



# A Constraint-Based System for the Design and Evaluation of Precision Machine Tools

by

Bradley S. Homann

Submitted to the Department of Mechanical Engineering  
on May 9, 1997  
in partial fulfillment of the  
requirements for the Degree of Master of Science

## ABSTRACT

Evaluation and optimization of precision machine tool concepts is a complex process. Most concepts are evaluated against existing machines and past design experiences, with little or no attention paid to formalized analysis methods. When formal methods are used, they are too time consuming to apply to more than one or two concepts. This results in an evaluation process where superior concepts are often overlooked or dismissed, resulting in inferior machine tool or costly modifications to a design late in the design cycle.

The lack of efficient formal methods has arisen from the belief that machine tool error motions are non-deterministic. While coupling does make prediction of error motions difficult, the majority of error motions resulting in tool-workpiece errors can be accounted for through the use of an error budget. This dissertation presents a computational tool for the rapid evaluation and optimization of precision machine tools based on parametric modeling of a given concept. Components common to many machine tools are incorporated into standard libraries, and the error motions and common design constraints associated with these components are captured within the component description. A machine design concept is defined in terms of these standard components, with interactions between the various components accounted for by standard interfaces between compatible components. Total tool-workpiece interaction errors are determined through the use of an error budget based on the Homogeneous Transformation Matrices. Design constraints are set with respect to any variable in the system, and optimization with respect to any subset of these constraints may be performed.

This results in an application that can be used to rapidly model, analyze, and optimize a series of precision machine tool concepts. Standard components may be incorporated into a variety of concepts, allowing rapid modeling of a given concept and avoiding the cost of completely respecifying the machine model every time the concept is altered. This increase in speed allows for an increased number of concepts to be evaluated, and the ability to optimize the concept for a given set of design constraints ensures a higher quality concept. This results in an increase in the design solution space and reduces the chance that a superior concept will be lost after the conceptual design phase is ended.

Thesis Supervisor: Anna Thornton  
Title: Assistant Professor of Mechanical Engineering

# Acknowledgments

My time at M.I.T. has been vaguely reminiscent of the 1997 Grand Prix of Monaco: lots of curves, no straitaways, plenty of rain, and an ending that seems a few laps premature. As in racing, individual success at M.I.T. requires the support of many people. Without their help and guidance, I would have never completed my work, or had the opportunity to thank them for their contributions to my education.

Many thanks to Leslie Regan for her constant support and concern. A brief visit to the graduate office always provided a bit of sanity and restored my motivation (in addition to keeping me out of trouble).

Thanks also to Dan Frey, for his willingness to stop whatever he was doing to help me solve my machine design problems. Dan's input cleared up more than a few problems, and provided valuable information that just doesn't seem to make it into papers or textbooks.

To all of my friends, thanks for all the good times, and the support through the bad. It's been a great two years.

Of course a big thanks to my advisor, Anna Thornton. Her input and guidance (and funding) got me started in this project and kept me pointed in the right direction. Most importantly, Anna never forgot that management is more than deadlines, schedules, and money, and for that I owe her big time.

Last, to mom and dad (who started this whole thing), thanks for many years of support, love, and understanding. You guys are the best.

Brad Homann  
M.I.T.  
May, 1997

# Table of Contents

<b>Title Page</b> .....	<b>1</b>
<b>Abstract</b> .....	<b>2</b>
<b>Acknowledgments</b> .....	<b>3</b>
<b>List of Figures</b> .....	<b>7</b>
<b>List of Tables</b> .....	<b>9</b>
<b>1.0 Introduction</b> .....	<b>10</b>
1.1 Background .....	10
1.2 Overview of Precision Machine Tools .....	13
1.2.1 Accuracy, Resolution, and Repeatability .....	13
1.2.2. Error Budgets .....	15
1.2.3. Abbe Error .....	16
1.2.4. Sensitive Directions .....	16
1.3 Overview of Precision Machine Design .....	18
1.3.1 Development of New Machines .....	18
1.3.2 Modification of Existing Machines .....	21
1.4 Existing Software for Precision Machine Design .....	22
1.4.1. Specific Machine Tool Design Applications .....	23
1.4.2. Applications for the Simulation of Machine Tool Assemblies .....	24
1.5 Research Overview .....	24
<b>2.0 Principles of Machine Tool Error Calculations</b> .....	<b>28</b>
2.1 Introduction .....	28
2.2 Sources of Machine Tool Errors .....	28
2.2.1 Geometric Errors .....	28
2.2.2 Thermal Errors .....	32
2.2.3 Loading Errors .....	33
2.2.4 Dynamic Errors .....	34
2.2.5 Summary of Error Motions .....	35
2.3 Error Motion Representation .....	36
2.3.1 Fixed Frame .....	36
2.3.2 Rotating Frame .....	37
2.3.3 Coordinate Frame Location .....	37
2.4 Error Propagation and HTM Representation .....	38
2.4.1 The Homogeneous Transformation Matrix .....	38
2.4.2 The Kinematic Machine Model .....	40
2.5 Machine Modeling and the System Error Budget .....	43
2.6 Example of the Machine Tool Design Process .....	45
2.6.1 Specification of Machine Requirements .....	45
2.6.2 Conceptual Development .....	45
2.6.3 Conceptual Evaluation .....	46
2.6.4 Example Summary .....	48

2.7 Chapter Summary .....	51
<b>3.0 Constraint-Based Design Methods .....</b>	<b>52</b>
3.1 Introduction .....	52
3.2 Design Constraint Methodology .....	53
3.2.1 Equality Constraints .....	54
3.2.2 Inequality Constraints .....	55
3.2.3 Continuous Constraints .....	56
3.2.4 Discrete Constraints .....	57
3.2.5 Constraint Network .....	57
3.3 Sources of Constraints .....	58
3.3.1 Theoretical Relationships .....	59
3.3.2 Physical Characteristics .....	59
3.3.3 Empirical Information .....	60
3.3.4 Heuristics / Design Knowledge .....	61
3.4 Constraint Example .....	62
3.5 Chapter Summary .....	66
<b>4.0 PMDA - Object Representations .....</b>	<b>67</b>
4.1 Introduction .....	67
4.2 Requirements of PMDA Objects .....	68
4.2.1. Component-Specific Parameters .....	68
4.2.2 Component-Connectivity Information .....	74
4.3 Object Hierarchy and Representations .....	76
4.3.1 Overview .....	77
4.3.2 Support Objects .....	79
4.3.3 Component Objects .....	83
4.3.4 Interface Objects .....	88
4.4 Chapter Summary .....	90
<b>5.0 PMDA - Functionality .....</b>	<b>91</b>
5.1 Introduction .....	91
5.2 PMDA Overview .....	91
5.3 Kinematic Machine Modeling .....	92
5.3.1 Creation of the Kinematic Model .....	93
5.3.2 HTM Calculation .....	93
5.4 Error Motion Modeling and Evaluation .....	94
5.4.1 Error Motion Representation .....	94
5.4.2 Determination of Total Tool-Workpiece Interaction Error .....	97
5.5 Constraint Evaluation .....	99
5.6 Concept Optimization .....	100
5.7 User Interface .....	100
5.8 Chapter Summary .....	101
<b>6.0 PMDA - Example Cases .....</b>	<b>102</b>
6.1 Introduction .....	102
6.2 The Rotary Bearing Object .....	102
6.2.1 The Rotational Motion Bearing .....	102
6.2.2 The Rotary Bearing Template Object .....	104
6.2.3 The NTN Rotary Bearing Object .....	110
6.2.4. Summary .....	112
6.3 Case Study - A Simple Machine Spindle .....	112
6.3.1 Component Selection .....	114
6.3.2 Spindle Modeling .....	117

6.3.3 Kinematic Modeling .....	117
6.3.4 Error Motion Modeling .....	117
6.3.5 Constraint Evaluation .....	120
6.3.7 Conclusion .....	124
6.4 Chapter Summary .....	124
<b>7.0 Conclusions .....</b>	<b>127</b>
7.1 Summary of Work .....	127
7.1.1 Kinematic Machine Modeling .....	127
7.1.2 Error Motion Modeling and Evaluation .....	128
7.1.3 Constraint Evaluation .....	128
7.2 Analysis of Approach.....	128
7.2.1 Advantages of the PMDA Approach .....	128
7.2.2 Disadvantages of the PMDA Approach .....	129
7.3 Future Research .....	129
7.3.1 Short Term Research .....	129
7.3.2 Long Term Research .....	130
7.4 Conclusion.....	131
<b>Bibliography.....</b>	<b>132</b>
<b>A. Sample Case.....</b>	<b>134</b>
A.1. Component and Interface Objects.....	134
A.2. Sample Case Second Run .....	134
A.3. Sample Case Output Files .....	134
<b>B. Error Motion Modeling.....</b>	<b>145</b>
B.1 Error Motion Representations in PMDA .....	145
B.2 Inclusion of Error Motions in PMDA Objects.....	145
B.3 Calculation of Error Motions in PMDA Structural Loops .....	149
<b>C. Kinematic Modeling .....</b>	<b>151</b>
C.1 HTM Calculation .....	151
C.2 Structural Loops .....	151
<b>D. Input and Output Files .....</b>	<b>155</b>
D.1. Input Files .....	155
D.2. Output Files .....	156
D.3. Sample Input and Output Files.....	158

# List of Figures

Figure 1.1: Example of Design Constraints .....	13
Figure 1.2: Accuracy, Repeatability, and Resolution.....	14
Figure 1.3: Abbe Error .....	16
Figure 1.4: Comparison of Sensitive vs Non-Sensitive Directions.....	17
Figure 1.5: General Product Design Cycle .....	19
Figure 1.6: Process for Modification of Existing Machine Tools .....	22
Figure 1.7: Overview of the Precision Machine Design Assistant.....	25
Figure 1.8: PMDA in the Machine Tool Design Cycle .....	26
Figure 2.1: Examples of Common Geometric Errors .....	29
Figure 2.2: Surface Finish and Straightness Errors.....	30
Figure 2.3: Parallelism and Perpindicularity Error .....	31
Figure 2.4: Example of Uniform and Non-Uniform Thermal Errors .....	33
Figure 2.5: Fixed Reference Frame Representation.....	36
Figure 2.6: Rotating Reference Frame Representation .....	37
Figure 2.7: Example of HTM Modeling.....	42
Figure 2.8: Sample Machine Tool Proposal.....	45
Figure 2.9: Sample Machine Tool Concepts .....	47
Figure 2.10: Concept with Specified Components.....	48
Figure 2.11: Sample Toolpoint Error Motion .....	49
Figure 2.12: Example of Toolpoint Error Prediction.....	50
Figure 3.1: Examples of Exact and Inexact Constraints .....	55
Figure 3.2: Example Constraint Network.....	58
Figure 3.3 Example of Empirical Information.....	61
Figure 3.4: Constraint Example for a Thin-Walled Pressure Vessel.....	63
Figure 4.1: Linear Error Motion Representation Example.....	73
Figure 4.2: Example of Shared Performance Information.....	76
Figure 4.3: Shaft Template and Catalog Objects .....	78
Figure 4.4: Equation Tree Representation .....	81
Figure 4.5: Standard Rotational Bearing.....	85
Figure 4.6: Standard Single-Diameter Circular Shaft.....	86
Figure 4.7: General Rotary Motion Chuck .....	86
Figure 4.8: Simple Single-Point Cutting Tool.....	87
Figure 5.1: PMDA Modeling Process .....	92
Figure 5.2: HTM Calculation Process.....	95
Figure 5.3: Error Motions in PMDA.....	96
Figure 5.4: Inner Ring Radial Runout Modeling.....	96
Figure 5.5: PMDA User Interface.....	100
Figure 6.1: Rotary Bearing Object Hierarchy .....	103
Figure 6.2: Required Rotational Motion Bearing Design Information.....	104
Figure 6.3: The Rotary Bearing Template Object.....	105
Figure 6.4: Rotary Bearing Constraints.....	106
Figure 6.5: Rotary Bearing Error Motions.....	118
Figure 6.6: Rotary Bearing Error Motion Modeling.....	109
Figure 6.7: The NTN Rotary Bearing Object.....	111
Figure 6.8: A Simple Machine Tool Spindle .....	113
Figure 6.9: Common Machine Spindle Error Motions.....	113
Figure 6.10a: Simple Machine Spindle Model.....	115
Figure 6.10b: Structural Loops for the Simple Machine Spindle.....	116
Figure 6.11: HTMs for the Simple Machine Spindle.....	118

Figure 6.12a: Toolpoint Displacement Errors for the Simple Machine Spindle.....	121
Figure 6.12b: Axial Displacement Errors for the Simple Machine Spindle.....	122
Figure 6.12c: Radial Displacement Errors for the Simple Machine Spindle.....	123
Figure 6.13: Constraints for the Simple Machine Spindle.....	125
Figure A.1: Sample Case Component Objects.....	135
Figure A.2: Sample Case Interface Objects.....	136
Figure B.1: Geometry of PMDA Error Motions.....	146
Figure B.2: PMDA Error Motion Equations.....	147
Figure B.3: Sample Error Motion Output.....	148
Figure B.4: Process for Calculating Error Motions in PMDA Objects.....	149
Figure B.5: Error Calculation Procedure.....	150
Figure C.1: Procedure to Calculate HTMs between Arbitrary Systems.....	152
Figure C.2: Example of Structural Loops.....	153
Figure C.3: Structural Loop Representation in PMDA.....	154
Figure D.1: Sample PMDA Catalog Record.....	156
Figure D.2: Sample PMDA Material Data File Record.....	157
Figure D.3a: Toolpoint Displacement Errors for the Simple Machine Spindle.....	171
Figure D.3b: Axial Displacement Errors for the Simple Machine Spindle.....	172
Figure D.3c: Radial Displacement Errors for the Simple Machine Spindle.....	173

# List of Tables

Table 3.1a: Design Variables for Iteration 1 .....	65
Table 3.1b: Constraint Evaluation for Iteration 1 .....	65
Table 3.2a: Design Variables for Iteration 2 .....	66
Table 3.2b: Constraint Evaluation for Iteration 2 .....	66
Table 4.1: Infix and Postfix Equation Representations.....	81
Table 6.1: Error Motion Values for the Example Machine Spindle .....	120
Table A.1: Error Motion Values for the Sample Case - Run #2.....	138

# **1.0 Introduction**

## **1.1 Background**

A machine tool is an integrated collection of mechanical components operating in a coordinated fashion in order to position a cutting tool with respect to a workpiece. Material is then removed from the workpiece until the desired part geometry is achieved. While the design and characterization of the individual components in a machine tool is a relatively mature field, the design, evaluation, and optimization of integrated machines is not. This is especially true at the conceptual level of design, where the majority of features, operational characteristics, and costs for a given machine are set. Berliner and Brimson [1991] state that up to 80% of the total life-cycle cost of a product is set during the conceptual stage of design. The need for increased accuracy, repeatability, and resolution, as well as a desire to decrease design cycle time and increase efficiency of the overall design process, has led to a re-evaluation of the methods used in the conceptual design of machine tools.

Traditionally, the dominant factor in the success of a machine tool design was the skill and experience of the individual designer or design team [Slocum, 1992]. Engineers and skilled machinists would develop new machine concepts based on heuristics and past design experience. Concepts were then modified and evaluated until a single concept was selected by the team for further development. Throughout the evaluation phase, basic theory or computational simulation are often ignored or used sparingly. The use of computational tools was limited to CAD drawings in the detailed stages of design, or simple calculations for individual component sizing or reaction force prediction. The use of more advanced computational techniques is limited to the detail design phase, when the geometric topologies and major operating characteristics of the machine are set. While this approach was viable for the most preliminary stages of development, any problems not identified by the design team at the conceptual level could result in a machine tool requiring expensive re-working, hand-fixturing, or modification.

The above inadequacies are a result of an incomplete exploration of the design space for the given machine tool design problem. Chan [1987] points out:

*Due to the constraints on time and resources, as well as the limited expertise available in an organization, designers are used to considering only one or two alternative. This procedure always results in a situation where a superior concept appears in the subsequent stages, but it is too late or too costly to implement it.*

The solution with respect to machine tool design is to provide designers with a comprehensive product for the rapid evaluation and optimization of design concepts. By increasing the number of concepts evaluated, the solution space is expanded and the opportunity for selection of improved concepts is increased. Unfortunately, computational tools allowing for the rapid evaluation of machine tool concepts do not exist, and the lack of suitable algorithms has precluded the creation of tools useful at the conceptual level. For this reason, current design methods continue to rely on heuristics and the experience of the design team.

A relatively new trend related to machine tool design is the retrofitting of existing machines with new, more accurate components. This process allows existing hardware to be used in order to avoid the investment of a new machine. For example, the replacement of gear-driven actuators on an existing milling machine with linear motors can increase the accuracy of the machine while avoiding the cost of purchasing a new milling machine and reducing maintenance costs. While the design and integration of these new components offers a much different challenge than the design of a completely new machine, the methods for evaluating these modifications is similar to the HTM approach described above. The influence of existing machine errors and operational conditions must be accurately accounted for, and the new components must be selected to ensure proper operation with existing components. As with constraint-based design (discussed above), formal tools and methods for the retrofitting of machine tools do not exist, and designers must still rely on experience and intuition for many retrofitting decisions.

The above approach for machine tool design arose from the belief that interactions between the components comprising a machine tool were not deterministic or predictable. However, while some aspects of machine tool behavior are not well understood, the majority of influences on overall machine errors are deterministic and can be predicted. Bryan [1984] states:

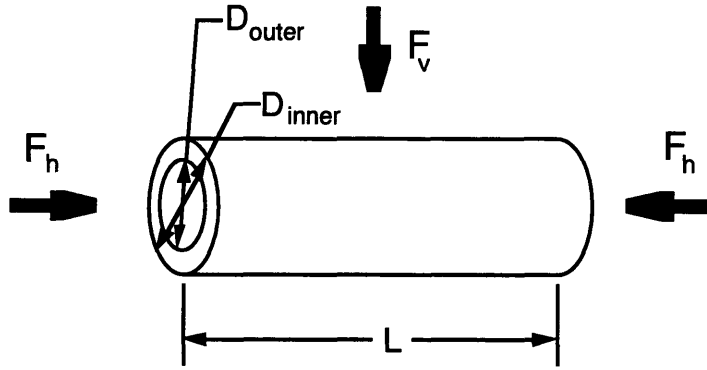
*The basic idea is that machine tools obey cause and effect relationships that are within our ability to understand and control and that there is nothing random or probabilistic about their behavior. Everything happens for a reason and the list of reasons is small enough to manage.*

Recent improvements in design methods propose basic machine error simulation and concept evaluation through the use of machine modeling via Homogeneous Transformation Matrices (HTM) on a component-by-component basis. By modeling each component as a rigid body, including the component errors in the local component coordinate frame, and then transforming these errors to a reference coordinate system, the total tool-workpiece interaction error for a given machine tool concept may be accurately predicted for a variety of operating conditions.

Errors that are easily accounted for, such as elastic beam-bending or thermal deflections, are determined theoretically, while more difficult or esoteric errors, such as joint interactions or complex dynamic errors, may be predicted and included based on experience or empirical data. This allows designers to evaluate a concept against the total allowable errors (error budget) for the machine. By directly comparing the errors associated with each proposed concept versus the error budget, a designer may determine the performance of a particular concept or compare a series of concepts against each other. This allows the designer to expand the solution space and evaluate concepts based on sound engineering theory, and avoids the need for decision-making based purely on intuition or experience.

While the above method allows for reasonable predictions of machine tool errors, the process for developing an HTM model is a complex and time consuming process [Frey, 1997]. Spreadsheet programs or general mathematical software is used for almost all necessary calculations and simulations; currently no commercially-available software packages allow for the explicit creation of HTM models. Any modification to the design concept requires extensive modification to the HTM model, and evaluation of independent concepts requires independent models. Information on component error motions, accuracy, or expected performance must be included manually, and comparison of individual components performing the same operation also require remodeling by the designer.

Another shortcoming of existing machine tool design methods is the lack of design optimization. To thoroughly evaluate several machine concepts, the effect of many factors, such as catalog-based parts, limitations on cost, or ease of manufacturability, must be included. These factors can be seen as a series of constraints on the design, and optimization subject to a complete set of constraints for a given concept yields the optimum design for that concept. While most traditional design methods are based on the definition and satisfaction of constraints, it is impossible to include all but the major constraints for a given machine concept without the assistance of computational methods. For example, the simplified connecting rod shown in Figure 1.1 requires 9 constraints to partially limit the design to a feasible design space. Increased complexity of the part, inclusion of other factors besides those listed, or a more complete limitation of the design space will result in a dramatic increase in the number of required constraints; as a result, non-computational methods become impractical or impossible. Unfortunately, few constraint-based design tools exist outside of the research environment, forcing the machine tool designer to invest the time necessary to develop an application in-house, or not leverage constraint-based methods.



**Constraints:**

**Geometry:**

- $L \geq L_{\min}$
- $L \leq L_{\max}$
- $D_{\text{outer, min}} \leq D_{\text{outer}} \leq D_{\text{outer, max}}$
- $D_{\text{inner, min}} \leq D_{\text{inner}} \leq D_{\text{inner, max}}$
- $D_{\text{inner}} \leq D_{\text{outer}}$

**Loading:**

$$F_{h, \max} \leq \sigma * \Pi((D_{\text{outer}} - D_{\text{inner}}) / 4)$$

$$F_{v, \max} \leq \sigma * L((D_{\text{outer}} - D_{\text{inner}}) / h)$$

**Operational:**

- $\text{RPM} \leq \text{RPM}_{\max}$
- $T \leq T_{\max}$
- Material must be compatible with support bearings

Figure 1.1: Example of Design Constraints

Recent advances in object-oriented architecture provide an opportunity for the automation of the evaluation and optimization process for machine tool concepts. By encapsulating machine tool component information in computational objects, the majority of engineering information necessary for the development of computational methods for machine design may be stored and accessed as needed. The following sections provide a more detailed overview of the existing machine tool design and retrofitting process, and offer some insight into the challenges facing machine designers.

## 1.2 Overview of Precision Machine Tools

### 1.2.1 Accuracy, Resolution, and Repeatability

The performance of a given machine tool is often measured by three quantities: accuracy, repeatability, and resolution. By classifying machines by these three quantities, the designer may directly compare the performance of a series of machines, or evaluate a specific

concept against the required tolerances of the machined part. Figure 1.2 contains a graphical explanation of these three quantities.

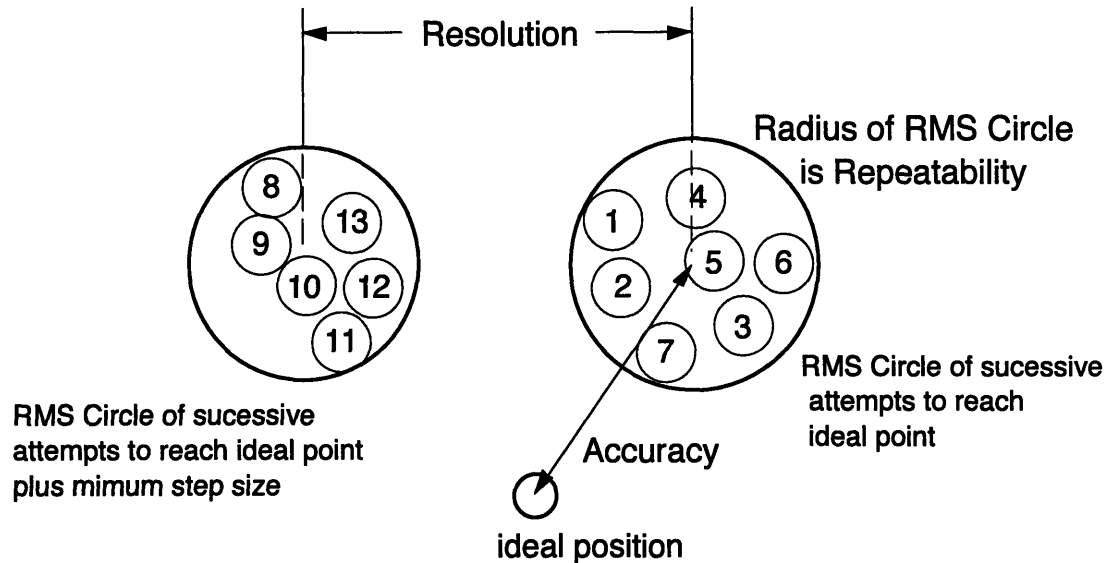


Figure 1.2: Accuracy, Repeatability, and Resolution

Accuracy is defined as the maximum error between any two positions in the working volume of the machine. For example, the accuracy of a Coordinate Measuring Machine (CMM) may be determined by programming the probe to translate to a certain location. The difference between the programmed location of the probe (measured with respect to a base coordinate system) and the actual probe location is the accuracy of the machine. Accuracy may be defined in terms of translational or rotational error motions, and is often measured at various locations within the working volume of the machine to determine variations in accuracy with respect to location within the volume.

Repeatability is the error between successive attempts to position the tool at a given location in the working volume. The maximum deviation between any of a series of attempts is the total repeatability of the machine. Because of the nature of the measurement, repeatability is often referred to in terms of probability of tool placement within a certain limit of the ideal (desired) location. Repeatability may be classified in terms of uni-directional or bi-directional approaches, with the latter accounting for backlash, hysteresis, and other directional effects.

Resolution is a measure of how precisely the machine can be controlled. The largest of the smallest programmable step sizes is the resolution of the machine. Resolution may be

thought of as a measure of the ideal performance of the machine, since the machine cannot achieve a smaller step size without the influence of errors in the individual components of the machine.

A precision machine tool differs from a standard machine tool in the tolerances placed on the above quantities. While there is no defined point for transition between standard and precision machines, the use of "precision" implies the ability to achieve tighter tolerances and a desire for highly accurate final part geometries. A typical number used to define the precision necessary for classification as a precision machine tool is an accuracy of 5.0  $\mu\text{m}$  or greater [Frey, 1997]. Any machine tool accurate to this general limit may be thought of as a precision machine tool.

### **1.2.2. Error Budgets**

One method of defining and evaluating the performance of a machine tool concept is through the use of an error budget. An error budget allows the designer to assign values to the errors associated with each component and determine the overall effect of these component errors on the total error at the tool-workpiece interface point. This result is then compared to the desired performance of the machine, and modifications to the machine concept are made if the error are considered unacceptable. The components most heavily influencing the tool-workpiece error may also be determined through the error budgeting process, allowing for concentration on the "problem areas" of the machine. Limits on best and worst case machine performance may also be determined by varying the performance of the individual components in the machine.

The first step in creation of an error budget is to develop a mathematical model of the complete machine tool. Each component is modeled as a rigid body, with the geometric relationships between components described by Homogeneous Transformation Matrices (HTM). The details concerning HTM modeling are described in Section 2. Once the model is completed, errors associated with each component are included and the effect on the tool point location (with respect to the workpiece) is determined. Total machine performance is determined by including all of the individual component errors at once, while the effects of each component can be determined by including the errors of each component individually. Upper and lower bounds on the error estimates for the machine may be determined by altering the individual component errors for best and worst case performance. Optimization of the concept takes place through the iterative process of evaluating the machine model and modifying the errors of each component until the desired level of performance is achieved.

As stated above, the major drawback of an error budget is the effort and time required to construct the machine model. A typical HTM model for a precision machine tool can require up to one month to construct and test [Frey, 1997]. Any changes made to the concept must be reflected in the model, and even small changes in the concept could result in major modifications requiring days to implement. Because of this, many designers avoid error budgets and rely on intuition and experience to evaluate concepts, or develop overly-simplistic error budgets of limited value. This practice often results in an inadequate investigation of a concept, and increases the chance that a problem in the design will not be corrected, or that a superior concept will not be selected for further development.

### 1.2.3. Abbe Error

Abbe error is the amplification of an error by a lever arm [Slocum, 1992]. A graphical example of Abbe error (also referred to as parallax error) is shown in Figure 1.3. In this example, a relatively small deflection at the bearing results in a large deflection at the end of the shaft. For this reason, Abbe error effects must be considered in the evaluation of any machine tool concept, and efforts must be made to minimize the amplification of errors due to a lever arm.

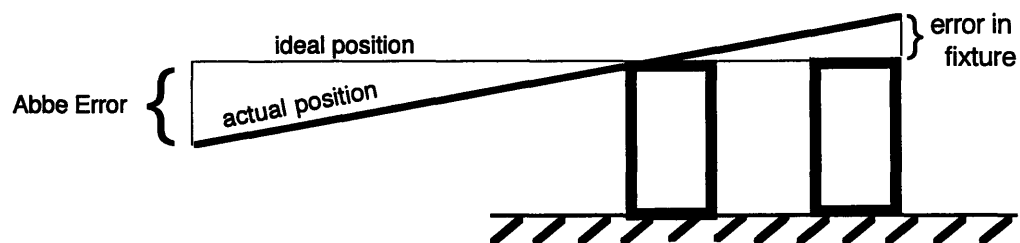
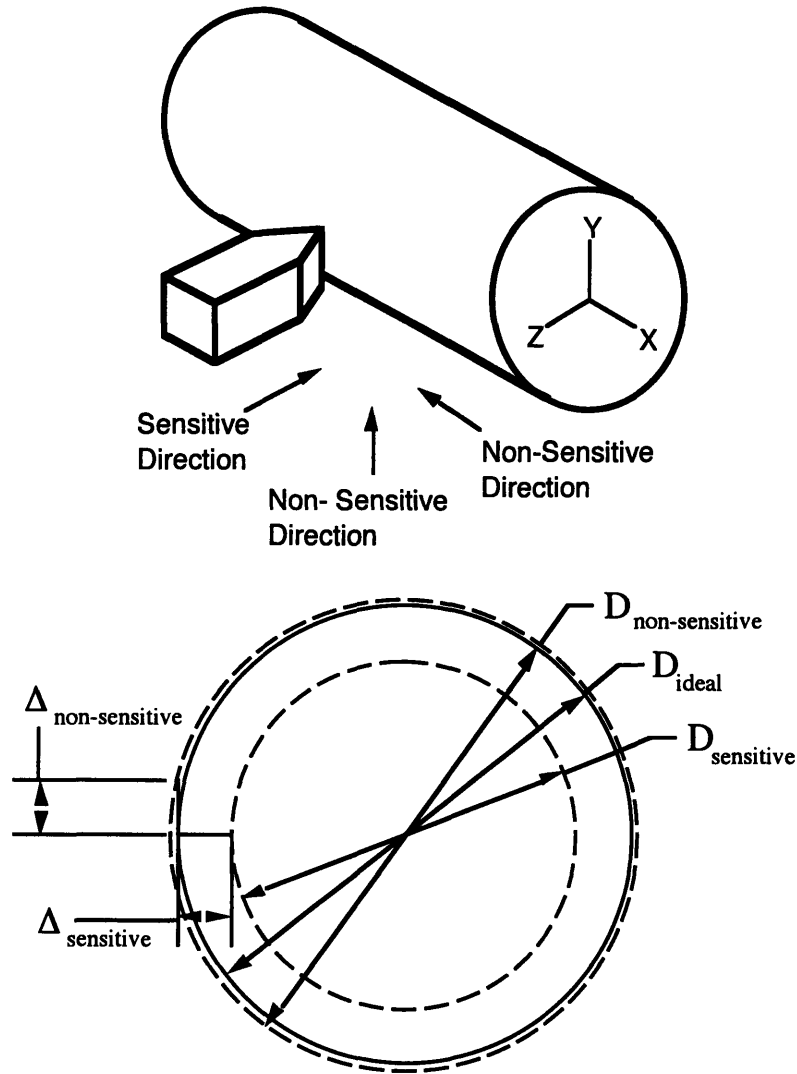


Figure 1.3: Abbe Error

An HTM-based error budget will account for Abbe error and allow the designer to determine which components are affected by or result in angular amplifications. For the conceptual design phase, however, a designer should attempt to develop concepts that are not sensitive to or dominated by Abbe effects.

### 1.2.4. Sensitive Directions

Tool-workpiece interaction errors are often described in terms of displacements and rotations in a coordinate frame attached to the tool. Figure 1.4 shows an example of these



$$\text{For: } \Delta_{\text{sensitive}} = \Delta_{\text{non-sensitive}}$$

$$\Delta D_{\text{sensitive}} = 8.4 * \Delta D_{\text{non-sensitive}}$$

**Figure 1.4: Comparison of Sensitive vs Non-Sensitive Directions**

errors for a tool-workpiece interaction for a simple metal lathe. The effect of two of these errors on the part is also shown in the figure. While both errors are assumed to be of equal magnitude, the resulting workpiece errors are not. In this example, a displacement along the Z-axis of the cylinder of magnitude  $\Delta$  results in a workpiece of diameter ( $D_{\text{ideal}} - 2\Delta$ ). A displacement along the Y-axis of  $\Delta$ , however, results in a diameter of ( $D_{\text{ideal}} + (0.24*\Delta)$ ). The error along the X-axis results in no diameter variation, and is also considered non-sensitive. This would not be the case, however, if the cutting tool was to remove material up to a specific point on the workpiece.

The above example illustrates the concept of sensitive directions. A sensitive direction is one in which an error motion of the tool will result in a substantial workpiece error (in terms of the total workpiece error). Analysis of sensitive directions provides an indication of what errors a designer should concentrate on; reducing  $\Delta$  in the above example would result in more accurate parts, while reducing  $\beta$  will have little effect on part accuracy (for the above example). For this reason, errors in non-sensitive directions may be neglected, and unnecessary modifications may be avoided. In practice, all errors are included in the error budget and sensitive directions are accounted for after the total error is determined. For conceptual development a designer must be aware of the issues of sensitive directions and avoid any designs that would result in an increase in sensitive-direction errors.

### **1.3 Overview of Precision Machine Design**

This section covers the existing methods for the development of precision machine tools. Topics include the process for design of new machines, the process for remanufacture and retrofit of existing machines, and the software tools currently available for use in precision machine design. An example of the complete machine tool design process, as well as the development and evaluation of an error budget, is presented in Chapter 2.

#### **1.3.1 Development of New Machines**

The design cycle for a new machine begins with the establishment of a set of operations or functions the machine will be required to complete. Once the required functions are identified and refined, and a target market is set, the design team develops a series of concepts capable of meeting the requirements. After successive iterations and evaluations of the concepts, the team selects one design for detail development and production. Prototyping and production phases follow. The design cycle is considered complete when the production run of the machine is concluded or no further modification to the machine is allowed. The individual phases of the design cycle are shown graphically in Figure 1.5.

#### **Concept Development**

The starting point for a machine design is the conceptual design phase. During this phase, the design team attempts to coalesce current technology into a machine that will satisfy all of the prescribed functional requirements. Ulrich and Eppinger [1995] describe a concept as

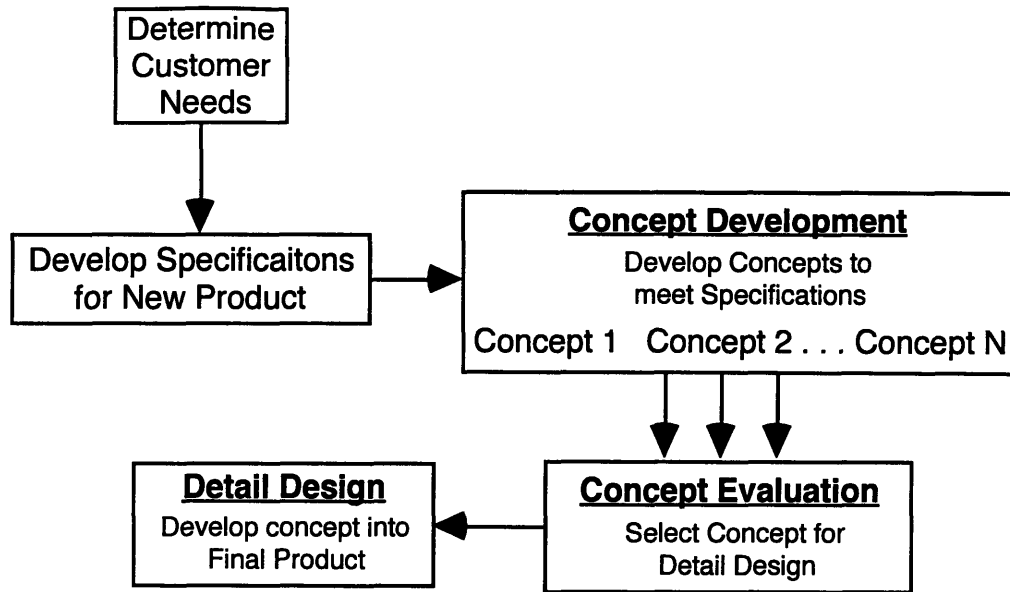


Figure 1.5: General Product Design Cycle

"a description of the form, function, and features of a product [p 16]". It is during this phase that the majority of major design decisions concerning the overall shape, function, performance, and cost of the machine will be made.

Concept development includes the determination of the overall size and layout, required degrees of freedom, and number and basic type of bearings, actuators, and drive components. Other factors, such as part fixturing, lubrication concerns, structural materials, and operator access are also considered. Concepts are generally developed as a series of sketches or models describing how the final integrated machine tool will operate. The analysis performed at this stage of development is limited to rough calculations for force and load estimates, preliminary component sizing, and rough structural geometry.

During this phase of design, the design team must rely on experience and engineering skill to devise concepts that will be able to perform the required tasks. While the analytical relationships necessary for concept evaluation exist, there are currently no analytical methods for concept generation. Often, new machine concepts are a combination of existing technology proven in previous application and new technology developed since the last generation of machines. The lack of formal methods for the concept generation phase is offset by the relatively low cost of generating concepts; this generally results in a large number of potential concepts from which a candidate for the detail design phase will be selected.

## Concept Evaluation

Concept evaluation is the process of determining the feasibility and expected performance of the concepts developed in the earlier phases of design. Estimates on machine accuracy, repeatability, resolution, cost, and other factors will be made for each concept. During this time, the initial assumptions made concerning component parts, materials, manufacturing, and operation are analyzed further to try and determine the problems or shortcomings of a given concept. A preliminary error budget (described above and illustrated in Section 2.5) is developed to assist in the evaluation of the error motions of each concept. Machine proportions are refined to more closely match the desired final geometry of the machine. Tests concerning the feasibility of unverified processes or components are conducted, and any analysis of existing machines is performed. Sensors and control systems are also selected during this stage, although these selections are only a rough indication of the types of systems to be used. This phase of design is completed when a final concept is selected for the detail design phase.

