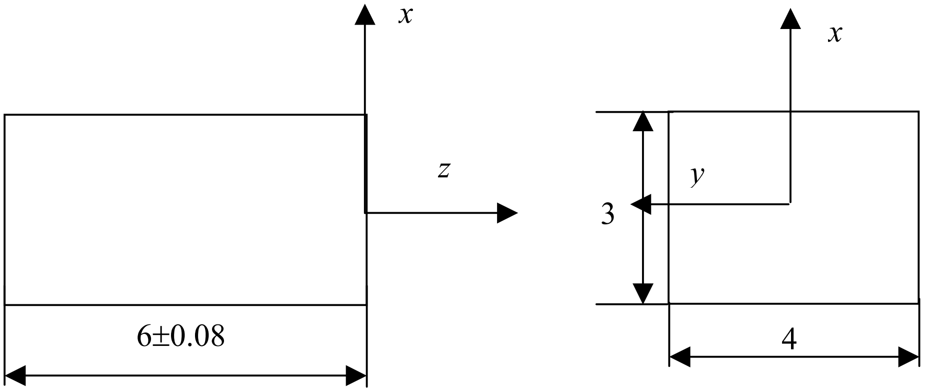


Figure 21: Planar feature (rectangular) with size tolerance

Figure 21 shows a simple size tolerance to a planar surface with LCS specified. Let us assume the left hand side surface is the implicit datum and the other side is controlled by the tolerance specification.

[6] In this case $(TU+T_p\text{-}dr)$ is from the other cylindrical feature

From Figure 21, we know the deviation parameters for the right hand side surface which is a non-size feature are $\Delta\theta_x$, $\Delta\theta_y$, Δz.

The following constraints are established by considering the deviation at the four extreme points (corners) of the plane which are nominally at (1.5, 2, 0), (1.5, -2, 0), (-1.5, 2, 0) and (-1.5, -2, 0).

$$-0.08 \le \Delta z + 1.5*\Delta\theta_y - 2*\Delta\theta_x \le 0.08,$$
$$-0.08 \le \Delta z + 1.5*\Delta\theta_y + 2*\Delta\theta_x \le 0.08,$$
$$-0.08 \le \Delta z - 1.5*\Delta\theta_y + 2*\Delta\theta_x \le 0.08,$$
$$-0.08 \le \Delta z - 1.5*\Delta\theta_y - 2*\Delta\theta_x \le 0.08.$$

The above constraints on deviation parameters result in a diamond shape in deviation space as follows:.

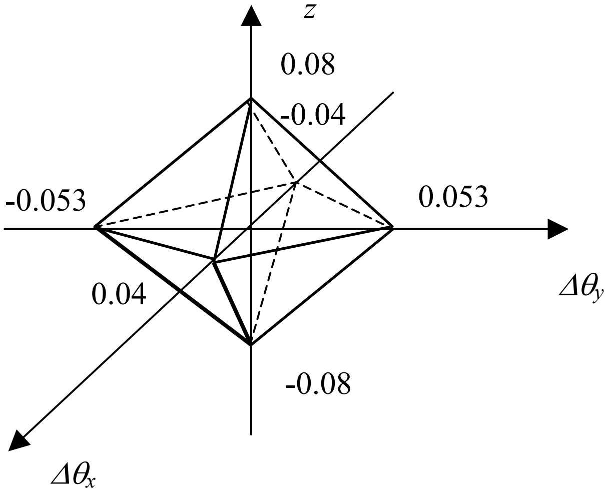


Figure 22: Deviation space for size tolerance on planar surface

Case 2: Cylindrical mating

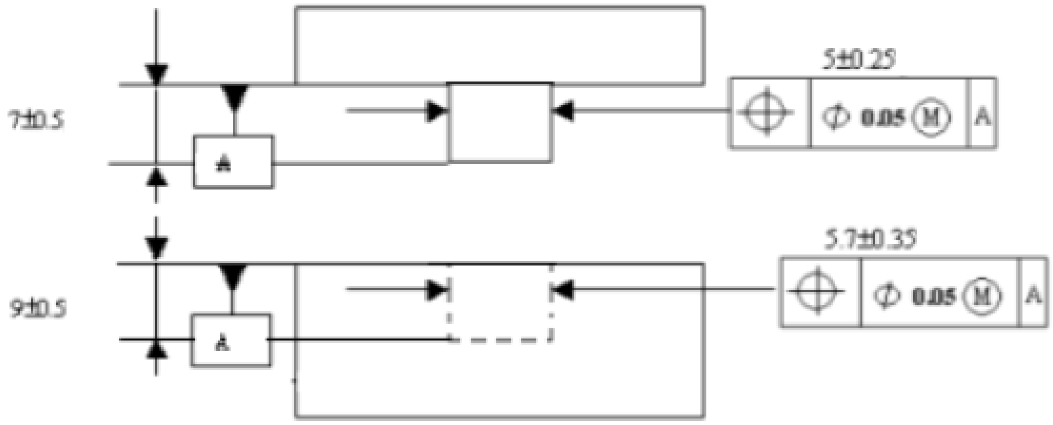


Figure 23: Assembly of two parts specified with positional tolerance

From the above specification, we know that the VCB of the external feature of the above part is equal to that of the internal feature of the bottom part. Following equations (3.3.3-1) and (3.3.3-2), the size of the VCB for the external cylindrical feature is 5+0.25+0.05= 5.3, and that of the internal cylindrical feature is .5.7-0.35-0.05= 5.3. Therefore, we know that the two parts are guaranteed to be assembled. Now let's see what constraints on deviation parameters we can get by our tolerance mapping, so we can computationally perform the tolerance analysis or synthesis.

Let $a = r + dr,\quad b = \Delta\theta_y\,,\ \ c = \Delta x,\ \ d = -\Delta\theta_x,\quad e = \Delta y,$

$$f_{\max} = \max\left\{ \frac{\left(\frac{\pm 1}{\sqrt{d^2 + b^2}}(ad^2 + ab^2) - be + cd \right)^2}{d^2 + b^2}, 0 \right\},\quad f_{\min} = \min\left\{ \frac{\left(\frac{\pm 1}{\sqrt{d^2 + b^2}}(ad^2 + ab^2) - be + cd \right)^2}{d^2 + b^2}, 0 \right\}.$$

The final set of constraints for the external feature is:

$$f_{\max} \le \left(\frac{5 + 0.25 + 0.05}{2} \right)^2,$$

$$-0.25 \le \quad dr \le 0.25.$$

When the toleranced feature is internal, as for example, for a hole, the constraints are:

$$f_{\min} \ge \left(\frac{5.7 - 0.35 - 0.05}{2} \right)^2,$$

$$-0.35 \le \quad dr \le 0.35.$$

The above equations connot be reduced further in compact form so no further derivation and/or plotting is possible. However, the inequalities as such could be used in computations.

4. Optimization Process

Formulation of the Cost Functions

The cost function to be minimized can have several components, including:

- Cost of Manufacturing,
- Cost of Assembling,
- Cost of Inspection,
- Cost related to product life span,
- Cost to meet functional requirements,
- Cost to meet assemblability, etc.

In this study, we have considered the Cost of Manufacturing in details and a new cost of manufacturing model [deviation-based cost of manufacturing] has been introduced. Apart from the c*ost of manufacturing,* we could not yet carry out further investigation into the *cost of assembly, inspection*, *product life-span related costs*, etc. Some authors have done context specific work on the above costs and, for example, Dong [DON97] has considered the effect of tolerance on cost related to product lifespan by constructing a functional performance index generated from the effect of tolerance. However, the work is applicable to very specific domains and cannot be used universally.

In our tolerance synthesis scheme, we have not considered these costs. In the proposed tolerance synthesis scheme, we have kept a generalized *cost of manufacturing model* so that the designer can easily introduce/modify the cost functions associated with each part in the artifact library [ROY01]. The cost function has been kept as a virtual function in the artifact class so that the specific cost models could be introduced in the instantiated object from a class of real cost functions.

We would consider the last two of the above cost factors (Cost to meet functional requirements and Cost to meet assembleability) as constrains rather than minimization criteria. We will now go into details of cost of manufacturing.

4.1 Cost of Manufacturing

We consider the cost associated with manufacturing a part for specified material, dimensions/geometric shape, a sequence of manufacturing operations, and the process capabilities. In general, the cost of manufacturing is a function of all the above entities. Many researchers have analyzed various issues associated with the cost of manufacturing in relation to tolerances and the cost of manufacturing as a decreasing function of a single tolerance parameter. Various functional forms (like the inverse power law, exponential decay, etc.) have been used as cost models that are generally monotonically decreasing functions of increasing in tolerance value. Effect of process capabilities on cost of manufacturing has also been well studied by several authors [DON97].

Also, since, in general, more than one operation is required to transform the raw blank into the final finished part, the cost of manufacturing is a function of the process sequence and of how much accuracy is achieved in each stages of operation. The cost is also affected by the setup error in each machining process. The total cost of production thus becomes a sum of the costs associated with each process [ROY97].

While none of these methods mentioned above could claim to be universal, there are several limitations with the one-parameter (single tolerance) cost of manufacturing formulation. In reality, a manufactured surface would rarely have a single tolerance value. Apart from a size tolerance, there could be geometric tolerances (form tolerance, orientation tolerance, etc) and it would be difficult to formulate a single parameter representing all these tolerances that could effectively be used for representing the cost of manufacturing. In order to circumvent the above problem, we propose a new formulation for the cost of manufacturing, which we would like to call the deviation-based model.

4.2 Deviation-based Cost Model of Manufacturing

In general, a part will have more than one feature, and is connected to other features of other parts and since each feature could possibly have up to six degrees of freedom, the scheme we have adopted uses six deviation parameters to represent the variations associated with each feature. In our earlier reports, we have used the small deviation torsors (SDT) for representing the deviations of features. We have also shown that these deviation parameters are related to the tolerance specifications through a functional mapping (a one to many relation: a tolerance specification defined a bounded region in the deviation space). In our approach, we propose to represent the cost of manufacturing as function of these deviation parameters. Thus, for example, the cost function $g(\delta)$ defined as a function of some tolerance value δ, would become a function of the six parameters $g(\delta) = g(\delta(\theta_x\ \theta_y\ \theta_z\ \ \delta_x\ \delta_y\ \delta_z),$ and then the cost functions could be treated as functions of the small deviation parameters associated with the surface/part.

We propose to model the cost of manufacturing a part as an explicit product of six functions of the six deviation parameters in the form:

$$\mathrm{DCOM}(d) = C_1(d_1)*C_2(d_2)*\ldots.C_6(d_6), \qquad (4.2\text{-}1)$$

where $d = (d_1, d_2, d_3, d_4, d_5, d_6) = (\theta_x , \theta_y , \theta_z , \delta_x , \delta_y , \delta_z)$ is a 6-componenet torsor representing the six deviation parameters characteristic of the feature.

The form of the individual functions can vary depending on the specific surfaces and manufacturing process. Some of the functions could be of similar form. Also depending on the nature/type of the feature, some of the functions will be constants (i.e. $\forall x, f(x) \equiv 1$) and could be eliminated. These functions correspond to the deviation parameters that are invariants of the surface.

For an example, for a planar surface there are only three independent parameters given by: $d = (d_x, 0, 0, 0, \theta_y , \theta_z)$ that affect the deviation of the surface from it's nominal shape; changes in the remaining three keep the surface invariant. The cost function can then be represented as:

$$DCOM(d) = C_x(d_x)*C_\theta(\theta_y)*C_\theta(\theta_z). \qquad (4.2\text{-}2)$$

Also, in this case, since the surface is symmetric about the *y* and *z* axes, the form the two functions for rotational deviations along these two directions are also same, namely C_θ.

We have mentioned earlier in this section that the deviation parameters are restricted by the tolerance specification. For the planar feature, this mapping forms a convex hull in the form of a diamond in the 3-D space. For a rectangular planar section with cross-section *('a' x 'b'*), (see section 3.3.2), the following are the constraints:

$$T_{SL} \le min\ (d_X + a*\theta_y + b*\theta_z\ ,\ d_X + a*\theta_y - b*\theta_z\ ,\ d_X - a*\theta_y + b*\theta_z\ ,\ d_X - a*\theta_y - b*\theta_z, \qquad (4.2\text{-}3)$$
$$T_{SU} \ge max\ (d_X + a*\theta_y + b*\theta_z\ ,\ d_X + a*\theta_y - b*\theta_z\ ,\ d_X - a*\theta_y + b*\theta_z,\ d_X - a*\theta_y - b*\theta_z), \qquad (4.2\text{-}4)$$

where $(T_{SL},\ T_{SU})$ are the lower and upper values of the tolerance parameter for the planar surface. Thus, the parameters $d_x\ \theta_y\ \theta_z$ are restricted. The cost function (3-2) will be restricted accordingly.

To illustrate the cost function, let us assume a generic function of the form: $C(x) = a + b/|x|$, where *x* is the deviation parameter and *a* and *b* are constants. For the planar case, we then get,

$$DCOM(d) = C_x(d_x)*C_\theta(\theta_y)*C_\theta(\theta_z)=(a_1+b_1/|d_x|) * (a_2+b_2/|\theta_y|) * (a_3+b_3/|\theta_z|)$$

To further simplify, we assume, $a=b=1$ and $a_2 = a_3,\ b_2 = b_3.$

Then we have

$$-T_{SL} \le\ (d_X + \theta_y + \theta_z) \le T_{SU},$$

$$-T_{SL} \le (d_X + \theta_y - \theta_z) \le T_{SU},$$
$$-T_{SL} \le (d_X - \theta_y + \theta_z) \le T_{SU},$$
$$-T_{SL} \le (d_X - \theta_y - \theta_z) \le T_{SU},$$

and $$DCOM(d) = (a_1+b_1/|d_x|) * (a+b/|\theta_y|) * (a+b/|\theta_z|).$$

Removing the z-parameter θ_z again so that we can have a visual representation of the cost function, we have $-T_{SL} \le\ (d + \theta) \le T_{SU}$ $\quad -T_{SL} \le (d - \theta) \le T_{SU}$ and

$$DCOM(d) = (a_1+b_1/|d|) * (a+b/|\theta|). \qquad (4.2\text{-}5)$$

In the $d\text{-}\theta$ plane, this would look like a tent bounded by four vertical planes (by the tolerance specification) approaching infinity along the two axes (Figure 24, Figure 25).

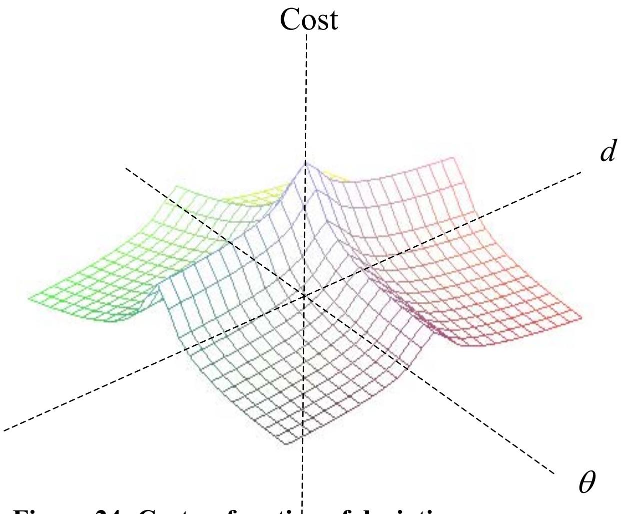


Figure 24: Cost as function of deviation parameters

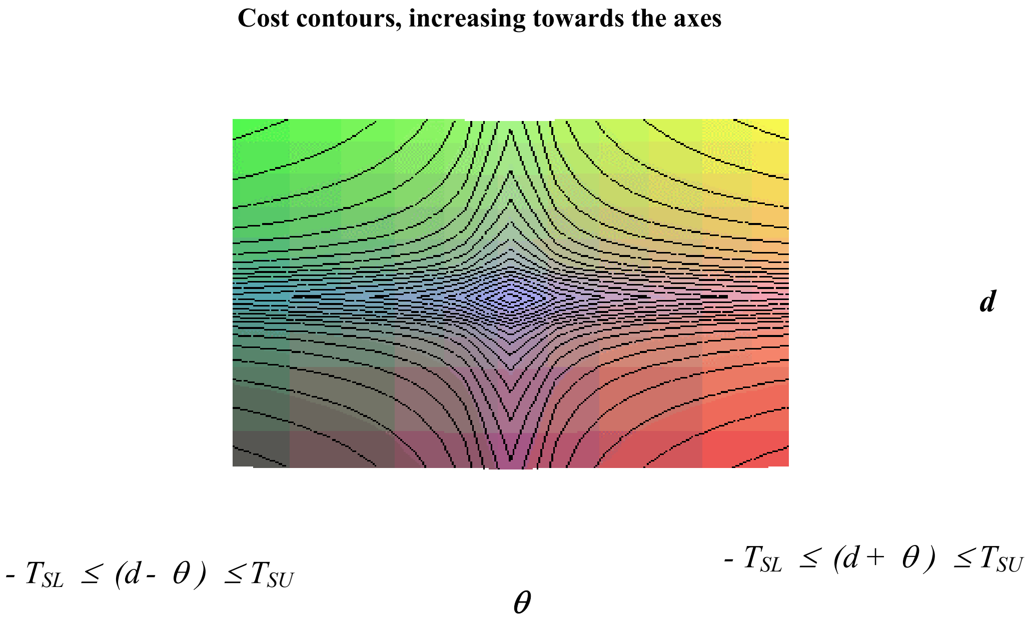


Figure 25: Cost contour lines and the bounds

It is to be noted that the actual form of the individual cost functions are yet to be determined. Also, it is evident that since the deviation parameters limit the tolerance values in some way or other, the cost functions of these deviation parameters should also be monotone non-increasing in nature.

4.3 Optimization Process

Based on the models described above, the optimization process could be summarized as below:

Minimize:

Cost of production: $\boldsymbol{C(x) = \Sigma_{(i \in all\_parts)}\Sigma_{(j \in all\_mating\_features)} C(x_{ij})}$, (4.3-1)

where $\boldsymbol{x_{ij}}$ is the deviation parameters of the $\boldsymbol{j^{th}}$ feature of part $\boldsymbol{i}$ in the assembly

Subject to:

Functional constraints of the form: $\boldsymbol{G(x) \leq 0, x \in R^n}$, (4.3-2)

Linear Assembliability Constraints of the form: $\boldsymbol{Ax = b}$, where $\boldsymbol{A = [a_{ij}]}$ is an $\boldsymbol{m \ x \ n}$ matrix, $\boldsymbol{a_{ij} \in R, \ x \in R^n}$, $\boldsymbol{b \in R^m}$, and $\boldsymbol{m}$ is the number of constraints. In general, $\boldsymbol{m}$ is less than the number of variables $\boldsymbol{n}$.

Deviation to tolerance mapping constraints: $\boldsymbol{FDP_k(x) \leq FTol_k(t), \ x \in R^n, \ k \in N^+}$ (4.3-3)

where ***FDP*** denotes Function of Deviation Parameter, ***FTol*** denotes Function of tolerance parameter subscript *k* is the number of constraint equations.

In order to look for a solution for the above minimization formulation that falls under the non-linear programming [NLP] category, we are proposing to develop suitable and efficient evolutionary algorithms [EA] because of the potential problems that could be anticipated in using standard gradient-based NLP solvers as detailed below.