It is at this stage of development that traditional methods are least robust. Often, the conceptual evaluation process is dependent on the experience and intuition of the design team and not on formal analysis methods. The volume of information necessary to properly evaluate a concept often results in only a small number of the original concepts being analyzed in an in-depth fashion. The time for development of a machine error budget is also a limiting factor, and each machine concept requires an almost completely new budget in order to accurately determine the expected machine performance. Manual development of an error budget also does not allow for substitution of various standard components in the model, since each component must be modeled as a rigid body and any changes are reflected throughout the entire model. For this reason, error budget are rarely developed for more than one or two of the most promising concepts.

It is also during this phase of design that constraint-based evaluation methods are most useful. In contrast to general constraint-based design, constraint-based evaluation methods rely only on mathematical constraints. For example, the statement *if a shaft exist, then it must be supported* is a production rule constraint [Thornton, 1993]. A mathematical constraint is a statement involving design parameters generally expressed in terms of a mathematical equation. For example, the length of a shaft must be equal to or larger than the distance between the supporting bearings,  $L_{\text{shaft}} \geq D(b_1 - b_2)$ . By setting a series of mathematical constraints on the design parameters of a given concept, the solution space for the concept is more clearly defined. As each constraint is added, the size of the solution space is decreased, until all design parameters are interconnected in a network of constraints resulting in a well-defined

solution space. The concept may then be optimized using standard or custom component libraries to provide the optimal configuration for the given constraint network. A more complete overview of constraint-based methods is offered in Chapter 3.

Because of the complexity of the above methods, conceptual evaluation is often based on the skill of the design team, and formal determination of machine performance through the use of error budgets, constraint-based evaluation, or similar methods is limited to only the concepts that appear to be the most promising. It is this lack of exploration of the design space for a given machine that often leads to selection of an inferior concept. For this reason, a formalization of the above design methods would yield an improvement in the conceptual evaluation process and an increase in the probability of selecting the most capable of a series of concepts.

## **Detail Design**

Detail design consists of the final selection of the actual components, materials, and processes that will comprise the new machine. The geometry is fixed and final drawings for the manufacture of each custom component are prepared. All support equipment is specified, and any remaining design work is completed. At this stage in development, a solid model of the final concept is often developed to evaluate operational clearances and other geometric concerns. Computational simulations, including dynamic modeling and finite-element analysis, are also conducted at this time to ensure the machine will perform as required. Any conflicts are resolved before prototyping begins. The prototype machine is then built and tested, and any required changes or modifications are made before production begins. In special cases in which a custom machine is produced or production volume is low, the prototyping phase may be eliminated completely and the first machine produced is considered part of the production run. It is also standard procedure for a prototype machine to be delivered to a customer if the prototype performs to the required level of accuracy.

### **1.3.2 Modification of Existing Machines**

A relatively new problem in the field of precision machine design is the modification of existing machines for the purpose of increasing machine performance and lifetime. Generally, drive and/or bearing systems are replaced to increase the accuracy and resolution of the machine tool, or to increase the operational speed of the machine axes. This allows for increased performance of the machining process related to the machine without the expense of purchasing a new machine tool.

The process for modifying an existing machine is shown in Figure 1.6. First, a kinematic model of the existing machine tool is constructed. Evaluation of this model provides an indication of the current performance of the machine tool, and allows for

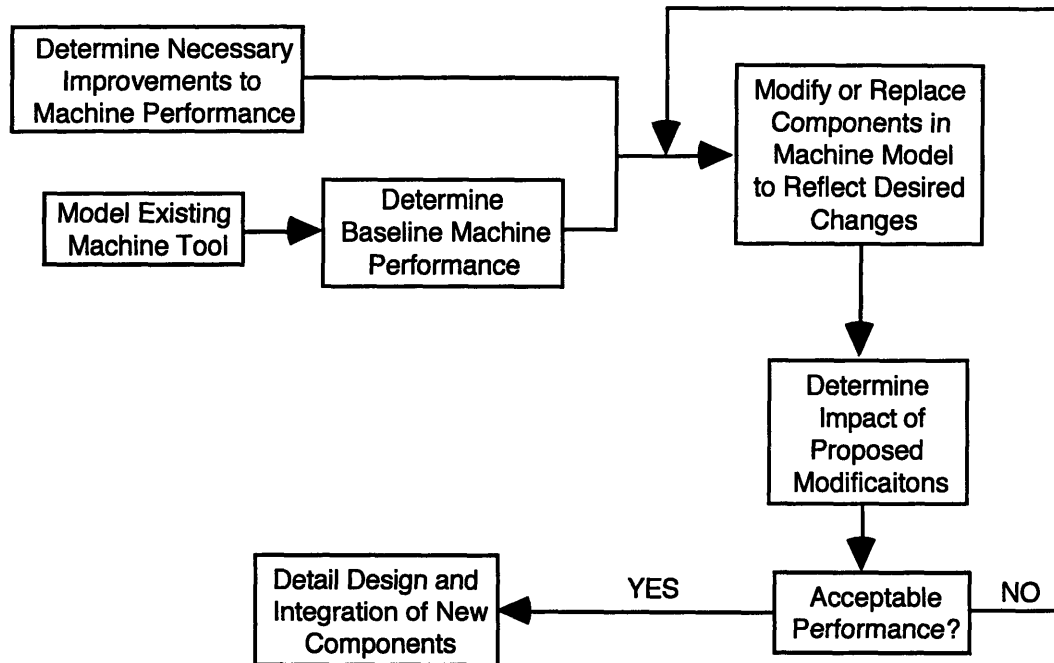


Figure 1.6: Process for Modification of Existing Machine Tools

the determination of a "baseline" of the performance of the machine tool. This model is then modified to reflect the changes resulting from the replacement or addition of new components. The new model is evaluated to determine the performance of the modified machine, as well as the feasibility of including the selected components. If the performance increase is acceptable, the proposed changes proceed to the detailed design phase (which is similar to the detail design phase described above). Unacceptable performance indicates that the selected components will not perform to the required level of performance, or that the integration of these components into the complete machine tool is not correct. By following the above process, the designer can quickly and confidently determine the modifications necessary to gain the desired increase in machine tool performance.

#### 1.4 Existing Software for Precision Machine Design

This section presents an overview of the existing software available for design and analysis of precision machine tools. Three major types of software currently exist: specific

applications for the design of machine tools, packages for the simulation of machine tool assemblies and sub-assemblies, and general software developed for non-machine tool specific applications.

#### **1.4.1. Specific Machine Tool Design Applications**

Of the existing packages developed specifically for the design of machine tools, all are non-commercial applications developed for research purposes. This section presents an overview of several of these packages:

- Chakrabarti et al. [1992] developed a system for the generation of machine tool concepts through the combination and adaptation of existing elements. The system captures the functionality of existing mechanisms and sub-assemblies and generates new concepts based on combinations and adaptations of these elements. Evaluation of these concepts is the responsibility of the designer.
- Gorti and Sriram [1995] developed a system for the geometric design of a part based on function-symbol and symbol-form mapping. Constraints, standard components, and design process knowledge are combined with geometric primitives to define the shape of the desired part.
- Li et al. [1996] presented a system for the kinematic design of mechanisms based on heuristic and qualitative representations. Devices are represented within the system and selected based on evaluation of the user-input requirements with the functional descriptions contained within the program.
- Colton and Dascanio [1991] develop the Intelligent Design System (IDS) for the geometric design of parts. The system allows the designer to specify geometry and function based on a library of features. These features contain geometric as well as functional data. Material type, manufacturing process, and other information may also be included in the part representation. Evaluation and verification of the design takes place in separate applications.
- Kannapan and Marshek [1990] propose a system for the modeling machine elements based on first order predicate and algebraic logic. These models are collected into libraries for use in the conceptual stages of design. Libraries may be used by designers

or to provide information for design verification, selection, and synthesis systems based on computational routines.

#### **1.4.2. Applications for the Simulation of Machine Tool Assemblies**

The simulation of a complete machine or machine sub-assembly requires detailed knowledge of the system in question, and is often reserved for the detailed stage of design. These packages may, however, be used for conceptual evaluation if enough detail is present in the concept description. Simulation may also be used to determine the feasibility of a sub-assembly or component for use in a complete machine.

- Spur et al. [1990] developed a system for the evaluation of machine tool designs based on a library of machine sub-assemblies and workpieces. The designer begins by selecting a desired workpiece profile, defining various parameters of the workpiece, and selecting the desired process. A three-dimensional geometric model is then constructed of the machine based on a standard component library, and simulation of the machining process provides details on the operation of the tool. Results are limited to kinematic descriptions of the machining process.
- Mahajerin [1988] presents an application for the simulation of spindle growth due to temperature effects. The spindle is modeled as a solid shaft with two bearings at a given location. The simulation provides information on the expected deflection of the spindle due to temperature rise in the bearings.
- The Machine Tool - Agile Manufacturing Research Institute is in the process of developing the Virtual Machine Tool. This application simulates and evaluates a range of processes executed on alternative topological machine configurations. Error compensation and vibrations suppression methods are also explored to improve performance of existing machine tools. The machine to be simulated must be described both kinematically and dynamically.

While the above packages provide a detailed investigation of machine tools or sub-assemblies, these simulations required detailed knowledge of the machine construction and layout. Much of this information is not specified in the conceptual phases of design; as a result, the above simulation packages are generally not practical for conceptual evaluation.

## 1.5 Research Overview

The purpose of this research is to identify the inadequacies of existing methods for the conceptual design and evaluation of precision machine tools and to develop a set of computational tools to address these inadequacies. To address the above needs, the Precision Machine Design Assistant (PMDA) is developed as an integrated software tool for the rapid evaluation of machine tool concepts. Figure 1.7 provides an overview of the goals and functionality of PMDA, while Figure 1.8 shows how PMDA can be integrated into the machine tool design cycle.

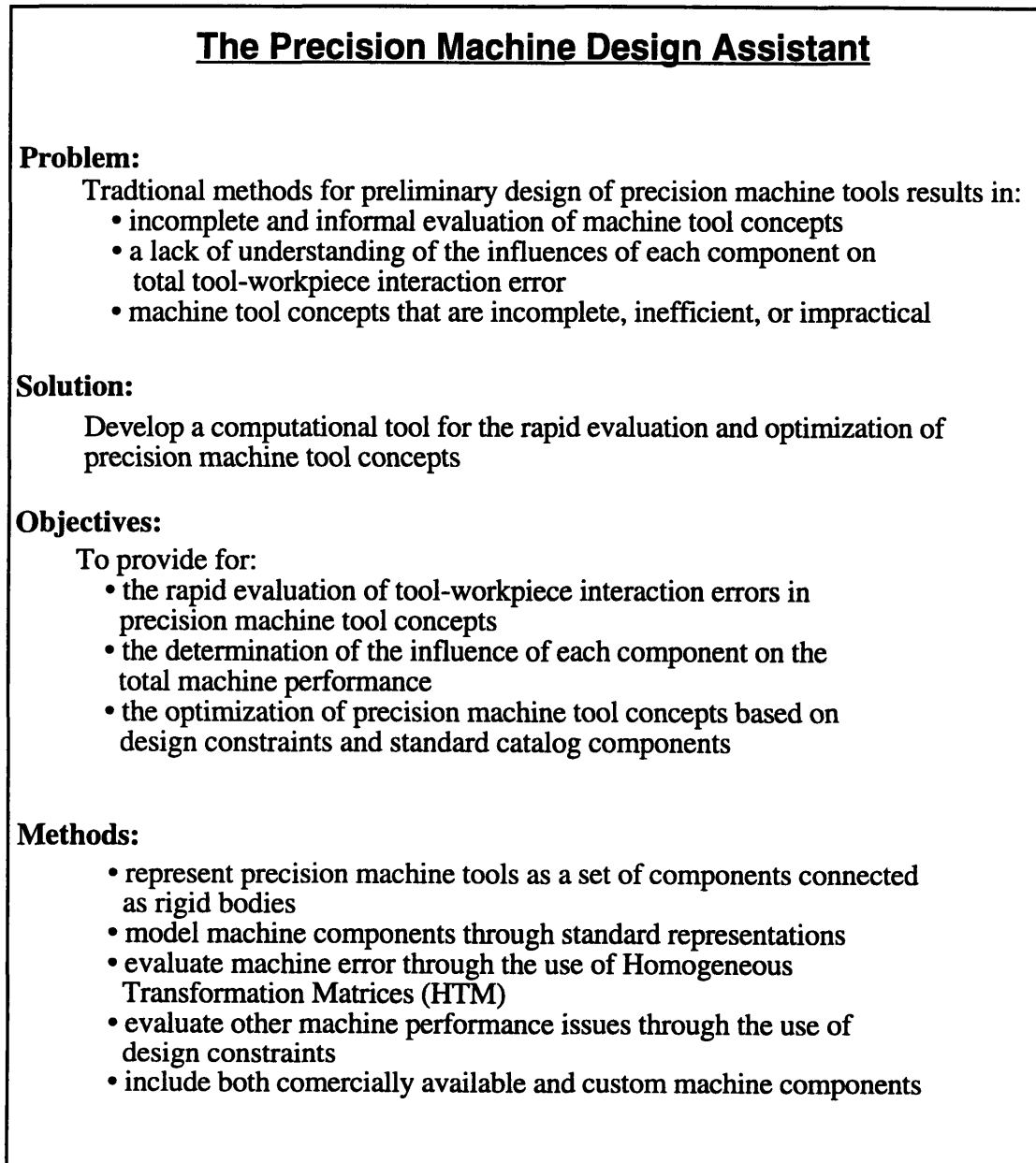


Figure 1.7: Overview of the Precision Machine Design Assistant

The functionality of PMDA is divided into four key areas: kinematic modeling, error evaluation, and constraint specification and evaluation, and concept optimization. A model of the proposed machine concept is created through the use of the error budget methods described above, and is based on a series of catalogs containing representative component parts. Constraints are set automatically according to the interactions specified between the selected components, or by the designer as necessary. Once the collection of constraints describing the entire machine is developed, the constraints may be evaluated on an individual basis, or the entire constraint "network" may be optimized.

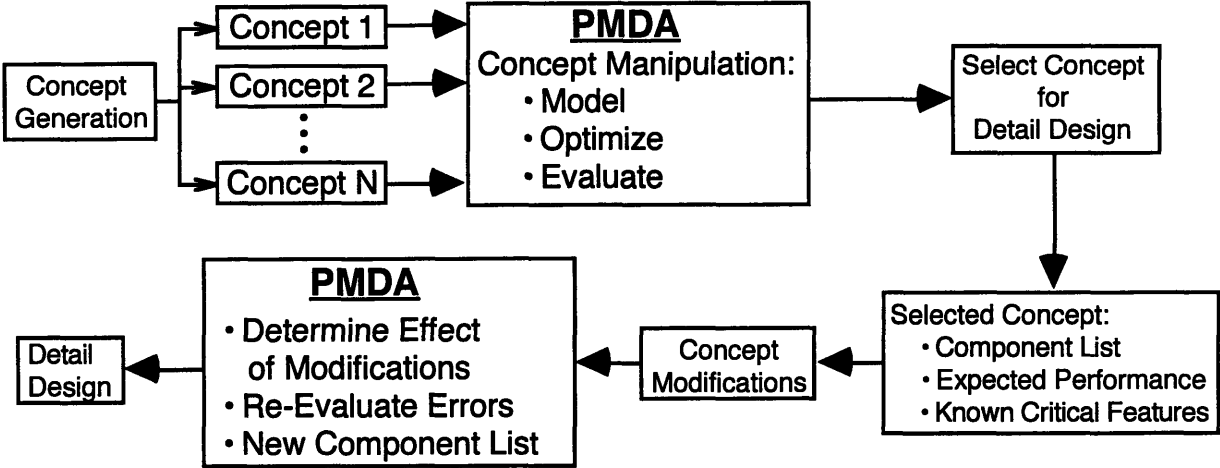


Figure 1.8: PMDA in the Machine Tool Design Cycle

This dissertation is presented in the following manner:

- Chapter 2 presents the principles of machine tool error calculations. Type of error motions and their sources are presented, as are the combinatorial rules employed in error budget formulations. A sample error budget for a simple mechanism is also presented as an example.
- Chapter 3 is an overview of constraint-based design. Mathematical constraints are defined for a variety of relationships, and the sources of constraints are reviewed. The section concludes with an overview of current constraint satisfaction algorithms, and an example constraint network for a simple mechanical assembly.

- Chapter 4 is a description of the component representations used in PMDA. An overview of standard mechanical components used in machine tools is offered, and the data contained in typical component catalogs is presented. The organization and hierarchy of PMDA objects is presented, as are the different levels and types of computational objects.
- Chapter 5 presents an overview of the functionality of PMDA. The methods for kinematic modeling, error motion modeling and evaluation, and constraint evaluation in PMDA are presented.
- Chapter 6 contains two examples related to the form and functionality of PMDA. The first example provides an overview of the information contained within the rotary bearing objects, and the methods for incorporating variations in components into the object hierarchy. The second example presents the determination of the total tool-workpiece interaction error for a simple machine spindle, and the evaluation of the constraint network associated with this spindle.
- Chapter 7 covers conclusions and recommendations for future work.

## **2.0 Principles of Machine Tool Error Calculations**

### **2.1 Introduction**

The basic principles governing the use and development of an error budget for machine tool conceptual evaluation was presented in Chapter 1. This chapter presents a more detailed analysis of the construction and use of error budgets. The primary sources of common machine tool error motions are described, as is a generic method for representing these motions with respect to a machine component. Rules for combining these errors into a comprehensive prediction of tool-workpiece interaction errors is also included. In addition, an example is provided to illustrate the above principles.

### **2.2 Sources of Machine Tool Errors**

Machine tool errors are classified into four major types: geometric, thermal, load-induced, and dynamic. A detailed understanding of these errors, their relative magnitudes, and their sources is critical to the formulation of any machine tool error budget. Failure to correctly account for and estimate any of these errors will result in an inaccurate error budget and a biased evaluation of the machine tool in question. The sources of machine tool errors discussed in this section are:

- **Geometric** - due to the deviation of form of the component
- **Thermal Errors** - deformations due to temperature gradients
- **Loading Errors** - deformations due to external or internal loads
- **Dynamic Errors** - due to friction and vibration

This section presents an overview of the sources of the errors listed above, as well as a summary of the effects of these errors on precision machine tools. Examples of these errors for typical commercially-available components are also provided.

#### **2.2.1 Geometric Errors**

Geometric errors are errors associated with the form of a particular component. These include errors due to straightness of linear surfaces, surface finish effects, twist, bow, and other errors due to imperfections in the manufacture of the component. In the scope of this

dissertation, geometric errors also include some forms of kinematic<sup>1</sup> or assembly errors, such as a failure to achieve parallelism and perpendicularity. Other kinematic errors, such as workpiece mounting or fixturing concerns, are treated as loading errors and are discussed in a later section.

Straightness error is defined as the deviation of a linear feature of a part from an ideal straight line. In a similar fashion, surface finish error is the deviation of the surface profile from an ideal planar profile. The difference in the above definitions is one of scale: straightness is often identified with macroscopic features, while surface finish is defined in terms of microscopic imperfections. Figure 2.1 illustrates the difference between straightness and surface finish. Straightness will affect the ability of an axis to position a carriage in a consistent manner, resulting in deviations along the bearing surface. Only linear motion



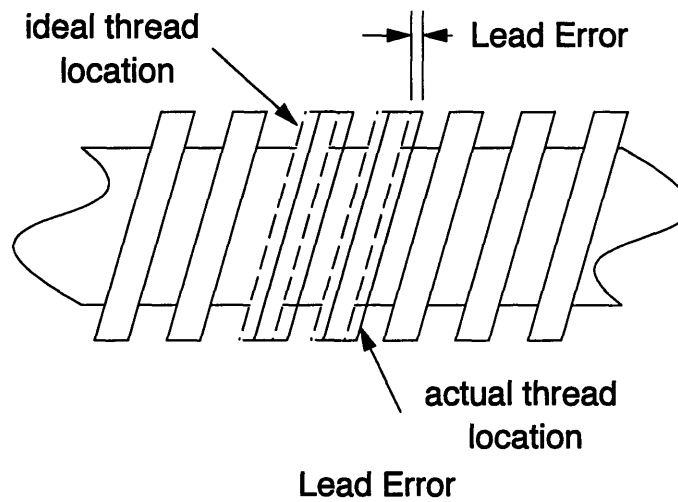
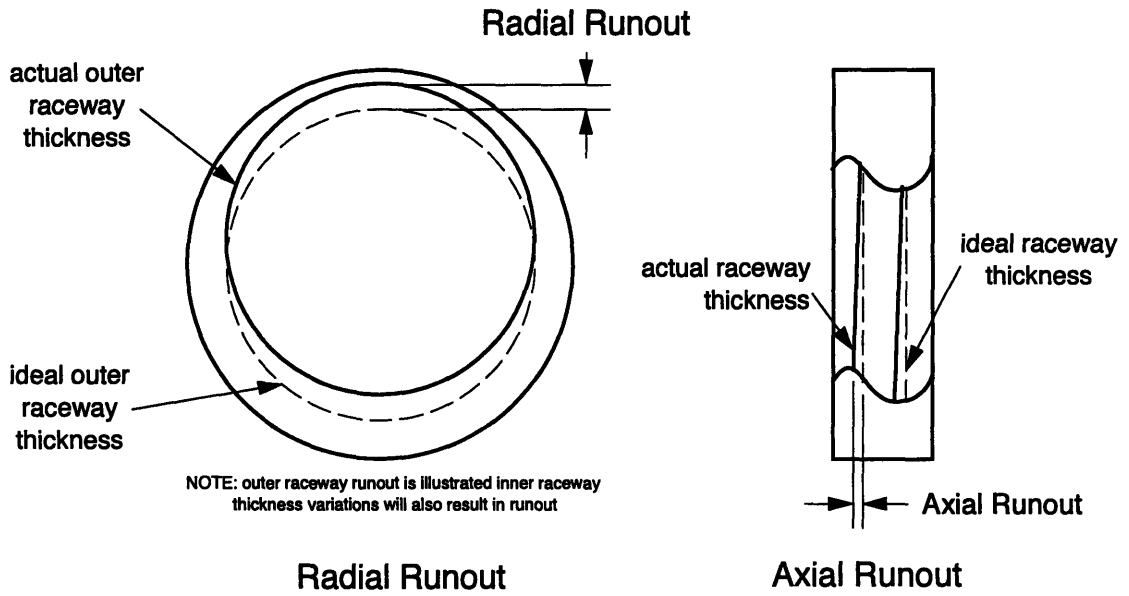
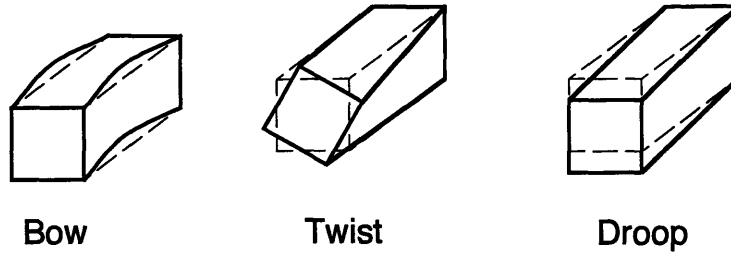
Figure 2.1: Surface Finish and Straightness Errors

components will have straightness-induced error motions, since rotary systems are not measured with respect to a straight line reference. Surface finish errors, however, affect all components involving mechanical contact, including rotary and linear bearings, since the mechanical contact between two imperfect surfaces will result in an imperfect motion.

Other geometric errors include "bow", twist, and droop in shafts and beams; radial and axial runout in rotary bearings; and lead errors, bow, and windup in leadscrews. Figure 2.2 provides graphical examples of some of these errors. All of these errors deal with manufacturing inaccuracies and can be accounted for through a variety of statistical or empirical methods discussed below. Care must be taken not to classify errors caused by loading as geometric errors; loading a leadscrew will result in twist along the longitudinal axis, which will be superimposed on the existing twist resulting from manufacturing inaccuracies. While

---

<sup>1</sup> "kinematic" refers to the mountings and connections of components



**Figure 2.2: Examples of Common Geometric Errors**

loading errors will alter the geometry of the part in a similar fashion to manufacturing errors, the root cause of the errors are fundamentally different and must be considered as such.

Geometric errors related to kinematics, also referred to as assembly errors, are errors due to the inaccuracies in the assembly of the machine tool. Unlike other geometric errors, which represent the effect of an individual component on the system, assembly errors represent the errors associated with the joining of multiple components. The major assembly errors dealt with in conceptual evaluation are parallelism and perpendicularity. Parallelism and perpendicularity errors result in Abbe effects for both linear and rotary motion parts.

Parallelism, as shown in Figure 2.3, is the angular error between two ideally parallel surfaces or lines and the actual surfaces or linear features. Perpendicularity is defined in a similar fashion but deals with perpendicular features. In both cases, the error is defined with respect to a datum surface; this datum surface must be at a known location to other datum surfaces of the machine tool, and the datum should not be altered once set.

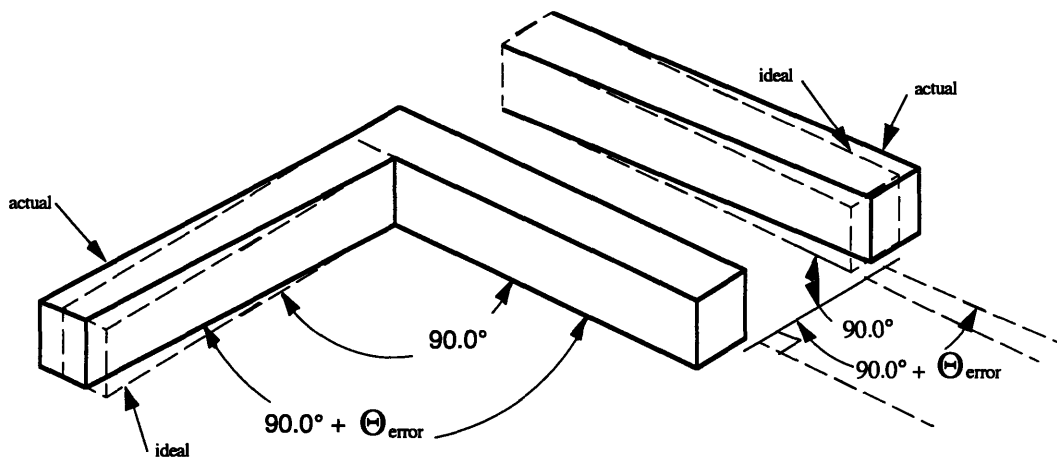


Figure 2.3: Parallelism and Perpendicularity Error

Geometric errors may be predicted and accounted for by empirical, statistical or heuristic methods. For custom parts that are familiar to the designer, existing machine tool parts representative of components of a given concept may be accurately measured (by CMM or other method), and the results used to predict average values for the geometric errors. For commercially-available components, tolerance information and the experience of applications engineers may be used to estimate best case, average, and worst case performance of a given component. Statistical methods may also be used to predict the average, best-case, and worst-case for a given type of component. Estimates may also be based on past experience and the

performance of existing machine tools, or on the intuition of the designer. In all cases, the manufacturing errors must be estimated or predicted and included, since no theoretical relationships currently exist for the determination of manufacturing errors.

### **2.2.2 Thermal Errors**

Thermal errors, also classified as thermally-induced geometric errors, are errors in geometry associated with variations in ambient temperature or temperature gradients in, on, or around the machine tool. While the variation in temperature results in changes in component geometry, thermal errors are considered separate from the previously described geometric errors since thermal errors result from variations in environment and not variations in manufacture. Because of this dependency, thermal errors will vary with respect to machine operating condition while geometric errors will remain essentially constant throughout the lifetime of the machine.

All thermal errors are a consequence of the relationship:

$$\Delta L = \alpha \Delta T L_0$$

where  $\alpha$  is the coefficient of thermal expansion of the material, and  $\Delta T$  is the deviation of temperature from the undeflected shape. Several different modes of this equation common to machine tools are shown in Figure 2.4. For a constant  $\Delta T$  caused by a change in equilibrium temperature, the result is an increase in the overall size of the part dimensions. Variable values of  $\Delta T$  throughout a given component result in a temperature gradient and more complex behaviors, including bending and location-dependent variations in component size.

Thermal gradients are often present in machine tools, resulting in a non-uniform expansion in the components and the machine tool structure itself. Sources of thermal gradients include heat generated by friction in individual components, variations in ambient temperature, coolant streams, motors, heat generated by cutting processes, and the body heat of the machine operator. Any time every component of the machine is not in thermal equilibrium, gradients will be present, and the effects must be accounted for to accurately predict the performance of the machine tool in question. For the purpose of error budget calculations, most ambient conditions (including the effects of ventilation systems, lights, the operator, and other "shop floor" factors) can be assumed constant, and only the gradients resulting from the operation of machine components included in the system error budget.

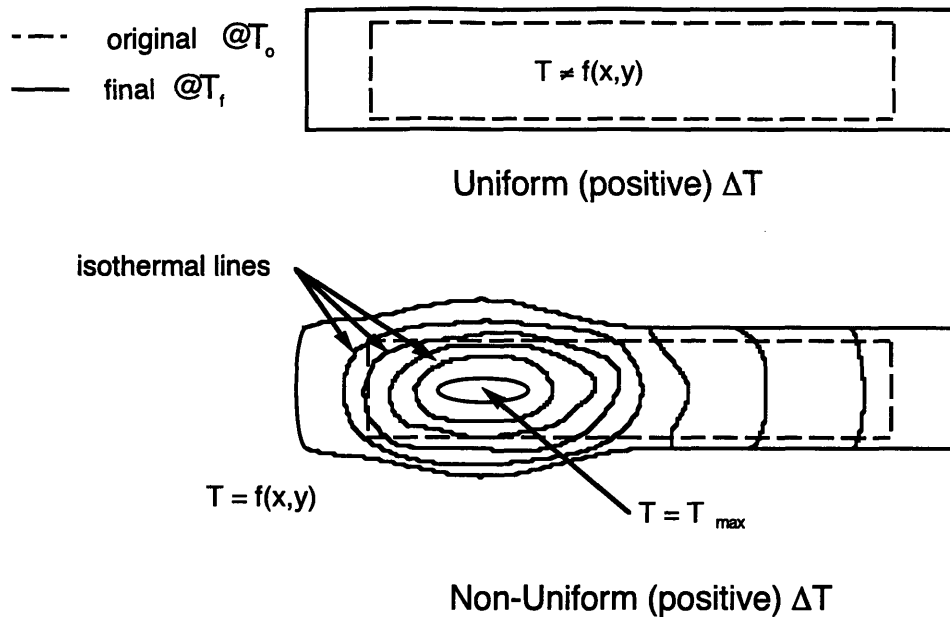


Figure 2.4: Example of Uniform and Non-Uniform Thermal Errors

Inclusion of thermal errors in the machine tool error budget is a complex process. In the conceptual evaluation phase, most thermal errors may be estimated based on experience and existing designs. Simple calculations based on preliminary assumptions (for example, constant  $\Delta T$  for a given part or low thermal conductivity between components) may be used, or more detailed analysis may be made through the use of computational methods. Representative rates of heat generation and temperature rise are also generally provided for commercially-available components. If large thermal gradient are expected, or the design is expected to be sensitive to thermal errors, the errors budget should be modified to account for each of these conditions.

### 2.2.3 Loading Errors

Loading errors include the deflections of each component of the machine (including machine structure) due to gravitational forces, mounting of individual components or the entire machine tool, and cutting forces. In many cases, these deformations are substantial and can be the most significant errors affecting the given machine tool.

Loading errors due to gravity are the result of the static weight of each component (including structural members) deflecting the machine tool structure itself. While the weight of these components (except the workpiece) are assumed to be constant during conceptual

evaluation, the location of these components will change during the operation of the machine. This will result in different deflections based on different machine "poses" or operating conditions. For this reason, an error budget must be constructed to allow evaluation of machine performance for a variety of operating conditions.

The errors due to the machining process are less straightforward than gravitationally-induced errors. For some machine tools, such as photolithography machines ("wafer-steppers") and waterjet cutters, the forces generated by the machining process (exerted on the cutting tool and supporting structure) are negligible, while the cutting forces generated by metal lathes and machining centers can be substantial. In machine tools where machining forces are expected to be significant, care must be taken to both correctly estimate the magnitude of these forces and determine the distribution of these forces and associated deflections between the structure supporting the tool and the structure supporting the workpiece. Inclusion of these effects in the system error budget may be accomplished by estimates based on past experience, measurements of existing machine tools, simulation of cutting processes, and/or calculations and modeling of the stiffnesses of the structures affected by these forces.

Errors caused by internal forces, such as assembly errors and bearing preload, are often the most neglected type of errors in precision machine tool design. Assembly (or mounting) errors are the result of forces generated by mounting of components to the machine or the machine to the floor. The most common mounting error is forced geometric congruence, where two parts of slightly different geometry are forced into alignment. This forced alignment generates internal stresses in the mated parts, resulting in the distortion of the parts from their ideal shape. For example, a linear bearing rail bolted to a curved base will deform to the shape of the base, and the resulting deflection will lead to errors in the rail, bearing, and attached components. These errors must almost always be estimated based on previous experience or measurement of existing designs. Bearing preload is another source of internal-force induced errors. In this case, a load is applied to a bearing to increase stiffness and fatigue life. This preload also leads to an increase in internal stresses and deflections of the bearing and surrounding structure. In the case of preload, however, the force applied to the bearing is known, and the resulting deflections may be incorporated into the system error budget.

#### **2.2.4 Dynamic Errors**

Dynamic errors refers to any error influencing or resulting from the movement of the machine. These errors are caused primarily by friction in the bearing systems and vibrations in the surrounding structure. In the case of bearing friction, the machine axis in question is unable to servo to the desired position due to high levels of static friction between the bearing

surfaces. This inability to move a machine axis to a desired position results in losses in machine resolution, repeatability, and accuracy. Dynamic errors due to vibration are much more difficult to account for. Vibrations from the floor, rotating mechanical components, fluid supply lines, motors, and other sources lead to unusual modes of vibration. These result in an inability to achieve the desired surface finish on the part being machined.

While the other three classes of errors are quasi-static, dynamic errors cannot be analyzed by a static model, and are highly dependent on a variety of machine operating conditions and characteristics. Dynamic errors are also often coupled; in the case of component vibrations, differences in the vibration frequencies will result in "beat", a phenomenon where the vibrations constructively and destructively interfere with each other in a time-dependent fashion. Dynamic errors are also strongly related to the characteristics of joints and mechanical component contact. While some simulation programs exist for simple dynamic error prediction, the majority of these errors are influenced by factors that are difficult to predict or not well understood. For these reasons, dynamic errors are difficult to include in the evaluation of machine tool errors during the conceptual design phase, and must often be neglected until the detail design or prototype testing phase of machine tool development.

Since dynamic errors are often neglected in the preliminary stages of design, this dissertation will not include dynamic error considerations. It should be noted, however, that the inclusion of dynamic errors in the system error budget is possible, and should be considered if predictions of the dynamic error motions for a machine concept are available.

### **2.2.5 Summary of Error Motions**

The total error motion of a machine is the result of geometric, thermal, loading, and dynamic errors. Geometric errors result from errors in the manufacturing process of the part, and must be accounted for empirically or through simulation. Thermal errors occur when a component is subjected to a temperature gradient, or when the ambient temperature the component operates under is not equal to the "original" ambient temperature (at which the characteristics of the component are defined). Loading errors are deflections resulting from externally or internally applied loads. Dynamic errors are the result of movement of machine parts, and result in vibrations that may interact ("beat") in unusual and unpredictable ways. All of these error sources must be accounted for in the design of precision machine tools, since any of the above sources may result in significant error motions between the tool and workpiece.

## 2.3 Error Motion Representation

Before the above errors may be included in a system error budget, a method for representing them must be developed. While the sources of errors influencing the performance of a precision machine tool are diverse, the effect of these errors is not: all errors generated by a component lead to deflections in the component. A common method of error motion representation compatible with the combinatorial rules introduced below is to model each component as a rigid body and the error motions as a series of translational and rotational transformations. These transformations take place about the coordinate system of the component, which allows all of the error motions associated with the component to be combined and reflected to the adjoining components. The details of this representation are presented below for both fixed and rotating coordinate frames.

### 2.3.1 Fixed Frame

The fixed reference frame model for a generic rigid-body component is shown in Figure 2.5. The coordinate system local to the component is located with respect to a global coordinate system, and is oriented so that the servo axis (axis of motion) is in the local X direction.

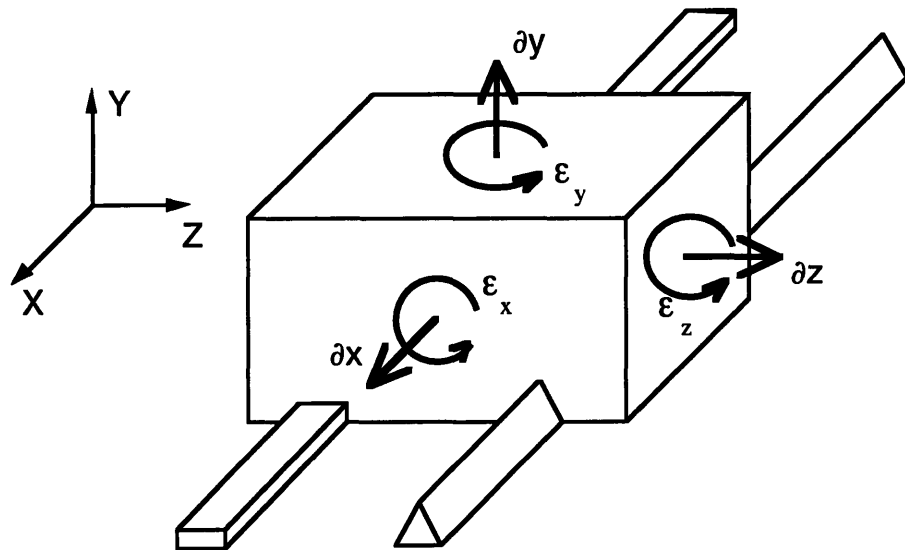


Figure 2.5: Fixed Reference Frame Representation

Vertical straightness error  $\partial y$  and horizontal straightness error  $\partial z$  are also defined with respect to the local coordinate frame. The three angular errors  $\epsilon_x$ ,  $\epsilon_y$ , and  $\epsilon_z$ , are represented as rotations about the local coordinate axes, and are referred to as roll, pitch, and yaw,

respectively. Since the component in question has six degrees of freedom, specification of  $\partial x$ ,  $\partial y$ ,  $\partial z$ ,  $\epsilon_x$ ,  $\epsilon_y$ , and  $\epsilon_z$  will completely describe the possible error motions of a linear system

### 2.3.2 Rotating Frame

The rotating reference frame representation is shown in Figure 2.6. As seen in the figure, the translational and rotational errors are defined with respect to the fixed reference frame. The exception is  $\epsilon_z$ , which is included in the rotational angle  $\Theta_z$  and can be represented as a rotational angle error. Ideally, the part rotates about the reference Z axis, but because of the errors it rotates about the deflected Z-axis. Note that all errors are a function of the rotation angle, since the rotation of the part will result in different axis orientations with respect to the (fixed) reference frame.