The problem formulated above has several layers of complexity that make the problem very difficult to solve using traditional methods. We have tried to formulate the constraints for a very simple, elementary, two dimensional, planar, 3-element assembly [Appendix-A] for which we need about 150+ variables, 80+ linear constraints, 10+ non-linear constraints. We assumed there were no constraints arising out of function requirements and assumed a fairly complex [exponential and inverse] cost function. We then tried to solve the problem using the Matlab© NLP solver **fmincon.** We got some solutions after several trials but the solution is not realistic unless we impose several restrictions on the deviation parameters as well as the tolerance parameters. We anticipate that a medium to reasonably large sized assembly with say a couple of hundred parts in a 3D space would give rise to several thousands of variables and constraints. Apart from that the matrix $\boldsymbol{A}$ in the linear assembleability constraints $\boldsymbol{Ax = b}$ could be very sparse depending upon the assembly, requiring special treatment.

In addition, it is our endeavor to establish a deviation hull as a suitable solution space for further mapping of the tolerance space. This task itself is quite complicated and we believe an approach using the EA techniques could give us some lead by, for example, storing the intermediate solutions and doing some analysis of the pattern. However, this area is open for further research.

Computer Implementation of the Tolerance Synthesis System

The proposed tolerance synthesis system has been considered as an object-oriented representation built in part on our earlier work [ROY01] where we introduced a generic artifact class that can be extended to include a tolerance class and other class structures. We developed the class structures using UML (Rational Rose 2000© - Enterprise Edition). We plan to do the full-scale implementation in C++ and add interfaces with 3D CAD packages such as AutoCAD©, ProE©, and SolidWorks© etc, so that the end user could interact with a user-friendly graphic environment for defining the product and carry out the desired tolerance related computations. At present the graphics display and other graphics driven design/tolerance synthesis aspects are being formulated for use with SolidWorks©. Appendix – B shows a part of the class diagrams of the basic entities in the synthesis scheme.

A very skeletal (but operational) tolerance synthesis method has been implemented under the Windows© environment as a core module without Windows GUI and CAD interfaces using a database programming system (Clipper 5.2©). In order to make manipulation of the model data easy, systematic, and structured, the entire data associated with each model is stored in a relational database system with several tables for each type of object/class. The structures of these tables, along with relational links between the tables, have been shown in Appendix–C. The field names in each table have been kept nearer to their real counterparts for ease of understanding.

The program is menu driven and operates in two phases. In the first phase, the designer would build the model (artifact) using standard artifact library elements and specifying other details. After a model has been established, governing equations for each independent path are generated and then the equations are converted into a set of MatLab© input files for optimization. Cost functions and constraints for the functional requirements are formulated separately as MatLab© input files. MatLab© is then used to solve the resulting nonlinear optimization problem for arriving at a set of optimal deviation parameters.

Details of the Example Model

In Appendix – D we present details of an example model including part/feature details, input data listing, generated data (partial list), generated MatLab© input files (mostly generated from the synthesis module), and some preliminary results. It has already been mentioned that we are using MatLab© to solve the cost minimization problem. In the present case, only a sample (fictitious) cost function has been used to test if the constraints generated could be properly treated in MatLab©. The optimized output results are of little significance here since those are based on fictitious cost functions. We are now in the process of selecting realistic cost functions and various models, so that some practical tolerance limits/bounds could be computed based on the optimization results.

Acknowledgement

This work is sponsored by the SIMA (Systems Integration for Manufacturing Applications) program at NIST and the RaDEO (Rapid Design Exploration and Optimization) program at DARPA. The authors also wish thank the reviewers for their valuable comments and suggestions.

Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipments, instruments, or materials are identified in this report in order to facilitate better understanding. Such identification does not imply recommendations or endorsement by the National Institute of Standards and Technology, nor does it imply the materials or equipment identified are necessarily the best available for the purpose.

References

[BAL00] E. Ballot, P. Bourdet, *A Mathematical Model of Geometric Errors in the case of Specification and 3D Control of Mechanical Parts.* Advanced Mathematical and Computational Tools in Metrology IV, 2000 World Scientific Publishing

[BAL98] E. Ballot, P. Bourdet, *A Computation Method for the Consequences of Geometric Errors in Mechanisms.* Geometric Design Tolerancing: Theories, Standards, and Applications, Hoda A. Elmaraghy (Editor), Chapman & Hall, pp 197-207, 1998.

[DAS97] D. Dasgupta, Z. Michalewicz, *Evolutionary Algorithms in Engineering Applications*,Springer-Verlag, 1997, ISBN: 3-540-62021-4

[DON97] Zuomin Dong, *Tolerance Synthesis by Manufacturing Cost Modeling and Design Optimization,* Advanced Tolerancing Techniques, Edited by Hong-Chao Zhang, John Wiley & Sons, Inc, pp 233-260, 1997.

[DRA99] Drake, Paul, Dimensioning and Tolerancing Handbook, McGraw-Hill, 1999

[GIL91] Gilbert, Olivier L., Representation of Geometric Variations Using Matrix Transforms for Statistical Tolerance Analysis in Assemblies, Master thesis, MIT, 1992

[LIN97] Shui-Shun Lin, Hsu-Pin (Ben) Wang and Chun (Chuc) Zhang, *Optimal Tolerance Design for Integrated Design, Manufacturing, and Inspection with Genetic Algorithms,* Advanced Tolerancing Techniques, Hong-Chao Zhang (Editor), John Wiley & Sons, Inc, pp 261-281, 1997.

[MAT98] L. Mathieu, A. Clement and P. Bourdet, *Modeling, Representation and Processing of Tolerances, Tolerance Inspection: A Survey of Current Hypothesis.* Geometric Design Tolerancing: Theories, Standards, and Applications, Hoda A. Elmaraghy (Editor), Chapman & Hall, pp 1-33, 1998.

[ROY97] Utpal Roy and Ying-Che Fang, *Optimal Tolerance Re-allocation for the Generative Process Sequence,* IIE Transactions, Vol 29, pp 37-44, 1997.

[ROY99] U Roy, Bing Li. *Representation and interpretation of geometric tolerances for polyhedral objects-II. Size, orientation and position tolerances*. Computer-Aided Design 31 (1999), pp 273-285.

[ROY00a] U. Roy, Bing Li, *Tolerance Synthesis by Manufacturing Cost Modeling and Design Optimization,* Advanced Tolerancing Techniques, Edited by Hong-Chao Zhang, John Wiley & Sons, Inc, pp 233-260, 1997.

[ROY00b] U. Roy, N. Pramanik, *Tolerance Representation and Synthesis Scheme* Interim Report # 4: Submitted to NIST, March 2000.

[ROY00c] U. Roy, N. Pramanik, *System Level Tolerancing: Standards and Implementation Issues: Tolerance Representation and Synthesis Scheme* Interim Report # 6: Submitted to NIST, July 2000.

[ROY00d] U. Roy, N. Pramanik, *System Level Tolerancing: Standards and Implementation Issues: Tolerance Representation and Synthesis Scheme* Interim Report #7: Submitted to NIST, September 2000.

[ROY00e] U. Roy, N. Pramanik, and H. Wang *System Level Tolerancing: Standards and Implementation Issues: Tolerance Representation and Synthesis Scheme* Interim Report #8: Submitted to NIST, March 2001.

[ROY01] U. Roy, N. Pramanik, R. Sudarsan, R. D. Sriram and K. W. Lyons*, Function-to-Form Mapping: Model, Representation and Applications in Design Synthesis,* Journal of Computer-Aided Design (accepted for publication).

[ROY02] Utpal Roy, and Balaji Bharadwaj. *Design with Part Behaviors: Behavior Model, Representation and Application*. Journal of Computer-Aided Design (in press).

Appendix – A: An Example of Deriving Assembliability Constraints

Mathematical Model of Tolerances

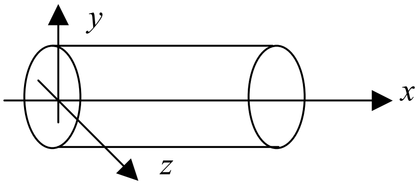
Various mathematical models for representation of components in an assembly have been studied so that the geometric variations arising out of small deviations of the features (faces) of the components could be analyzed. We adopted the small displacement torsor scheme to represent the deviation of a real surface with respect to the corresponding nominal surface at a point on a feature/face of a component. Based on this scheme, an effective representation of kinematic variations for the mating faces/features could be formulated as elaborated hereinafter.

A torsor is a tensor-type representation of small displacements and rotations (corresponding to the six degrees of freedom associated with a space point on a surface embedded in R^3). If we introduce three small displacements along the three local principal axes at a point A on a surface as $\boldsymbol{\delta} = [\delta_x \;\; \delta_y \;\; \delta_z]^T$ and three rotations $\boldsymbol{\theta} = [\theta_x \;\; \theta_y \;\; \theta_z \;\;]^T$ then the torsor $T_{A/R}$ at A with reference to a datum R can be represented as:

$$T_{A/R} = [\theta_A \quad \delta_A\,]^T = [\theta_x \; \theta_y \; \theta_z \quad \delta_x \; \delta_y \; \delta_z\,]^T.$$

Torsors transform according to standard rotation and translation rules of axes. However, the components are small in comparison to the dimension of the components.

The torsor representation of deviations of real surfaces with respect to the nominal surface for all possible regular surface geometries used in engineering design have been well studied. It has been found that, there can be some component (out of the possible six) variations along which the surface is kept invariant with respect to the nominal surface. These components are called independent components and may be varied for considering possible configurations. As for example, for a cylindrical surface, a translation along the axis keeps the surface invariant as does a o a rotation about the axis. Thus the torsor representation for the deviation is:



$$T_{A/R} = [\theta_A \quad \delta_A\,]^T = [R_x \; \theta_y \; \theta_z \quad I_x \; \delta_y \; \delta_z\,]^T$$

where *I* stands for independent movement (in this case along the axis of the cylinder) and *R* stands for free rotation (here, about the axis of the cylinder). We can arbitrarily make these quantities zero so that they do not enter into the equations. In addition, we can vary these parameters for various constraint satisfactions keeping the surfaces invariant.

The small-displacement torsor mentioned above could be used to represent three types of deviations associated with the mating of components in an assembly. Two torsors associated with each component and a gap torsor at the mating. The two torsors associated with each surface are: a displacement torsor and a deviation torsor as shown in Figure 26. The displacement torsor is an intrinsic variation torsor that is specific to the element (as for example, displacement arising out of the variation of the radius of a cylindrical surface). The displacement torsor represents the deviation of the real surface from the nominal surface. The gap torsor between two mating features represents possible geometric variations between the mating features, as depicted in Figure 26.

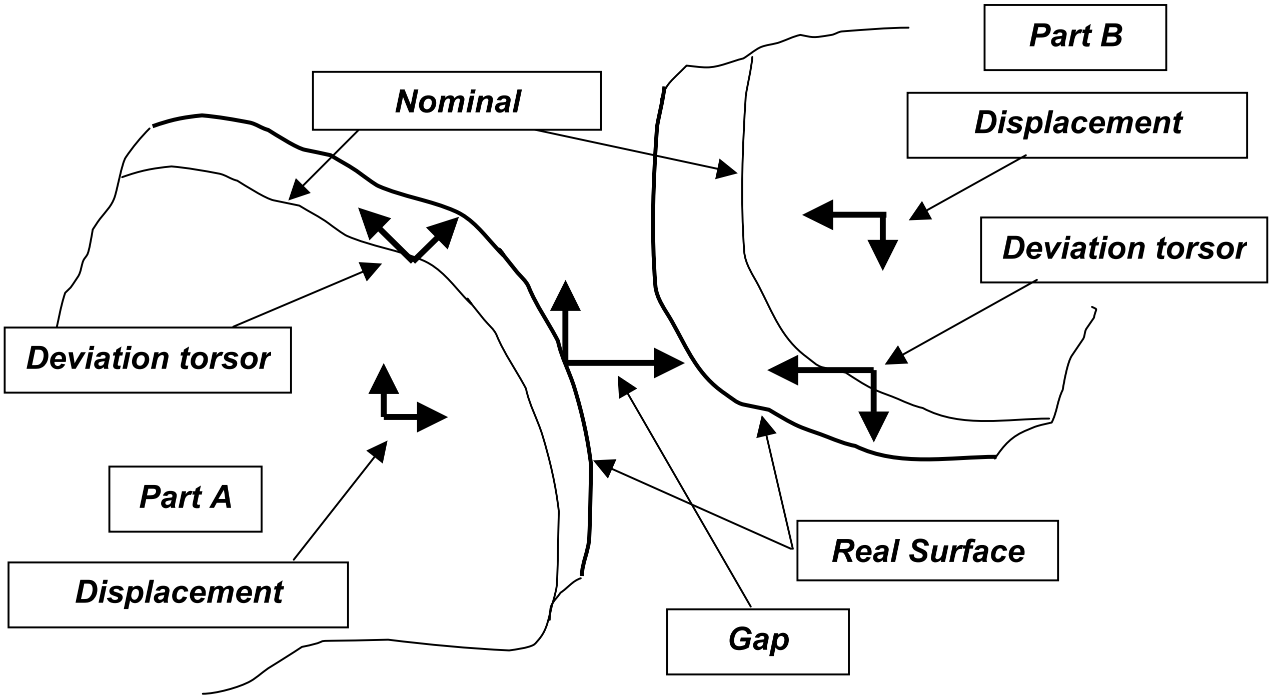


Figure 26: Interacting torsors at mating surfaces

The torsor scheme indicated above could be used to form equations based on loops connecting components in an assembly. Torsors associated with each component feature would be kept in the artifact library in the local coordinate system of the component. The gap torsor associated with two mating features would be formed based the functional and/or kinematic requirements as established in the design specifications. In order to add the torsors in a loop to form equations connecting the components of the torsors, a mechanism would be necessary to transform the torsors defined in the local coordinate systems of the individual components into a global reference frame. Since a torsor is represented by a *1x6* vector, the transformations required would be 6x6 transformation matrices.

Torsor Transformation Rules

The small displacement torsor $T_N = [\theta_N \quad \delta_N]^T = [\theta_x\, \theta_y\, \theta_z \quad \delta_x\, \delta_y\, \delta_z]^T$ defined at a point *N* on a surface could be expressed at a point *M* on a different surface by applying the required displacement and rotation to the vector *MN*. The transformation is a *6x6* matrix given by:

$$D = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & -\Delta z & \Delta y & 1 & 0 & 0 \\ \Delta z & 0 & -\Delta x & 0 & 1 & 0 \\ -\Delta y & \Delta x & 0 & 0 & 0 & 1 \end{array}\right],$$

where $(\Delta x \;\; \Delta y \;\; \Delta z)^T$ is the vector from *M* to *N*.

The transformation is given by $\boldsymbol{T_M = DT_N}$ and it can be explicitly written as

$$T_M = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \\ \delta_x + (\Delta y\,\theta_z - \Delta z\,\theta_y) \\ \delta_y + (\Delta z\,\theta_x - \Delta x\,\theta_z) \\ \delta_z + (\Delta x\,\theta_y - \Delta y\,\theta_x) \end{bmatrix} = \begin{bmatrix} \theta_N \\ \\ \delta_N + \overrightarrow{MN} \times \theta_N \end{bmatrix}.$$

The torsor scheme for the modeling of the small geometric variations of connecting features/faces of an artifact has been elaborated with a simple example of a rectangular block (Part-3) enclosed within two channels (Part-1 and Part-2) (Figure 27, Figure 28). Apart from the simplified configuration, further simplification has been done as detailed below so that the relevant transformations and preliminary computations could be checked manually.

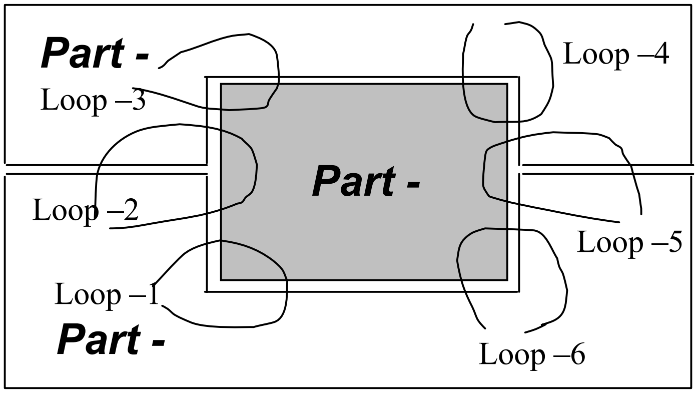


Figure 27: Example problem with three blocks

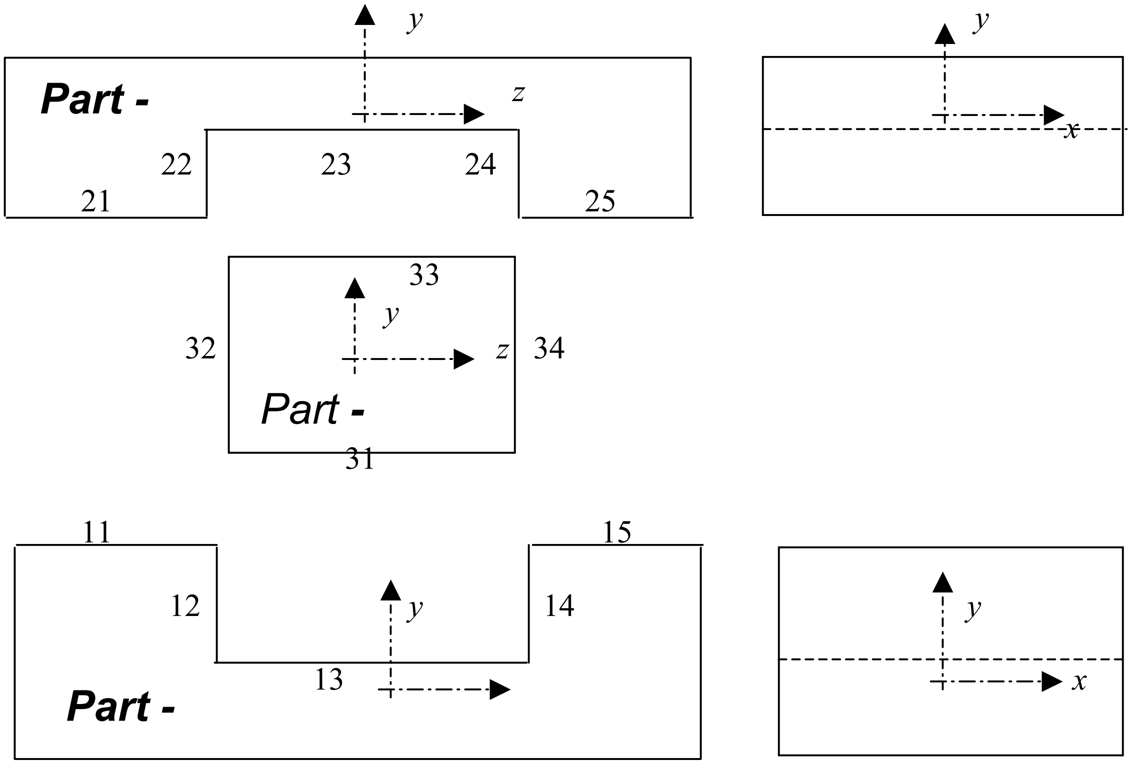


Figure 28: Example problem with feature numbers and LCS

The simplifications are:

a) All variations take place along the vertical plane only (with 3 possible variations: *x*-rotation, *y*-displacement and *z*-displacement) to keep the formulation simple.

b) There is no displacement torsor for each part. Only the surface deviation torsors and the gaps are taken into consideration.

c) The bottom surface of Part 1 has been assumed to be a fixed datum.