### 2.3.3 Coordinate Frame Location

An important, and often overlooked, factor in the creation of a system error budget is the location of the coordinate frame for a given component. For the above examples of linear

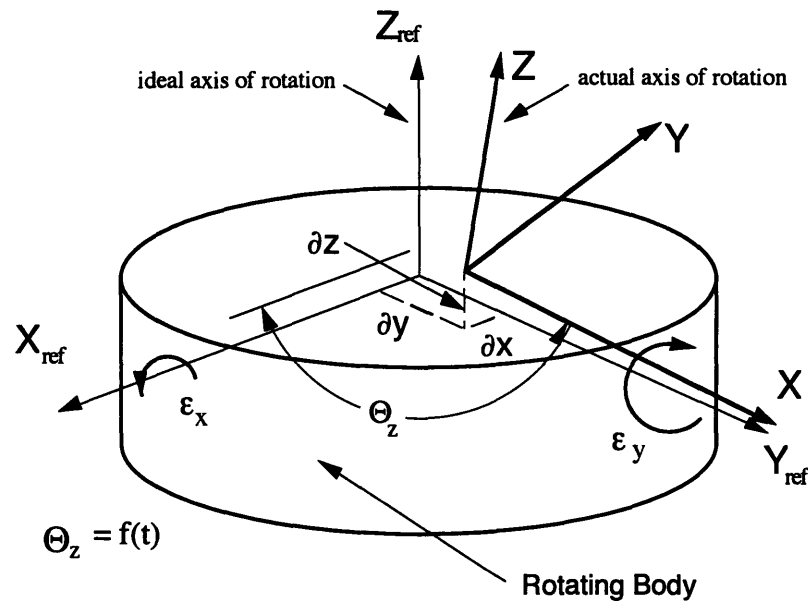


Figure 2.6: Rotating Reference Frame Representation

and rotary motion systems, the reference coordinate frame is located at the roll center of the component. This allows for a convenient convention, but results in difficulty for verification of results later in the prototyping and production phases of the design cycle. For this reason, it

is common to associate more than one coordinate frame with a given component. The additional coordinate frames are often located on the surface of the component, or at a point on the component where measurements will be easily made. These additional reference frames may be included in the kinematic model of the machine tool as separate components rigidly attached to the "base" component. As with the reference coordinate frame, error motions may be defined with respect to this new frame to allow for more complex representations of machine errors.

## 2.4 Error Propagation and HTM Representation

Section 2.1 presented the sources of error in a precision machine tool that must be included in a system error budget to provide meaningful results for conceptual evaluation. A method for general representation of these errors for fixed and rotating reference frames is described in Section 2.2. This section details a method for modeling the spatial relationships between component in a machine tool, allowing for the combination of component errors into a total tool-workpiece interaction error. This is accomplished through the use of the Homogeneous Transformation Matrix (HTM). Use of the HTM, the interpretation of each element of this matrix, and the set of combinatorial rules concerning HTMs, is presented.

### 2.4.1 The Homogeneous Transformation Matrix

The homogeneous transformation matrix is a mathematical description of the transformation between two Cartesian coordinate systems [Foley, 1990]. The reason for using a matrix is seen in by observing that all transformations are a combination of translations and rotations. For the 2-dimensional case, translation of a point is:

$$x' = x + \partial x \qquad y' = y + \partial y$$

where  $\partial x$  and  $\partial y$  represent the amount of translation. Rotation is described by:

$$x' = x \cdot \cos \Theta - y \cdot \sin \Theta \qquad y' = x \cdot \sin \Theta + y \cdot \cos \Theta$$

where  $\Theta$  is the angle of rotation. Transformations for the 3-dimensional case are similar and found in [Foley, 1990]. For both cases, the above formulas may be applied to any point in space to achieve the desired transformation. Unfortunately, translation takes place through

addition while rotation takes place through multiplication, requiring two separate routines for the transformation of a given point in traditional (x,y) coordinates. The above transformations may be accomplished via matrix multiplication, however, if homogeneous coordinates are used [Foley, 1990]. The representation of a point in homogeneous coordinate does not alter the location of the point in space, and allows for the transformation of the point in a standard manner, regardless of the type of transformation.

The 3-D homogeneous representation of a point is given by:

$$p = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}$$

where W is referred to as the homogeneous coordinate. One unusual characteristic of homogeneous coordinates is that any homogeneous point that is a multiple of another represents the same point in space. For this reason, W is almost always set equal to one to avoid problems with interpretation.

Homogeneous representations allow the translational and rotational transformations of a given point to be performed through multiplication by a series of 4-by-4 HTMs. The notation:

$${}^{ref}T_{local}$$

denotes the transformation between the local coordinate system and the reference coordinate system, i.e. the coordinates of a point with respect to the local system are transformed to a representation in the reference system.

The HTMs required for Cartesian transformations are:

For a rotation of  $\Theta_x$  about the X axis:

$${}^{ref}T_{local} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \Theta_x & -\sin \Theta_x & 0 \\ 0 & \sin \Theta_x & \cos \Theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For a rotation of  $\Theta_y$  about the Y axis:

$${}^{\text{ref}} \mathbf{T}_{\text{local}} = \begin{bmatrix} \cos \Theta_y & 0 & \sin \Theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \Theta_y & 0 & \cos \Theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For a rotation of  $\Theta_z$  about the Z axis:

$${}^{\text{ref}} \mathbf{T}_{\text{local}} = \begin{bmatrix} \cos \Theta_z & -\sin \Theta_z & 0 & 0 \\ \sin \Theta_z & \cos \Theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For translation by  $\partial x$ ,  $\partial y$ , and/or  $\partial z$ , the HTM is:

$${}^{\text{ref}} \mathbf{T}_{\text{local}} = \begin{bmatrix} 1 & 0 & 0 & \Delta_x \\ 0 & 1 & 0 & \Delta_y \\ 0 & 0 & 1 & \Delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Each of these HTMs will transform a given series of points from one coordinate frame to another. Multiple rotations and translations may be accomplished by multiplying the HTMs in the desired order of transformation to obtain a single, composite HTM. The order of rotation must be observed, however, since the final HTM is dependent on the order of the transformations applied to the point. Appendix C provides more information concerning the calculation of HTMs for transformations between two coordinate systems.

## 2.4.2 The Kinematic Machine Model

The purpose of the development of a machine model is to determine the total tool-workpiece interaction error given the error motions associated with each component of the machine. Once a method for transforming points represented in one coordinate frame to

another is developed, the necessary model may be constructed. One method of machine modeling, referred to as "kinematic modeling", is to consider each component as a rigid body and include all of the error motions associated with the component in the component model. This component is then "attached" to the adjoining components through an HTM, allowing the error motions to be transformed through the components to a reference coordinate frame. Since the HTMs are between two separate component coordinate frames, the joints between components must either be assumed to be rigid or must be included as separate components.

The process for developing a kinematic model of a simple machine tool is shown in Figure 2.7. For simplicity, only a two-dimensional model is constructed, with all Z coordinates assumed to be zero; 3-dimensional models are developed in a similar fashion. The kinematic relationships between the various components, and the location of the coordinate frame (all are assumed to be at the geometric center of the component) are shown in Figure 2.7b. Two structural loops, shown in Figure 2.7C, define the "error paths" between the reference coordinate frame and either the toolpoint or the workpiece. The common point of interest is the point where the toolpoint contacts the workpiece. This represented the desired state of zero tool-workpiece interaction error; this condition, however, rarely occurs due to errors associated with each individual component.

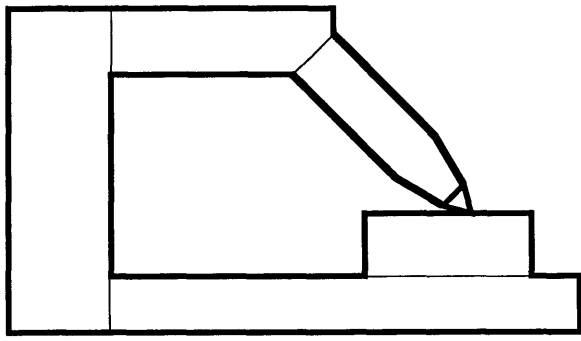
The total HTM for the tool-reference loop is the matrix product of all of the HTMs in the structural loop:

$${}^{ref} HTM_{tool} = {}^{ref} T_{T1} * T1 T_{T2} * T2 T_{tool}$$

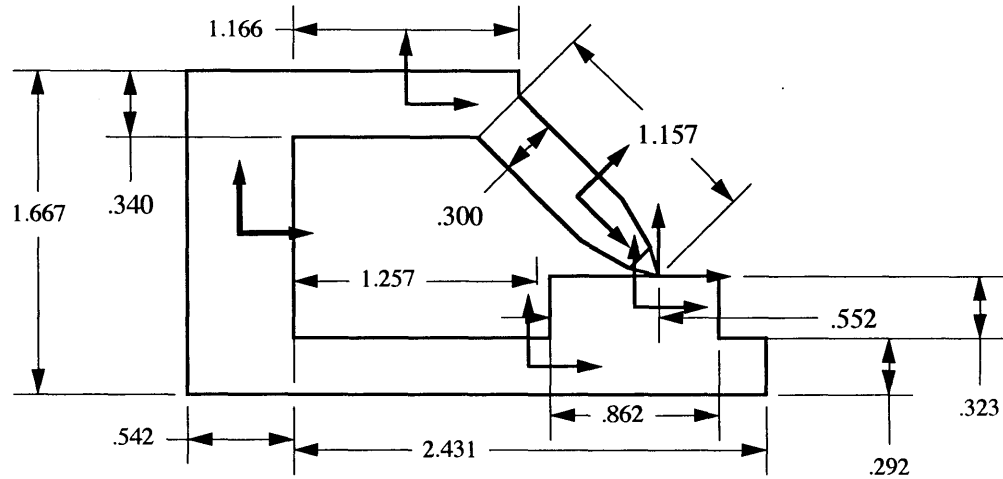
Similarly, the HTM for the workpiece-reference loop is:

$${}^{ref} HTM_{workpiece} = {}^{ref} T_{W1} * W1 T_{W2} * W2 T_{workpiece}$$

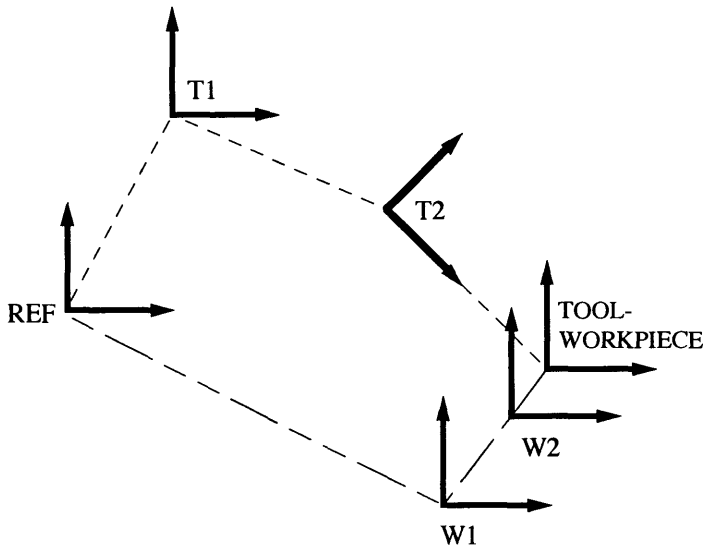
Note that the HTMs defined above and shown in Figure 2.7D represents the ideal transform between the reference coordinate system and the tool-workpiece interaction point; for this reason, both total HTMs are identical. For inclusion of the error motions of each component, the tool-workpiece interaction point is modified vectorially in each coordinate frame. For example, the toolpoint is originally defined in terms of the tool coordinate system. The error motions of the tool are then added vectorially to the coordinates of the tool-workpiece interaction point, resulting in the deflected position of the toolpoint due to the errors associated with the tool. This new point is then transformed to the T2 coordinate system, where the error motions of the T2 component are added vectorially. This process continues until the reference coordinate frame is reached. The process is repeated for the workpiece loop, and the vector



A) Machine Concept



B) Kinematic Relationships



C) Structural Loops & Coordinate Frames

$$\begin{aligned}
 {}^{W2}HTM_{TW} &= \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.121 \\ 0.0 & 1.0 & 0.0 & 0.161 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} &
 {}^{T2}HTM_{TW} &= \begin{bmatrix} 0.707 & -0.707 & 0.0 & 0.578 \\ 0.707 & 0.707 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \\
 {}^{W1}HTM_{W2} &= \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.541 \\ 0.0 & 1.0 & 0.0 & 0.307 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} &
 {}^{T1}HTM_{T2} &= \begin{bmatrix} 0.707 & 0.707 & 0.0 & 0.886 \\ -0.707 & 0.707 & 0.0 & -0.473 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \\
 {}^{GND}HTM_{W1} &= \begin{bmatrix} 1.0 & 0.0 & 0.0 & 1.468 \\ 0.0 & 1.0 & 0.0 & -0.688 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} &
 {}^{GND}HTM_{T1} &= \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.854 \\ 0.0 & 1.0 & 0.0 & 0.663 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \\
 {}^{HTM}_{Loop W} &= \begin{bmatrix} 1.0 & 0.0 & 0.0 & 2.149 \\ 0.0 & 1.0 & 0.0 & -0.219 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} &
 {}^{HTM}_{Loop T} &= \begin{bmatrix} 1.0 & 0.0 & 0.0 & 2.149 \\ 0.0 & 1.0 & 0.0 & -0.219 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}
 \end{aligned}$$

D) Homogeneous Transformation Matrices

Figure 2.7: Example of HTM Modeling

difference between the toolpoint and the workpiece (both in reference frame coordinates) is the total tool-workpiece interaction error. An example of this process is found in Section 2.4.2.

This process of kinematic modeling of machine tool concepts has several advantages over other forms of modeling. First, construction of the model is relatively simple; while the spatial relationships between components are sometimes difficult to establish, the mathematics and concepts behind the modeling process are straightforward. Second, the model is reusable in a variety of situations. Changes in error motions (both magnitude and direction) require a simple re-transformation of the point from the change to the reference frame. Modifications to the actual component or machine model also require changes only to the elements between the change and the reference frame. In this way, much of the model may be reused without complex recomputation of the HTMs. Kinematic modeling also provides fairly accurate results, assuming the error motions for each component are carefully determined. Finally, this method of modeling is relatively easy to implement computationally, and allows for the rapid analysis of multiple machine tool concepts.

One of the difficulties of the above process is the necessity to recreate the kinematic model for each "pose" of the machine. For example, if the workpiece is relocated 0.2 units in the X-direction, the model will be altered. The tool-workpiece interaction point will change (due to the profile of the workpiece), resulting in a change in the probe (or tool) position. The location of the origin of the workpiece will also change. This results in a need to recalculate the kinematic model; this recalculation must take place every time a component of the machine is repositioned. Evaluation of the performance of a machine tool requires the analysis of many poses of the machine, since errors (especially Abbe Error) will change with respect to component positions. For this reason, the process of kinematic modeling is a time-consuming process, and requires a high level of automation to be feasible for evaluation of a large number of concepts.

## **2.5 Machine Modeling and the System Error Budget**

The method for machine modeling via HTMs and the determination of tool-workpiece interaction errors is a straightforward, but time-intensive, process. The steps involved in this process are outlined below and illustrated in the following example:

### **Kinematic Machine Modeling**

A kinematic model of the machine is created. This involves the determination of each HTM for the connected components of the machine. Starting with the tool point, the HTM

between adjoining components are determined, in series, until the reference coordinate frame for the entire machine is reached. A similar process is completed for the workpiece and associated components. The end result is a series of HTMs describing the location of the toolpoint and the workpiece with respect to the machine reference frame. This process must be repeated for each geometric operating condition of the machine, unless the location of each component can be specified by modifying the elements of the affected HTMs. For example, the HTM for a linear carriage moving in the  $\pm X$  direction may be modified by altering the X translational term in the HTM defined for transformations between the carriage and the supporting structure. The HTMs of the components attached to this carriage will not change, since they are defined with respect to the location of the carriage.

### **Error Motion Modeling**

Error motions are incorporated into each local coordinate frame. For translational errors, this is accomplished by decomposing each error into components along the X, Y, and Z axes of the local frame, summing all errors along the axis, and incorporating the total error motion for each axis into the translational elements of the local HTM. Transformation of a point through this HTM will then result in the inclusion of the translational error motions in the transformation. Rotational errors are included in a similar manner, with each rotation requiring a decomposition into rotations about the local coordinate axes. This decomposition is not necessary if the local coordinate frame is chosen to be coincident with the axes of motion of the component, since error rotations will take place along or perpendicular to axes of motion. Further details on the selection of coordinate frames is found in [Slocum, 1992].

### **Determination of Tool-Workpiece Interaction Error**

The toolpoint location is set with respect to the workpiece, and the point of contact is multiplied through the tool-ground and workpiece-ground loops. The result is the toolpiece location with respect to ground and the workpiece location (at the point where the tool should be contacting the workpiece) with respect to ground. The difference between these two points is the total tool-workpiece interaction error.

### **Model Modification and Sensitivity Analysis**

The above process may be repeated to account for variations in error motion magnitude and direction, machine pose, or component types. Altering the magnitudes or directions of error motions requires the error budget to be recomputed from Step 2, while a change in machine orientation or component locations requires rearrangement of the kinematic model from the point of change to the tool or workpiece.

## 2.6 Example of the Machine Tool Design Process

This section presents a comprehensive machine tool design example to illustrate the current methods used for development and evaluation of machine tool concepts. The example begins with a sample concept proposal, and proceeds through the conceptual development and evaluation phase. One example concept is selected, and the kinematic model for this proposed machine is developed. The example concludes with an evaluation of the error budget based on the performance of a set of sample components. Detail design is not covered. For simplicity, only a 2-dimensional model of the machine will be developed, although the techniques illustrated also apply in 3-dimensions.

### 2.6.1 Specification of Machine Requirements

As stated in Chapter 1, the first stage of the machine tool design cycle involves the specification of the requirements of the new machine. Often, the initial requirements are stated in the form of a proposal. Proposals for the development of machine tools are simple statements expressing the most basic functions the machine must perform. A typical proposal, and the one used for development of the example machine, is shown in Figure 2.8. This proposal defines the basic need of the customer, without needlessly limiting the design of the machine tool.

**Proposal:**

Development of a machine tool to measure the surface profile of a part. The required working volume is a cube 0.3 units on each side. Measurements must be accurate to 0.001 units at every point in the machine working volume.

Figure 2.8: Sample Machine Tool Proposal

### 2.6.2 Conceptual Development

Once the proposal for a new machine is defined, machine tool concepts capable of satisfying the requirements imposed by the proposal are developed. The development of machine concepts is described above, and is a process closely related to the experience of the designer. In this phase of machine tool design, sketches of possible configurations for the machine are prepared and refined until a decision concerning the feasibility of each concept may

be made. Two more detailed example concepts, with brief descriptions of the operation, advantages, and disadvantages of each concept, are shown in Figure 2.9. In the development of an actual machine tool, many more concepts would be developed and analyzed to allow for a more exhaustive evaluation of the possible designs of the proposed machine; for the purposes of this example, it will be assumed that these two concepts represent the most feasible of the proposed concepts.

At this stage in the conceptual development process, the rough details of each concept will be included in the concept description. As more details are associated with each concept, the possible performance of the machine tool becomes more evident. Once a designer feels reasonably confident that a given concept will meet the desired level of performance, a kinematic model of the concept is developed and evaluated. One possible layout for the angle-probe concept is shown in Figure 2.10; this concept will be evaluated in the following section.

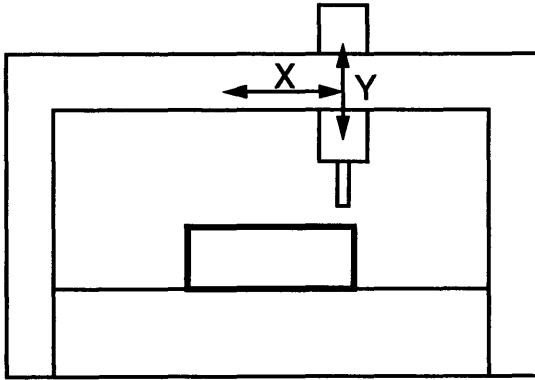
### **2.6.3 Conceptual Evaluation**

The conceptual evaluation process is described in Section 1.2.4. At this stage of the design process, each concept would be modeled to determine the expected performance of each concept based on expected performance of the components included in the machine tool concept. The kinematic model for the angle-probe concept is shown in Figure 2.7; the process for developing this model is discussed in Section 2.3. Note that this model only applies to the machine pose shown in the figure; altering the position of the probe and/or workpiece will require a re-creation of the kinematic model.

At this point, all of the error motions associated with each component would be included in the kinematic model, and the location of the toolpoint and the workpiece (transformed through the appropriate structural loop) would be determined with respect to the reference coordinate frame. After this total error is determined, the errors resulting from each component may be determined by setting the errors for all components except the one to be evaluated to zero. For this example, only the errors associated with the probe servo system are included in the error budget; other errors may be included in a similar manner, and should be included to determine the overall performance of the machine. The servo errors, and the resulting deflection of the toolpoint, are illustrated in Figure 2.11.

Figure 2.12 shows the step-by-step process for inclusion of the probe servo errors into the machine tool error budget. For purposes of example, it will be assumed that this error is the only error associated with the machine; in reality, each component error would be included in a similar fashion to determine the total tool-workpiece interaction error based on the complete machine concept. As can be seen in the figure, the total tool-workpiece interaction error

**CONCEPT 1- Moving Bridge-Type**

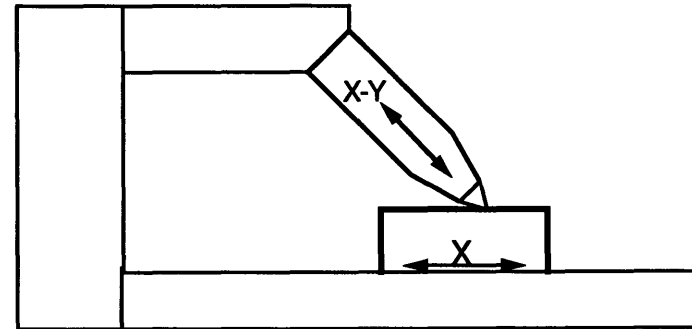


**Description:** Concept uses a closed-loop, bridge-type structure. Workpiece is stationary, with the probe and probe structure moving in both the X and Y axes.

- Advantages:**
- bridge loop avoids Abbe errors
  - stationary workpiece limits error motions to probe structure
  - less prone to thermal errors than C-structure

- Disadvantages:**
- bridge design is structurally indeterminate
  - probe servo system is complex
  - bridge blocks access to workpiece

**CONCEPT 2 - Angle-Probe Type**



**Description:** Concept uses an open-loop structure (modified C-structure) with the probe extended from the top horizontal arm as shown. Workpiece is servo-ed in the X-direction, while elevation is controlled by the extension and retraction of the probe within the probe structure.

- Advantages:**
- C-loop is structurally simple and easily evaluated
  - No stacking of axes
  - mechanically simple

- Disadvantages:**
- C loop is prone to Abbe Errors
  - probe-axis system limits size of working volume
  - probe makes changing workpiece difficult

Figure 2.9: Sample Machine Tool Concepts

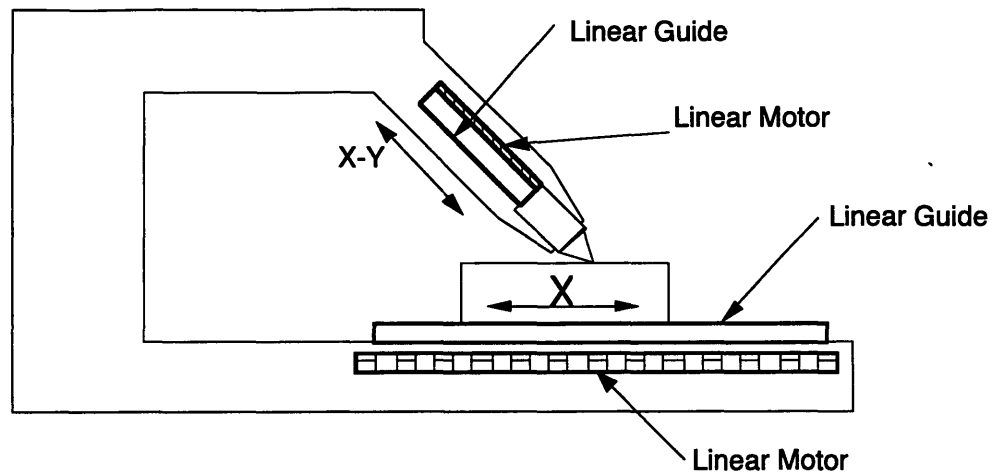


Figure 2.10: Concept with Specified Components

resulting from the probe servo system is well above the required level of accuracy. For this reason, the selected servo system is not acceptable, and must be redesigned to provide the necessary accuracy. Once the new system is defined and the associated errors are determined, the error budget may be re-evaluated to determine if the new system meets the required specifications.

#### 2.6.4 Example Summary

This example illustrates the complexity of concept evaluation for precision machine tools. The angle of the probe results in a transformation of the servo errors in a manner that is not easily predicted by inspection. The inclusion of other error motions complicates the process, and the tool-workpiece interaction error for the entire machine tool cannot be determined without mathematical assistance. For this reason, the HTM process described above should be applied to machine tool concepts to determine the effects of each component (and the associated errors) on the performance of each concept. This allows direct comparison between concepts, and reduces the chance that an inferior concept will be selected.

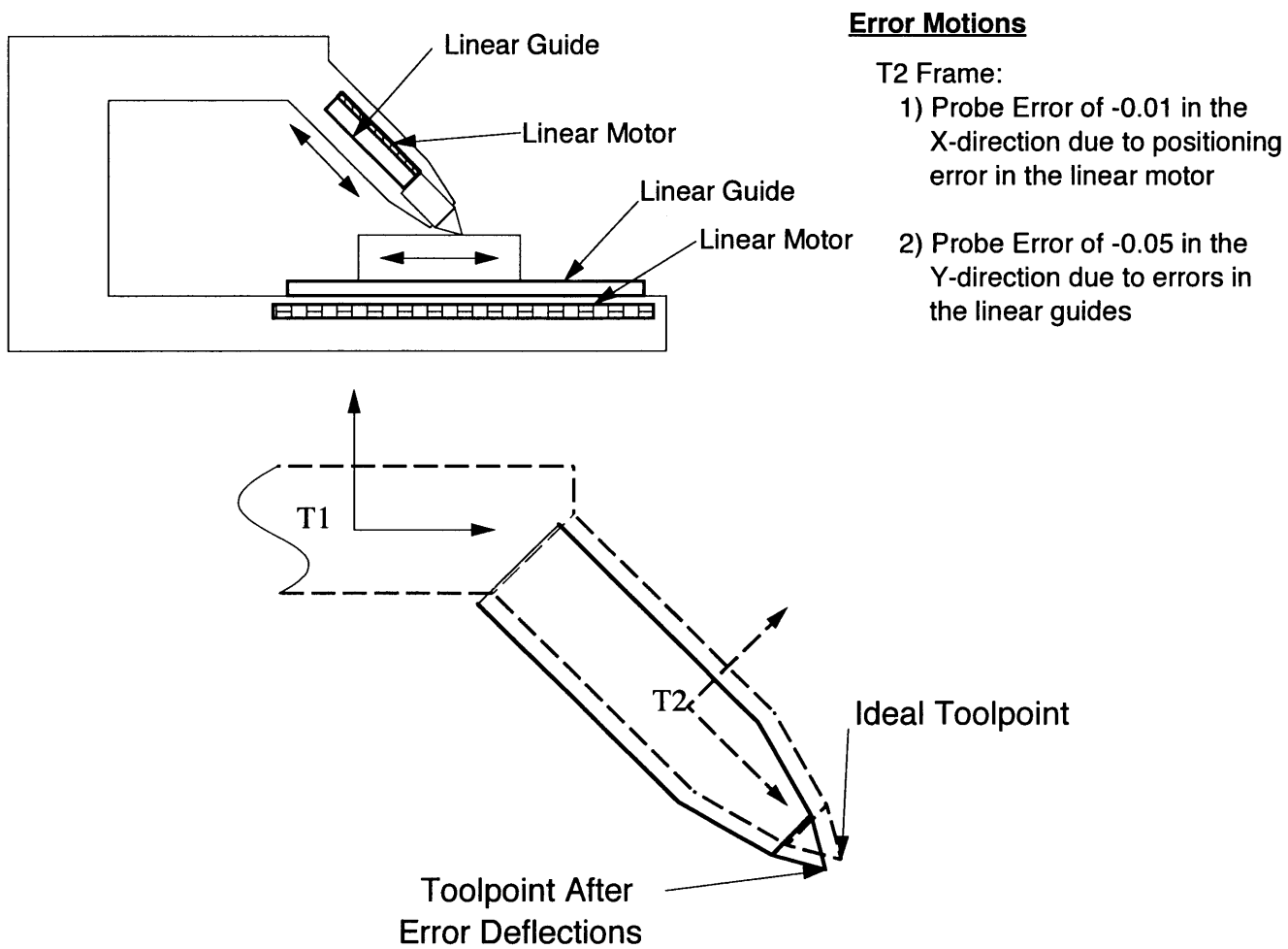
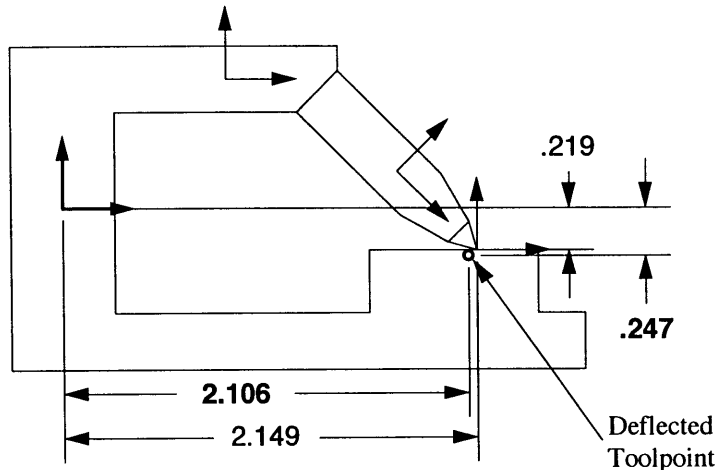


Figure 2.11: Sample Toolpoint Error Motion



Bold numbers indicate the actual toolpoint location  
Regular numbers indicate the ideal toolpoint location

**STEP 1**

Transform Toolpoint to Probe Coordinates:

$$\text{Point (probe)} = \begin{bmatrix} 0.707 & -0.707 & 0.0 & 0.578 \\ 0.707 & 0.707 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 0.578 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix}$$

**STEP 2**

Add Probe Error Components

$$\text{Point (probe)} = \begin{bmatrix} 0.578 \\ 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} + \begin{bmatrix} -0.01 \\ -0.05 \\ 0.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0.568 \\ -0.05 \\ 0.0 \\ 1.0 \end{bmatrix}$$

Coordinates of Toolpoint in Toolpoint Reference Frame =

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

**STEP 3**

Transform Point to T1 Coordinates

$$\text{Point (T1)} = \begin{bmatrix} 0.707 & 0.707 & 0.0 & 0.886 \\ -0.707 & 0.707 & 0.0 & -0.473 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 0.568 \\ -0.05 \\ 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 1.252 \\ -0.91 \\ 0.0 \\ 1.0 \end{bmatrix}$$

**STEP 4**

Transform Point to Ground Coordinates

$$\text{Point (GND)} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.854 \\ 0.0 & 1.0 & 0.0 & 0.663 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 1.252 \\ -0.91 \\ 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 2.106 \\ -0.247 \\ 0.0 \\ 1.0 \end{bmatrix}$$

**STEP 5**

Compare to Ideal Coordinates

$$\text{Delta} = \begin{bmatrix} 2.149 \\ -0.219 \\ 0.0 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 2.106 \\ -0.247 \\ 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 0.043 \\ -0.028 \\ 0.0 \\ 1.0 \end{bmatrix}$$

This indicates that a deflection in the probe of (-.01, -0.05, 0) results in a toolpoint error of (0.048, -0.028, 0) with respect to the Reference coordinate system

Figure 2.12: Example of Toolpoint Error Prediction

## **2.7 Chapter Summary**

The sources of machine tool errors is presented, and the basic requirements of representing error motions in a machine model are detailed. A method for creating a system error budget through the use of HTMs is described. As the location of the toolpoint or workpiece-point is transformed through the structural loop, errors associated with each component are included. The result is a determination of the total error between toolpoint and workpiece point based on the errors of each component in the machine tool. This method allows for a direct comparison between machine tool concepts, since the tool-workpiece interaction errors are specified with respect to a reference coordinate frame.

## **3.0 Constraint-Based Design Methods**

### **3.1 Introduction**

A constraint is a relationship involving one or more parameters of a given design concept [Chan, 1987]. Development of any product is essentially a process involving the specification and satisfaction of this set of constraints. The requirements of the new design represent the initial set of constraints, and as the design cycle proceeds the set of constraints is modified to provide a complete description of all possible solutions to the design problem. The responsibilities of the design team include the modification of the constraints associated with the design, and the development of a product concept that is capable of meeting these constraints in an "optimum" fashion.

The process of satisfying all of the constraints and determining this optimum solution, however, is extremely complex. This complexity is due to the coupling of constraints and the inclusion of an individual design parameter in multiple constraints [Watton, 1991]. The definition of some constraints is also difficult, since the exact relationships that must be represented (especially those dealing with aesthetics or user-interface issues) are obscure and/or difficult to specify (i.e. a product must "look appealing"). The number of constraints associated with a design is also large, and manually recording all of the necessary relationships is time consuming and prone to error. For these reasons, constraints are often represented as mathematical relationships (consisting of one or more variables associated with the design concept) for use in computational design tools. This allows for the manipulation and evaluation of a large number of constraints associated with a design. These tools ensure a complete and systematic exploration of the constraints associated with a given concept, and allow for the optimization of a concept within the design space defined by the constraint set.

For the purpose of conceptual evaluation and optimization, constraints provide a method of mathematically describing the form, features, and function of a given concept. Each individual constraint is defined as a mathematical relationship between one or more of the design variables, often resulting in a single design variable being included in more than one constraint. The result is a collection of constraints related by common design variables referred to as a constraint network. An example of a constraint network for a simple part is shown in Figure 3.2. Evaluation of this network results in a determination of which constraints are violated and which design variables must be altered to resolve these violations. Through the use of optimization routines, the variables of the constraint network may be set to optimum values, and the best design solution for the given set of constraints may be determined.

This chapter presents an overview of the use of constraint-based methods in the evaluation and optimization of machine tool design concepts. Section 3.2 covers the details of constraint-based methods, including the mathematical representation of constraints. Section 3.3 deals with the sources of constraints as related to precision machine tools, and the methods for incorporating this information into a constraint network representing the complete machine tool. The remaining section of this chapter provides a brief example illustrating the use of constraints in conceptual evaluation.

### 3.2 Design Constraint Methodology

Design constraints may be represented in a variety of forms. Each of these forms provides information on the performance and characteristics of the design under consideration. For conceptual evaluation and optimization through computational methods, constraints must be represented in a computationally-friendly form [Thornton, 1993]. Two types of constraints meet this requirement: those based on condition/action rules and those based on mathematical equations.

Condition/action (C/A) rules, sometimes referred to as production rules, provide for constraints based on logic. An example of a logical-based constraint is "if an electric motor is included in the machine, the motor must have a power supply". Constraints of this type are typically employed in software for the initial development of machine concepts, or the evaluation of a concept based on the physical requirements of the included components. Development and use of C/A-based constraints typically requires various levels of artificial intelligence; for this reason, C/A-based constraints are considered beyond the scope of this thesis, and the design concepts under evaluation are assumed to be physically realistic and consistent.

The second, and more common, form of mathematical constraints are based on mathematical equations. These constraints represent mathematical relationships between the design variables associated with a specific concept. An example is Newton's Second Law of Motion:

$$F \propto d(mv) / dt$$

which describes a constraint relating the time rate of change of momentum of a component to the force exerted on the component. Mathematical constraints may be expressed as equalities and inequalities consisting of one or more parameters related to the design, and may be

classified as discrete or continuous based on the limitation placed on the design variables included in the constraints.