The three parts 1, 2, and 3, local axes and contact surfaces have been shown in the same sketch. Surfaces have been designated as <part_number><serial_number> such that surfaces on part 1 are numbered as 11, 12, 13, 14, etc. For generating equations connecting the variations of the mating surfaces, six loops have been considered (Figure 27).

With the above simplifications, we are left with following configuration:

Number of mating surfaces	:	14 (all planer)
Number of gaps between mating surfaces	:	8
Number of surface parameters	:	14 x 3 = 42
Number of gap parameters	:	8 x 3 = 24
Total number of parameters	:	42 + 24 = 66
Number of loops	:	6
Number of equations from the loops	:	6 x 3 = 18

The six loops are as follows:

Loop-1: 12→32→31→13
Loop-2: 11→21→22→32→32→12
Loop-3: 22→32→33→23
Loop-4: 24→34→33→23
Loop-5: 15→25→24→34→34→14
Loop-6: 14→34→31→13

The associated torsors in their local coordinate systems are:

$\boldsymbol{T_{11}}$, $\boldsymbol{T_{12}}$, $\boldsymbol{T_{13}}$, $\boldsymbol{T_{14}}$, $\boldsymbol{T_{15}}$, $\boldsymbol{T_{21}}$, $\boldsymbol{T_{22}}$, $\boldsymbol{T_{23}}$, $\boldsymbol{T_{24}}$, $\boldsymbol{T_{25}}$, $\boldsymbol{T_{31}}$, $\boldsymbol{T_{32}}$, $\boldsymbol{T_{33}}$, $\boldsymbol{T_{34}}$.

Typically, each of these torsors would have similar 6-component forms like:

$$\boldsymbol{T_{11} = [\theta_{11}\ \ \delta_{11}]^T = [\theta_{x11}\ \ 0\ \ 0\quad 0\ \ \delta_{y11}\ \ \delta_{z11}]^T}\ ,$$
$$\boldsymbol{T_{12} = [\theta_{12}\ \ \delta_{12}]^T = [\theta_{x12}\ \ 0\ \ 0\quad 0\ \ \delta_{y12}\ \ \delta_{z12}]^T}.$$

The gap torsors are:

$\boldsymbol{G_{11/21}}$, $\boldsymbol{G_{12/32}}$, $\boldsymbol{G_{13/31}}$, $\boldsymbol{G_{14/34}}$, $\boldsymbol{G_{15/25}}$, $\boldsymbol{G_{22/32}}$, $\boldsymbol{G_{23/33}}$, $\boldsymbol{G_{24/34}}$,

with typical forms given by:

$$\boldsymbol{G_{11/21} = [\theta g_{11/21}\quad \delta g_{11/21}]^T = [\theta g_{x11/21}\ \ 0\ \ \ 0\qquad 0\ \ \delta g_{y11/21}\ \ \ \delta g_{z11/21}]^T}.$$

We now consider the Loop - 1 (12→32→31→13) and write the torsor equation for the loop as:

$$T_{12} + G_{12/32} + (-T_{32}) + T_{31} + (-G_{13/31}) + (-T_{13}) = 0.$$

Transforming all the terms in the above equation in Part–1's frame of reference, the equation becomes,

$$T_{12} + (D_{12/23})_{12}\, G_{12/32} + (D_{32})_{12}\,(-T_{32}) + (D_{31})_{12}\, T_{31} + (D_{13/31})_{12}\, G_{13/31} + (D_{13})_{12}\,(-T_{13}) = 0,$$

where the ***D's*** are the corresponding transformations with respect to the Part - 1.

Writing the *D's* in full, and carrying out the operations, the above equation becomes,

$$
\begin{array}{llllll}
[\theta_{x12} & 0 \;\; 0 & 0 & \delta_{y12} & \delta_{z12}\,]^T & + \\
[\theta_{gx12/32} & 0 \;\; 0 & 0 & \delta_{gy12/32} & \delta_{gz12/32}\,]^T & - \\
[\theta_{x32} & 0 \;\; 0 & 0 & \delta_{y32} + \theta_{x32}(z_{32} - z_{12}) & \delta_{z32} + \theta_{x32}(y_{32} - y_{12})\,]^T & + \\
[\theta_{x31} & 0 \;\; 0 & 0 & \delta_{y31} + \theta_{x31}(z_{31} - z_{12}) & \delta_{z31} + \theta_{x31}(y_{31} - y_{12})\,]^T & + \\
[\theta_{gx13/31} & 0 \;\; 0 & 0 & \delta_{gy13/31} + \theta_{gx13/31}(z_{31} - z_{12}) & \delta_{gz13/31} + \theta_{gx13/31}(y_{31} - y_{12})]^T & - \\
[\theta_{x13} & 0 \;\; 0 & 0 & \delta_{y13} + \theta_{x13}(z_{13} - z_{12}) & \delta_{z13} + \theta_{x13}(y_{12} - y_{12})]^T & \\
\multicolumn{6}{l}{= [\,0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0\,]^T.}
\end{array}
$$

Equating each component to zero gives six scalar equations. In this present case, there are only three equations due to the restricted space of the example.

$$\theta_{x12} + \theta_{gx12/32} - \theta_{x32} + \theta_{x31} + \theta_{gx13/31} - \theta_{x13} = 0,$$

$$\delta_{y12} + \delta_{gy12/32} - \delta_{y32} - \theta_{x32}(z_{32} - z_{12}) + \delta_{y31} + \theta_{x31}(z_{31} - z_{12}) + \delta_{gy13/31} + \\ \theta_{gx13/31}(z_{31} - z_{12}) - \delta_{y13} + \theta_{x13}(z_{13} - z_{12}) = 0,$$

$$\delta_{z12} + \delta_{gz12/32} - \delta_{z32} - \theta_{x32}(y_{32} - y_{12}) + \delta_{z31} + \theta_{x31}(y_{31} - y_{12}) + \delta_{gz13/31} + \\ \theta_{gx13/31}(y_{31} - y_{12}) - \delta_{z13} + \theta_{x13}(y_{12} - y_{12}) = 0.$$

Each of the six loops will give three similar equations. Altogether eighteen equations would be formed connecting the sixty-six parameters.

Let us repeat the procedure of forming the equations for Loop–2 (11→21→22→32→32→12).

The torsor equation for the Loop - 2 is:

$$T_{11} + G_{11/21} + (-T_{21}) + T_{22} + (G_{22/32}) + (-T_{32}) + T_{32} + (G_{32/12}) + (-T_{12}) = 0,$$

i.e., $T_{11} + G_{11/21} + (-T_{21}) + T_{22} + (G_{22/32}) + (G_{32/12}) + (-T_{12}) = 0.$

Transforming all the terms in the above equation in Part–1's frame of reference, the equation becomes,

$$\boldsymbol{T_{11} + (D_{11/21})_{11}\, G_{11/21} + (D_{21})_{11}\,(-T_{21}) + (D_{22})_{11}\, T_{22} + (D_{22/32})_{11}\, G_{22/32} +}$$
$$\boldsymbol{+ (D_{32/12})_{11}\, G_{22/12} + (D_{12})_{11}\,(-T_{12}) = 0,}$$

where the ***D's*** are the corresponding transformations with respect to the Part-1.

Writing the D's in full, and carrying out the operations, the above equation becomes,

$$\begin{aligned}
&[\theta_{x11} \quad 0 \quad 0 \quad 0 \quad \delta_{y11} \quad \delta_{z11}]^T + \\
&[\theta_{gx11/21} \quad 0 \quad 0 \quad 0 \quad \delta_{gy11/21} \quad \delta_{gz11/21}]^T - \\
&[\theta_{x21} \quad 0 \quad 0 \quad 0 \quad \delta_{y21} + \theta_{x21}(z_{21} - z_{11}) \quad \delta_{z21} + \theta_{x21}(y_{21} - y_{11})]^T + \\
&[\theta_{x22} \quad 0 \quad 0 \quad 0 \quad \delta_{y22} + \theta_{x22}(z_{22} - z_{11}) \quad \delta_{z22} + \theta_{x22}(y_{22} - y_{11})]^T + \\
&[\theta_{gx22/32} \quad 0 \quad 0 \quad 0 \quad \delta_{gy22/32} + \theta_{gx22/32}(z_{22} - z_{11}) \quad \delta_{gz22/32} + \theta_{gx22/32}(y_{22} - y_{11})]^T + \\
&[\theta_{gx32/12} \quad 0 \quad 0 \quad 0 \quad \delta_{gy32/12} + \theta_{gx32/12}(z_{32} - z_{11}) \quad \delta_{gz32/12} + \theta_{gx32/12}(y_{32} - y_{11})]^T - \\
&[\theta_{x12} \quad 0 \quad 0 \quad 0 \quad \delta_{y12} + \theta_{x12}(z_{12} - z_{11}) \quad \delta_{z12} + \theta_{x12}(y_{12} - y_{11})]^T \\
&= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T.
\end{aligned}$$

Equating each component to zero gives following three equations:

$$\theta_{x11} + \theta_{gx11/21} - \theta_{x21} + \theta_{x22} + \theta_{gx22/32} + \theta_{gx32/12} - \theta_{x12} = 0,$$