### 3.2.1 Equality Constraints

The most common form of constraints are equality constraints. These represent exact comparisons between two or more design parameters. Typically, these parameters are represented by variables having specific numeric values, or logical statements representing specific qualities of form or function (for example, black = 1, red = 2). An example of an equality constraint relating the nominal size of a bolt to the nominal size of a hole is:

$$d_{\text{bolt}} = d_{\text{hole}}$$

A similar relationship involving multiple parameters is:

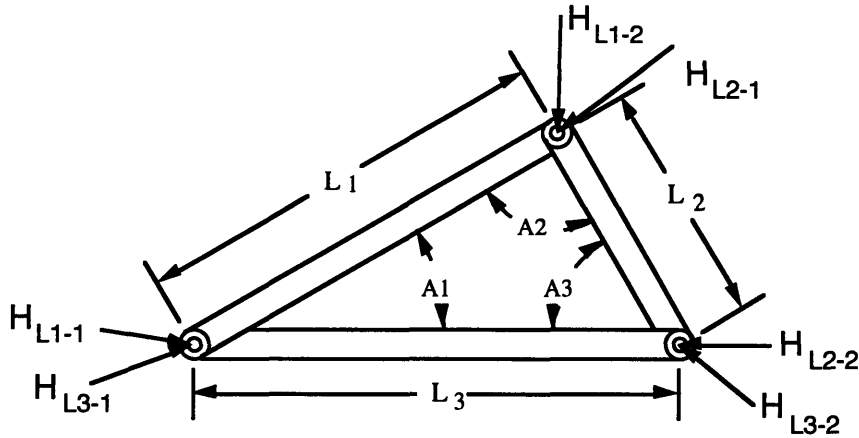
$$\tan \Theta = 2 \cot \Theta (M^2 \sin 2\beta - 1) / M^2 (\gamma + \cos 2\beta - 2)^2$$

Equality constraints may be the result of physical principles, design requirements, geometry, performance, or other characteristics of the design [Colburn, 1990]. Any equation representing exact relationships may be included as an equality constraint, provided the variables included in this constraint exist in the concept representation.

The main difficulty in the use of equality constraints arises from the exact nature of the constraint. In some instances, the constraint is truly exact; for the 3-bar system shown in Figure 3.1a, the constraints listed are physically exact (within manufacturing tolerances). The thermal deflection constraints described in Figure 3.1b, however, are computationally exact but not physically exact, since variations in the material behavior will result in variations in deflection for both each individual beam and beams with "identical" geometry. The result of this variation is a possible report of a violated constraint when the actual magnitude of the violation is insignificant and may be ignored. For this reason, some equality constraints should have an associated tolerance or allowable violation limit, similar to the range of acceptable values described by inequality constraints.

---

<sup>2</sup> this is the oblique shock wave equation for supersonic aerodynamics



**Constraints:**

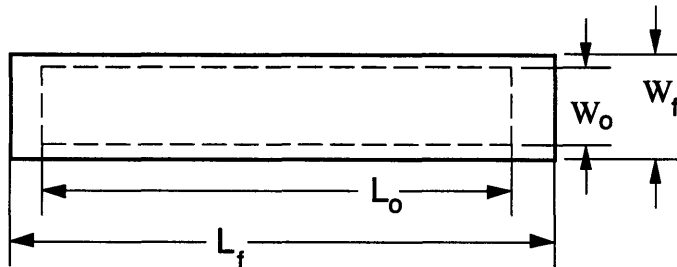
$$A1 + A2 + A3 = 180^\circ$$

$$L_2^2 = L_1^2 + L_3^2 - 2(L_1 * L_3 * \cos(A_1))$$

$$H_{L1-1} = H_{L3-1} \quad H_{L3-2} = H_{L2-2}$$

$$H_{L1-2} = H_{L2-1}$$

(A) Three-Bar System



**Constraints:**

$$L_f = L_o * \alpha \Delta T$$

$$W_f = W_o * \alpha \Delta T$$

(B) Thermal Constraints

Figure 3.1: Examples of Exact and Inexact Constraints

**3.2.2 Inequality Constraints**

Inequality constraints are more complex than equality constraints. An inequality constraint represents a range of possible values of the associated variables for which the constraint is considered valid. This range may be single bounded, providing for only one limit

on the constrained parameters, or double bounded, providing an upper and lower limit for the parameters. For example, the diameter of a shaft may be limited by a single bounded constraint by:

$$d_{\text{shaft}} \geq d_{\text{min}}$$

or limited by a double bounded constraint by:

$$d_{\text{min}} \leq d_{\text{shaft}} \leq d_{\text{max}}$$

As with equality constraints, more complex relationships constraining multiple variables and equations are possible.

Inequality constraints are useful in limiting the parameters of a design to a range of values. Like equality constraints, inequality constraints are typically the result of physical phenomenon ( $F_{\text{applied}} \leq F_{\text{max}}$ ), design limitations ( $F_{\text{applied}} \geq 10 \text{ N}$ ), or geometric relationships. Because of the allowed range of values, inequality constraints are often applied to components whose size or performance does not require exact values. The range offered by an inequality also allows for an increase in the robustness of the design, since the absolute limit of performance is not specified as a fixed number. Inequality constraints also do not suffer from the same evaluation difficulties as equality constraints.

### 3.2.3 Continuous Constraints

Constraints may also be classified by representation into continuous and discrete categories. A continuous constraint is one in which the relationship between the equations in the constraint is defined mathematically, and may take on any value according to the values included in the constraint. An example of a continuous constraint is:

$$\text{Volume} = (\pi * (D^2/4) * h)$$

where the values of  $D$  and  $h$  in the above example may be discrete or continuous; this does not alter the nature of the constraint, however, since the relationship between the two equations is defined for all valid values of  $D$  and  $h$ .

Continuous variables are often associated with custom-designed components, overall machine geometry, and other design parameters that are not considered by the design team to be fixed or of limited sizes. For example, if a single-diameter circular shaft is to be custom

ground, it may be any diameter the grinding machine can produce; in this case, the diameter of the shaft would be represented as a continuous variable. If the shaft is to be purchased from a catalog, however, the diameter is limited by the available sizes as defined by the manufacturer. In this case, a discrete constraint is needed.

### **3.2.4 Discrete Constraints**

Discrete constraints are more complex and less flexible than continuous constraints. A discrete constraint is one that results from a discrete relationship between the equations in the constraint. An example is a look-up table, where the value of one variable is dependent on another in a manner that is not easily classified by mathematics. Because of this inability to mathematically define these relationships, discrete constraints are more difficult to implement in a computational manner.

### **3.2.5 Constraint Network**

Each constraint represents a limitation on the values of the associated variables. Each variable, however, may be included in more than one constraint, resulting in multiple limitations on the value of a given variable. The result is a collection of constraints interconnected by common variables referred to as a constraint network. An example network for a simple design is shown in Figure 3.2. The coupling of constraints results in several difficulties. First, the constraint network is complex for all but the most simplistic representations. This complexity makes the construction, evaluation, and modification of the constraints in the network difficult. The satisfaction of all constraints in a network also presents a difficult problem, since altering the value of one variable to satisfy a certain constraint may result in the violation of another constraint associated with that variable. Often, this problem requires the modification of one or more constraints in the network in order for all constraints to be satisfied. A similar problem arises concerning optimization; because of the large amount of coupling, it is often difficult to determine which variables should be altered, and by how much. Several computational algorithms, including genetic algorithms and simulated annealing, may be applied to constraint networks for optimization of the associated variables (see [Thornton, 1993]).

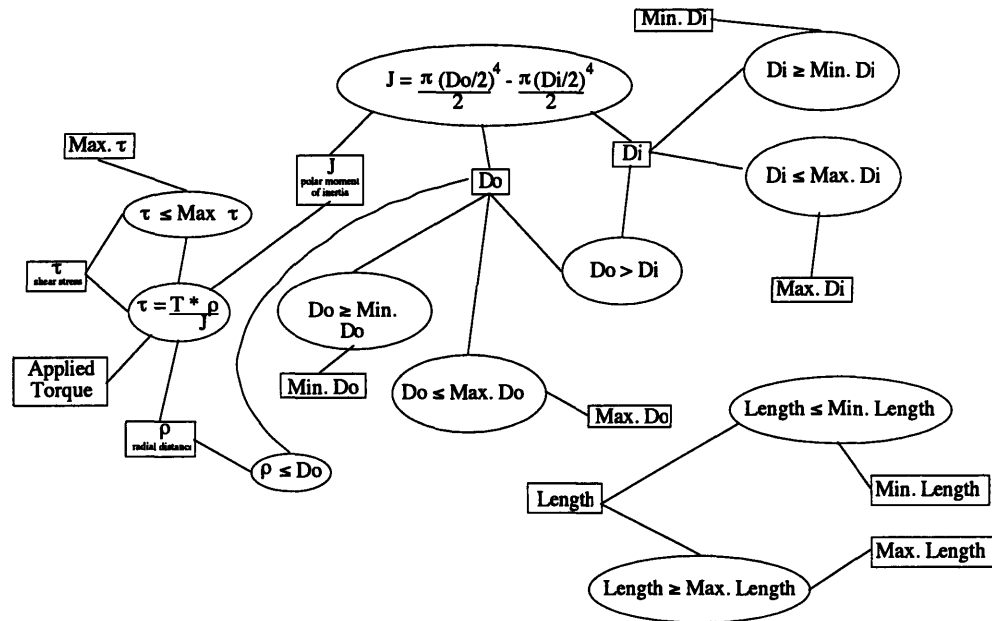
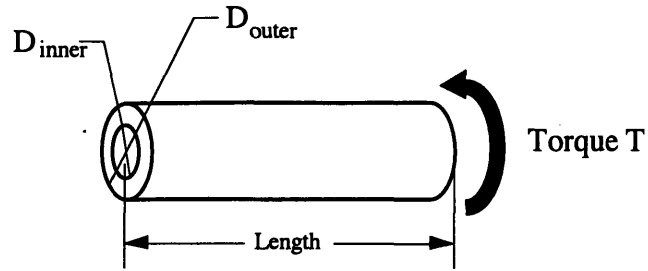


Figure 3.2: Example Constraint Network

### 3.3 Sources of Constraints

The constraints associated with product design in general and precision machine design in specific arise from a variety of factors. Engineering theory, graphs and charts from catalogs, and the knowledge and experience of the design team are a few examples. While the above factors are diverse, constraints are the result of one of the following sources:

- theoretical relationships
- physical characteristics
- empirical information
- heuristics/designer knowledge

An understanding of the above sources of constraints is necessary before the constraints associated with machine tool design may be developed and applied. This section provides a brief description of each of the above sources, including examples of common constraints arising from each source.

### **3.3.1 Theoretical Relationships**

Theoretical relationships provide a detailed description of nearly all aspects of mechanical behavior, and provide exact relationships that define the limitations imposed by the laws of nature. Since the majority of theoretical relationships are represented in the form of equations, they are ideal sources for constraints on machine tool design concepts. Examples of constraints resulting from theoretical relationships are Bernoulli's equation, Newton's Second Law of Motion, and beam buckling equations. Examples of theoretically-induced constraints are found in the pressure vessel example described in Section 3.4.

While the majority of theoretical relationships useful in concept evaluation are relatively simple, it is possible to have complex theoretical constraints. For example, the thermal deformation of a beam may be described by:

$$\Delta L = \alpha L_0 \Delta T$$

This equation represents a valid constraint on the design, as the change in length of the beam must follow the above relationship. However, if a complex temperature gradient is expected, or a more complex thermodynamic determination of heat flows is desired, a more complex method of determining the thermal deflection of the beam may be included. While the method for determining the thermal deflection of a beam may differ, the constraint on thermal deformation still exists; the beam still deflects due to thermal effects.

### **3.3.2 Physical Characteristics**

Physical characteristics of a machine tool provide constraints relating to the geometry or physical properties of the machine. Unlike theoretical constraints, constraints resulting from physical characteristics represent the spatial make-up of a component, and not the limits based on nature. Constraints representing the desired and required geometry of the component are the result of physical characteristics, as are limitations imposed by the design team on appearance, color, and aesthetic features.

An example of constraints arising from physical characteristics is a standard metallic washer. A designer may wish to use a 1 inch diameter bolt, which constrains the desired inner diameter of the washer to 1 inch:

$$D_{\text{inner}} = 1.0''$$

This constraint also forces the outer diameter of the washer to obey the constraint:

$$D_{\text{outer}} \geq D_{\text{inner}} \geq 1.0''$$

since the outer diameter cannot be less than the inner diameter. The first constraint is an example of a desired geometric constraint, while the latter constraint is required by physical necessity and the existence of the washer.

### **3.3.3 Empirical Information**

The majority of parts comprising a precision machine tool are standard parts available from a supplier. Each of these standard parts operates in a specific fashion, and the manufacturer of the part typically provides information on the performance of their product. This information is often based on experiments performed on a large number of parts, and the data is often presented in tables, graphs, or equations with correction factors based on operational conditions. This information must be incorporated into the evaluation of any machine tool concept employing a standard component in order to insure the part operates within allowable limits and its potential values will still satisfy the design requirements.

Unlike theoretical and physical relationships, empirical information is not easily incorporated into mathematically-represented constraints. The information provided by the manufacturer is often not easily represented as equations. For this reason, charts and graphs often must be decomposed into look-up tables or similar computationally-friendly representations. An example is the roller bearing raceway hardness factor graph shown in Figure 3.3. This graph represents information that must be incorporated into any constraints concerning the raceways of rotary bearings, but is not easily represented in a computational manner. Several of the methods available for the inclusion of empirical information in constraints are described in Chapter 5.

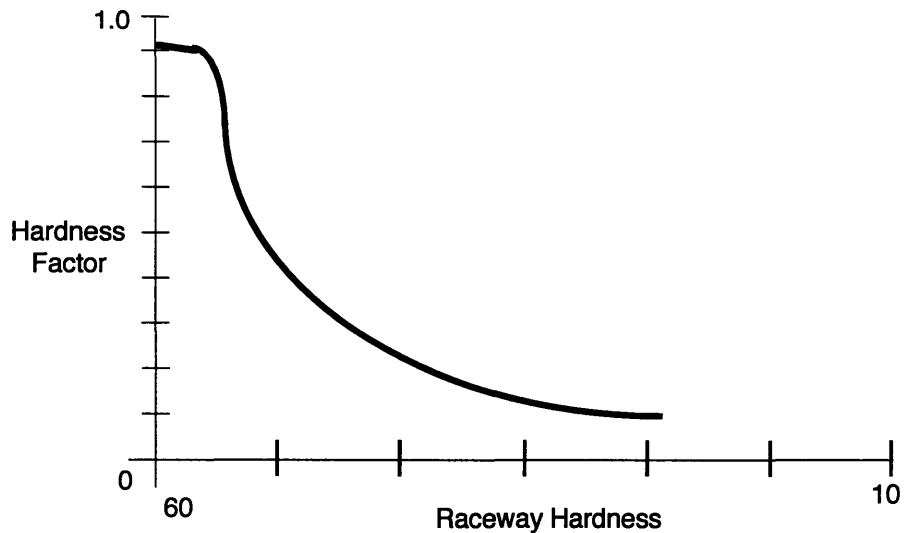


Figure 3.3: Example of Empirical Information

### 3.3.4 Heuristics / Design Knowledge

The combined knowledge of the design team provides an additional source for constraints on a proposed design. These constraints are the result of the experience of the individual designers, as well as the feedback from end-users, component suppliers, and others involved in the design cycle. One example of a constraint resulting from heuristics is the Golden Rectangle Rule, which states that machine carriages should have an aspect ratio of 1.618:1 or less to provide stability when driven through the roll center of the carriage. While this rule is not a requirement, it does provide the design team with a sound basis for component sizing, as well as an indication of the required size and shape of the carriage.

Like constraints resulting from empirical information, constraints based on heuristic information are difficult to classify mathematically. There are several methods to approach this problem. One method involves the determination by the designer of what variables should be influenced by the heuristic knowledge. The designer may then create a set of constraints to enforce the required relationships. Another method is to set variables equal to constant values set by the designer. Other heuristic knowledge, such as poop performance between certain series of components, may be included on a "per-component" basis in order to alert the designer of possible conflicts.

### 3.4 Constraint Example

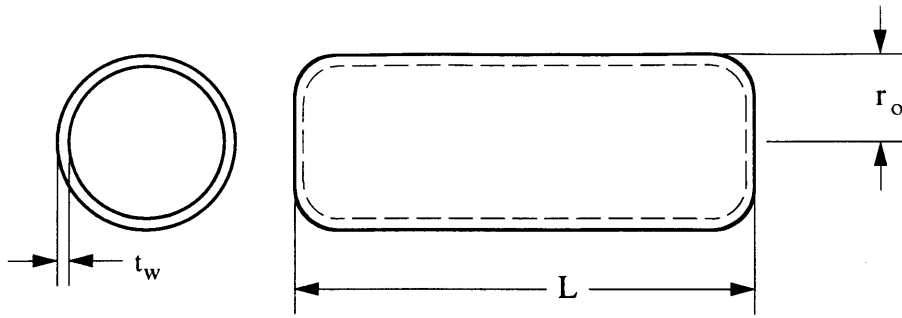
This section presents a comprehensive example illustrating the development and use of constraints for a simple design. The system selected for the example is a closed thin-walled cylindrical steel pressure vessel, as seen in Figure 3.4a. The basic mechanics of thin-walled cylindrical pressure vessels is found in [Popov, 1990]. The specified requirements for this vessel are as follows:

- withstand pressurization up to 0.80 MPa
- wall deflection at 0.80 MPa of less than 0.5 mm
- outside radius of at least 1000 mm

Additional requirements are not addressed, although the methods demonstrated in the example can be applied for a design problem involving many more limitations.

The variables and constraints associated with the design of the cylinder are shown in Figure 3.4b. The variables represent all features of the design that must be specified or determined to meet the above requirements, while the constraints represent the limitations on these features. The geometric variables describe the geometric features of the vessel pertinent to fulfilling the requirements stated above, while the engineering variables describe the stresses, strains, and other factors associated with the performance of the vessel. Likewise, the sources of constraints include physical and theoretical relationships; empirical and heuristic constraints do not apply and are not included. Variables and constraints relating to other factors, such as appearance or more complex geometric features, are not included.

Tables 3.1a and 3.1b show the results for the first iteration of the constraint network. This iteration begins by selecting values for the design variables  $r_o$ ,  $t_w$ , and  $P_i$ . Once these values are set, the equations included in each constraint may be calculated, and the constraint evaluated. The existence of violations in the constraint system indicate that the design described by the selected values of the design variables is not valid for the given set of constraints. This situation may be corrected by either modifying the values of the design variables, or by altering the violated constraints. In this example, the constraint set will remain constant, indicating that resolution of the constraint violation must be accomplished through the selection of new design variables.



(A) Cylindrical Thin-Walled Pressure Vessel

**Variables:**

**Geometry:**

Outside Radius:  $r_o$   
 Wall Thickness:  $t_w$

**Engineering:**

Modulus of Elasticity:  $E$   
 Poisson's Ratio:  $\nu$   
 Maximum Allowable Stress:  $\sigma_{allow}$   
 Internal Pressure:  $P_i$   
 Hoop Stress:  $\sigma_h$   
 Hoop Strain:  $\epsilon_h$   
 Longitudinal Stress:  $\sigma_l$   
 Longitudinal Strain:  $\epsilon_l$

**Constraints:**

**Design Specifications:**

$r_o \geq 1000\text{mm}$   
 $P_{max} \leq 0.80\text{ MPa}$   
 $\Delta r_o \leq 0.5\text{mm @ } P_{max}$   
 $\sigma_{max} = 165\text{ MPa}$   
 $E = 200\text{ GPa}$   
 $\nu = 0.25$

**Geometric:**

$t_w > 0.0$   
 $r_o > 0.0$   
 $r_i < r_o$   
 $r_o - r_i = t_w$

**Engineering:**

$$\sigma_1 = \frac{Pr_i}{t} \quad \sigma_1 \leq \sigma_{max}$$

$$\sigma_2 = \frac{Pr_i}{2t} \quad \sigma_2 \leq \sigma_{max}$$

$$\epsilon_1 = \frac{\sigma_1}{E} - \nu \frac{\sigma_2}{E}$$

$$\Delta = \epsilon_1 r_o$$

(B) Pressure Vessel Variables & Constraints

Figure 3.4: Constraint Example for a Thin-Walled Pressure Vessel

<u>Variable</u>	<u>Value</u>
$r_o$	1000 mm
$t_w$	5 mm
$P_i$	0.80 MPa
$E$	200 GPa
$\nu$	0.25

Table 3.1a: Design Variables for Iteration 1

<u>Constraint</u>	<u>LHS</u>	<u>Operator</u>	<u>RHS</u>	<u>Violation</u>
$r_o \geq 1000$ mm	1000 mm	$\geq$	1000 mm	No
$P_{i,max} \leq 0.80$ MPa	0.80 MPa	$\leq$	0.80 MPa	No
$E = 200$ GPa	200 GPa	$=$	200 GPa	No
$\nu = 0.25$	0.25	$=$	0.25	No
$t_w > 0.0$	5.0 mm	$>$	0.0	No
$r_o > 0.0$	1000 mm	$>$	0.0	No
$r_i = r_o - t_w$	995 mm	$=$	995 mm	N/A
$\sigma_1 = (P_i * r_i) / t_w$	159.2 MPa	$=$	159.2 MPa	N/A
$\sigma_2 = (P_i * r_i) / 2*t_w$	79.6 MPa	$=$	79.6 MPa	N/A
$\epsilon_1 = (\sigma_1 / E) - \nu(\sigma_2 / E)$	6.965E-4 m/m	$=$	6.965E-4 m/m	N/A
$\Delta r_o = \epsilon_1 * r_o$	0.6965 mm	$=$	0.6965 mm	N/A
$\sigma_1 \leq \sigma_{max}$	159.2 MPa	$\leq$	160.0 MPa	No
$\sigma_2 \leq \sigma_{max}$	79.6 MPa	$\leq$	160.0 MPa	No
$\Delta r_o \leq 0.5$ mm	0.6965 mm	$\leq$	0.5 mm	<b>YES</b>

Table 3.1b: Constraint Evaluation for Iteration 1

The second iteration for the constraint network is shown in Table 3.2. The values for the design variables are modified, and the constraint violations are resolved. This indicates that the selected design is valid for the given set of constraints, and that further interactions (based on the above set of constraints) are not required.

The above example illustrates one of the difficulties concerning the use of constraints in design. The lack of completeness of the above constraint network results in a design that is far below the optimum design. This is determined by the fact that the hoop and lateral stresses are

<b>Variable</b>	<b>Value</b>
$r_o$	1050 mm
$t_w$	10 mm
$P_i$	0.80 MPa
$E$	200 GPa
$\nu$	0.25

Table 3.2a: Design Variables for Iteration 2

<b>Constraint</b>	<b>LHS</b>	<b>Operator</b>	<b>RHS</b>	<b>Violation</b>
$r_o \geq 1000\text{mm}$	1050 mm	$\geq$	1000 mm	No
$P_{i,\text{max}} \leq 0.80 \text{ MPa}$	0.80 MPa	$\leq$	0.80 MPa	No
$E = 200 \text{ GPa}$	200 GPa	$=$	200 GPa	No
$\nu = 0.25$	0.25	$=$	0.25	No
$t_w > 0.0$	10.0 mm	$>$	0.0	No
$r_o > 0.0$	1050 mm	$>$	0.0	No
$r_i = r_o - t_w$	1040 mm	$=$	1040 mm	N/A
$\sigma_1 = (P_i * r_i) / t_w$	83.2 MPa	$=$	83.2 MPa	N/A
$\sigma_2 = (P_i * r_i) / 2*t_w$	41.6 MPa	$=$	41.6MPa	N/A
$\epsilon_1 = (\sigma_1 / E) - \nu(\sigma_2 / E)$	3.64E-4 m/m	$=$	3.64E-4 m/m	N/A
$\Delta r_o = \epsilon_1 * r_o$	0.382 mm	$=$	0.382 mm	N/A
$\sigma_1 \leq \sigma_{\text{max}}$	83.2 MPa	$\leq$	160.0 MPa	No
$\sigma_2 \leq \sigma_{\text{max}}$	41.6 MPa	$\leq$	160.0 MPa	No
$\Delta r_o \leq 0.5 \text{ mm}$	0.382 mm	$\leq$	0.5 mm	No

Table 3.2b: Constraint Evaluation for Iteration 2

well below the allowable value, indicating that the vessel is stronger than necessary. The constraints necessary to indicate the level of "over-design", however, are not included in the above constraint network, which prevents the constraint of the design variables against over-design. This is a result of the lack of completeness in the constraint set, and illustrates the

necessity of specifying a set of constraints that limits all aspects of the design space. Once the constraint network is verified to be complete for the given problem, the network may be optimized through a number of methods; an overview of optimization methods for constraint networks is found in [Thornton, 1993].

### **3.5 Chapter Summary**

This chapter presents an overview of the development of constraints for use in the process of design evaluation and optimization. The sources of constraints include theoretical relationships, physical characteristics, empirical knowledge, and heuristics. Each constraint for a given design is defined as an mathematical relationship between two equations, and the inclusion of variables in more than one constraint results in relationships among constraints, referred to as a constraint network. Once the constraint network for a given design is developed, a design (represented by a set of design variables) may be evaluated against the constraint set to determine the validity of feasibility of the design. Optimization of the design variables may take place to determine the optimum value of the design variables for the given set of constraints.

## **4.0 PMDA - Object Representations**

### **4.1 Introduction**

Chapter 3 showed that machine tool design can be seen as a problem involving the specification and satisfaction of a set of design constraints. The information necessary for the development and evaluation of these constraints is often associated with each specific component selected for use in the machine tool. Because of this, development of machine tool concepts can be modeled as an object-oriented process, where each object contributes a set of design variables and design constraints to the overall machine tool concept. At the early stages of design, each component in the machine is specified in generic terms, since only the basic functionality of the component is necessary for development of the initial concept. As the design progresses, however, additional information associated with each component is included in the model. This process continues until the detail phase of design, where each component is completely defined, usually through the selection of a commercially-available component from a manufacturer catalog. Throughout this process, the increasing levels of information provide the opportunity for increasingly accurate predictions of the total machine tool performance, cost, and operational characteristics.

The procedures for machine modeling in PMDA, as well as the constraint-satisfaction systems and the machine modeling routines, are also object-oriented. Each component is modeled as an independent computational entity based on a standard representation of the associated mechanical component. Increasing levels of detail for each object are provided by additional levels of abstraction. The connections between components, referred to as interfaces, are also represented as objects, as are all of the variables, equations, and constraints associated with each component. This results in a completely object-oriented approach to machine modeling, evaluation, and optimization.

This object-oriented approach has several advantages over other methods of model development (such as procedural-based methods). First, this approach is similar to the actual process of machine design, allowing a designer to model concepts in terms of the individual components included in the concept. A second advantage is the completeness and consistency of component representations; each case of a given type of component (rotary bearing, linear bearing, carriage) will be represented in exactly the same manner, resulting in standard methods of component modification and manipulation, computational efficiency, and a complete understanding of the information that must be provided for each specific component. Use of component objects also simplifies the process of incorporating component information based on commercially-available parts. This allows the designer to evaluate or optimize a given

concept based on actual manufacturer-supplied parts, ensuring that each concept will be realistic and commercially available.

## 4.2 Requirements of PMDA Objects

The object representation for a given component must contain all of the information associated with the component required by the design software. Information related to the connectivity of the component to the remainder of the system model must also be included. For the process of conceptual evaluation and optimization, this information includes data on part geometry, operation, and performance, as well as heuristic data and other information concerning the functionality of the component. While components vary widely in their form and function, the required information for a given object may be divided into two main categories: component-specific parameter and component-connectivity parameters. The following is a description of these types of information and the methods for representing the related information.

### 4.2.1. Component-Specific Parameters

While the components comprising a typical machine tool will vary widely in their construction and function, the information that must be included in a generic object to describe each component is similar regardless of the specific design of the component. For a given component, the required object information may be classified in the following categories:

- **Geometry** - all geometric features necessary to describe the shape of the part
- **Loading** - all applied loads, as well as all maximum allowable loads, methods for load calculation/distribution, load factors, etc.
- **Operational** - lubrication, cooling requirements, vibration effects, etc.
- **Design Heuristics** - rules based on the experience of the designer, such as the ability of a component to carry more than the specified maximum load
- **Error Motions** - non-desired motions resulting from the operation of or loads applied to the component
- **Miscellaneous** - purchase or maintenance costs, catalog number, etc.

## **Geometry**

Geometry refers to any information concerning the unloaded dimensions or spatial relationships of the component (deflections due to loading are considered error motions and are dealt with separately). This includes the complete set of basic dimensions for the component, as well as dimensional tolerance and surface finish information. The basic dimensions represent the size and shape of the component, and are stored as single variables in the component template. All dimensional information necessary to completely represent the size and shape of the component must be included. Tolerances, which indicate the conformity of the actual part dimensions to the ideal dimensions, may be represented as single values for maximum or minimum deviations, or as equations describing the statistical distribution of tolerances associated with a specific feature of the part. Surface finish, which is a measure of the smoothness of a surface, should be included when the surface of the part must conform to a certain standard.

Geometric tolerance information can be used in the modeling of component errors (see Chapter 5). While the magnitude of certain error motions are directly related to the magnitude of a given geometric tolerance, error motions are distinctly different from geometric tolerances, and must be treated as such. For this reason, error motions are treated as a separate class of information, and are discussed below.

## **Loading**

The information related to the reaction of a component to external and internal forces includes several factors. First, the maximum loading possible for a component describes the externally-applied load at which the component is expected to fail. This may be represented as one absolute value, as several values for different operating conditions, or as a statistical distribution indicating the likelihood of failure at a given load. For example, maximum loads for linear or rotary bearings are often specified for both dynamic and static conditions. Since maximum loads are often expressed in terms of explicit numbers, they may be represented in objects as one or more variables. These variables will later be included in the constraints limiting the actual loads on the component. In the case of different maximum loads for different loading conditions, care must be taken to include the loading conditions for which each maximum load is valid.

A second type of loading information is the methods for calculating the load placed on a component. For example, a shaft supported by two rotary bearings will exert a force on each of the bearings; the division of this force, and the resulting radial and axial forces placed on

each bearing is often determined by a set of equations. These equations may either be derived from basic statics and dynamics, or be supplied by the manufacturer of the component based on component-specific reactions to special loading situations. In the case of basic statics, the equations may be included directly in the object, while the equations supplied by a specific manufacturer must be included as special cases. In either situation, the equations determine the actual load on the components, and will be included in the constraints on maximum load. Loading information of this type will also be used to determine load-induced error motions, which are described in a separate section below.

The third type of loading information required in standard component representations involves preload. Preload is the load placed on a component to increase the stiffness of the component through deformations (preload is explained in detail in [Slocum, 1992]). While preload is often thought of in terms of linear or rotary bearings, it may be applied to any component (including the structure of the machine tool itself) to increase stiffness. The assembly of the machine tool may also result in the generation of preload, since the static weight of components can result in significant force on the supporting components even when the machine is at rest. Preload affects a component in several ways, including the increase in static and dynamic friction and increases in the deflections of certain features of the component. Loading deflections and maximum loads may also be affected by preload, and major changes in performance may result if the preload is sufficiently high.

Including the effects of preload in an object is, unfortunately, not a simple matter. While the actual value of the preload may be represented as a single variable, the effects of this force require more complex treatment. Often, manufacturers will provide empirically-derived data to help a designer predict the reaction of a component to preload forces. If this data is not available, the designer must develop estimates to account for preload effects.

## **Operational**

Operational parameters describe two types of information related to the normal function of a component: requirements for normal operation of the component and the effects of normal operation of the component. The requirements for normal operation include the various features necessary to support the operation of the component. Examples include the following:

- lubrication requirements - type of grease or oil, required flow rates
- cooling requirements - type of coolant, flow rate, cooling capacity
- electrical power supply requirements - peak and continuous requirements
- level of vibration isolation - dampening requirements
- level of thermal insulation - insulation and environmental requirements

These requirements may be often be expressed in simple terms, such as variables specifying volume per unit time or kilowatts per hour, or in terms of acceptable levels of heat or vibration transmission to the component. The actual requirement of the inclusion of these support components and supplies, however, must either be explicitly included in the model by the designer, or included based on condition/action rules (see Section 3.1). Once the proper support equipment is specified, the constraints on operation based on the above operational concerns may be set and included in the overall machine model.

## **Design Heuristics**

While the majority of information concerning a component is based in engineering theory and may be described through mathematics, some constraints are the result of experience or intuition. This type of design knowledge, referred to as heuristic information, is of great value for the design and development of machine tools. Preliminary concept generation is often dependent entirely on heuristic information, since detailed engineering analyses or computational simulations cannot be applied to a machine that is not yet completely detailed. The development of user-interfaces, aesthetic issues in design, and ergonomic factors are also heavily dependent on heuristics.

In terms of conceptual evaluation and optimization, heuristic information can provide a variety of "practical" measures against which the machine is evaluated. An example is the maintenance of a machine tool; once a machine is assembled, certain areas must be accessible to maintenance personnel. If access to the parts typically requiring service is blocked or severely limited, routine service may require disassembly of certain parts of the machine. Another example is the use of a tool chuck; if the area around the chuck does not allow proper access, manually changing a tool will be a difficult and time-consuming process. While these types of requirements are easily stated in words, they are difficult to represent computationally. Often, heuristic-based requirements must be entered into the component representations by the designer as a set of constraints. The designer must set the constraints on the variables associated with the given heuristic information, and may have to define new variables or equations to ensure the correct relationships are enforced.

## **Error Motions**

While error motions are associated to several of the other basic types of template information discussed above, it is often useful to consider error motions as separate entities. This allows each error motion to be simulated independently and then combined through

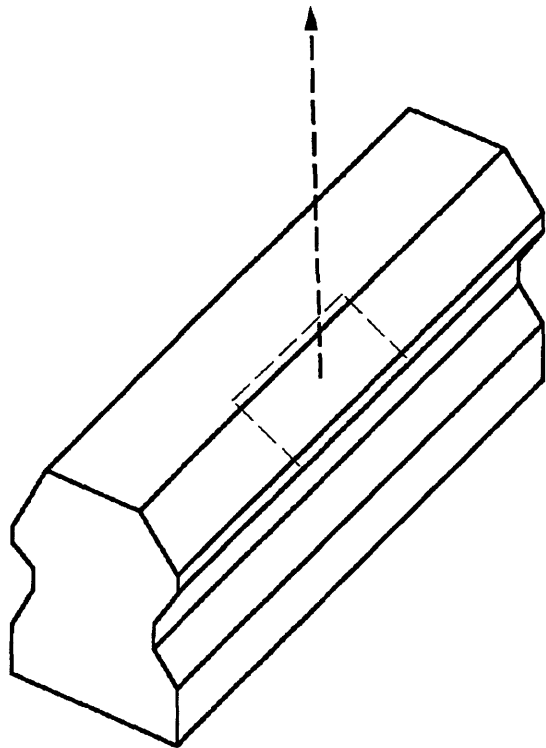
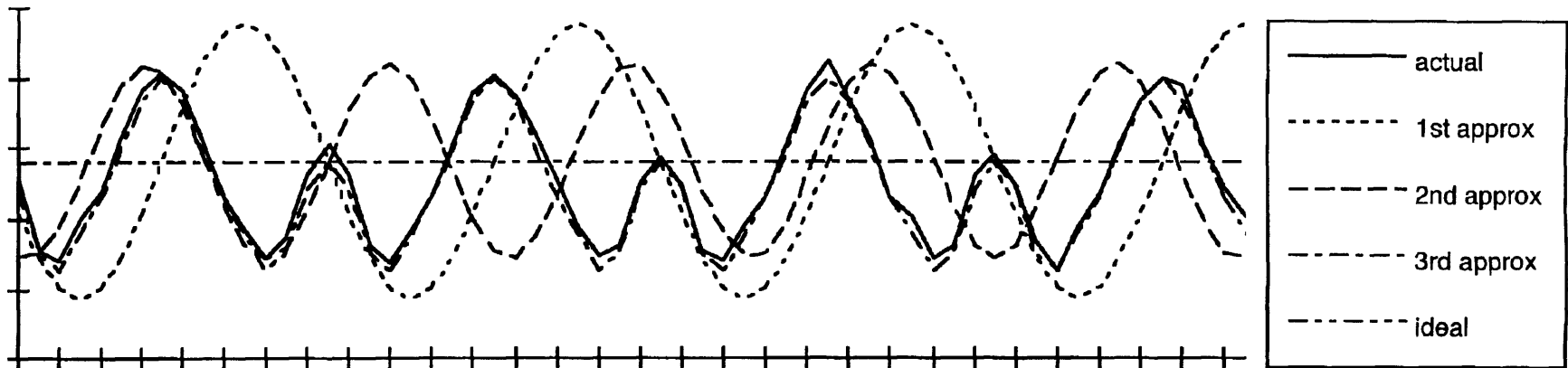
standard computational routines. This also allows the contributions to each error motion to be isolated and altered as necessary to determine the overall affect on total machine tool error.

The sources of general component error motions are discussed in Chapter 2, as is a method for combining the individual error motions for a given component into a total error motion for that component. These methods of error combination require a method for determining and representing each individual error motion associated with a component.

In order to accurately determine the effect of error motions on the operation of a component, a model of each error motion must be developed. This model will be highly dependent on three factors: the actual motion being modeled, the intended function and operation of the specific component, and other factors such as statistical information regarding the magnitudes of the error motions. For example, part of a linear bearing rail is shown in Figure 4.1 with one possible associated error motion. This motion results from rail straightness errors in the direction perpendicular to the rail. While straightness errors are easily described, they are not easily modeled due to the wide variations possible in the actual magnitude of the error motion as a function of position along the rail. The variation between bearing systems of the same type from the same manufacturer must also be incorporated, and error motions for similar components may vary widely depending on the method of manufacture of the given components.

In the example shown in Figure 4.1, the form of the bearing rail may be modeled in several manners. The first approximation uses a standard sinusoidal wave to model the shape of the rail. The second approximation is a combination of sine and cosine functions, resulting in a more accurate modeling of the rail profile. The third approximation is a combination of sine and cosine functions of different amplitude and frequency, and is the most accurate when compared to the actual form of the rail. This model is, however, a much more complex representation and may be prohibitively expensive to develop or implement (especially for use in a conceptual evaluation environment). And, while the above models approximate the shape of the bearing rail, they do not accurately predict the error motion of the bearing block, which will undergo both the  $\pm X$  displacements and "slope" deflections shown graphically in Figure 4.1. For this reason, the error motion of the bearing block resulting from the straightness error of the bearing rail is not only a function of the rail shape, but of the length of the bearing block.