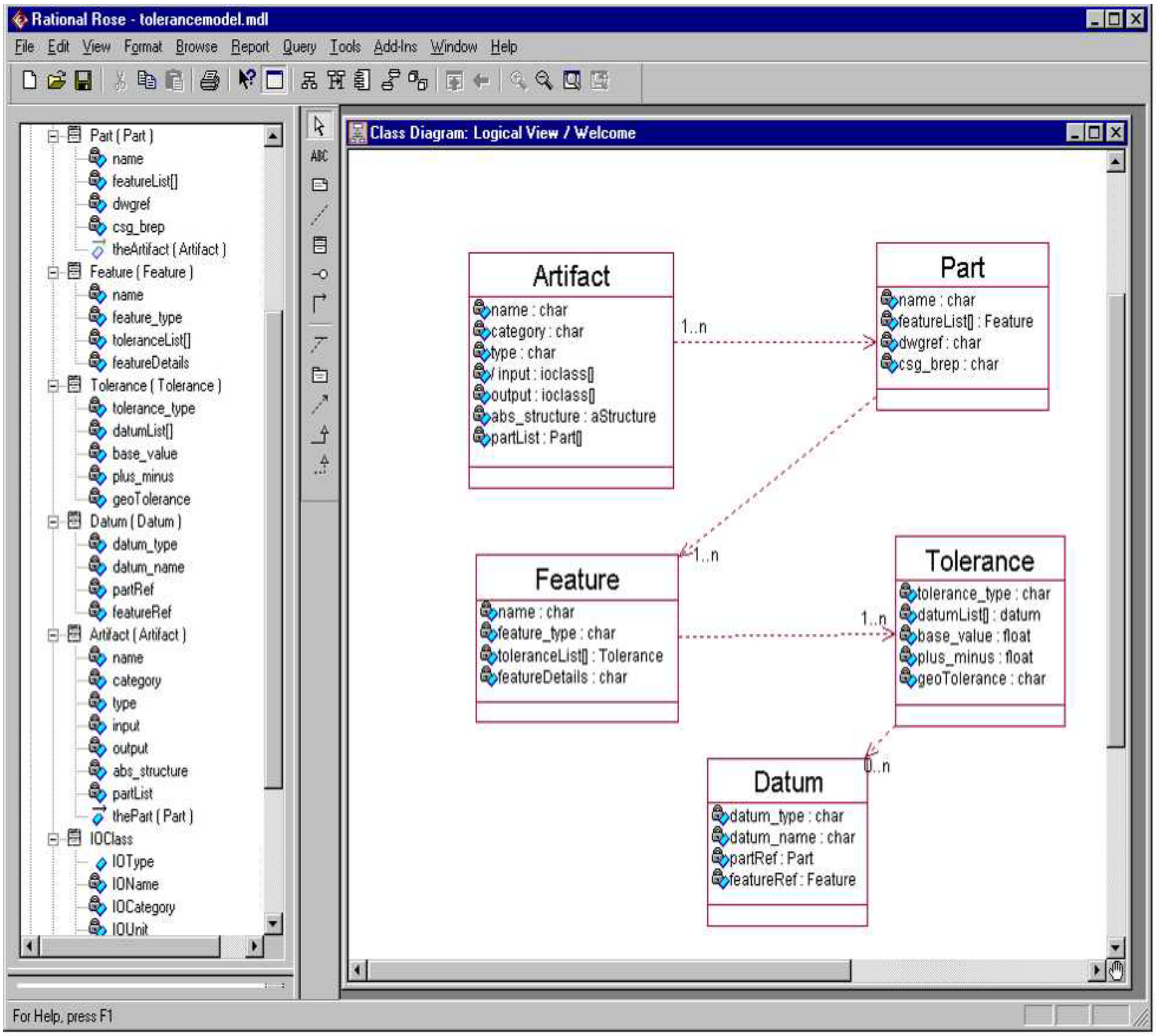
$$\delta_{y11} + \delta_{gy11/21} - \delta_{y21} - \theta_{x21}(z_{21} - z_{11}) + \delta_{y22} + \theta_{x22}(z_{22} - z_{11}) + \delta_{gy22/32} +$$
$$+ \theta_{gx22/32}(z_{22} - z_{11}) + \delta_{gy32/12} + \theta_{gx32/12}(z_{32} - z_{11}) - \delta_{y12} - \theta_{x12}(z_{12} - z_{11}) = 0,$$

$$\delta_{z11} + \delta_{gz11/21} - \delta_{z21} - \theta_{x21}(y_{21} - y_{11}) + \delta_{z22} + \theta_{x22}(y_{22} - y_{11}) + \delta_{gz22/32} +$$
$$\theta_{gx22/32}(y_{22} - y_{11}) + \delta_{gz32/12} + \theta_{gx32/12}(y_{32} - y_{11}) - \delta_{z12} - \theta_{x12}(y_{12} - y_{11}) = 0.$$

Similar procedures would yield the rest of the equations for the remaining four loops.

The gap parameters have to be generated / constrained based on the physical nature of mating between the pair of surfaces and/or design functional requirements. Converting a functional requirement into some constraint binding the gap parameters is a very complex task and currently not many generalized treatments have been published (as far as knowledge of this author goes). However, many people are carrying out work for this goal. In our present work, we have assumed that the mapping from functional requirements to constraints binding the gap parameters have been carried out, and the corresponding constraint equations have been obtained.

Appendix – B: Class diagrams of the basic entities in the synthesis scheme



Appendix - C: Artifact Database Tables and Relations

Table **ART**

Purpose Main Artifact Master File. Serves as the Artifact Library.
Both compound and Primitives are stored here.

Index ART_NAM + ART_CAT

Link -

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	Char	10	0	Artifact Name
	ART_CAT	Char	5	0	Category
	ART_TYP	Char	5	0	Type
	PRIMITIVE	Logical	1	0	Is it a primitive?
	DWGREF	Char	15	0	Drawing Reference
	LCS_NODE	Num	3	0	Local Coordinate system
	CENTER	Char	3	0	Center Node (if any)

Table **ARTLOOP**

Purpose Stores Loops and Paths that are generated for each artifact for the connectivity graph

Index ART_NAM + LOOP +LOOP_EL

Link ART --> ARTLOOP (one to many)

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	Char	10	0	Artifact name
	LOOP	Char	3	0	Loop counter
	LOOP_EL	Num	3	0	Loop sub-counter
	LOOPS	Char	254	0	Loop String

Table **ARTTREE**
Purpose The artifact link definition.
Index ART_NAM + LOOP + FROM_ART + FROM_FACE
Link ART --> ARTTREE (one to many)
Structure

Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
ART_NAM	Char	10	0	Artifact name
LOOP	Char	3	0	Loop number
FROM_ART	Char	10	0	From part
FROM_FACE	Char	3	0	From feature
TO_ART	Char	10	0	To part
TO_FACE	Char	3	0	To feature
LOOP_EL	Char	3	0	Loop element

Table **ARTEQN**
Purpose Generated list of all variables as used in the constraints equations are kept here
This table serves as a link between the variable names and its physical position
Index ART_NAM+VAR_TYP+FROM_ART+FROM_FACE+TO_ART+TO_FACE+DOF
Link ART --> ARTEQN (one to many)
Structure

Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
ART_NAM	Char	10	0	Artifact name
VAR_TYP	Char	1	0	Variable type
FROM_ART	Char	10	0	From part
FROM_FACE	Char	3	0	From feature
TO_ART	Char	10	0	To Part
TO_FACE	Char	3	0	To feature
DOF	Char	1	0	Degrees of freedom
VAR_NAM	Char	15	0	Variable name
VAR_SLN	Num	4	0	Serial number
VAR_XNAM	Char	7	0	Equivalent array element name: z(sln)
VAR_VALU	Num	10	5	Value (computed/specified)
KNOWN_VALU	Logical	1	0	Known Value?

Table **ARTF2N**

Purpose This table lists the feature to node links. Each feature of each part is defined as a set of connected nodes.

Index ART_NAM + FACE_NO + NODE_NUM

Link ART --> ARTF2N (one to many)

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	Char	10	0	Artifact name
	FACE_NO	Char	3	0	Face/feature number
	NODE_NUM	Char	3	0	node number
	DX	Num	10	5	delta-x
	DY	Num	10	5	delta-y
	DZ	Num	10	5	delta-z
	RX	Num	10	5	rot-x
	RY	Num	10	5	rot-y
	RZ	Num	10	5	rot-z

Table **ARTFACE**

Purpose Table defining each feature on a part

Index ART_NAM + FACE_NO

Link ART --> ARTPART (one to many)

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	C	10	0	Artifact name
	FACE_NO	C	3	0	Face/Feature number
	FACE_TYPE	C	3	0	Feature type
	CENTER	C	3	0	Center node(if any)
	DATUM	C	5	0	Datum reference, if any
	DOF	C	6	0	degrees of freedom
	DX	N	10	5	delta-x
	DY	N	10	5	delta-y
	DZ	N	10	5	delta-z
	RX	N	10	5	rot-x
	RY	N	10	5	rot-y
	RZ	N	10	5	rot-z
	DCX	N	8	3	orientation direction cosine-x
	DCY	N	8	3	orientation direction cosine-y
	DCZ	N	8	3	orientation direction cosine-z

Table **ARTPART**

Purpose Table of parts of each compound artifact

Index ART_NAM + ART_REQD

Link ART --> ARTPART (one to many)

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	C	10	0	Artifact name
	ART_REQD	C	10	0	Requires (artifact reference)
	TERMINAL	L	1	0	Is this a terminal?
	CENTER	C	3	0	Center (node), if any
	DOF	C	6	0	Degrees of freedom six '1' or '0'
	DX	N	10	5	delta-x
	DY	N	10	5	delta-y
	DZ	N	10	5	delta-z
	RX	N	10	5	rot-x
	RY	N	10	5	rot-y
	RZ	N	10	5	rot-z
	DCX	N	8	5	orientation direction cosine-x
	DCY	N	8	5	orientation direction cosine-y
	DCZ	N	8	5	orientation direction cosine-z

Table **ARTNODE**

Purpose Nodal coordinates of all nodes on a part

Index ART_NAM + NODE

Link ART --> ARTNODE (one to many)

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	C	10	0	Artifact name
	NODE	C	3	0	Node Number
	CX	N	10	3	x-coord
	CY	N	10	3	y-coord
	CZ	N	10	3	z-coord

Table **ARTIO**

Purpose Table of I/O (Input/Output) parameters for a part / artifact

Index ART_NAM + IONAME

LinkART --> ARTNODE (one to many)

Structure	Variable Name	Type	Len	Dec	Details/ Purpose/ Remarks
	ART_NAM	C	10	0	Artifact name
	IOTYPE	C	5	0	I/O Type
	IONAME	C	10	0	I/O Name
	IOCAT	C	5	0	Category
	IOUNIT	C	5	0	Unit
	IOVALU	N	10	3	Value

Appendix – D: Tolerance Synthesis Example Model

In this section we present preliminary results for an example artifact with three blocks (consisting of parts 1, 2, 3) as shown in Figure 29. This artifact as represented in the program is shown along with local coordinate systems for each feature in Figure 30.