Similar difficulties are associated with the modeling of other error motions, and, as with much of design, the level of complexity of the error motion model must reflect the necessary accuracy of the machine model. For conceptual evaluation and optimization, error motion representations do not require the level of accuracy of computational simulations, but care must be taken to model component error motions in a consistent manner. It is often acceptable and prudent to use simple geometric representations to model error motions at this



**1st Approximation:**  $= (1 - (\sin(8 \cdot B_4)/4))$

**2nd Approximation:**  $= (1 - (((\sin(11 \cdot B_4)/4) + (\cos(11 \cdot B_4)/4))))/2$

**3rd Approximation:**  $= (\text{ABS}((((\sin(4 \cdot B_4))^2 + (\cos(8 \cdot (B_4))^4))))^{0.2})$

The ideal flat surface of the bearing rail is actually a complex shape, similar to the "actual" curve in the graph above. This non-ideal shape is the result of errors in the rail manufacturing process.

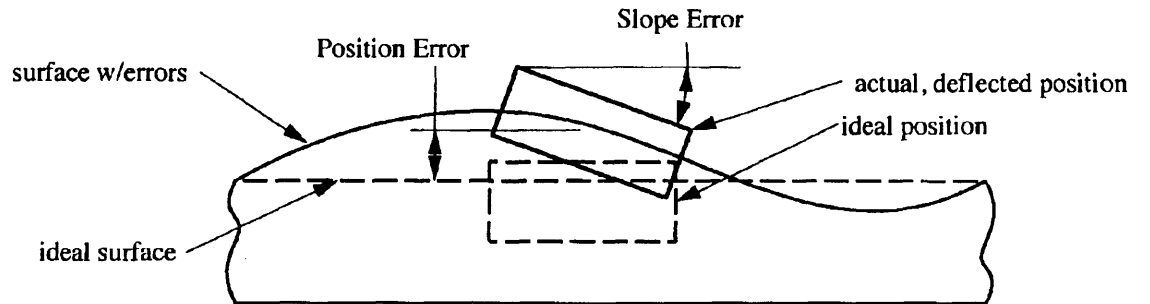


Figure 4.1 : Linear Error Motion Representation Example

stage of design. Some error motions may be represented in terms of one or two variables, while others may be modeled in terms of simple equations. For example, eccentricity of the cross-section of a shaft can be modeled as an ellipse, with major and minor axes and angle of rotation set in order to match the specific eccentricity of the shaft being modeled. Statistical data supplied by manufacturers may then be used to determine the values of these variables, or the designer may set the values based on expected error values or past design experience. Details of these representations for a shaft and a rotary bearing are described in Chapter 6, as are basic methods for determining the values for inclusion in the error motion models. Methods for error motion representation for PMDA components are found in the appendices.

## **Miscellaneous**

There is a potentially wide set of information not included in the previous categories. Cost of purchase and maintenance, catalog numbers, and manufacturer and source information may be considered miscellaneous information. This should not imply, however, that this information is trivial or may be ignored. Data of this nature must be included in a machine model to ensure a complete evaluation and/or optimization is possible, and to ensure the constraint set associated with the machine tool is complete.

### **4.2.2 Component-Connectivity Information**

The second type of basic information that must be contained in a machine tool model concerns the connection of components. In a precision machine tool, each component is connected to at least one other component through mechanical means. These connections influences the performance and behavior of the connected components in several ways. First, the connection defines the geometric arrangements of components in the machine; for the methods presented in Chapter 2, the location of each component is represented with respect to the component closer to ground in the structural loop. Second, the connection between components constrains the operation and performance of the connected components. For example, the rotational speed of a shaft must be the same as the rotational speed of the attached rotary bearing. Third, the actual joint between components may influence the performance or operation of one or both of the connected components. These factors must be included in any machine tool model to provide the information necessary to accurately construct and evaluate the model.

## **Geometry of the Connection**

In order to specify a connection between two components in a machine tool, the geometry of the connection must be specified regardless of the type of connection or the type of components. The required geometric information includes the position of the origin of each component and the angular rotations of the axis system of one component with respect to the attached "base" component. This information is easily and completely represented in the form of an HTM, which describes the geometric position of one axis system with respect to a reference axis system (see Chapter 2). For this reason, a component object should include an HTM or similarly complete description to specify the location of the object with respect to the "base" object. The objects must also possess the routines necessary for the construction of the HTM based on the locations of the components being connected.

## **Operational / Performance Information**

The performance and operational information associated with a connection between two components describes the parameters that are related between components due to the connection. Examples are common rotational speeds, ratios between geared objects, and forces transmitted between objects. Any instance where performance information from one component is reflected onto the performance parameters of another must be included. Multiple examples are illustrated by the connection of a linear motion bearing to a machine carriage, shown in Figure 4.2. The forces applied to the carriage in the Y and Z directions are transmitted to the bearing, while the X-direction force results in bearing motion. This motion results in movement of both the bearing block and the carriage, and the acceleration and velocity of the bearing must equal the velocity and acceleration of the carriage. Other shared information includes the heat transfer between the bearing block and the carriage, vibration transmitted through the system, and error motions in the bearing altering the position and orientation of the carriage.

As seen in the above example, all information common to the operation and performance of connected components must be included in the description of the connection. This ensures that all variables, performance, and operational information is consistent and accurate throughout all components in the machine model.

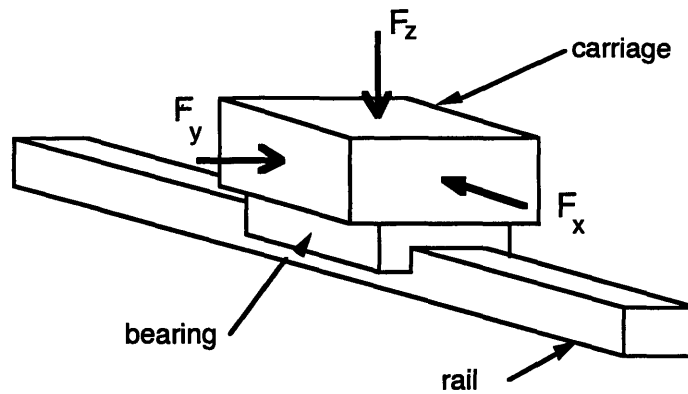


Figure 4.2: Example of Shared Performance Information

### Joint Information

The physical nature of the joint between two components can influence the performance or operation of the connected components in many complex ways. An example is the vibrations transmitted through the joint between a tool holder and the machine structure. The vibrations may be transmitted with little or no loss of amplitude, or the joint may damp the vibrations by several orders of magnitude. Many factors influence this behavior, such as the connection methods (bolts, adhesive, etc.), materials, and use of dampening layers. Because of the complex nature and lack of understanding concerning the effects of joints, prediction of joint interactions is often dependent on the experience and knowledge of the design team. For this reason, joint effects are not included in PMDA, although inclusion through design heuristics is possible (see above).

### 4.3 Object Hierarchy and Representations

Once the information required to model each component in a machine tool is defined, the computational representations of these components can be developed. This section presents the details concerning the form and functionality of the component and interface objects in PMDA.

### 4.3.1 Overview

The object hierarchy for PMDA is presented in Figure 4.2. The top level object, common to all component object, is the Block class. The Block object stores all of the information common to each component in the machine, including the component HTM and basic overall geometry. Each Block object defines the connectivity of components in the machine model, but contains no information as to the specific component it is associated with. This component-specific information is housed in the objects that inherit the Block object. The next level of abstraction represents generic component objects, or "templates". These are the basic definitions of component objects in PMDA; they contain a complete representation of each mechanical and performance-related feature of the component, as well as any other required information. Note, however, that these template objects only provide the definition of the component and the pointers required for the variables, equations, and constraints associated with the various features of the component. Actual values for these specific variables, as well as the equations and constraints describing the given component, are provided by the third level of objects. This level represents the information specific to the particular instance of the object. Both custom and manufacturer-specific information is found at this level. Custom objects are included as "blank" templates that allow the user to define the information specified by the template class. Manufacturer-specific objects represent catalog-based, commercially-available components, and are discussed below.

The general method for inclusion of a commercially-available, manufacturer-specific component into PMDA is as follows. The generic component type is determined; for example, a circular shaft manufactured by Berg would be represented by the standard circular shaft template object (see Figure 4.3). This standard shaft object contains all of the variable, equation, and constraint information associated with the shaft as required by the evaluation and optimization routines in PMDA. The information related to this specific shaft (as manufactured by Berg) is housed in a separate, but related, object. Figure 4.3 illustrates this example for a subset of the shaft template data.

As seen in the figure, the template class supplies the storage space required to house the shaft information. The actual instances of these variables, equations, however, are housed in the manufacturer-specific object, illustrated in right side of the figure. This object is essentially a subclass of the component template class, and provides the means for completing the template object. Each manufacturer of a given component type is included as a separate subclass of the given template object. This ensures that the numeric values, equations, and constraints included in the given component object accurately reflect the information supplied by the

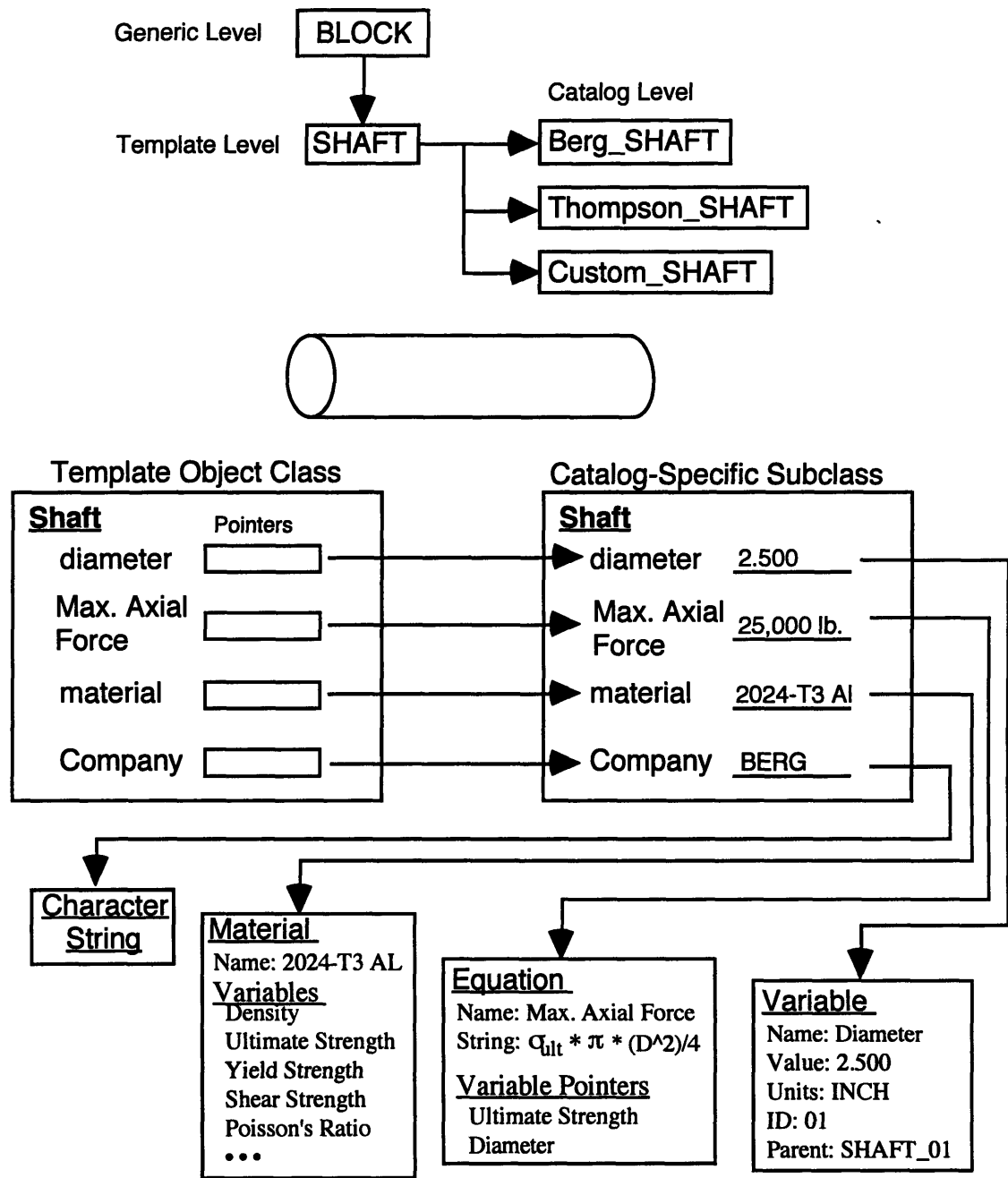


Figure 4.3: Shaft Template and Catalog Objects

manufacturer for this type of object. This method is applied to all component objects in PMDA that will use manufacturer-supplied data.

This organization of objects is selected for several reasons. First, the use of a generic superclass for all machine components allows for a standard representation of common data, and avoids the needless repetition of information in each class object. Second, this organization provides for increasing levels of detail, so that subcomponents of slightly differing type may be implemented at one of several levels. This approach also results in a minimum of repetition of information, and allows for a minimum of development effort for inclusion of new components. Finally, the object hierarchy provides a method of including manufacturer-specific catalog data into the standard template classes in an efficient and easily-implemented manner.

For the incorporation of catalog values, each object must have the input routines necessary to read in the required values. For example, in the shaft example above, the values for diameter and maximum axial force would be read in by the associated shaft object directly from the Berg shaft catalog. The string for the company name would be read as well, and the material would be stored temporarily as a character string and used later to set the pointer to a material object after the material object is read from the material catalogs (this may take place before or after the shaft data is read). Equations specific to the shaft are defined in the object itself, as are the associated constraints.

### **4.3.2 Support Objects**

The amount of information necessary to completely define a common machine tool component or the connection between two or more components is extensive. Descriptions of the types of information are found in the previous section. This information may be represented as fixed numbers, ranges of numbers, equations, lists, and pointers to other objects. In order to use this information in computational evaluation and optimization routines, standard representations must be developed. Standard representations ensure that the information pertaining to each computational object is included in the machine model in a consistent and complete manner, and decreases the modeling time by providing standard methods for the incorporation of information into each component model.

The standard objects for representation of information in PMDA are referred to as support objects, since they provide support for the inclusion of information in each component object. PMDA includes several types of support objects to provide for the incorporation of the various forms of information. These include objects representing the following:

- variables
- materials
- equations
- constraints
- ground

Each of these objects provides a standard representation of the associated information, allowing the incorporation and use of common engineering data in a standard for. The remainder of this section provides a description of each support object.

## Variables

The most common type of support object is the variable. Variable objects provide storage space for a single double-precision number and the information associated with that number. This information includes the name and symbol of the variable, the object that has direct control of the variable (referred to as the "parent"), and the system of units (metric/SI or English) for the variable. A linked-list of pointers to all of the equations the variable is associated with is also included, as is a flag to indicate if the value of the variable has been altered during the latest iteration of an optimization procedure. The above information is included to allow for easy identification of the variable; this is especially important in large constraint networks, or when several component representations are based on one template object. The latter case may result in several identical variable objects, and the additional information ensures the correct variable object will be identified.

## Equations

For use in PMDA, an equation is defined as one or more variables related through common mathematical operators. These include addition, multiplication, and exponent operators, as well as the logical operators "greater than", "less than", or "equal to". For computational simplicity, equations in PMDA are written in Reverse Polish Notation (RPN), or "postfix", as opposed to the standard, or "infix", representation. Several examples of postfix equations suitable for use in PMDA are shown (with their infix counterparts) in Table 4.1:

Infix Representation	Postfix Representation
$A + 4$	$A 4 +$
$(2.0 * W) / (T \leq 4) + C$	$W 2.0 * T 4 \leq^3 / C +$
$(\text{Pressure} * \text{Length}) / (\text{Area} * \text{Young's Modulus})$	$\text{Pressure Length} * \text{Area Young's Modulus} * /$
$\text{Force} = \text{mass} * \text{acceleration}$	$\text{mass acceleration} * \text{Force}^4 =$

Table 4.1: Infix and Postfix Equation Representations

<sup>3</sup> the expression  $(T \leq 4)$  evaluates to 1 if the value of T is less than or equal to 4, 0 otherwise

<sup>4</sup> this equation evaluates to 1 if Force is equal to mass \* acceleration, 0 otherwise

In order to store equations in PMDA, an equation object is developed. This object contains the equation stored in a binary tree (explained below), as well as associated information including the equation name, a pointer to the "parent" object that controls the equation object, and the numeric value of the equation determined the last time the equation was evaluated. A flag to indicate modification since the last optimization cycle is also included, as is a character string to store the equation in a user-friendly format.

One method available for the computational representation of equations is binary trees. A binary tree consists of nodes, each of which has two "children" and one "parent". Each node may house a variety of information depending on the application, but each node must have exactly two children and one parent. The exceptions are the "root" node, which is the beginning of the tree, and the "null" nodes which terminate a given branch of the tree. For example, the equation:

$$2 * 4 +$$

may be stored in the form of a binary tree as seen in Figure 4.4. This representation results in each variable node (shown as circles in Figure 4.4) having an operator "parent" and no children (except the null nodes, which are used for termination of the tree). The operator nodes each have two children and one parent; the children may either be another operator or a variable. If one child of an given operator is another operator, the entire tree structure below the child operator is one argument to the parent operator. This structure is required for equation representation in binary trees, since every operator implemented in PMDA requires two arguments, and an argument cannot take arguments (this would result if two variable nodes

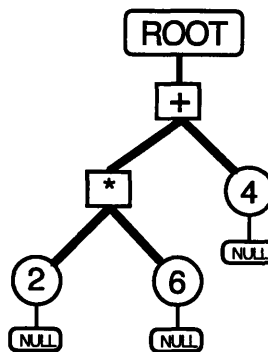


Figure 4.4: Equation Tree Representation

were joined in a parent-child manner). Evaluation of the equation proceeds from the bottom leftmost node, and proceeds upwards and to the right. Details on the manipulation and evaluation of binary tree equations is found in [Sedgewick, 1992].

Implementing equations in a binary tree format has several advantages. First, the tree structure is easily built as the equation is input. As each substring of the equation is incorporated into the equation object, the type of the substring (variable or operator) is determined and a node is created based on this type. The nodes are then pushed to or popped from a stack as necessary to create the tree structure (stack operations are discussed in [Sedgewick, 1992]). Second, the algorithms for the manipulation and evaluation of binary trees are well developed and tested. While the use of binary trees is not limited to mathematical equations, the algorithms may easily be applied to equation manipulation and evaluation. Binary trees also offer a flexible and stable representation for any mathematical equation, regardless of the size or complexity of the equation. The representation is also standard for any equations, and does not require modification for especially complex terms or variables.

## Constraints

Design constraints, discussed in detail in Chapter 3, provide a method for placing limitations on the variables associated with a design. For use in PMDA, constraints are represented as equality or inequality relationships between two equations. These equations (discussed above) may be simple, one-term constants, complex relations describing the behavior of a component, or combinations of less complex equations. The equality and inequality operators, called constraint operators, currently supported are: =, !=, >, <, ≥, ≤. Several examples of valid constraints for use in PMDA are:

$$\begin{aligned}
 [A] < [4]^\dagger \\
 [\alpha L_0 * \Delta T *] \geq [\Delta L] \\
 [m a * F =] = [Newton\_flag]
 \end{aligned}$$

Note that the third constraint statement is a check on the agreement of the truth (non-violation) of the left-hand side equation with the value of `Newton_flag`, not a process to set the value of the flag itself.

Constraints, like equations, are defined as an independent objects in PMDA. The structure of the constraint object includes pointers to the two equations and a string to store the constraint operator. The equations may either contain only variables associated with the object containing the constraint, or they may contain variables from other object. Evaluation of the constraint takes place in routines built into the constraint object itself.

---

† the "[ ]" notation indicates this is a PMDA equation object

## Materials

The material of each component or part of a component is specified within PMDA by a material object. The material properties included in the material object are:

- density,  $\rho$
- ultimate strength,  $\sigma_{ult}$
- yield strength,  $\sigma_y$
- shear strength,  $\tau_{xy}$
- coefficient of thermal expansion,  $\alpha$
- compressive strength,  $\sigma_{c\_ult}$
- modulus of elasticity,  $E$
- shear modulus,  $G$
- Poisson's ratio,  $\nu$

Like other support objects in PMDA, the material object is a standard representation. This allows for an infinite number of materials to be included in PMDA through the use of material catalogs. Each material in the catalog may be stored as a separate material object; the only limitation is that the catalog must describe the material in terms of the properties listed above.

## Ground

The Ground object is a block used to mark the origin of the complete machine model. Ground is essentially just a Block object with the origin fixed at (0,0,0,1). This allows the development of interfaces to the origin, and the evaluation of HTM loops to the origin using the routines contained in the Block object.

### 4.3.3 Component Objects

Chapter 4 discusses the use of standard component representations to model the components associated with a machine tool concept. The advantages of standard representations are: the ability to incorporate a number of similar components into the machine model in a consistent manner, the opportunity to select components from a catalog of commercially-available components, and the computational simplicity associated with the use of standard objects. For these reasons, the machine components comprising the machine model in PMDA are represented in as a set of standard components. The library of components currently implemented include:

- rotary bearing
- circular shaft
- circular workpiece
- three-tooth chuck
- single-edge cutting tool

In addition to these specific objects, an object containing information common to all of the machine elements is also developed. This object, known as a "block", contains information on

HTMs, coordinate systems, and associated support information for the inclusion of the component in the computational computer model. Each of the above standard representations, including the "block" object, is explained below. The details of the Block Object and the Rotary Bearing Object are discussed in Chapter 6 to provide an example of the representation of components in PMDA; the detailed descriptions of all other objects is found in the appendices.

## **Block Object**

The Block object is the most general object for the representation of machine component information. As seen in Figure 4.2, the block class is a superclass to all component objects. This allows all data common to all machine elements to be stored in a single class, and avoids repetition or exclusion of required information. The information stored in the block class includes:

- the origin and normalized axes of the component coordinate system
- the HTM relating this block to the previous block in the structural loop
- the maximum geometry (bounding box) of the component
- miscellaneous information, including id number, name, and system of units

Details of the implementation of the block class may be found in Chapter 6.

For the purpose of HTM modeling of the machine, the block class provides all of the necessary information for the creation of the HTM model. As the structural loop is defined, the required information for the creation of the HTM for a given block is passed to the block by the main program (see the discussion of the assembly class below). The block object then calculates the HTM to transform coordinates local to the block to the previous block in the structural loop. This process is discussed in detail in Chapter 5.

## **Rotary Bearing Object**

One of the primary component found in precision machine tools is the rotary bearing. Rotary bearings provide for unlimited rotational motion about the axis of the bearing, with either the inner or outer ring fixed. The basic layout of a rotary bearing is shown in Figure 4.5. The outer and inner rings provide a surface on which the rolling elements ride, with ball spacing preserved by the retainer (commonly called the "cage"). Rolling elements may be either ball bearings, cylindrical or spherical rollers, or "needles" (essentially high aspect-ratio cylindrical rollers). A number of variations on the bearing shown in the figure are possible,

including the modification of the raceways to provide for the support of both radial and axial forces. Details on the rotary bearing object are found in the case study in Chapter 6.

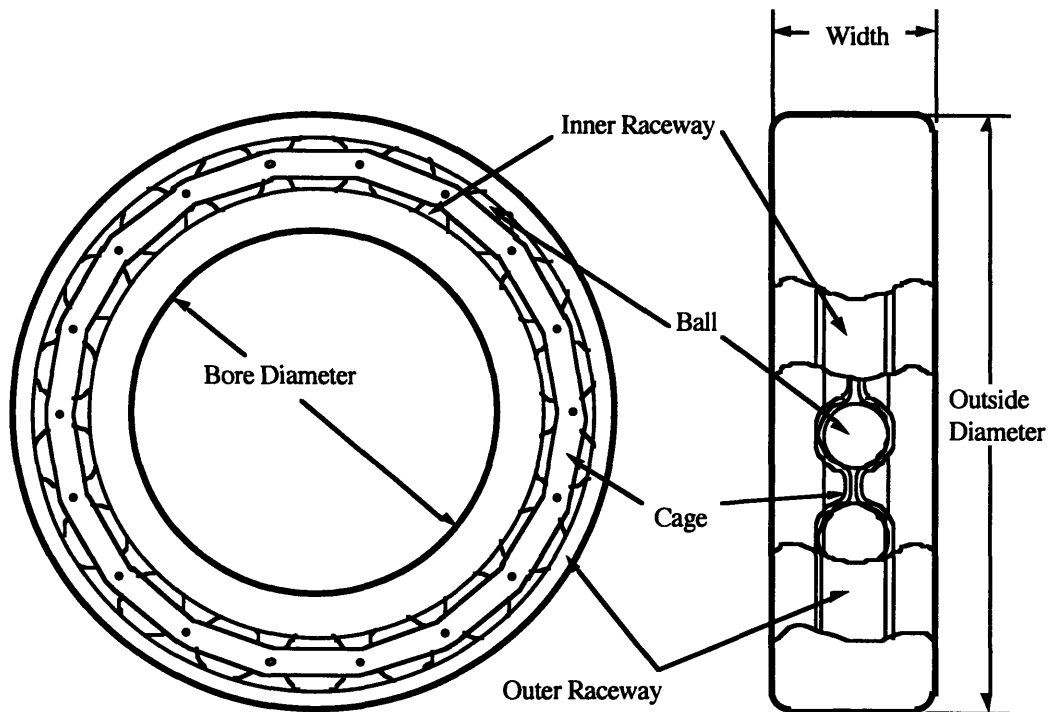


Figure 4.5: Standard Rotational Bearing

### Shaft Object

The Shaft object provides a model of a simple circular shaft. An example is shown in Figure 4.6. The information associated with the shaft object is:

- geometry (diameter, length)
- tolerance (diameter/length variations, eccentricity, etc.)
- material
- operating conditions (rpm, temperature)
- error motions (manufacturing defects, loading/temperature deflections)

The shaft object is developed in a similar manner as the rotary bearing object, with a template storing the pointers necessary for a complete representation, and the manufacturer-specific object supplying the necessary values, equations, and constraints.

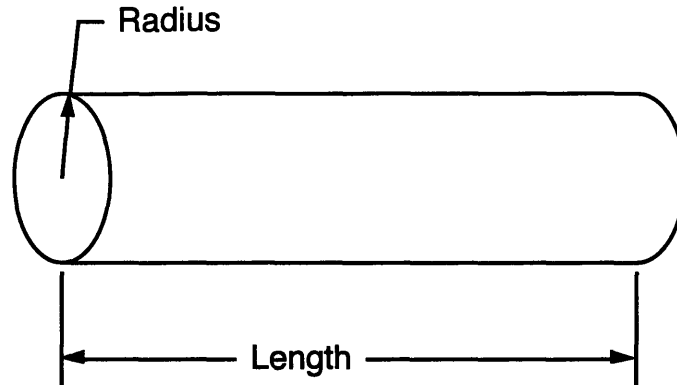


Figure 4.6: Standard Single-Diameter Circular Shaft

### Chuck Object

For rotary motion machining processes, the workpiece must be secured to the spindle shaft in a manner that allows the application of cutting forces to the workpiece. This is most commonly accomplished through the use of a rotary motion chuck. The chuck is essentially a multi-jawed clamp that attaches to the spindle shaft and hold the workpiece in the jaws. Figure 4.7 shows a typical 3-jaw chuck; other configurations with four or more sets of jaws are possible.

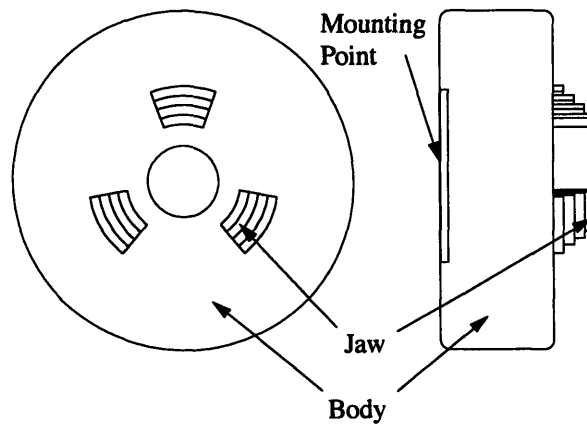


Figure 4.7: General Rotary Motion Chuck

## Cutting Tool Object

The cutting tool is the actual component that removes material from the workpiece. In order to ensure long life of the cutting edge, most tools are made of extremely hard material (for example, carbide steel or diamond-impregnated steel). For this reason, the cutting tool object in PMDA is assumed to be rigid and not subject to deflections due to cutting processes. This approximation is reasonable for light cutting forces. For simplicity, the only tool currently implemented in PMDA is a single-point cutting tool as shown in Figure 4.8. Because of the above assumptions, the cutting tool object may also be used to model a measurement probe or similar device.

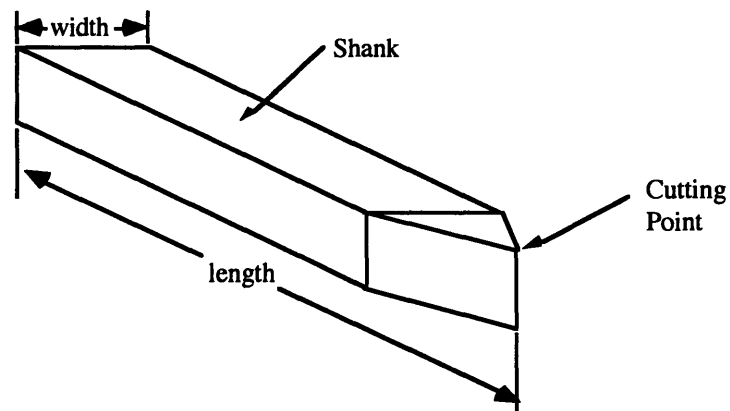


Figure 4.8: Simple Single-Point Cutting Tool

## Workpiece Object

The workpiece object in PMDA is modeled as a single-diameter circular cylinder, similar to the shaft class described above. The only use of the workpiece object is to provide a description of the location of tool-workpiece interaction. Since this point is generally defined by the user, deflections of the workpiece are neglected in PMDA. The exception is the application of cutting forces and the related deflection of the support structure for the workpiece. This error motion is included to allow for the prediction of deflections related to the cutting process.

#### 4.3.4 Interface Objects

As stated in the previous section, two basic types of information are required for the development of an error budget for a given machine tool: component-specific information and component-connectivity information. In PMDA, connectivity information is contained in interface objects. An interface object is a computational representation of the physical interaction between two or more compatible mechanical components. For example, a circular workpiece is connected to a rotary-motion chuck through a clamping process; this process, and the resulting orientation of the workpiece with respect to the chuck must be completely described in PMDA to provide the information necessary for construction, evaluation, and optimization of the associated machine tool concept.

Before the interface classes are developed, the necessary connections between PMDA component objects must be determined. Currently, the following object connections are supported by PMDA:

- Shaft - Chuck
- Tool - Ground
- Rotary Bearing - Ground
- Workpiece - Chuck
- Rotary Bearing - Shaft

The interface connections to ground represent a connection of the object to the global origin; all other components are connected with respect to another component object. The above classes represent all possible connections between PMDA object; no other interfaces are currently allowed.

The interface provides several types of information. First, the complete collection of interfaces in the machine tool model define the structural loop through each machine component from tool to workpiece. Second, interfaces provide for the sharing of common information. For example, a rotary bearing attached to a shaft must have the same rotational velocity as the shaft. The interface between the bearing and shaft is responsible for this sharing of information. Third, the interface class provides for equations, constraints, and error motions associated with the connection of components. An example is the interface between a workpiece and a chuck. Mounting errors between the workpiece and the chuck will result in workpiece error motions; these error motions are determined by the interface, since they are not a result of either the workpiece or the chuck but the interface between them.

Unfortunately, unlike the Block class for component objects, the interface classes are almost completely unique in each instance and require a separate and complete representation for each of the above classes. This is a result of the unique features of each interface class and the requirement that interface objects contain all information related to the connection of

components. This sections provides an overview of the function of each interface object in PMDA.

### **Shaft - Chuck Interface**

The Shaft-Chuck interface provides for the connection of a Shaft object to a Chuck object through a simple rigid-body connection. Each interface object may be related to only one Shaft Object and one Chuck Object. The joint is assumed to be rigid, with no mounting errors between the shaft and chuck. As a result, the only information contained in the Shaft-Chuck interface is rotational velocity and pointers to both the Shaft object and the Chuck Object.

### **Workpiece - Chuck Interface**

As described above, a rotary-motion chuck secures a workpiece to the spindle shaft through a clamping process, allowing the workpiece to be machined with various cutting tools. The ideal clamping process results in a rigid-body connection between the workpiece and chuck; while this connection is not perfectly rigid, the clamping force necessary results in a connection that is much more rigid than other joints in the machine. For this reason, the Workpiece-Chuck Interface in PMDA is assumed to be perfectly rigid.

This does not imply that the connection is ideal, however. Error motions due to the clamping process are still possible, and should be included in the interface object to accurately predict the error motions of the entire machine tool. The current version of PMDA does not include error motions in the interface classes.

### **Rotary Bearing - Shaft Interface**

The most complex interface included in PMDA controls the connection between a shaft and the rotary motion bearings that support it. Because a shaft is often supported by more than one bearing, this interface must be able to include multiple bearing and determine which bearings are "fixed" and which "float" (allowing for thermal expansion of the shaft). For simplicity, the interface between the shaft and associated bearings is assumed to be perfectly rigid, with mounting errors specified by the designer and included in the Shaft HTM.

## **Tool-Ground Interface**

Because of the nature of the error budgeting process, where the errors between the toolpoint and workpiece are determined, the Tool Object in PMDA is assumed to be attached to the reference coordinate frame. This greatly simplifies one of the two loops (tool-ground and workpiece-ground) required for error budget evaluation. The Tool-Ground Interface is a perfectly rigid connection between the tool object and ground block; no error motions are generated by the interface.

## **Rotary Bearing-Ground Interface**

The second structural loop is currently defined in PMDA with respect to a rotary bearing object. This allows the fixed bearing of a shaft-bearing combination to be referenced to the machine tool origin. Like the Tool-Ground Interface, the Rotary Bearing-Ground Interface is perfectly rigid and does not allow error motions between the two components.

## **4.4 Chapter Summary**

This chapter presents the object hierarchy used in PMDA. The advantages of an object-oriented approach to machine modeling, evaluation, and optimization is discussed, followed by a discussion of the requirements for completely describing both mechanical components and the connections between these components. The organization of the mechanical objects included in PMDA is presented, and specific information concerning object representations for rotary bearings, circular shafts, chucks, workpieces, and cutting tools is described. Interface objects are also discussed. Additional details of these objects is found in the appendices.

## **5.0 PMDA - Functionality**

### **5.1 Introduction**

Chapter 4 introduced the methods for representing mechanical components and the connections between these components. Once the component objects are created and the connections are defined, the kinematic model of the machine may be created, error motions and constraints may be evaluated, and the machine concept may be optimized. These processes are divided in PMDA into four main areas of functionality:

- **Kinematic Machine Modeling** - the creation of the model describing the machine tool concept
- **Error Motion Modeling and Evaluation** - the process for simulating individual component error motions, and determining both individual total machine error and the sensitivity to individual error motions
- **Constraint Evaluation** - creation and evaluation of the constraint network for each component and the entire machine tool
- **Concept Optimization** - optimization of the concept based on the constraint network and error evaluation results

Operation of PMDA takes place through a graphical-user interface, with a three-dimensional display of the machine model. Output files containing the error motions of various parts, as well as the total tool-workpiece interaction error and constraint violations, are also created. The functionality of the user-interface is also described in this chapter.

### **5.2 PMDA Overview**

The process for modeling a machine tool in PMDA is shown in Figure 5.1. The process begins with the reading of a set of input files (see Chapter 6); these include both user input files (modified by the user) and PMDA data files. Once the files are read, PMDA constructs the kinematic model of the machine (based on the geometry specified in the user input files), creates the constraint network, and creates the necessary error motion models. The total tool-workpiece interaction error is then calculated and the constraint network is evaluated. The error motions and constraint violations are written to a set of output files for off-line analysis.

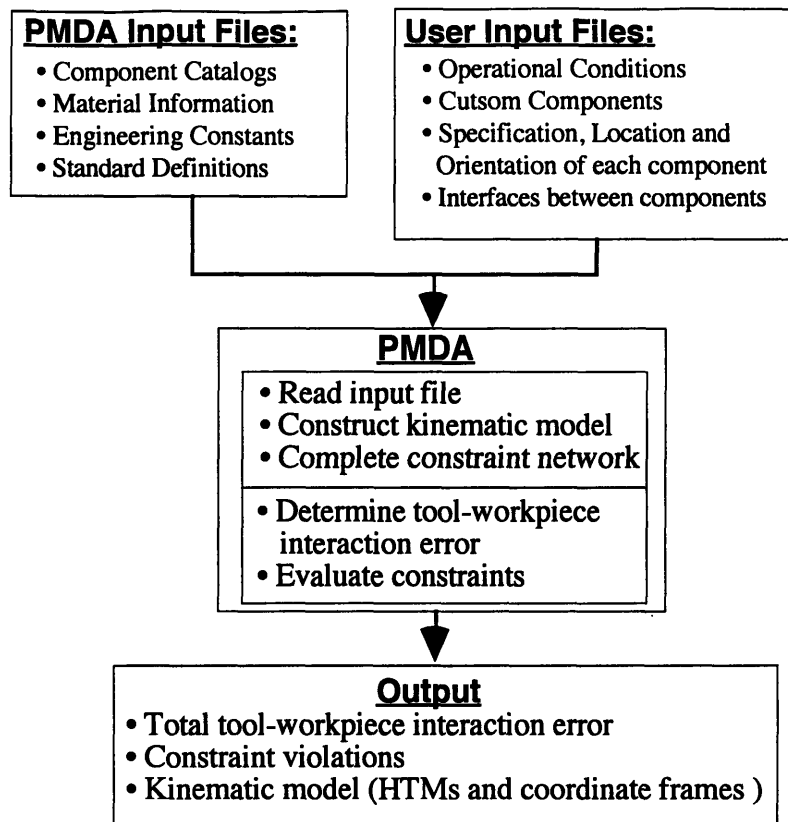


Figure 5.1: PMDA Modeling Process

### 5.3 Kinematic Machine Modeling

The kinematic machine model provides the information necessary to completely describe the position and connectivity of each component included in the machine tool assembly. The method of kinematic modeling used in PMDA is based on the use of Homogeneous Transformation Matrices and is presented in Chapter 2. The kinematic model serves two purposes in PMDA. First, the process of error evaluation requires a definition of the relative locations of each component in the machine tool. This is accomplished through the use of HTMs to determine the transformation between two connected components. The second purpose of the kinematic model is to provide verification of the kinematic feasibility of the concept. Verification is accomplished through the interface classes, since a non-feasible connection between components will not be allowed by the interface object.

### **5.3.1 Creation of the Kinematic Model**

The process for developing a kinematic model of a given machine concept is described in general terms in the example found in Section 2.6. For PMDA, this process is performed in the four steps outlined below. First, each component object in the machine must be defined. This does not require the specification of other numeric values associated with the objects; only the existence of the component and the location and orientation of the component is required. The second step involves the specification of the interfaces between the various components. As each interface is specified, the third step is completed automatically. This third step is the calculation of the HTM by the block class; this process is discussed in the following subsection. Once all of the interfaces and HTMs are set for all of the components in the machine concept, the final step of defining the structural loops may be completed. This step results in the specification of the component "trails" between the workpiece and ground and the cutting tool and ground. Details concerning the representation of structural loops in PMDA are found in Appendix C.

While the above process is acceptable for a static machine, time-varying effect, such as bearings and motors, will exhibit changes in position and orientation as a function of time. To account for this, the kinematic modeling routines and component representations in PMDA include the routines necessary to determine the relative position of moving elements with respect to stationary elements. This allows for the inclusion and modeling of moving parts and the errors associated with movement, such as radial bearing runout. These routines do, however, require that the kinematic model be recomputed at each time step. The HTMs between components will change, but the definition of structural loop and the interfaces between components are not altered.

### **5.3.2 HTM Calculation**

The specific process for calculating the HTM between two arbitrary coordinate systems is discussed in Section 2.4. The requirements for HTM calculation include the location of the origin and orientation of the axes for two component objects and the understanding of which component contains the "original" axis system and which component contains the "final" axis system. In PMDA, the Block object contains all of the routines necessary for HTM calculation, as well as the location and orientation of the component axis system.

The process for calculating the HTM between two Block objects is shown in Figure 5.2. The information as to which component is the original and which is the final is defined by the structural loop, with the component "upstream" in the loop being designated as the original

component. An example of this process is found in Appendix C. The HTMs for all components in a structural loop are generated automatically when the structural loop is defined, and recalculated as necessary when the location or orientation of a component is altered.

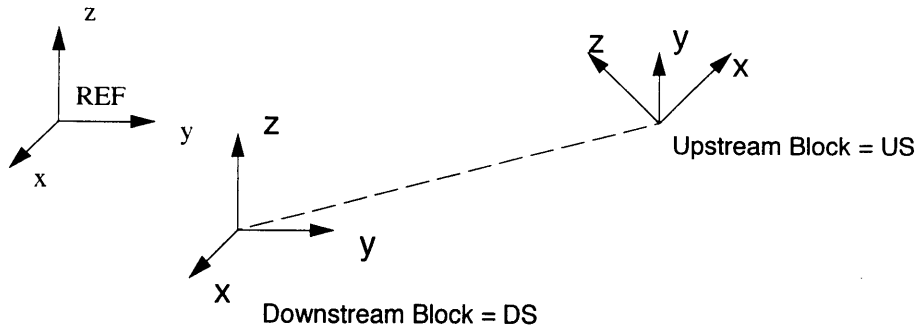
## **5.4 Error Motion Modeling and Evaluation**

The general process of evaluation of machine tool errors is presented in detail in Chapter 2. The specific process used in PMDA is similar to the example in Section 2.6; as the tool-workpiece interaction point is transformed to each successive component, the errors associated with that component are vectorially added to the point. The resulting point is compared to the ideal tool-workpiece interaction point (defined by the total HTM) to determine total tool-workpiece interaction error. The modeling of individual component errors is performed in the associated component object, while the routines for error evaluation are stored in the assembly object.

Two types of error analysis are currently implemented in PMDA. The first determines the total tool-workpiece interaction error based on the error motions of the components in the assembly. This type of analysis is similar to that found in Section 2.6. The second form of analysis is sensitivity analysis, which determines the sensitivity of the tool-workpiece interaction error to the error motion of each individual component. By determining which components have the greatest impact on the tool-workpiece interaction error, design efforts may be concentrated on the necessary machine components.

### **5.4.1 Error Motion Representation**

As stated above, the representations for the error motions associated with each component in PMDA are stored in the appropriate component object. Error motions representing mounting errors are not currently included in PMDA. Figure 5.3 provides a list of the error motions currently included in PMDA. Note that PMDA does not predict the actual error motions of a given component; this requires complex physical simulation of the manufacturing process and mapping of existing components, and is beyond the scope of this dissertation. Instead, the error motion models used in PMDA provide a representation of the expected performance of a "generic" component, and are represented by simple mathematical models that closely mimic, but do not accurately simulate, the error motion of the component.



**Step 1 :** Create Matrices Based on X, Y, and Z Components of Block Axes

$$DS\_Mat = \begin{bmatrix} Xd-x & Yd-x & Zd-x & 0 \\ Xd-y & Yd-y & Zd-y & 0 \\ Xd-z & Yd-z & Zd-z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad US\_Mat = \begin{bmatrix} Xu-x & Yu-x & Zu-x & 0 \\ Xu-y & Yu-y & Zu-y & 0 \\ Xu-z & Yu-z & Zu-z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**Step 2:** Transpose DS\_matrix

$$DS\_Mat = \begin{bmatrix} Xd-x & Xd-y & Xd-z & 0 \\ Yd-x & Yd-y & Yd-z & 0 \\ Zd-x & Zd-y & Zd-z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**Step 3:** Multiply Matrices

$$HTM = \begin{bmatrix} Xd-x & Xd-y & Xd-z & 0 \\ Yd-x & Yd-y & Yd-z & 0 \\ Zd-x & Zd-y & Zd-z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} Xu-x & Yu-x & Zu-x & 0 \\ Xu-y & Yu-y & Zu-y & 0 \\ Xu-z & Yu-z & Zu-z & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**Step 4:** Add Translations

$$HTM = HTM + \begin{bmatrix} 0 & 0 & 0 & Tx \\ 0 & 0 & 0 & Ty \\ 0 & 0 & 0 & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

X-x = x-component of block's X axis in Reference Coordinate System  
(this is 1 for the ground block and all blocks with an  
x-axis in the same direction)

**Figure 5.2: HTM Calculation Process**

**Rotary Bearing:**

Inner Ring Radial Runout  
Outer Ring Radial Runout  
Axial Runout

**Chuck:**

Workpiece Center Offset  
Jaw Center Offset

**Tool:**

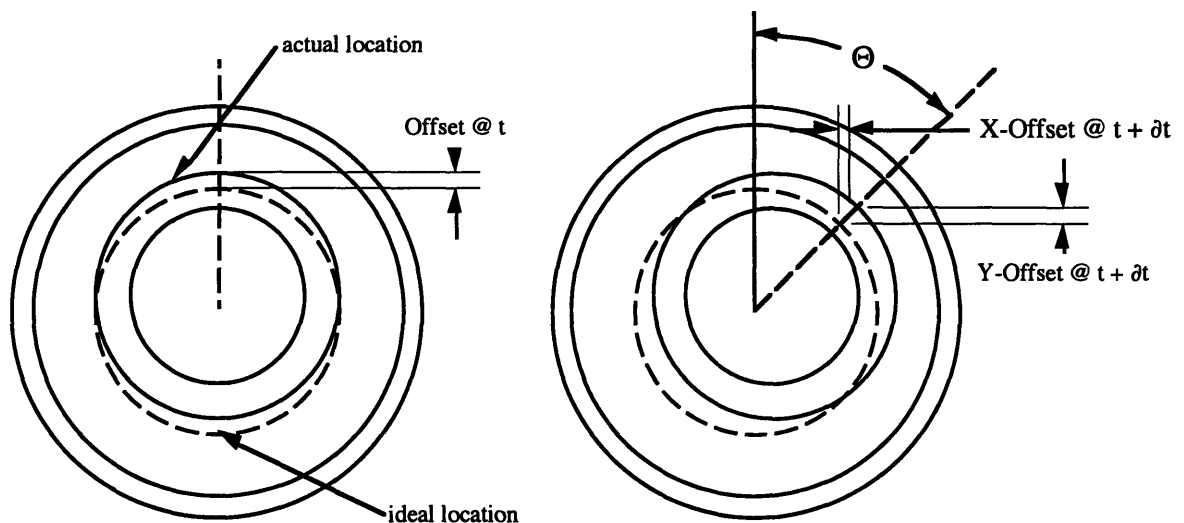
X-Straightness Error  
Y-Straightness Error  
Z-Straightness Error

**Shaft:**

Eccentricity  
Surface Finish  
Center Offset  
Diameter Variation  
Thermal Deflection

Figure 5.3: Error Motions in PMDA

An example of error motion modeling dealing with radial runout in a rotary bearing is found in Figure 5.4. The inner ring radial runout of a rotary bearing is essentially a variation in the thickness of the inner raceway of the bearing. This thickness variation results in an "out-



**Model:**

- 1) Offset is a constant determined by the inner raceway radial runout tolerance
- 2) Multiply Offset by  $\cos(\theta)$  for X-offset of shaft
- 3) Multiply Offset by  $\sin(\theta)$  for Y-offset of shaft

Figure 5.4: Inner Ring Radial Runout Modeling

of-round" error motion, with the bearing (and associated shaft) being displaced from the ideal center location. The model selected for use in PMDA for this error motion is shown in Figure 5.4. As the bearing rotates, this thickness variation will result in the ideal location of the center of the shaft-bearing sub-assembly moving in the manner shown. For each value of  $\Theta$ , the

resulting error motion will be calculated and included in the error budget for the machine. A similar representation for each error motion is included in PMDA, as described in the appendices.

#### **5.4.2 Determination of Total Tool-Workpiece Interaction Error**

The first type of error analysis included in PMDA is the determination of tool-workpiece interaction error. Typically, machine tools modify workplaces through the application of cutting forces via a cutting tool. The tool contacts the workpiece at a given location for the removal of material, and the difference between the ideal location of the tool and the actual location of the tool (both defined with respect to the workpiece) is the total tool-workpiece interaction error of the machine tool. The sources of this error is discussed in detail in Chapter 2.

PMDA calculates the total machine tool error in the following way. The first step is to define the ideal toolpoint location with respect to the workpiece. This is accomplished through the kinematic modeling process defined above, and the selection of an arbitrary point where the tool will contact the workpiece. This point (defined with respect to the reference, or "global", coordinate frame) represents the position of the tool with respect to the workpiece for a machine tool with no error motions. Once this point is determined, the error motions of each component in the structural loops (tool-ground and workpiece-ground) may be included into the analysis by recomputing the location of the toolpoint and workpiece-point locations through the respective structural loops. The position vector between these new locations, as modified by the component error motions, represent the actual position of the tool with respect to the workpiece. The components of this vector are the total tool-workpiece interaction errors.

Since the total tool-workpiece interaction error is a result of the error motions of each component in the machine tool concept, each error motion of each component must be accounted for. Inclusion of each component error motions is performed in the manner illustrated in the example in Section 2.6. At each local coordinate frame, the location of the point of interest is modified by the error motions of the given component, and then transformed to the next component through the appropriate HTM. This process continues until the reference coordinate frame is reached.

As stated in the previous subsection, determination of the effects of time-dependent components on the effect of machine tool errors require special treatments. In the case of component error motions, each time-varying error motion is computed with respect to a time scale, so that each increment of time provides an updated error motion. An example is the rotary bearing object described in the previous chapter. The effects of inner and outer ring

runout will result in an eccentricity error motion, illustrated previously in Figure 5.4. As the angle  $\Theta$  changes, the x- and y-direction error motions change. This effect is modeled by specifying the radial runout error motions with respect to  $\Theta$ . The remainder of the process for calculating tool-workpiece interaction error is unchanged.

Several alternate methods exist for the calculation of the effects of error motions. One method involves the modification of the HTM between components to represent the error motions; the resulting HTMs contain all error motions associated with a particular component, and transformation of the tool-workpiece interaction point to the reference frame will automatically include the effects of error motions. This method is not selected for use in PMDA since each change in error motions will require a recomputation of all HTMs in the kinematic model. While this process is not computationally prohibitive, the difficulty of verification of the kinematic model and performing error sensitivity analysis is increased. A second method for error prediction uses several HTMs for each coordinate system: one to define ideal positions, one to define the "pose" of the machine, and one to represent the effects of error motions. This process is similar to the approach defined above, with the addition of the "pose" HTM; this HTM is included to include the position of the component with respect to a "home" location. This method provides identical results to the methods included in PMDA, and was not selected based on the preference of the author.

## **Sensitivity Analysis**

Sensitivity analysis refers to the process by which the effect of each individual error motion on the total tool-workpiece interaction error is determined. This type of analysis provides the designer with information concerning the effect of each component and each error motion on the overall performance of the machine tool. In this way, a designer can determine which components in the machine tool are most critical to the performance of the concept, which parts of the machine tool will benefit from further review, and which parts do not require redesign. Currently, this form of analysis is performed in PMDA in the same manner as the determination of the total tool-workpiece interaction error explained above, with only the error motions from one component in the kinematic model "active" during an analysis cycle. This is accomplished through the use of flag variables in the component objects which determine if an error motion representation is or is not included in the error analysis.

## 5.5 Constraint Evaluation

Constraints are described in detail in Chapter 3, and their inclusion in PMDA is accomplished through constraint objects presented in Chapter 4. Currently, all constraints are included in the component or interface object templates; for this reason, user-defined constraints are not supported. As stated in Chapter 4, each constraint object contains pointers to two equations, which may be made up of variables local to the object containing the constraint or variables from other objects. For example, the constraint limiting the rotational speed of a rotary bearing is made up of two equations; one representing the operational speed of the bearing, and one representing the maximum speed of the shaft. While the maximum speed is a variable contained within the rotary bearing object, the operational speed is a variable contained in the bearing-shaft interface. Because of this, the information concerning the location of each variable in an equation (and constraint) is contained in the associated component or interface object.

Evaluation of the constraint network occurs in three steps. First, the constraints are created; currently, this task is performed when the object housing the constraint is created. As each constraint is created, it is added to a linked-list contained in the associated object. In this way, a list of all constraints in an object is created. After all of the constraints and the constraint list are created, the equation pointers (and associated variable pointers) are set. This avoids the problems associated with copying information and ensures the variables and equations included in the constraint are correct and up-to-date. Once all of the necessary pointers are set, the constraint network is complete.

Before the network may be evaluated, however, the values associated with each variable in each constraint must be set. These values may be set by several methods, including explicit specification of the value (rotary bearing bore diameter), calculation by equation (rotary bearing L10 lifetime), or look-up table (material properties). The specification of the maximum and minimum values in a constraint (i.e. "limits"), is dealt with either through user input (max. rpm or max cost), catalog or object input (limit speeds or loads), or through specification in the PMDA source code (min. cost = 0).

Evaluation of the constraint network takes place automatically after the tool-workpiece interaction error is calculated for a machine concept. Violations are reported through an output file, as seen in Appendix D. Currently, PMDA does not incorporate a method for altering variables based on constraint violation; for this reason, the values associated with a constraint violation must be altered manually, and the simulation must be re-run.

## 5.6 Concept Optimization

Once the constraint network is constructed, it may be applied to the optimization of the associated concept. This process involves the determination of the optimum value for each design variable given the constraints on the design. Constraint-based optimization methods are discussed in detail in [Thornton, 1993]. Currently, PMDA does not include any optimization routines.

## 5.7 User Interface

User input to PMDA is accomplished through a graphical user interface. A sample of this interface is shown in Figure 5.5. Components are added to the assembly object through the use of pull-down menus, and as each object is created an information window displaying

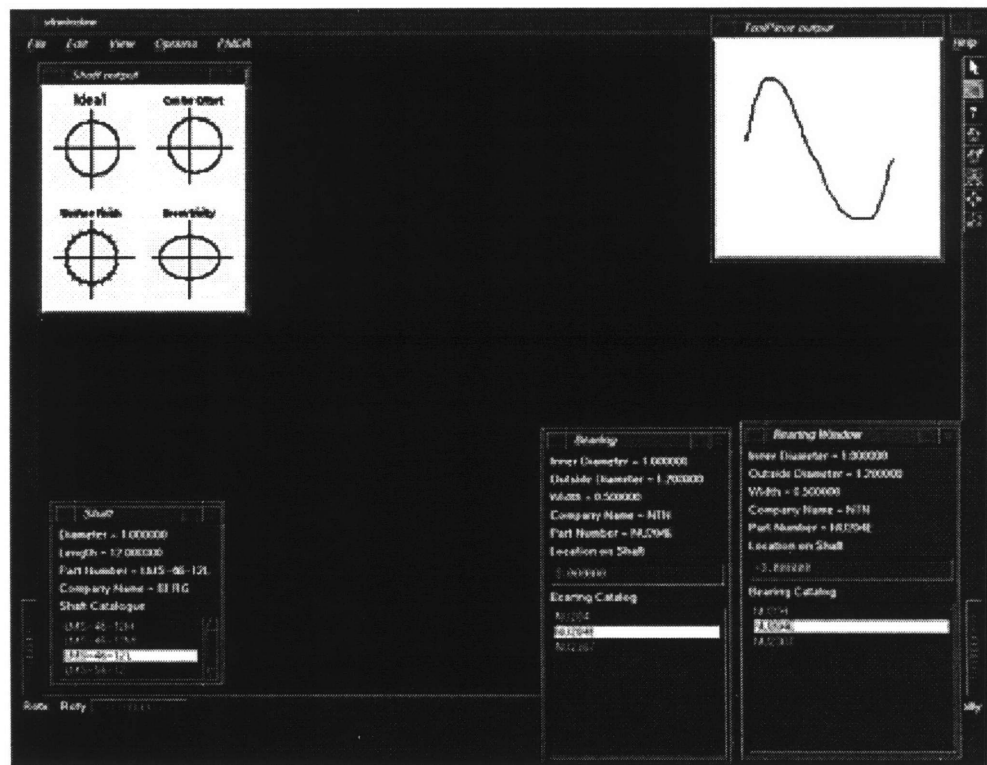


Figure 5.5: PMDA User Interface

the basic geometry, location, and available catalog parts is opened. Interface objects are also created through the use of pull-down menus, and the information necessary for the completion of the interface between components is entered in a pop-up window.

## **5.8 Chapter Summary**

The functionality of PMDA is divided into three distinct areas. Kinematic machine modeling is accomplished through the use of interface objects and HTMs. The interface objects define the connectivity between the components, and the HTMs define the geometric relationship between the axes systems of the components. The error motion modeling and evaluation routines allow for the determination of the total tool-workpiece interaction error based on the errors of each component in the machine. The sensitivity of the tool-workpiece error to the error motions of each component may also be determined. The constraint evaluation routines allow for the development of the constraint network for the concept, and for the evaluation of these constraints based on limit set by the user. This network also allows for the application of optimization routines to the machine tool concept. These three areas of functionality allow for the direct comparison of the performance of a series of machine tool concepts.

## **6.0 PMDA - Example Cases**

### **6.1 Introduction**

This chapter presents two examples of the design and functionality of the Precision Machine Design Assistant (PMDA). The first example illustrates the structure of a template object and one associated manufacturer-specific object. This serves to explain the development of the object hierarchy in PMDA, the methods for incorporation of component information into the object representations, and the inclusion of the object in the overall operation of PMDA. The second example presents the modeling and evaluation of a simple machine tool sub-assembly. A simplified rotational motion machine spindle is modeled to determine the tool-workpiece interaction error for a given set of error motion models. The constraint network for this model is also constructed and evaluated to determine the feasibility and performance of the individual components in the assembly. This example illustrates the use of PMDA in the conceptual evaluation process, and provides insight into how the PMDA system can provide a direct comparison between a set of machine tool concepts.

### **6.2 The Rotary Bearing Object**

This section presents the development of the object representations associated with rotational motion bearings. The basic design of a rotary motion bearing is discussed, as is the standard engineering information necessary to incorporate a typical rotary bearing into a machine tool concept. As with all PMDA objects, the rotary bearing objects are divided into one template object and multiple manufacturer-specific objects. This hierarchy is shown in Figure 6.1. One of these manufacturer objects is discussed to illustrate the structure of PMDA component objects, and provide an example of how engineering information is incorporated into PMDA objects.

#### **6.2.1 The Rotational Motion Bearing**

Rotary motion bearings are described in Chapter 4, and are discussed in detail in a number of mechanical design texts including [Slocum, 1992] and [Shigley, 1989]. Rotary motion bearings have one or more rows of rolling elements contained within two concentric rings. The rolling elements move along the surfaces of these rings, called races, and allow for

the support of pure radial, pure axial, or a combination of axial and radial forces depending on the design of the bearing. The result is a mechanical component capable of

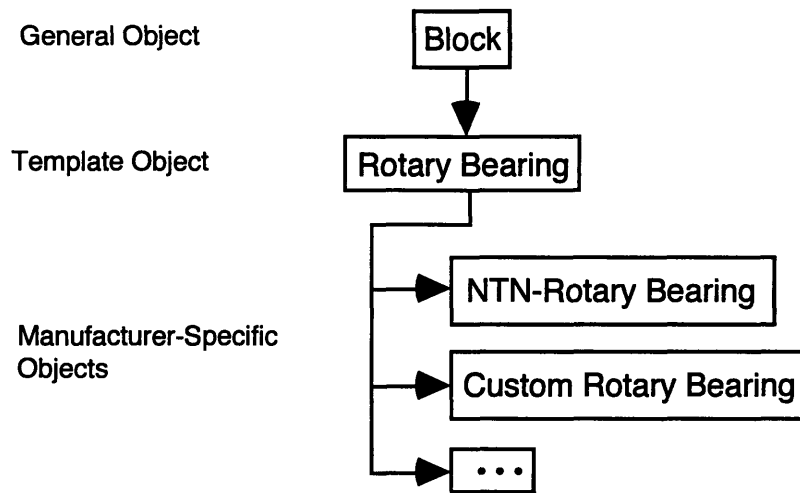


FIGURE 6.1: Rotary Bearing Object Hierarchy

providing unlimited radial motion about the longitudinal axis of the component. A typical rotary motion bearing is shown in Figure 4.5.

The standard engineering information used for the sizing and incorporation of rotary motion bearings is summarized in Figure 6.2. This list represents the complete set of information required by the designer before the performance and operational characteristics of a given rotary bearing can be determined. For conceptual design and development, only some of this information needs to be specified. For example, the  $L_{10}$  lifetime is a standard quantity that predicts the number of rotations at which 10 percent of a given type of bearing will fail for a specific loading condition.  $L_{10}$  life is always specified in concept development, since it provides a key relationship between the critical bearing design parameters of load capacity and lifetime. Information on bearing mounting procedures, however, is generally excluded from the preliminary design phase, since it is often assumed that the selected bearing can be mounted in the machine as required. While there is no rule for the division of information between preliminary and detail design phases, the topics listed with an asterisk in Figure 6.2 are typically required for the conceptual development phase; because of this, these topics are included in the rotary bearing objects used in PMDA.

### **Included in PMDA**

- Bearing Geometry
- Contact Angle
- Tolerances
- Limit Speed
- Limit Load
- Limit Lifetime
- Bearing Arrangement
- Required Operational Load
- Required Operational Speed
- Required Performance
- Operational Profile  
(continuous or intermittent)

### **Not Included in PMDA**

- Bearing Fit
- Bearing Mounting
- Shaft (Bore) Fit
- Bearing Friction
- Bearing Temperature Rise
- Lubrication Methods
- Cooling Systems
- Loading Variations
- Maintenance Requirements
- Installation Procedures
- Preload

**Figure 6.2: Required Rotational Motion  
Bearing Design Information**

## **6.2.2 The Rotary Bearing Template Object**

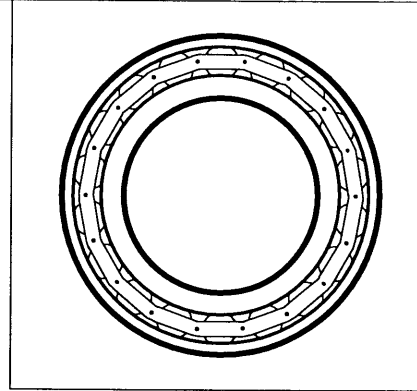
The rotary bearing template object in PMDA provides a generic representation of a rotational motion bearing. Currently, the types of rotary bearings included in PMDA are: deep groove, angular contact, self-aligning, cylindrical roller, and needle roller. These bearings may have a one (single-row) or two (double-row) rows of rolling elements. Tapered, spherical, and thrust bearings are not currently implemented in PMDA, and all rotary bearings are assumed to support only radial loading (axial loading must be  $\leq 10\%$  of the radial load). The characteristics included in the rotary bearing template object are:

- basic geometry (outside diameter, bore diameter)
- bearing geometry (contact angle, load center)
- axial and radial load factors
- limit speed and limit loadings
- operating conditions (temperature, rpm)
- tolerances (runout, dimensional deviations)
- life ratings
- error motion predictions
- constraints

Figure 6.3 shows the specific information associated with the above categories. This information is stored as a series of pointers to variable objects (described above), a set of

---

# Rotary Bearing Template Object



## Variables

### Geometry

Bore Diameter  
Outside Diameter  
Width  
Load\_Center  
Contact\_Angle  
Arrangement

### Operating Conditions

Axial\_Load (applied)  
Radial\_Load (applied)  
Operational\_Temperature  
RPM

### Operating Life ( $L_n$ )

$L_n$  (basic rated life - n% failure)  
 $L_n_h$  (basic rated life - hours)  
Life\_Factor ( $L_n$  correction)  
Speed\_Factor ( $L_n$  correction)  
 $a_1, a_2, a_3$  (generic  $L_n$  corrections)

### Miscellaneous

Mass  
ABEC\_Class  
ID #  
Maximum\_Allowable\_Temperature  
Limit\_Speed\_Oil  
Limit\_Speed\_Grease  
Gamma (roller vs ball factor)

### Load Calculations

- Axial\_Load\_Factor
- Radial\_Load\_Factor
- Static\_Axial\_Load\_Factor
- Static\_Radial\_Load\_Factor
- Basic\_Dynamic\_Rated\_Load
- Basic\_Static\_Rated\_Load
- Dynamic\_Equivalent\_Radial\_Load
- Dynamic\_Equivalent\_Axial\_Load
- Static\_Equivalent\_Radial\_Load
- Static\_Equivalent\_Axial\_Load
- Max\_Static\_Equivalent\_Load
- Max\_Dynamic\_Equivalent\_Load

### Tolerances

- $\Delta D_{M_P}$  (single mean plane bore diameter deviation)
- $V_{D_P}$  (single radial plane bore diameter deviation)
- $K_{I_A}$  (inner ring radial runout)
- $S_D$  (face runout with bore)
- $S_{I_A}$  (inner ring axial runout)
- $\Delta B_S$  (inner ring width deviation)
- $V_{B_S}$  (inner ring width variation)

## Interface

- pointer to a Bearing\_Shaft Interface object
- pointer to Ground Object

---

Figure 6.3: The Rotary Bearing Template Object

constraint objects, and pointers to two interface objects (see Chapter 4). The details of the template object are discussed below.

### **Variable Pointers**

The majority of the rotary bearing template object is made up of pointers to variable objects, as listed in Figure 6.3. Variable objects are described in detail in Chapter 4. Geometry, operational conditions, loadings and performance information, tolerances, and miscellaneous characteristics concerning the bearing can be described with a single numeric value, and are represented by variable objects contained within the complete PMDA assembly.

Note that the actual variable objects are not necessarily created or specified by the rotary bearing object. For example, the rotational speed variable (RPM) *pointer* in a given rotary bearing object will point to a variable *object* created and controlled by the assembly object. This minimized the number of duplicate objects, avoid difficulty with updating multiple objects, and allows for the automatic constraining of variables within multiple objects (i.e. the rotational speed of a shaft and a bearing must be equal if the variable object representing this speed is referred to by both the shaft and the bearing object).

### **Interfaces**

The interface objects available in PMDA are described in detail in Chapter 4. Two interfaces are associated with the rotary motion bearing object: the bearing-shaft interface and the bearing-ground interface. These interfaces are accessed through two pointers in the rotary bearing object, and provide the information necessary to create the HTMs, define the structural loops, and indicate the connectivity between the bearing and the associated components in the machine assembly. The actual interface objects are created and controlled by the assembly object, with the connection between the component objects and the interface object handled automatically within the assembly object.

### **Equations and Constraints**

The representation, implementation, and use of constraints in conceptual design is discussed in Chapter 3. Since many of the constraints associated with the performance and operation of rotary bearings are common to all types of rotary bearings, the rotary bearing template object contains all of the constraints associated with rotary bearings in PMDA. The constraints contained in the rotary bearing template object are listed in Figure 6.4. Note that the

variables and equations contained within these constraints may be contained in either the template object or in the manufacturer-specific object.

### **Physical**

- Bearing Mass
  - Mass  $\geq$  Max. Mass
  - Mass  $\leq$  Min. Mass
- Outside Diameter
  - OD  $\leq$  Max. OD
  - OD  $\geq$  Min. OD
- Bore Diameter
  - BD  $\leq$  Max. BD
  - BD  $\geq$  Min. BD
- Bearing Width
  - BW  $\leq$  Max. BW
  - BW  $\geq$  Min. BW
- ABEC Class
  - ABEC  $\geq$  Min. ABEC
- Bearing Cost
  - Cost  $\leq$  Max. Cost

### **Operational**

- Bearing Loading
  - Dynamic Equiv. Radial Load  $\leq$  Max. Rated Radial Load
- Temperature
  - Operational Temp  $\leq$  Max. Operational Temperature
- Rotational Speed
  - Rotational Speed  $\leq$  Limit Speed Oil    OR  
Limit Speed Grease
- $L_n$  Lifetime
  - $L_n \geq$  Min.  $L_n$
  - $L_n \leq$  Max.  $L_n$
  - $L_{n_h} \geq$  Min.  $L_{n_h}$
  - $L_{n_h} \leq$  Max.  $L_{n_h}$

Figure 6.4: Rotary Bearing Constraints

## **Error Motions**

PMDA uses simple geometric representations of error motions to account for the variations involved in the operation and performance of a given mechanical component. The methods for modeling and including error motions are discussed in Chapters 2 and 5. Like the constraints described above, the modeling of error motions in PMDA occurs in the template object. The rotary bearing object currently contains error motion models for axial, outer ring, and inner ring runouts, and inner and outer race surface finish effects. Illustrations of these error motions are shown in Figure 6.5, while the models used to account for these errors are shown in Figure 6.6. The actual equations for these models are found in Appendix B. As stated in Chapter 5, these models are not meant to accurately match the

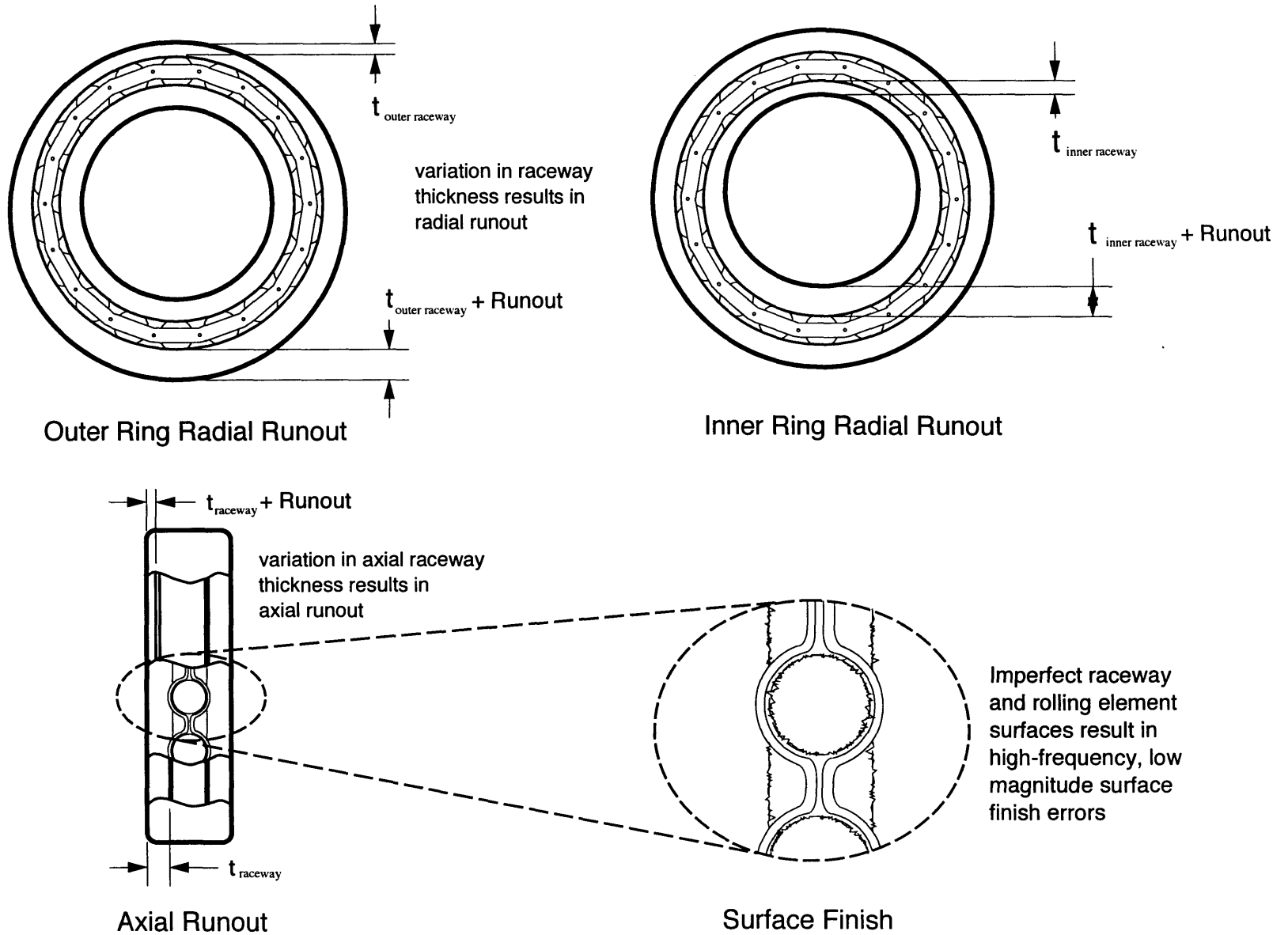
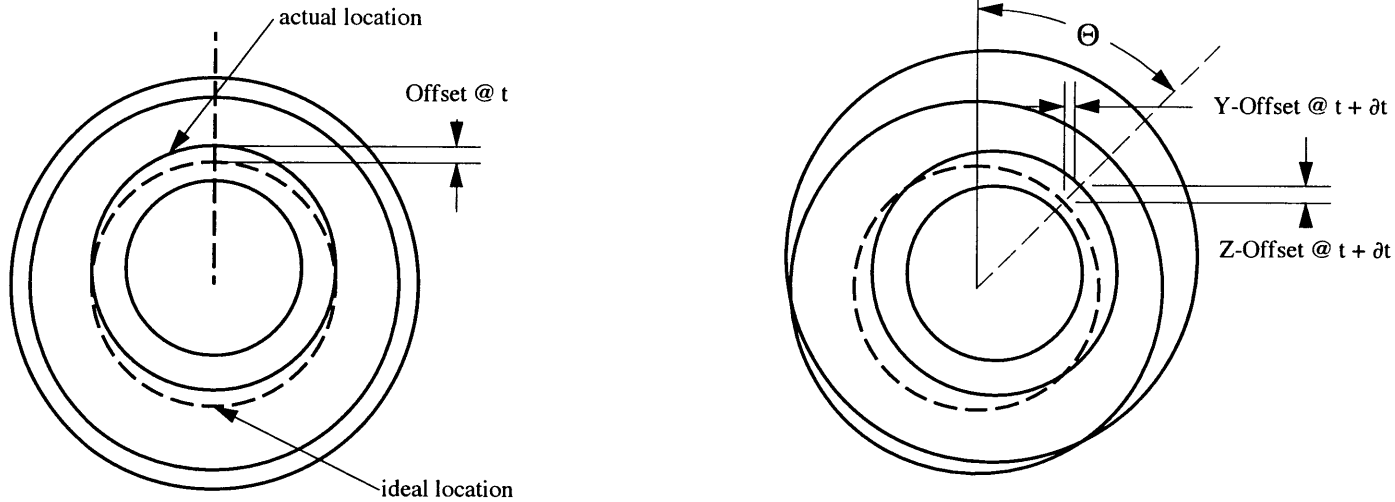
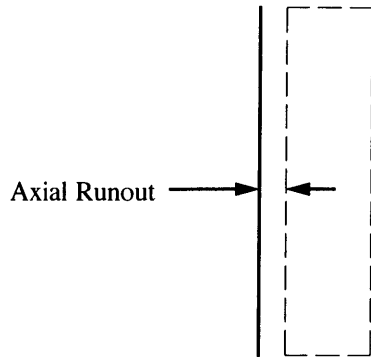


Figure 6.5: Rotary Bearing Error Motions



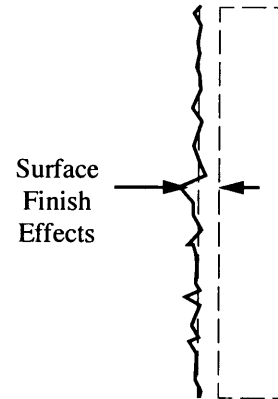
**Radial Runout Model (Inner and Outer Ring):**

- 1) Offset is a constant determined by the rotary bearing inputs
- 2) Multiply Offset by  $\cos(\theta)$  for Y-offset of shaft
- 3) Multiply Offset by  $\sin(\theta)$  for Z -offset of shaft



**Axial Runout Model:**

- 1) Error is a constant determined by the inner and outer raceway axial runout tolerance



**Surface Finish Model:**

- 1) Error is a a fraction of the Axial Runout Error multiplied by a randomly-generated number

Figure 6.6: Rotary Bearing Error Motion Modeling

performance of a selected bearing, but to provide a universal representation of common errors present in all rotary motion bearings.