The program first generates the artifact tree from the input configuration (Figure 31) and then all independent paths/loops are generated in an exhaustive manner (Figure 32). Each path/loop is used to generate assemblability constraints. As has been shown earlier these assemblability constraints are in the form $KX = b$. For ease of handling these constraints in Matlab©, a sparse matrix representation has been used for the *K*-matrix. This implementation has a built-in procedure to generate cost functions based on deviation parameters. The cost function generated for this example is shown in Figure 33.

For solving the optimization (minimization) problem as formulated in this program, we are using "**fmincon**", a nonlinear constrained optimization facility in Matlab©. The program generates a "dot m file" for running the optimization in Matlab©. Results of the optimization run are depicted in Figure 34 as a distorted shape of the original (nominal) artifact shown in Figure 30.

These deviation parameters signify the amount permissible variations of each critical feature of the assembly within the prescribed ranges of assembly and functional constraints. In order to arrive at a specific tolerance specification we need to carry out following steps:

- Identify candidate features for tolerancing,
- Determine datum reference frames (DRF),
- Decide tolerance types,
- Generate Tolerance Specification from Deviation to Tolerance Mappings for the corresponding features.

This work is in progress and we will report actual mapping to a particular tolerance specification in our next NIST IR.

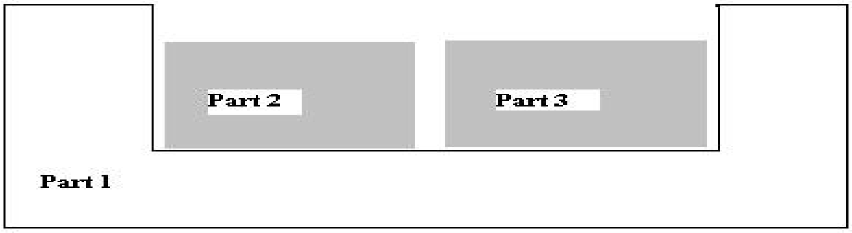
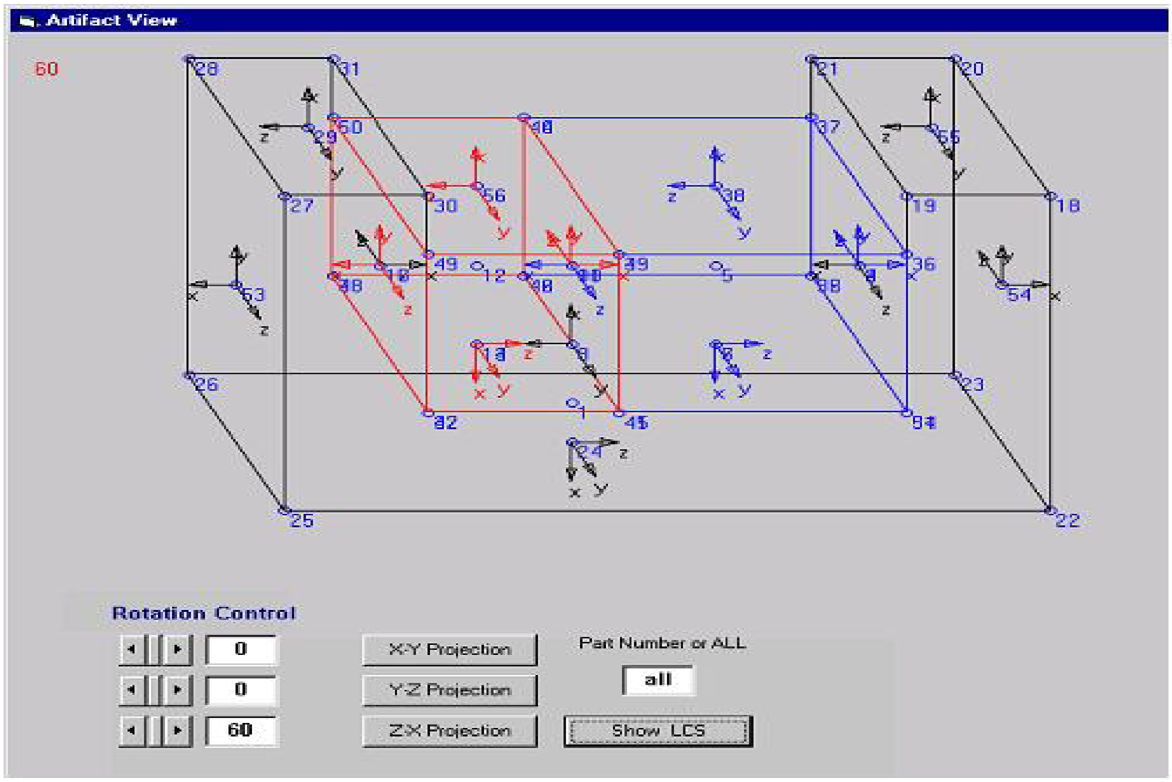


Figure 29: Block Artifact



Rotation Control: 0, 0, 60

Part Number or ALL: all

Figure 30: Nominal shape with local coordinate systems (LCS)

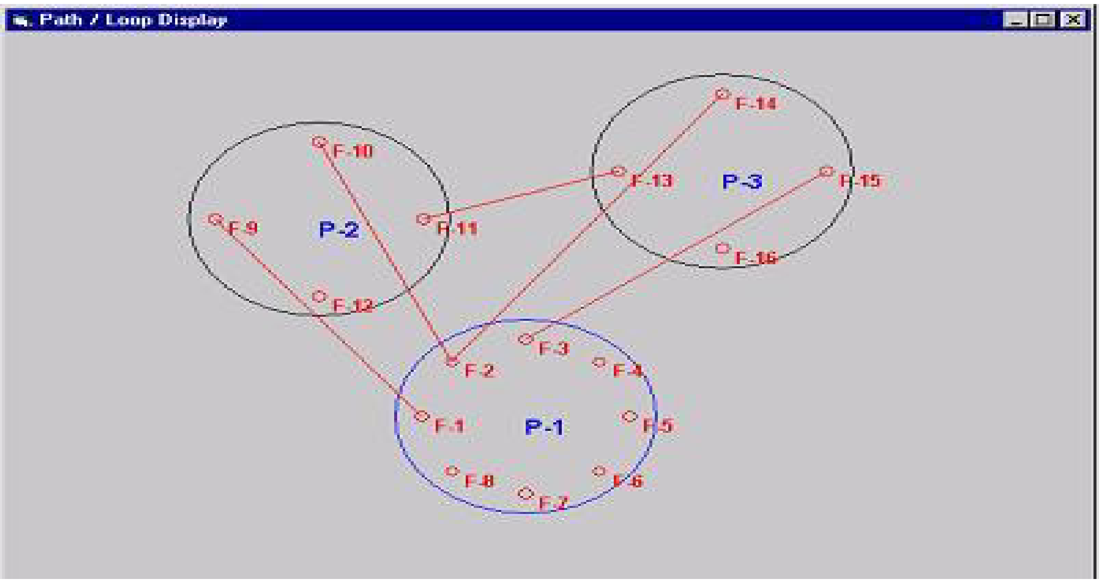
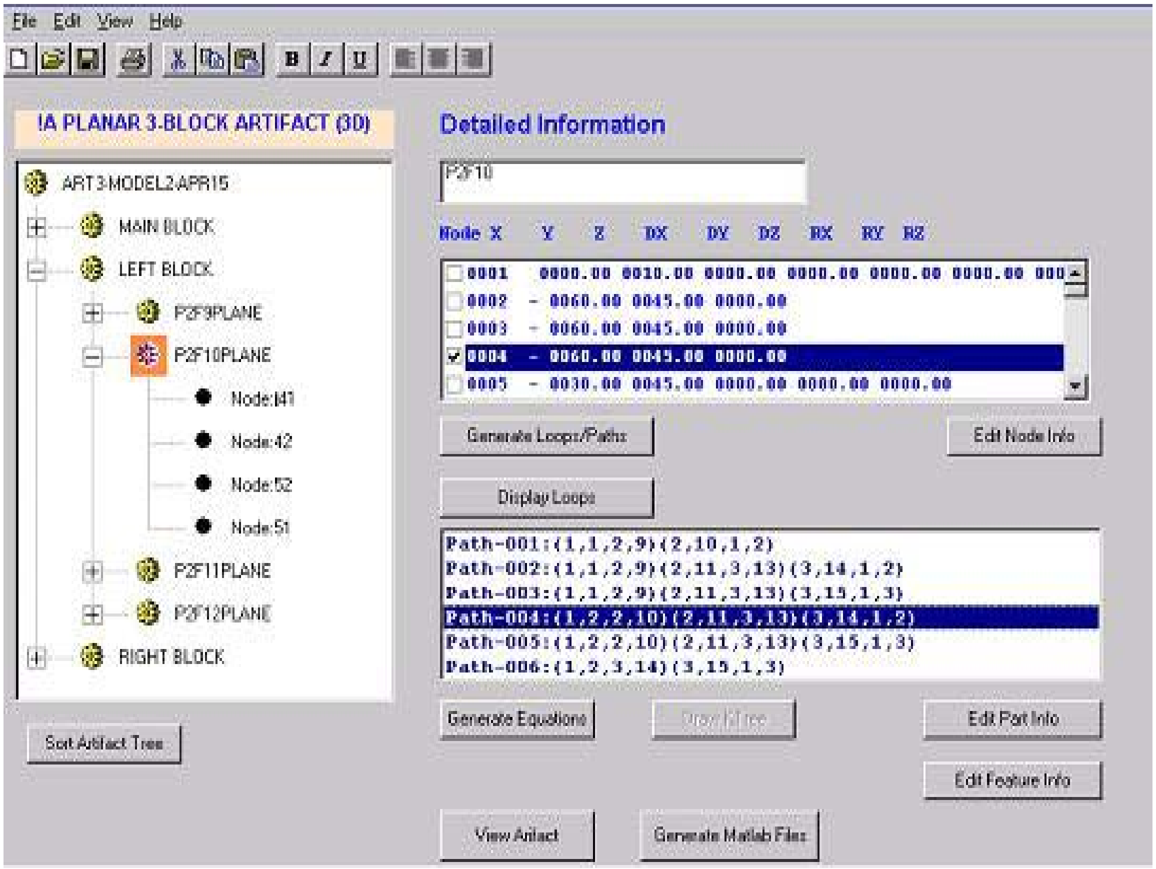