### **6.2.3 The NTN Rotary Bearing Object**

As stated in the previous section, the manufacturer-specific rotary bearing objects in PMDA are subclasses of the rotary bearing template object. This allows for the variation in manufacturer data to be incorporated on the lowest possible level, with no modification to the template class. One of the manufacturer-specific subclasses included in PMDA represents bearings from the NTN Corporation. While the example presented in this section is specific to objects of the rotary bearing type, the approach described may be used for incorporating information into the template class for any machine tool component.

As stated in Chapter 4, the rotary bearing template class defines the variables required to describe a general rotary bearing object. The majority of the values of these variables, however, are determined by the manufacturer-specific object. For the NTN rotary bearing, the sources of these values are divided into five categories, as listed in Figure 6.7. The catalog-specific values are those that are explicitly set as numbers in the catalog (i.e. geometry of limit speeds). Equation-specified values are those that are set according to a common equation. These equations (usually presented in the engineering section of the bearing catalog) are common for all NTN rotary bearings, and may be "hard-coded" into the NTN rotary bearing object. The values determined by these equations, however, will change for each specific bearing, and must be included as equations and not as static values. User-specific values are those that depend on the overall assembly, and are set automatically by PMDA based on the operational characteristics of the machine. PMDA-specific values are used for computational bookkeeping, are transparent to the user.

The table-specified values are values that are dependent on look-up tables or graphs. Like the equations described above, these tables and graphs are common to all NTN rotary bearings. This does not imply, however, that the values are common across all types of NTN rotary bearings; the values differ, but all bearing objects from NTN have tabular or graphical information stored in a similar manner. Because of this, the tables and graphs may be represented in the NTN rotary bearing object, with variations accounted for in small look-up tables or type-specific parts of the object. While this approach results in some extraneous information for each bearing object, the unnecessary information does not result in a large increase in object size. By including this information in the NTN object, the need for a subclass of NTN bearings is eliminated.

# NTN Rotary Bearing Object

## Catalog-Specified Values

### Geometry

- Bore Diameter
- Outside Diameter
- Width
- Load\_Center
- Contact\_Angle

### Miscellaneous

- Mass
- ABEC\_Class
- Maximum\_Allowable\_Temperature
- Limit\_Speed\_Oil
- Limit\_Speed\_Grease

### Tolerances

- $\Delta D_{M_P}$
- $V_{D_P}$
- $K_{I_A}$
- $S_D$
- $S_{I_A}$
- $\Delta B_S$
- $V_{B_S}$

### Loading

- Basic\_Dynamic\_Rated\_Load
- Basic\_Static\_Rated\_Load

### Bearing Life (Ln)

- $a_1$
- $a_2$
- $a_3$

## Equation-Specified Values

### Loading

- Dynamic\_Equivalent\_Radial\_Load  
 $P_r = F_r * X + F_a * Y$
- Dynamic\_Equivalent\_Axial\_Load  
 $P_a = F_a + 1.2 * F_r$
- Static\_Equivalent\_Radial\_Load  
 $P_{or} = F_r * X_o + F_a * Y_o$
- Static\_Equivalent\_Axial\_Load  
 $P_{oa} = F_a + 2.7 * F_r$

### Bearing Life (Ln)

- $L_n$   
 $L_n = a_1 * a_2 * a_3 * (C / P)^Y$
- $L_{n_h}$   
 $L_{n_h} = [(L_n / (a_1 * a_2 * a_3))^{1/Y} * f_n * 500]^Y$
- Life\_Factor  
 $f_h = f_n * (C / P)$
- Speed\_Factor  
 $f_n = (33.3 / \text{RPM})^{1/Y}$

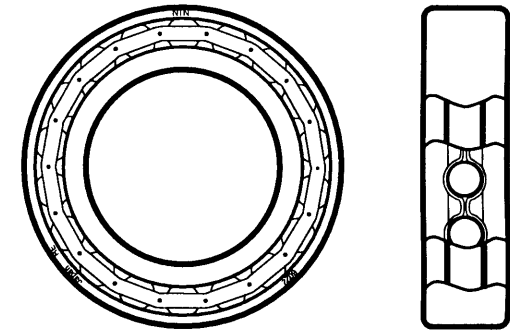
### Operating Conditions

- Operational\_Temperature  
 $Q = 1.05E-5 * M * \text{RPM}$   
T from thermal equilibrium
- Coefficient\_of\_Friction  
 $\mu = (2 * M) / (P * d)$

## PMDA-Specified Values

### Miscellaneous

- ID



## Table or Graph-Specified Values

### Loading

- Axial\_Load\_Factor
- Radial\_Load\_Factor
- Static\_Axial\_Load\_Factor
- Static\_Radial\_Load\_Factor

### Operating Conditions

- Coefficient\_of\_Friction

### Miscellaneous

- gamma

## User Specified Values

### Operating Conditions

- Axial\_Load (applied)
- Radial\_Load (applied)
- RPM

### Geometry

- Arrangement

Figure 6.7: The NTN Rotary Bearing Object

#### 6.2.4. Summary

The process for developing a template class and one manufacturer-specific class was presented. The rotary bearing template object is a standard representation of the information required to fully describe a general rotary motion bearing. A set of variables, equations, and constraints provide the relationships necessary to specify the bearing information pertinent to conceptual evaluation. The manufacturer-specific object determines the values of these variables, and allows the variation between manufacturers to be accounted for. This structure allows the routines in PMDA to operate with a standard set of component objects, while providing for the variation in components between manufacturers.

### 6.3 Case Study - A Simple Machine Spindle

One of the most common sub-assemblies used in machine tools is the rotary motion spindle. Spindles provide the rotational motion necessary for machining, either through the rotation of a workpiece or the rotation of a cutting tool. The functionality and operation of PMDA will be illustrated in this section through the modeling of a simple machine spindle. A kinematic model of the spindle will be created using PMDA, and the total tool-workpiece interaction error for an arbitrary rotation angle will be determined based on the performance of the associated components. The constraints associated with several of the components will also be evaluated to determine if the spindle assembly meets the desired level of performance.

A diagram of a simplified spindle for the rotation of a workpiece is shown in Figure 6.8. The shaft of the spindle is supported by two rotational motion bearings. The forward bearing, referred to as the "fixed bearing", is rigidly connected to the spindle housing and maintains the position of the shaft with respect to the housing. The other bearing, referred to as the "floating bearing", is allowed to move slightly along the longitudinal axis of the spindle to allow for the thermal expansion and contraction of the spindle shaft. The chuck is connected rigidly to the spindle shaft, and secures the workpiece to the spindle assembly through the use of clamps ("jaws"). The spindle is rotated through the use of a motor (not shown) attached to the end of the shaft.

Several error motions common in machine tool spindles are shown in Figure 6.9. Each of these errors arises from errors in either the individual mechanical components or the connection between components, or from the operation and setup of the spindle. *Axial runout*

is the result of thermal growth in the shaft and the axial runout of the fixed bearing. *Radial runout* in the spindle is due to the radial runout of the fixed bearing and the off-center mounting

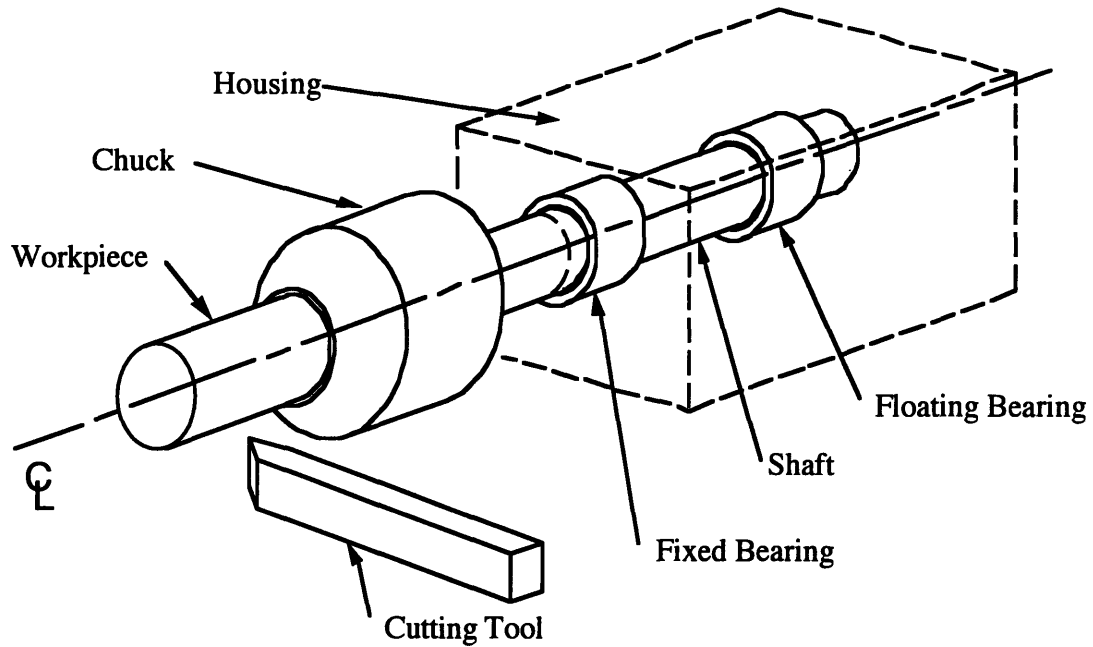


Figure 6.8: A Simple Machine Tool Spindle

errors between the shaft, chuck, and workpiece. *Droop* is the result of the weight of the workpiece and chuck causing the shaft to deflect. Note that the errors are dependent on the rotational angle of the spindle (with respect to a fixed reference), and that the rotation of the spindle will result in changes in the deflection of the workpiece and chuck.

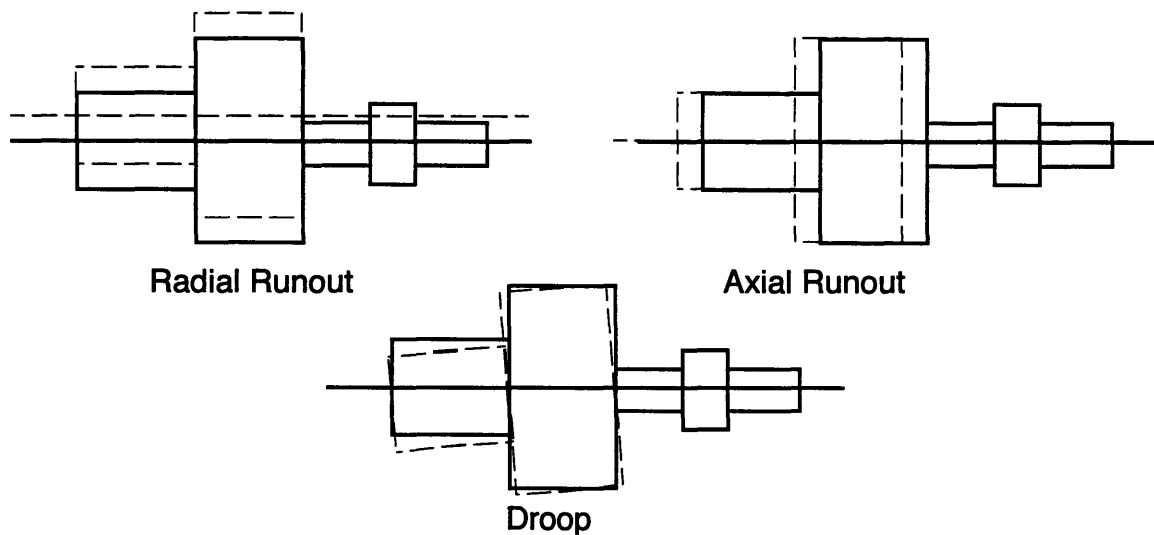


Figure 6.9: Common Machine Spindle Error Motions

Figure 6.10a shows the geometry and operating conditions for the example spindle assembly. The location and orientation of the reference frames for each component are shown, as are the operating conditions for the assembly. The values for calculation of error motions are also shown; details on the calculations are found in Appendix B. The connections between components is assumed to be rigid, although mounting errors between components are included in the component objects. The "floating" bearing is assumed not to affect the error motions of the spindle assembly and is not included in the machine model. Figure 6.10b shows the structural loops associated with the example spindle. The tool loop consists only of the tool object (and the global reference frame), while the workpiece loop contains the workpiece, chuck, shaft, and rotary bearing objects.

The following sections present the method for analyzing this simple machine spindle with PMDA. The four steps involved are component selection, kinematic modeling, error motion calculation, and constraint evaluation. These steps are described in Chapter 5, and are common for the analysis of any machine concept using PMDA.

### **6.3.1 Component Selection**

The process of evaluating a machine concept with PMDA begins with the selection of the components that will comprise the model. For the simple machine spindle, the following components are selected:

- 1 shaft (BERG LS-1-12L)
- 1 rotary bearing (NTN NU7204)
- 1 chuck (custom, 60mm X 160 mm dia., 3-tooth jaw)
- 1 workpiece (custom, 80 mm X 60 mm dia.)
- 1 cutting tool (custom, 80 mm X 20 mm X 20 mm)

These components comprise of all of the objects necessary to model the spindle. The shaft and rotary bearing are selected (via an input file) from catalogs read by PMDA when the program is started. The remaining objects are custom, and are specified in the input file (see Appendix A). The set of valid materials is also read from a material data file that may be modified by the user; the material data file is found in Appendix A. Details concerning the object representations of the above components are found in Chapter 4.

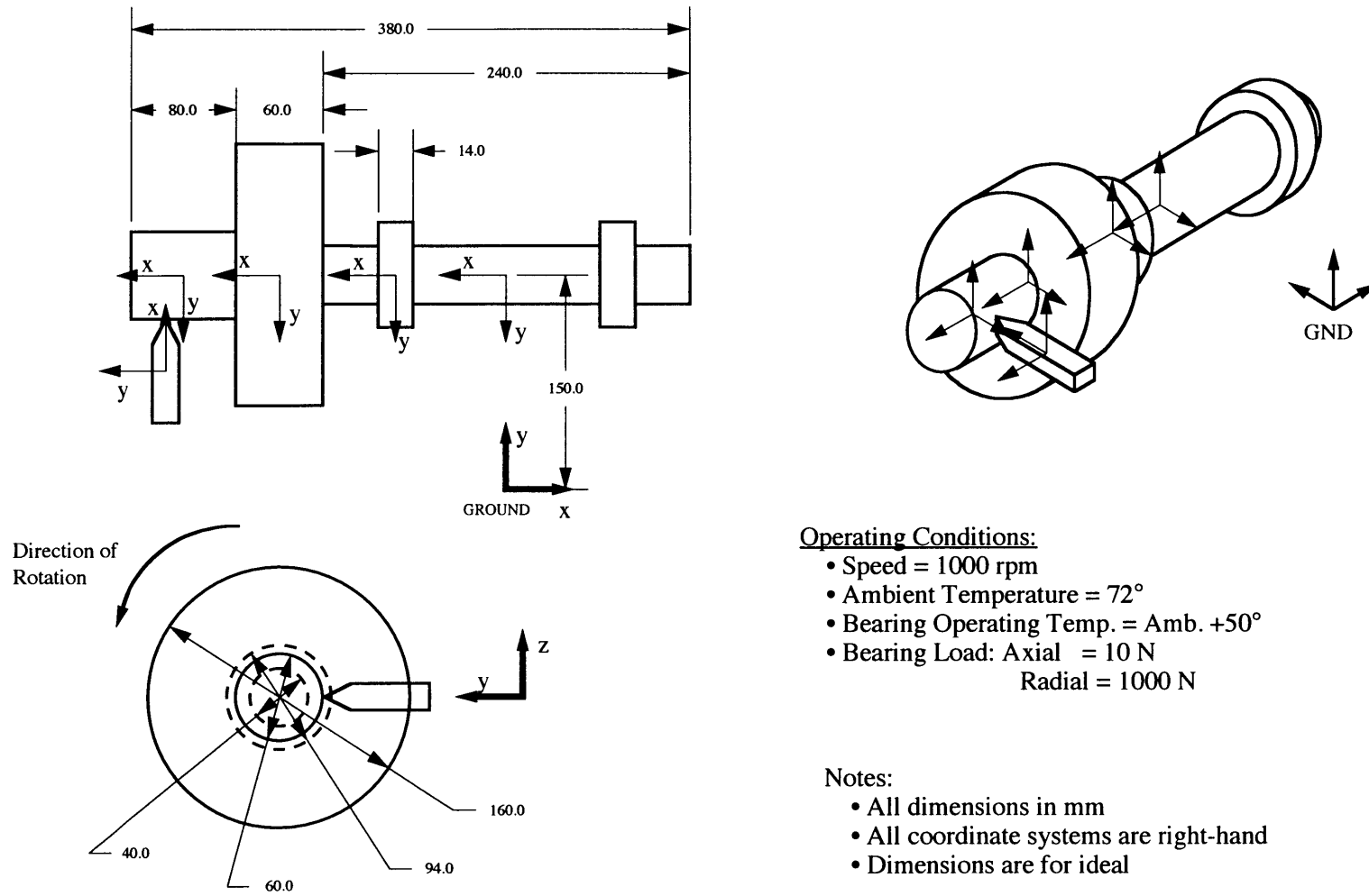
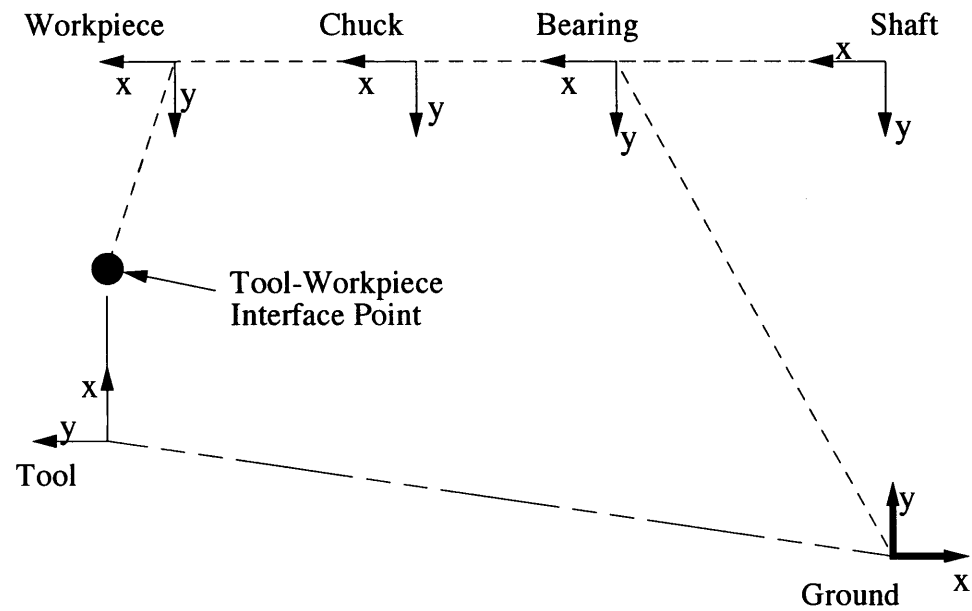


Figure 6.10a: Simple Machine Spindle Model



**Workpiece Loop:**

Workpiece --> Chuck --> Shaft --> Bearing --> Ground

**Tool Loop:**

Tool --> Ground

Figure 6.10b: Structural Loops for the Simple Machine Spindle

### 6.3.2 Spindle Modeling

Once the necessary mechanical components are specified and the component objects are created, the kinematic model of the machine is developed. For the example spindle, the 5 interfaces created are:

- Rotary Bearing - Ground
- Shaft - Rotary Bearing
- Chuck - Shaft
- Workpiece - Chuck
- Tool - Ground

The details for each interface are found in Chapter 5 and in Appendix A.

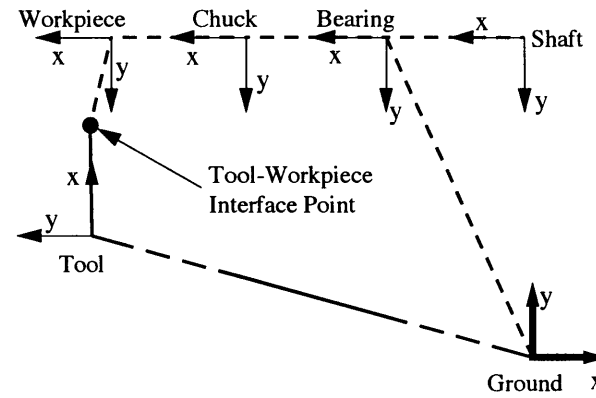
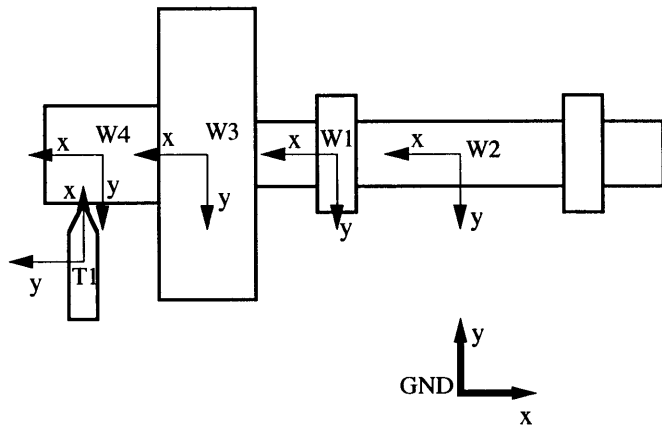
The second step in the creation of the kinematic model involves the specification of the position and orientation of each component. For the machine spindle example, the positions and orientations of each component are read in from an input file. Once this step is completed, the HTMs are created automatically by PMDA; the algorithm for automatic generation of HTMs is found in Appendix C. The positions and orientations of each component in the machine spindle are shown in Figure 6.10a.

### 6.3.3 Kinematic Modeling

Once the geometry of the spindle is determined, and the necessary component and interface objects are created, the structural loops can be defined. The structural loops for the machine spindle are shown in Figure 6.10b, and are defined through the input file for the machine spindle. Both the tool-ground and workpiece-ground loops are defined in this manner. Structural loops are explained in Chapter 5, and the implementation of structural loops in PMDA is explained in Appendix C. The HTMs for the spindle assembly at any spindle rotation angle  $\Theta$  are shown in Figure 6.11.

### 6.3.4 Error Motion Modeling

The process of evaluation of tool-workpiece interaction error is described in detail in Chapters 2 and 5. The process involves the determination of the tool-workpiece interaction point under ideal conditions, the transformation and modification of this point through each structural loop in the machine model, and the comparison of the ideal points with the "error-modified" points. For the machine spindle example, the tool-workpiece interaction point is



$${}^{W3}HTM_{W4} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 70.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$${}^{W2}HTM_{W3} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 150.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$${}^{W1}HTM_{W2} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & -90.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$${}^{GND}HTM_{W1} = \begin{bmatrix} -1.0 & 0.0 & 0.0 & -90.0 \\ 0.0 & -1.0 & 0.0 & 180.0 \\ 0.0 & 0.0 & 1.0 & 80.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$${}^{T1}HTM_{TW} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & TL/2 \\ 0.0 & 1.0 & 0.0 & TW/2 \\ 0.0 & 0.0 & 1.0 & TH/2 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$${}^{W1}HTM_{TW} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & X_{TL} \\ 0.0 & 1.0 & 0.0 & WW/2 \\ 0.0 & 0.0 & 1.0 & Z_{TL} \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

TL = Tool Length    TH = Tool Height  
 TW = Tool Width    WW = Workpiece Width  
 X<sub>TL</sub> = X Position of Toolpoint  
 Z<sub>TL</sub> = Z Position of Toolpoint

$${}^{GND}HTM_{T1} = \begin{bmatrix} 0.0 & -1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 1.5 \\ 0.0 & 0.0 & 1.0 & 12.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Figure 6.11: HTMs for the Simple Machine Spindle

determined by the position of the tool object (set in the input file). This point represents the ideal toolpoint location, and is assumed to be the contact point of the tool with the workpiece. The toolpoint is then transformed to the workpiece and stored as the ideal workpiece point. In the global reference frame, the ideal toolpoint and the ideal workpiece point will be coincident.

Once the ideal points are set, the actual points are determined through the transformation and modification of the point location through each object in the structural loop. The complete set of error motions included in the machine spindle example, as well as the characteristic values of these error motions, are shown in Table 6.1. Note that the values selected are not meant to be representative of real error motions; instead, they are selected to provide large deflections that are easily observed and to match the relative magnitudes of the other error motions involving errors in a specific (global) axis. A second run, with reasonable

<b>Object</b>	<b>Error Motion</b>	<b>Value</b>	<b>Angle</b>
<b>Workpiece</b>	none	N/A	N/A
<b>Chuck</b>	Jaw Center-Offset	5.0 mm	$\Pi$ / 4.0
	Mounting Offset	0.32 mm	$\Pi$ / 7.28
<b>Shaft</b>	Eccentricity (X/Y)	0.1815/-0.0821 mm	0° / 90°
	Center-Offset	5.0 mm	$\Pi$
	Thermal Gradient	+50° from ambient	N/A
<b>Rotary Bearing</b>	Inner Ring Runout	2.5 mm	7.0 $\Pi$ / 4.0
	Outer Ring Runout	2.5 mm	7.0 $\Pi$ / 4.0
	Axial Runout	.008 * RND #	N/A
	Surface Finish (min/max)	0.0082 / 0.0022	N/A
<b>Tool</b>	X-Straightness	0.0243 mm/mm	N/A
	Y-Straightness	0.0671 mm/mm	N/A
	Z-Straightness	0.189 mm/mm	N/A

Table 6.1: Error Motion Values for the Example Machine Spindle

error motion values, is presented in Appendix A. Details of the error motions models are found in Appendix B. Since the error motions of the workpiece loop vary with respect to the rotation angle of the spindle, the errors and associated workpiece point deflection is calculated for a full revolution of the spindle in 1-degree increments. The objects in the tool loop and the associated errors are assumed to be static with respect to time, and required only one calculation to determine the deflection of the toolpoint.

The deflections resulting from the above error motions are graphed in Figure 6.12. The error motions associated with the tool object are assumed to be simple deflections, and result in a static translation of the toolpoint for any rotation angle of the spindle. This error is shown in Figure 6.12a. The results of the spindle error motions are shown in Figures 6.12b and 6.12c. The axial deflection of the ideal workpoint, resulting from the axial runout and surface finish effects of the bearing and the thermal expansion of the shaft, is shown in Figure 6.12b. This figure shows the displacement of the ideal workpoint with respect to the global reference frame. Note that the thermal expansion and axial bearing runout result in a static translation of the workpoint, while the surface finish effects change with respect to rotation angle. Figure 6.12c shows the results of the center-offset, runout, and eccentricity errors from the objects in the workpiece structural loop on the global Y and Z location of the workpoint. Note that the Z error is simply the Y error rotated 90 degrees, and that the angles of the offsets (see Table 6.1) impact the overall shape "traced" by the workpoint. These figures represent the overall performance of the machine spindle with respect to the included components, and may be used in the analysis or comparison of similar machine spindle concepts.

### 6.3.5 Constraint Evaluation

Constraints are discussed in detail in Chapter 3, while Chapter 5 presents the methods for including constraints in PMDA. Currently, only the tool and rotary bearing objects contain complete constraint sets; for this reason, the machine spindle example only deals with constraints on the tool and rotary bearing.

The complete set of constraints associated with the machine spindle, as well as the values selected for the example and the violations as reported by PMDA, and shown in Figure 6.13. The actual PMDA output file for the constraint-evaluation routines is found in Appendix D. Once the values for the constraints are specified (see Chapter 5), and the constraint objects are created (which is handled automatically by PMDA), the network may be evaluated. For the sample machine spindle, several constraints were reported as violated (indicated by bold type in Figure 6.13). For the rotary bearing component, the L10 lifetime (which predicts the operational life of the bearing) is indicated to be higher than the maximum value. This indicates that the bearing is "underutilized", and that a bearing with a lesser lifetime (and most likely a lower cost) may be used. The cost of the bearing is also in excess of the upper bound, suggesting that a lower cost bearing may be used for this application. The violations for the tool object indicate that the x, y and z straightness values are far in excess of the acceptable values, indicating that this particular tool is not an acceptable choice for the given set of constraints.

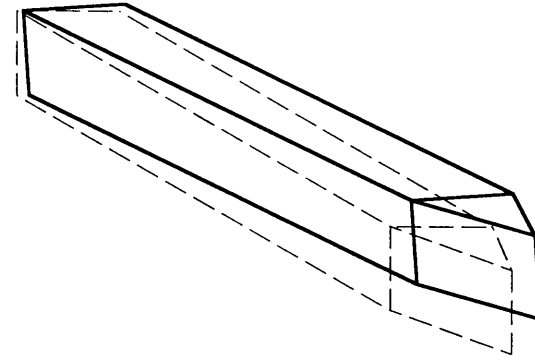
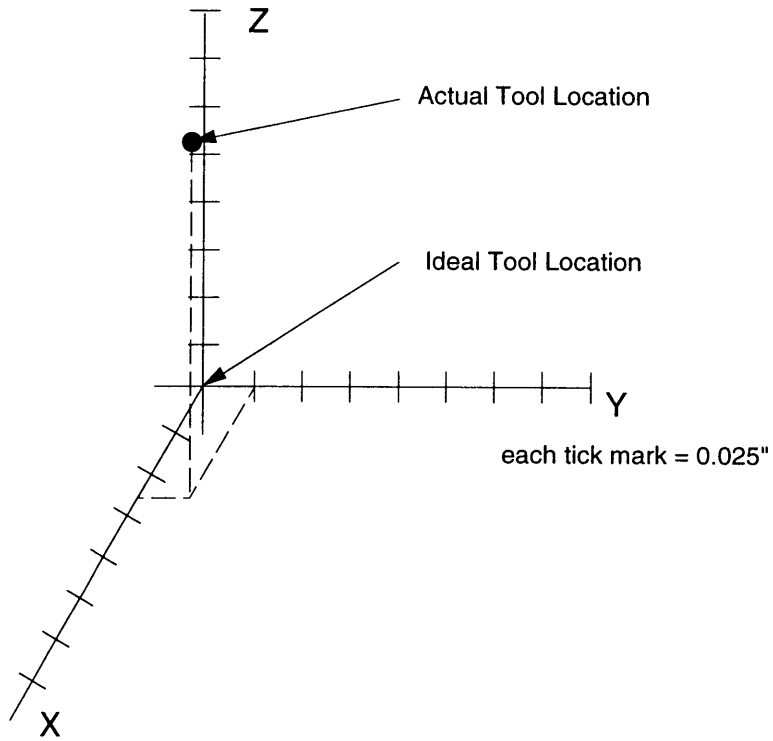


Figure 6.12a: Tool point Displacement Errors for the Simple Machine Spindle

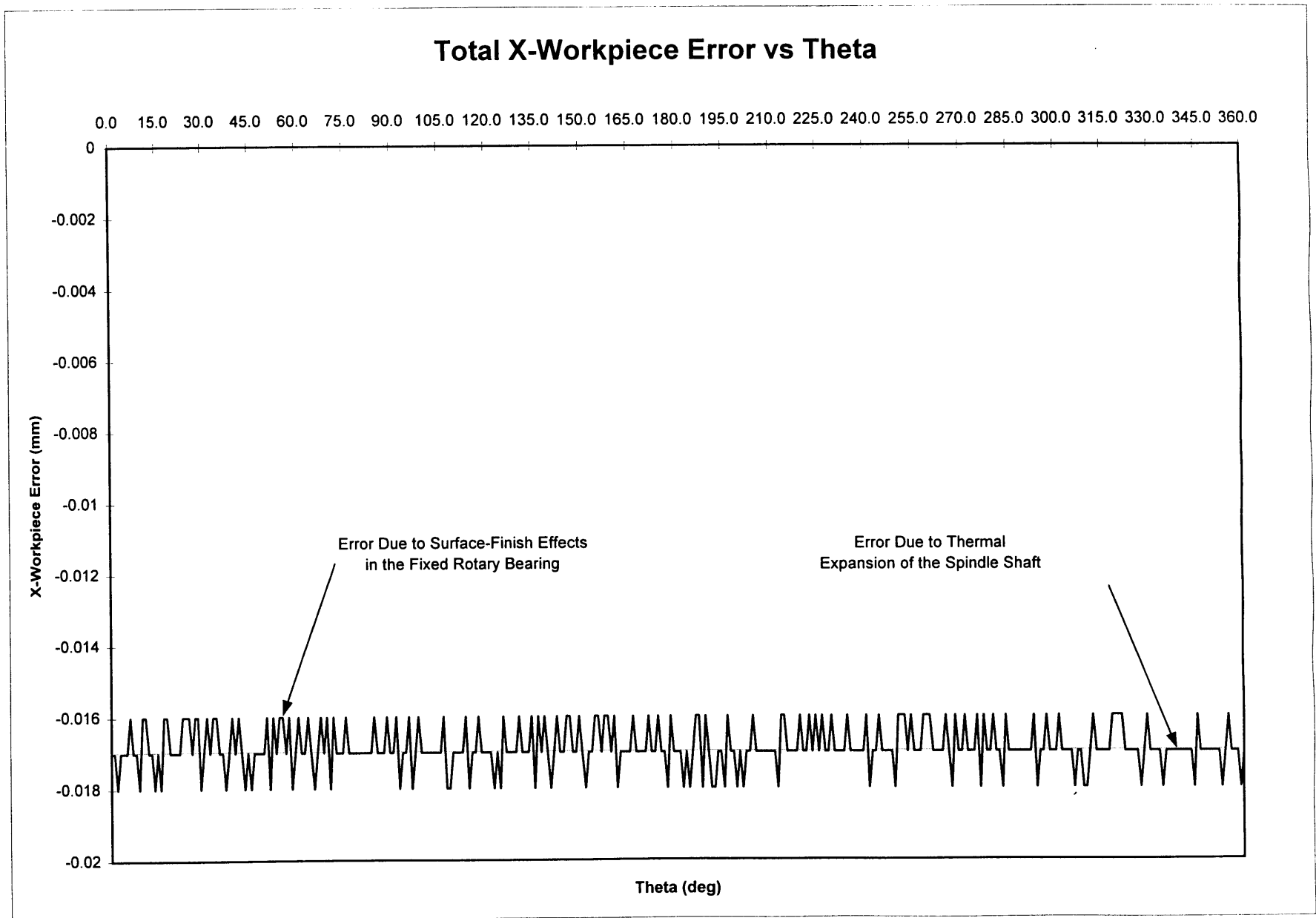


Figure 6.12b: Axial Displacement Errors for the Simple Machine Spindle

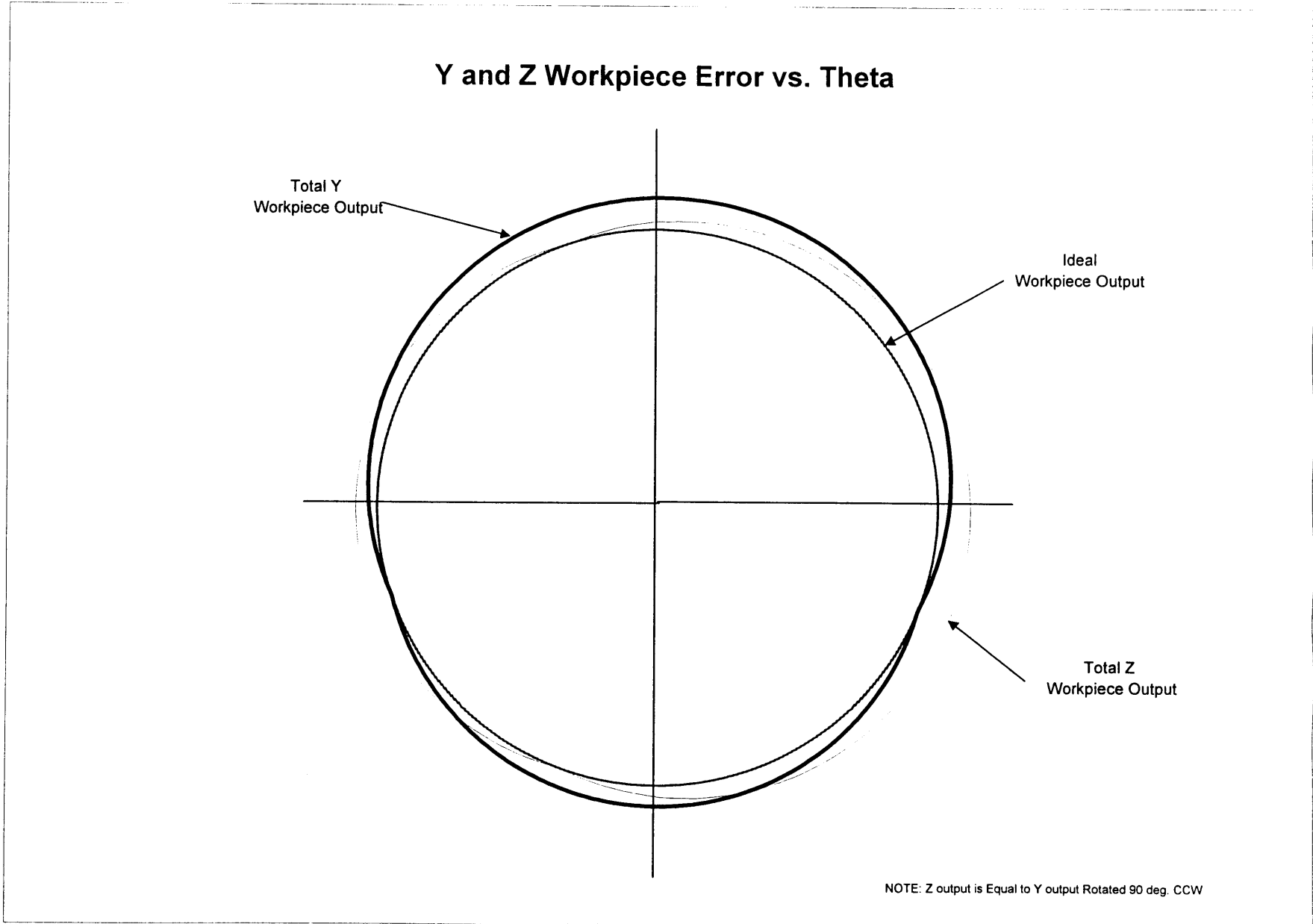


Figure 6.12c: Radial Displacement Errors for the Simple Machine Spindle

### **6.3.7 Conclusion**

While the results above provide an indication of the performance of the machine spindle, care must be taken while analyzing the data. As stated in Chapter 1, PMDA is not meant to provide data which can be directly compared against empirical data obtained from an existing machine. Instead, PMDA provides a simple, easy-to-implement tool for the evaluation of machine tool concepts and assemblies. The error motion models used above provide a simple approximation of the actual function of the machine tool, but do not accurately model the true motion of a machine spindle. For this reason, care must be taken in the selection of error motion models to ensure the required level of accuracy is achieved without an unnecessary increase in model complexity or evaluation time. Likewise, the network of constraints provides information that is only as accurate and useful as the values included in the constraints, and care must be taken in the selection of constraint values and in the interpretation of the constraint evaluation results. For example, the constraints associated with the straightness of the cutting tool indicate that the cutting tool selected for the example case is not an acceptable choice. If the servo system for the tool loop is capable of determining the actual toolpoint location, however, and is not based on the fixturing point (as in the example above), the tool may be an acceptable choice regardless of the tool straightness error.