Figure 31: Feature to feature connectivity diagram

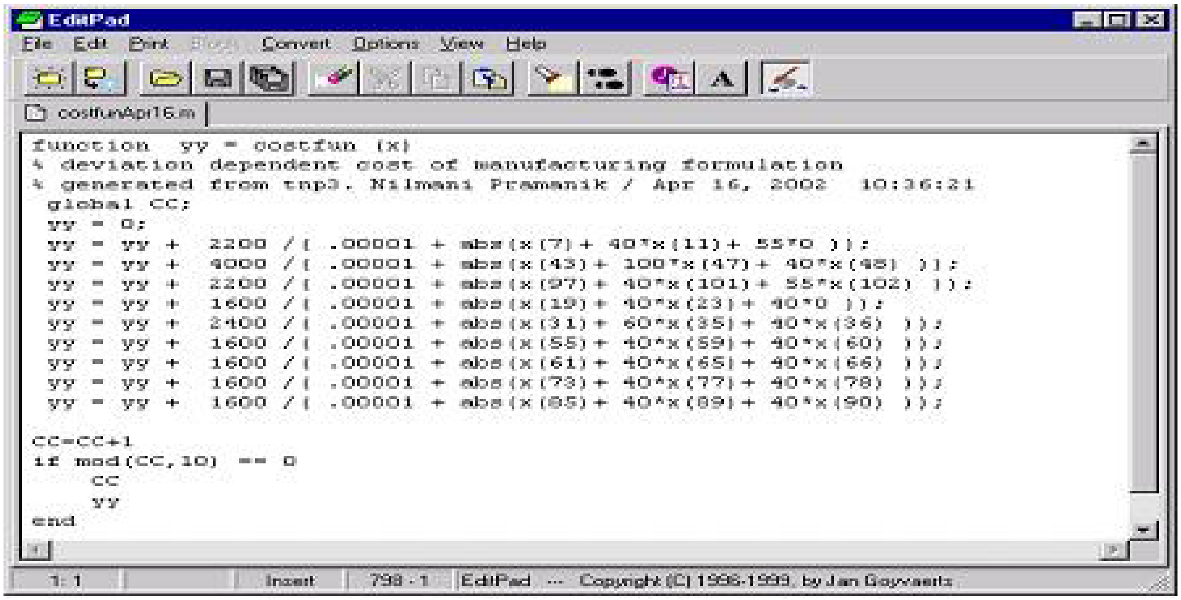


P2F10

```
Node X       Y       Z       DX      DY      DZ      RX      RY      RZ
□ 0001   0000.00 0010.00 0000.00 0000.00 0000.00 0000.00 000
□ 0002  - 0060.00 0045.00 0000.00
□ 0003  - 0060.00 0045.00 0000.00
☑ 0004  - 0060.00 0045.00 0000.00
□ 0005  - 0030.00 0045.00 0000.00 0000.00 0000.00
```

```
Path-001:(1,1,2,9)(2,10,1,2)
Path-002:(1,1,2,9)(2,11,3,13)(3,14,1,2)
Path-003:(1,1,2,9)(2,11,3,13)(3,15,1,3)
Path-004:(1,2,2,10)(2,11,3,13)(3,14,1,2)
Path-005:(1,2,2,10)(2,11,3,13)(3,15,1,3)
Path-006:(1,2,3,14)(3,15,1,3)
```

Figure 32: Artifact Control main Screen

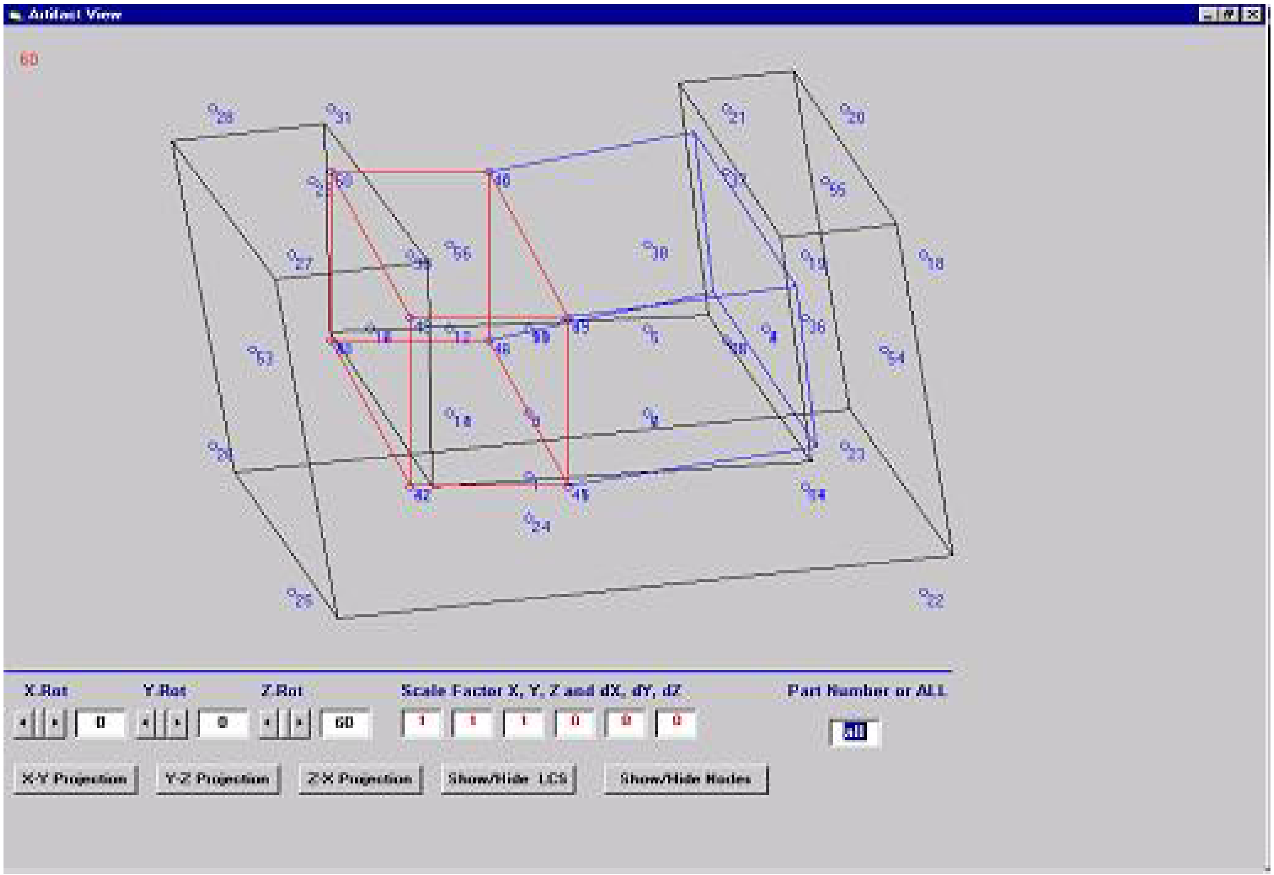


```
function  yy = costfun (x)
% deviation dependent cost of manufacturing formulation
% generated from tnp3. Nilmani Pramanik / Apr 16, 2002  10:36:21
 global CC;
 yy = 0;
 yy = yy +  2200 /( .00001 + abs(x(7)+ 40*x(11)+ 55*0 ));
 yy = yy +  4000 /( .00001 + abs(x(43)+ 100*x(47)+ 40*x(48) ));
 yy = yy +  2200 /( .00001 + abs(x(97)+ 40*x(101)+ 55*x(102) ));
 yy = yy +  1600 /( .00001 + abs(x(19)+ 40*x(23)+ 40*0 ));
 yy = yy +  2400 /( .00001 + abs(x(31)+ 60*x(35)+ 40*x(36) ));
 yy = yy +  1600 /( .00001 + abs(x(55)+ 40*x(59)+ 40*x(60) ));
 yy = yy +  1600 /( .00001 + abs(x(61)+ 40*x(65)+ 40*x(66) ));
 yy = yy +  1600 /( .00001 + abs(x(73)+ 40*x(77)+ 40*x(78) ));
 yy = yy +  1600 /( .00001 + abs(x(85)+ 40*x(89)+ 40*x(90) ));

CC=CC+1
if mod(CC,10) == 0
     CC
     yy
end
```

1: 1 | Insert | 798 - 1 | EditPad --- Copyright (C) 1996-1999, by Jan Goyvaerts

Figure 33: Sample cost of Manufacturing Function



X.Rot 0

Y.Rot 0

Z.Rot 60

Scale Factor X, Y, Z and dX, dY, dZ 1 1 1 0 0 0

Part Number or ALL all

Figure 34: Optimized configuration