## **6.4 Chapter Summary**

The above example provides a detailed analysis of the functionality and performance of PMDA. The collection of rotational motion bearing information into a common, reusable set of computational objects was presented. The hierarchy of the rotary motion bearing necessary to include the variation between manufacturer-specific parameters without the repetition of data results in two levels of objects. The first level contains the generic information, equations, and constraints necessary to define a standard rotary-motion bearing. The specific data associated with the generic object, however, is contained within another, lower level of object specific to bearings from a given manufacturer. The resulting object structure ensures all information necessary for the inclusion of a rotary bearing in a PMDA assembly is present without the needless (and computationally expensive) repetition of data.

A simple machine spindle provided an example of one application of PMDA to the evaluation of machine tool assemblies. The kinematic model of the spindle was created with a minimum of user input. The total tool-workpiece interaction error for one complete revolution

**Tool Object:**

$X\_Straightness \leq Max\_X\_Straightness$   
 $0.189 \leq 0.0005$   
 $Y\_Straightness \leq Max\_Y\_Straightness$   
 $0.0671 \leq 0.0006$   
 $Z\_Straightness \leq Max\_Z\_Straightness$   
 $0.0243 \leq 0.0007$   
 $Tool\_Mass \leq Max\_Tool\_Mass$   
 $0.23864 \leq 0.3$   
 $Tool\_Mass \geq Min\_Tool\_Mass$   
 $0.23864 \geq 0.001$   
 $Tool\_Length \leq Max\_Tool\_Length$   
 $80 \leq 100$   
 $Tool\_Length \geq Min\_Tool\_Length$   
 $80 \geq 0.001$   
 $Tool\_Width \leq Max\_Tool\_Width$   
 $20 \leq 30$   
 $Tool\_Width \geq Min\_Tool\_Width$   
 $20 \geq 0.001$   
 $Tool\_Height \leq Max\_Tool\_Height$   
 $20 \leq 30$   
 $Tool\_Height \geq Min\_Tool\_Height$   
 $20 \geq 0.001$   
 $Applied\_X\_Loading \leq (Height\ Width\ Sigma\_Comp.\ **)$   
 $0 \leq 280000$   
 $Applied\_X\_Loading \leq (Height\ Width\ Sigma\_Yield.\ **)$   
 $0 \leq 280000$

**Rotary Bearing Object:**

$L\_10\_Life \leq Max\_L10\_Life$   
 $3.04862e+09 \leq 1e+06$   
 $Bearing\_Cost \leq Max\_Bearing\_Cost$   
 $30.00 \leq 20.00$   
 $Bearing\_Mass \leq Max\_Bearing\_Mass$   
 $0.1 \leq 1.0$   
 $Bearing\_Mass \geq Min\_Bearing\_Mass$   
 $0.1 \geq 0.0$   
 $Bearing\_Outside\_Diameter \leq Max\_Bearing\_Outside\_Diameter$   
 $47 \leq 50$   
 $Bearing\_Outside\_Diameter \geq Min\_Bearing\_Outside\_Diameter$   
 $47 \geq 0.0$   
 $Bearing\_Bore\_Diameter \leq Max\_Bearing\_Bore\_Diameter$   
 $20 \leq 40$   
 $Bearing\_Bore\_Diameter \geq Min\_Bearing\_Bore\_Diameter$   
 $20 \geq 0.0$   
 $Bearing\_Width \leq Max\_Bearing\_W$   
 $14 \leq 30$   
 $Bearing\_Width \geq Min\_Bearing\_W$   
 $14 \geq 0$   
 $Bearing\_ABEC\_Class \geq Min\_Bearing\_ABEC\_Class$   
 $3 \geq 1$   
 $Dynamic\_Equivalent\_Radial\_Load \leq Basic\_Dynamic\_Rated\_Load$   
 $1000 \leq 14500$   
 $Operational\_Temperature \leq Max\_Operational\_Temperature$   
 $73 \leq 180$   
 $Bearing\_RPM \leq Limit\_Speed\_Grease$   
 $10000 \leq 17000$   
 $Bearing\_RPM \leq Limit\_Speed\_Oil$   
 $10000 \leq 23000$   
 $L\_10\_Lifetime \geq Min\_L10\_Lifetime$   
 $3.04862e+09 \geq 5000$   
 $L10\_Lifetime\_Hours \leq Max\_L10\_Lifetime\_Hours$   
 $5075.96 \leq 1e+06$   
 $L10\_Lifetime\_Hours \geq Min\_L10\_Lifetime\_Hours$   
 $5075.96 \geq 3000$

Figure 6.13: Constraints for the Simple Machine Spindle

of the spindle was determined through the inclusion of individual component errors, and a set of constraints associated with some of the objects included in the assembly provided an indication of the level of feasibility of the components included in the spindle assembly. With proper selection of error motion models and the constraints associated with the component objects, a machine tool concept can be analyzed and evaluated to determine the level of performance with respect to the form of the concept. Multiple concepts may be evaluated based on a common library of component objects to ensure the evaluation process is consistent across all concepts. The results of these analyses may then be used to determine which concept or concepts warrant additional attention, and which concepts are not likely to achieve the desired level of performance.

## **7.0 Conclusions**

### **7.1 Summary of Work**

This dissertation presents the development of a software tool for the rapid evaluation and optimization of precision machine tool concepts. This tool allows for a reduction in the time required to determine the performance of a specific machine tool concept, and helps formalize the process of selecting one of a set of concepts for continued development. Chapter 1 presented the challenges involved in development of precision machine tools, and some common methods for the classification of error in these machines. The challenges of new machine development and modification of existing machines is also discussed. Chapter 2 discussed the sources of error motions in machine tools, and the methods for representing these errors in both fixed and rotating coordinate frames. Methods for modeling machine tool error motions through the use of Homogeneous Transformation Matrices were presented, and a comprehensive example of the development and evaluation of a machine tool concept was constructed. Chapter 3 discussed constraint-based methods for the evaluation of general design concepts. The sources and methods of representing design constraints were presented, and the use of constraints in the design process was discussed. The computational representation of machine tool components was discussed in Chapter 4. The requirements of these objects, along with the additional objects necessary to store this information and describe the connection between components, is presented. The functionality of PMDA was discussed and illustrated in Chapters 5 and 6. The methods of machine modeling, total tool-workpiece interaction error, and constraint evaluation are discussed and applied to two example cases. The following sections briefly describe the key areas addressed by PMDA.

#### **7.1.1 Kinematic Machine Modeling**

The development of kinematic models for machine tool concepts is achieved through the use of Homogeneous Transformation Matrices (HTM). Once the location and orientation of each component is specified and the connections between components is determined, the process of HTM calculation is automatic. This allows for the rapid re-evaluation or modification of the machine tool model.

Standard catalog-based component objects are also included in the model. As each component is defined in the model, the specific information associated with the component is set either by the designer or by selection of a manufacturer-specific, catalog-based component. This ensures the model is based on realistic, commercially-available components.

### **7.1.2 Error Motion Modeling and Evaluation**

Evaluation and determination of the error motions associated with a specific machine tool concept is achieved through the inclusion of basic error motion representations in each mechanical component object. The impact of these errors on the total tool-workpiece interaction error for the machine concept is determined by transforming the ideal point of tool-workpiece contact through the HTMs associated with the tool and workpiece structural loops. The impact of each error motion or each component on the tool-workpiece interaction error may also be determined by sensitivity analysis.

### **7.1.3 Constraint Evaluation**

The constraints on a given machine tool concept are included in the component and interface objects in the machine concept model. The variables and equations associated with these constraints are automatically set by PMDA. Evaluation is also completely automated, and results in a list of violated constraints based on the components included in the machine model.

## **7.2 Analysis of Approach**

As stated previously, PMDA is an object-oriented tool for the rapid evaluation of machine tool concepts. Kinematic modeling is performed through the use of HTMs, and evaluation of error motions and constraints on the design are conducted through the use of standard representations of mechanical components, design information, and design constraints. The advantages and disadvantages of this approach are discussed below.

### **7.2.1 Advantages of the PMDA Approach**

The PMDA approach has several advantages. First, the use of a library of standard components is closely related to the actual process of machine design, where standard mechanical components (both generic and manufacturer-specific) are selected to perform certain functions in the machine. Machine tool concepts are generally designed in terms of these components, with the surrounding structure developed to connect the various components. Second, the process of kinematic modeling through the use of HTMs is rapid and relatively simple compared to other methods of machine modeling (for example, finite elements). The mathematics of HTMs is also well-understood and can be implemented for any two

components as long as coordinate system information is provided. Third, error motion modeling through HTMs and standard component error motions is a deterministic process that allows for the inclusion of error motions in any of the components included in the structural loop. Error motion magnitudes can be altered easily, and sensitivity analysis can be performed without altering the machine model or the component objects. Finally, the use of standard constraint objects allows for the standardization of constraint-manipulation and evaluation routines, and decreases the time necessary to create and evaluate a constraint network.

### **7.2.2 Disadvantages of the PMDA Approach**

There are several disadvantages to the PMDA approach. First, kinematic modeling through HTMs requires that each component be modeled as a rigid body. This does not accurately reflect the true nature of machine components, and can be inaccurate under certain machine operating conditions. Second, error motion modeling using the PMDA methods requires that error motions for each component be represented in a standard fashion. While these representations can closely model error motions, some inaccuracy will result from the use of standard representations. This inaccuracy would be reduced if a more detailed modeling approach, such as FEA, was employed. Finally, the use of standard components and constraint representations requires that each object be modeled in a similar fashion. This assumes that each user of PMDA will only require the information included in each standard object, and that additional information does not need to be included. If additional information must be included, a second program for the evaluation of the concept must be used.

## **7.3 Future Research**

The first version of PMDA proved that the rapid evaluation of precision machine tool concepts based on kinematic modeling, error motions, and design constraints, is feasible and applicable to the precision machine tool design process. The possible expansion of PMDA in both short-term and long-term timescales is presented below.

### **7.3.1 Short Term Research**

#### **Expansion of Component Library**

The component library currently includes only a small subset of the components found in a typical machine tool. For this reason, the component library should be expanded to include a more complete set of mechanical elements, such as linear motion components, servo

systems, and drive components. The number and type of manufacturer-specific components should also be increased to ensure the representative set of available mechanical components is complete. This will increase the scope of PMDA and allow for modeling of a wider variety of machine tool concepts.

### **Error Motion Representations**

The current set of error motions included in PMDA are limited to static or quasi-static representations developed specifically for a given component object. The error motion sets for each component are also not complete. The areas for future research concerning error motions includes:

- inclusion of additional error motions for each component to provide a more accurate model of the performance of the component
- standard representations for error motions common to more than one component to allow for a simplification of error motion representation
- inclusion of error motions in the interface classes to model assembly errors
- increased error motion accuracy through the analysis of existing machine components and tools
- inclusion of errors associated with the support structure of the machine

### **Expansion of Constraint Set**

Currently, PMDA contains only a limited number of constraints. This results in an incomplete description of the feasibility of the concept under evaluation. Expanding the constraint set to limit all variables associated with the concept will result in a more complete evaluation of the given concept.

### **7.3.2 Long Term Research**

Several possible avenues of long-term future research exist, including the following:

#### **Interface with Commercial Software Packages**

The integration or interfacing of PMDA with commercial software packages would increase the functionality of PMDA two ways. First, by incorporating a solid-modeling package with PMDA, a more exact definition of geometry is possible. As the solid model is modified, PMDA could determine the effects of the modifications, or PMDA could provide information on necessary geometric modifications. Second, simulation codes (such as FEA

programs) would allow for a more accurate prediction of machine tool errors, including the incorporation or prediction of errors in the machine structure. More accurate predictions would also be possible for bearing errors, thermal deflections, dynamic errors, or any other error that can be simulated based on the information included in PMDA.

### **Detail Design**

PMDA is designed for use in the preliminary phases of design. The inclusion of standard component libraries, however, makes PMDA an attractive candidate for the detailed design phases. By integrating PMDA with a geometric modeling or CAD program, and expanding the object catalogs to include a large number of manufacturers and components, PMDA could play an integral part in the specification, selection, and integration of components in the final stages of machine design.

## **7.4 Conclusion**

A tool for the rapid evaluation of machine tool concepts has been developed. The integration of kinematic modeling, error motion modeling, constraint sets, and catalog-based standard component representations allows for a program that can assist in the selection of one component from a set of possible components. By further expanding and developing the features of PMDA, a substantial amount of the uncertainty currently involved in precision machine design can be reduced or eliminated. This will allow for a reduction in the design cycle time, a reduction in development costs for new and modified machine tools, and an overall improvement in efficiency of the precision machine tool design cycle.

## Bibliography

- Bryan, James B. The Power of Deterministic Thinking in Machine Tool Accuracy. Proceeding of the First International Machine Tool Engineers Conference, Tokyo, Japan. 1984.
- Chakrabarti, A., T.P.Bligh, and T. Holden. Towards a Decision-Support Framework for the Embodiment Phase of Mechanical Design. *Artificial Intelligence in Engineering*, 7:21-36, 1992.
- Chan, Weng Tat, and Boyd D. Paulson, Jr. Exploratory Design using Constraints. *AI EDAM*, 1(1):59-71, 1987.
- Colburn, Eric R., and James R. Rinderle. Design Relations. Proceedings of the Second ASME Design Theory and Methodology Conference, Chicago, Illinois, September, 1990.
- Colton, Jonathan S., and John L. Dascanio II. An Integrated, Intelligent Design Environment. *Engineering with Computers* 7, 11-22, 1991.
- Ellis, Margaret A., and Bjarne Stroustrup. The Annotated C++ Reference Manual. Addison-Wesley Publishing Company, Reading, MA. 1990.
- Foley, James D., A. van Dam, S. K. Feiner, and J. F. Hughes. Computer Graphics - Principles and Practice, 2nd ed. Addison-Wesley Publishing Company, Reading, MA. 1990.
- Frey, Daniel. personal interview, 1997.
- Gorti, Sreenivasa R., and Ram S. Sriram. From Symbol to Form: A Framework for Conceptual Design. *Engineering Applications of Artificial Intelligence*, pp. 853-870. 1995.
- Johnsonbaugh, Richard, and Matrin Kalin. Applications Programming in C. MacMillian Publishing Company, New York. 1989.
- Kannapan, Srikanth M., and Kurt M. Marshek. An Algebraic and Predicate Logic Approach to Representation and Reasoning in Machine Design. *Journal of Mechanical Machine Theory*, Vol. 25, No. 3, pp 335-353, 1990.
- Li , C. L., S.T. Tan, and K.W. Chan. A Qualitative and Heuristic Approach to the Conceptual Design of Mechanisms. *Engineering Applications in Artificial Intelligence*, Vol. 9, No. 1, pp. 17-31, 1996.
- Mahajerin, E. *Computer Modeling of Spindle Growth*. Proceedings of the 4th Biennial International Machine Tool Technology Conference, Sept. 1988. pp 255-274.
- NTN Corporation. Ball and Roller Bearing Catalog No. 2200II/E. 1990.

- Popov, Egor. Engineering Mechanics of Solids. Prentice-Hall, Inc., New Jersey, 1990.
- Sedgewick, Robert. Algorithms in C++. Addison-Wesley Publishing Company, Reading, MA. 1992.
- Shigley, Joseph Edward, and Charles R. Mischke. Mechanical Engineering Design. 5th Ed. McGraw-Hill, New York. 1989.
- Slocum, Alexander H. Precision Machine Design. Prentice-Hall, Inc., New Jersey, 1992.
- Spur, G., S. Knupfer, and A. Schule. Evaluation of Machine Tool Design Using Simulation Systems. ASME, PED Modeling of Machine Tools: Accuracy, Dynamics, and Control Winter Annual Meeting of the American Society of Mechanical Engineers, Dallas, TX. Vol 45, pp 47-53. 1990.
- Thornton, Anna. *Constraint Specification and Satisfaction in Embodiment Design*. PhD Thesis, Dept. of Engineering, University of Cambridge, June 1993.
- Thornton, Anna. *A Support Tool for Constraint Processes in Embodiment Design*. Mechanical Engineering Dept., Massachusetts Institute of Technology, 1996.
- The Timkin Company. Bearing Selection Handbook. 1989.
- The Torrington Company. Service Catalog: Fafnir Ball, Torrington Roller, Torrington Needle Roller, and Kilian Precision Machined Bearings. 1988.
- Ulrich, Karl T., and Steven D. Eppinger. Product Design and Development. McGraw-Hill, Inc. New York, 1995.
- Watton, John D., and James R. Rinderle. Identifying Reformulations of Mechanical Parametric Design Constraints. *AI EDAM*, 5(3):173-178, 1991.

## **A. Sample Case**

This section presents the details of the sample case discussed in Chapter 6. Specifics concerning the component and interface objects are presented. A second run of the sample case using more representative error motion values is discussed, and the input and output files for the original sample case are included.

### **A.1. Component and Interface Objects**

The specifications for the component objects used in the sample case are found in Figure A.1. These objects represent all of the components included in the sample case, and are both catalog-based and custom items. The specifications for the interface objects are found in Figure A.2. These figures contain all of the information required by PMDA to construct the spindle model, with the exception of the location and orientation of the coordinate frame for each component.

### **A.2. Sample Case Second Run**

The sample case presented in Chapter 6 uses large error values in order to illustrate the effects of each error motion and provide data that is easily displayed in the graphs found in Figures 6.12a, 6.12b, and 6.12c. In reality, the error motions included in PMDA models will be several orders of magnitude smaller, and for precision machines may be on the order of several microns ( $\mu\text{m}$ ). A second run of the sample case was completed with the more realistic error values shown in Table A.1. The output for this second run is included in Appendix D. Total modification of the sample case for inclusion of the new values took less than five minutes, demonstrating the speed of machine modeling and concept evaluation using PMDA.

### **A.3. Sample Case Output Files**

The output files for the original machine spindle run are listed below. The following files are included (see Appendix D for an explanation of the files):

- PMDA.OUT
- PMDA\_HTMS.OUT
- PMDA\_CONSTRAINTS.OUT

**Component Objects:****Rotary Bearing:**

- Manufacturer: NTN
- Catalog Number: NU7204
- Type: Angular Contact - Ball Bearing
- Bore: 20.0 mm
- Outside\_Dia: 47.0 mm
- Width: 14.0 mm
- Dynamic Rated Load: 14500 N
- Static Rated Load: 8400 N
- Limit Speed - Oil: 23000 rpm
- Limit Speed - Grease: 17000 rpm
- Mass: 0.100 kg
- ABEC\_Class: 3
- Contact Angle: 30°
- Units: METRIC
- Tolerances:
- Delta\_D\_M\_P: -0.004 mm
- V\_D\_P: 0.004 mm
- K\_I\_A: 0.003 mm
- S\_D: 0.004 mm
- S\_I\_A: 0.008 mm
- S\_E\_A: 0.008 mm
- Delta\_B\_S: -.120 mm
- V\_B\_S: 0.0025 mm

**Shaft:**

- Manufacturer: BERG
- Catalog Number: LMS-46-12L
- Diameter: 20.0 mm
- Length: 240.0 mm
- Material: ASTM-A514 Steel
- EccX: .50 mm
- EccY: .40 mm
- Offset: 1.2 mm
- DiamError: .04 mm
- Units: METRIC

**Chuck:**

- Manufacturer: CUSTOM
- Catalog Number: 0001
- Diameter: 160.0 mm
- Jaw\_Diameter: 70.0 mm
- Height: 60.0 mm
- Material: 2014-T6 Aluminum
- ID Number: 3

**Workpiece:**

- Manufacturer: CUSTOM
- Catalog Number: 0010
- Diameter: 60.0 mm
- Length: 80.0 mm
- Material: 2014-T6 Aluminum
- ID Number: 6

**Tool:**

- Manufacturer: CUSTOM
- Catalog Number: 0100
- Length: 80.0 mm
- Height: 20.0 mm
- Width: 20.0 mm
- Material: ASTM-A514 Steel
- ID: 9

Figure A.1: Sample Case Component Objects

**Interface Objects:****Bearing-Ground Interface**

Rotary Bearing: NTN NU7204

Ground Block: Ground\_0001

Variable Pointers

- id

Name: B\_G\_Interface\_01

Units: METRIC

**Shaft-Chuck Interface:**

Shaft: Shaft\_LMS\_46\_12L

Chuck: Chuck\_0001

Variable Pointers†:

- rpm
- id

Name: S\_C\_Interface\_01

Units: METRIC

† All Variable Pointers point to Assembly Variables

**Tool-Ground Interface:**

Tool: Tool\_0100

Ground Block: Ground\_0001

Variable Pointers:

- id

Name: T\_G\_Interface\_01

Units: METRIC

**Bearing-Shaft Interface:**

Shaft: Shaft\_LMS\_46\_12L

Bearings:

- NTN\_NU\_7204\_01

Fixed Bearing: NTN\_NU\_7204\_01

Variable Pointers†:

- rpm
- id;
- angle\_x
- angle\_y
- angle\_z
- axial\_load
- radial\_load

Number of Bearings: 1

Name: B\_S\_Interface\_01

Units: METRIC

**Chuck - Workpiece Interface:**

Chuck: Chuck\_0001

Workpiece: Workpiece\_0010

Variable Pointers †:

- rpm
- id
- axial\_load
- radial\_load

Name: C\_W\_Interface\_01

Units: Metric

† All Variable Pointers point to Assembly Variables

Figure A.2: Sample Case Interface Objects

The output files for the second run (using more realistic error motion values) are found in Appendix D.

<b>Object</b>	<b>Error Motion</b>	<b>Value</b>	<b>Angle</b>
<b>Workpiece</b>	none	N/A	N/A
<b>Chuck</b>	Jaw Center-Offset	0.035 mm	$\Pi$ / 1.0
	Mounting Offset	0.002 mm	$\Pi$ / 5.37
<b>Shaft</b>	Eccentricity (X/Y)	0.0035/-0.0008 mm	0° / 90°
	Center-Offset	0.00713 mm	$\Pi$
	Thermal Gradient	+50° from ambient	N/A
<b>Rotary Bearing</b>	Inner Ring Runout	0.008 mm	6.2 $\Pi$ / 4.13
	Outer Ring Runout	0.0038 mm	9.21 $\Pi$ / 3.3
	Axial Runout	.00352 * RND #	N/A
	Surface Finish (min/max)	0.00082 / 0.00022	N/A
<b>Tool</b>	X-Straightness	0.000243 mm/mm	N/A
	Y-Straightness	0.000671 mm/mm	N/A
	Z-Straightness	0.000189 mm/mm	N/A

Table A.1: Error Motion Values for the Sample Case - Run #2

## PMDA.OUT

```
*****  
** The Precision Machine Design Assistant **  
** Version 1.0 - Copyright 1997 M.I.T. **  
** Code written by: Brad Homann **  
** Anna Thornton **  
*****  
*****
```

Assembly Name: SPINDLE

\*\*\*\*Structural Loops\*\*

\*\*Tool Loop:

Tool: CUSTOM 0100  
Length: 80 mm  
Width: 20 mm  
Height: 20 mm  
Material: ASTM-A514  
ID: 9

Tool  
Location: (-240,110,80,1)

\*\*Workpiece Loop:

Workpiece: CUSTOM 0010  
Diameter: 60 mm  
Length: 80 mm  
Material: 2014-T6  
ID: 6

--More--(82%)

Workpiece  
Location: (-220,180,80,1)

Chuck: CUSTOM 0001  
Diameter: 160 mm  
Jaw Diameter: 70 mm  
Height: 60 mm  
Material: 2014-T6  
ID: 3

Chuck  
Location: (-150,180,80,1)

Shaft: BERG LMS-46-12L  
Diameter: 40  
Length: 240  
Material: Steel  
Units: METRIC  
ID: 2

Shaft  
Location: (0,180,80,1)

NTN NU7204 Ang\_Contact\_Ball  
Bore: 20 mm  
Outside\_Dia: 47 mm  
Width: 14 mm  
C: 14500 N  
Co: 8400 N  
L.S.\_Oil: 23000 rpm  
L.S.\_Grease: 17000 rpm

Mass: 0.1 kg  
ABEC\_Class: 3 Units: METRIC  
Special:  
ID: 1  
#  
Location: (-90,180,80,1)

\*\*\*\*Active Error Motions\*\*\*\*

- \* Chuck
- \* Shaft
- \* Bearing
- \* Tool

IDEAL Tool-Workpiece Interaction Location: (-240,150,80,1)  
Actual WorkpiecePoint Locations written to file: PMDA\_ERRORS.OUT  
Constraint Information written to file: PMDA\_CONSTRAINTS.OUT

## PMDA HTMS.OUT

```
*****  
**          The Precision Machine Design Assistant          **  
**          Homogeneous Transformation Matrices            **  
**                      for                                **  
**                                                                **  
**          ASSEMBLY: SPINDLE                             **  
*****
```

TOOL HTM:

Matrix :

```
| 0 -1 0 -240 |  
| 1 0 0 110  |  
| 0 0 1 80   |  
| 0 0 0 1    |
```

END TOOL HTM

WORKPIECE HTM:

Matrix :

```
| 1 0 0 70 |  
| 0 1 0 0  |  
| 0 0 1 0  |  
| 0 0 0 1  |
```

END WORKPIECE HTM

CHUCK HTM:

Matrix :

```
| 1 0 0 150 |  
| 0 1 0 0   |  
| 0 0 1 0   |  
| 0 0 0 1   |
```

END CHUCK HTM

SHAFT HTM:

Matrix :

```
| 1 0 0 -90 |  
| 0 1 0 0   |  
| 0 0 1 0   |  
| 0 0 0 1   |
```

END SHAFT HTM

ROT\_BEAR HTM:

Matrix :

```
| -1 0 0 -90 |  
| 0 -1 0 180 |  
| 0 0 1 80   |  
| 0 0 0 1   |
```

END ROT\_BEAR HTM

# PMDA CONSTRAINTS.OUT

```
*****
**          PMDA Constraint Violation Report          **
**                                                    **
**          for Assembly: SPINDLE                    **
*****
```

```
###Tool      :
  Catalog Number: 0100
  ID Number    : 9
```

```
*****
****V I O L A T I O N S****
*****
```

```
Constraint: MAX_X STRAIT
  SYM strait_x # <= SYM max_x_strait #
    0.189 <= 0.0005
Constraint: MAX_Y STRAIT
  SYM strait_y # <= SYM max_y_strait #
    0.0671 <= 0.0006
Constraint: MAX_Z STRAIT
  SYM strait_z # <= SYM max_z_strait #
    0.0243 <= 0.0007
```

```
-----
-- Complete Constraint Set  --
-----
```

```
Constraint: MAX_MASS
  Parent:    TOOL
  Symbolic:  SYM mass # <= SYM max_m #
  Numeric:   0.23864 <= 0.3
  Value:     1
```

```
Constraint: MIN_MASS
  Parent:    TOOL
  Symbolic:  SYM mass # >= SYM min_m #
  Numeric:   0.23864 >= 0.00123
  Value:     1
```

```
Constraint: MAX_LENGTH
  Parent:    TOOL
  Symbolic:  SYM length # <= SYM max_l #
  Numeric:   80 <= 100
  Value:     1
```

```
Constraint: MIN_LENGTH
  Parent:    TOOL
  Symbolic:  SYM length # >= SYM min_l #
  Numeric:   80 >= 0.00123
  Value:     1
```

```
--More--(38%)
```

```
Constraint: MAX_WIDTH
  Parent:    TOOL
  Symbolic:  SYM width # <= SYM max_w #
  Numeric:   20 <= 30
  Value:     1
```

```
Constraint: MIN_WIDTH
```

```

Parent:      TOOL
Symbolic:    SYM width # >= SYM min_w #
Numeric :    20 >= 0.00123
Value:       1

Constraint:  MAX_HEIGHT
Parent:      TOOL
Symbolic:    SYM height # <= SYM max_h #
Numeric :    20 <= 30
Value:       1

Constraint:  MIN_HEIGHT
Parent:      TOOL
Symbolic:    SYM height # >= SYM min_h #
Numeric :    20 >= 0.00123
--More--(47%)
Value:       1

Constraint:  MAX_XC_LOADING
Parent:      TOOL
Symbolic:    SYM app_l_x # <= SYM height width sigma_c * * #
Numeric :    0 <= 280000
Value:       1

Constraint:  MAX_XT_LOADING
Parent:      TOOL
Symbolic:    SYM app_l_x # <= SYM height width * sigma_y * #
Numeric :    0 <= 280000
Value:       1

Constraint:  MAX_X_STRAIT
Parent:      TOOL
Symbolic:    SYM strait_x # <= SYM max_x_strait #
Numeric :    0.189 <= 0.0005
Value:       0

Constraint:  MAX_Y_STRAIT
Parent:      TOOL
Symbolic:    SYM strait_y # <= SYM max_y_strait #
Numeric :    0.0671 <= 0.0006
Value:       0

Constraint:  MAX_Z_STRAIT
Parent:      TOOL
Symbolic:    SYM strait_z # <= SYM max_z_strait #
Numeric :    0.0243 <= 0.0007
Value:       0

```

##### End Tool Constraint Report #####

```

####Rotational Bearing:
Catalog Number: NU7204
ID Number: 1

```

```

*****
****V I O L A T I O N S****
*****
--More--(65%)
Constraint: MAX_L10
SYM L_n_tmp # <= SYM max_L10 #
3.04862e+09 <= 1e+06

Constraint: COST
SYM cost # <= SYM max_cost #

```

30 <= 20

```
-----  
-- Complete Constraint Set --  
-----  
  
Constraint:  MAX_MASS  
Parent:      ROT_BEARING  
Symbolic:   SYM m # <= SYM max_m #  
Numeric :   0.1 <= 1  
Value:      1  
  
Constraint:  MIN_MASS  
Parent:      ROT_BEARING  
Symbolic:   SYM m # >= SYM min_m #  
Numeric :   0.1 >= 0  
Value:      1  
  
Constraint:  MAX_OD  
Parent:      ROT_BEARING  
Symbolic:   SYM D # <= SYM max_od #  
Numeric :   47 <= 50  
Value:      1  
  
Constraint:  MIN_OD  
Parent:      ROT_BEARING  
Symbolic:   SYM D # >= SYM min_od #  
Numeric :   47 >= 0  
Value:      1  
  
Constraint:  MAX_BD  
Parent:      ROT_BEARING  
Symbolic:   SYM b # <= SYM max_bd #  
Numeric :   20 <= 40  
Value:      1  
  
Constraint:  MIN_BD  
Parent:      ROT_BEARING  
Symbolic:   SYM b # >= SYM min_bd #  
Numeric :   20 >= 0  
Value:      1  
  
Constraint:  MAX_WIDTH  
Parent:      ROT_BEARING  
Symbolic:   SYM w # <= SYM max_w #  
Numeric :   14 <= 30  
Value:      1  
  
Constraint:  MIN_WIDTH  
Parent:      ROT_BEARING  
Symbolic:   SYM w # >= SYM min_w #  
Numeric :   14 >= 0  
Value:      1  
  
Constraint:  MIN_ABEC  
Parent:      ROT_BEARING  
Symbolic:   SYM ABEC_class # >= SYM min_Abec #  
Numeric :   3 >= 1  
Value:      1  
  
--More--(91%)  
Constraint:  LOADING  
Parent:      ROT_BEARING
```

```

Symbolic:  SYM Pr # <= SYM Cr #
Numeric :  1000 <= 14500
Value:     1

Constraint: TEMPERATURE
Parent:     ROT_BEARING
Symbolic:  SYM Top # <= SYM Tmax #
Numeric :  73 <= 180
Value:     1

Constraint: RPM_GREASE
Parent:     ROT_BEARING
Symbolic:  SYM rpm # <= SYM L_S_grease #
Numeric :  10000 <= 17000
Value:     1

Constraint: RPM_OIL
Parent:     ROT_BEARING
Symbolic:  SYM rpm # <= SYM L_S_oil #
Numeric :  10000 <= 23000
Value:     1

Constraint: MAX_L10
Parent:     ROT_BEARING
Symbolic:  SYM L_n_tmp # <= SYM max_L10 #
Numeric :  3.04862e+09 <= 1e+06
Value:     0

Constraint: MIN_L10
Parent:     ROT_BEARING
Symbolic:  SYM L_n_tmp # >= SYM min_L10 #
Numeric :  3.04862e+09 >= 5000
Value:     1

Constraint: MAX_L10_h
Parent:     ROT_BEARING
Symbolic:  SYM Lnh # <= SYM max_L10_h #
Numeric :  5075.96 <= 1e+06
Value:     1

Constraint: MIN_L10_h
Parent:     ROT_BEARING
Symbolic:  SYM Lnh # >= SYM min_L10_h #
Numeric :  5075.96 >= 3000
Value:     1

Constraint: COST
Parent:     ROT_BEARING
Symbolic:  SYM cost # <= SYM max_cost #
Numeric :  30 <= 20
Value:     0

##### End Rot_Bearing Constraint Report #####

```

## **B. Error Motion Modeling**

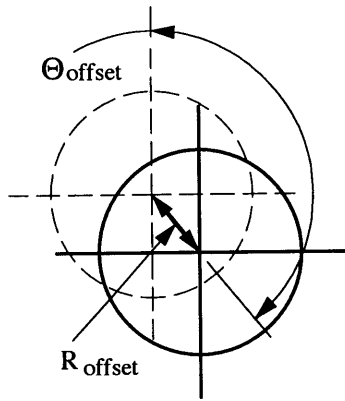
Error motions for machine components are discussed in Chapters 2 and 5. This section presents the methods for representing these motions in PMDA. The mathematical details of these representations, as well as the generalized methods for including different error motion models in PMDA, is presented.

### **B.1 Error Motion Representations in PMDA**

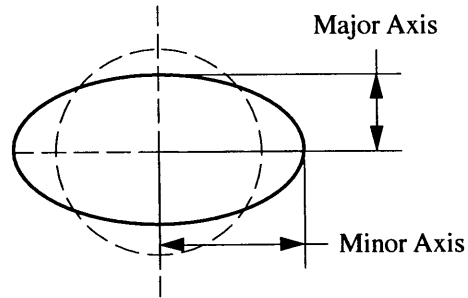
The geometric error motions currently included in PMDA are shown graphically in Figure B.1. These models represent all of the error motions that may be included in the given object; inclusion of other types of error motions requires modification of the PMDA source code. The equations for modeling each of these error motions are shown in Figure B.2. Each error motion is represented as an equation that determines the difference between the ideal (error-free) position of a point on the surface of the object and the position as altered by the error motion. The result is the change in the point location with respect to the ideal point. This concept is illustrated for radial runout in Figure B.3.

### **B.2 Inclusion of Error Motions in PMDA Objects**

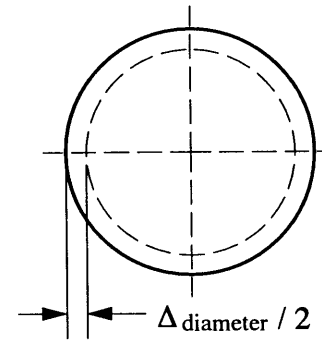
The inclusion of error motions in PMDA require the following steps. First, the equation pointer to house the error motion equation is included in the component object. The equation object itself is created when the component object is created, and the equation used to calculate the error motion is read into this equation object. Equations may either be



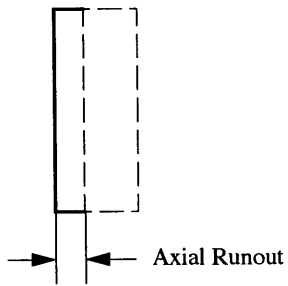
Center Offset & Radial Runout



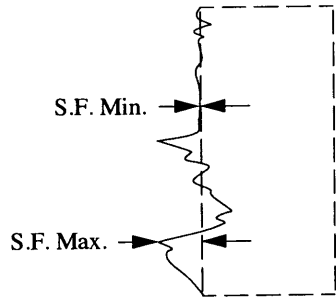
Eccentricity



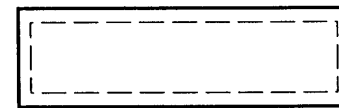
Diameter Variation



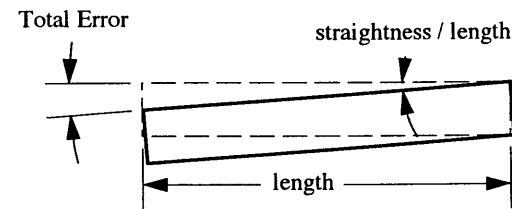
Axial Runout



Surface Finish



Thermal Expansion



Straightness Error

Figure B.1: Geometry of PMDA Error Motions

**Center Offset & Radial Runout**

$$r_{\text{off}} \cos_{\text{TMB}} * 0.5 r_{\text{off}} r_{\text{off}} * \cos_{\text{TMB}} \cos_{\text{TMB}} * * r_{\text{off}} r_{\text{off}} * - d 2.0 / d 2.0 / * + ^ +$$

**where:**

r\_off = radial offset (or radial runout)

COS\_TMB = cosine (object rotation angle minus radial offset angle)

d = original (perfect) diameter

**Eccentricity**

$$0.5 x_{\text{ecc}} x_{\text{ecc}} * y_{\text{ecc}} y_{\text{ecc}} * * y_{\text{ecc}} y_{\text{ecc}} * \cos_{\text{T}} \cos_{\text{T}} * * x_{\text{ecc}} x_{\text{ecc}} * \sin_{\text{T}} \sin_{\text{T}} * * + / ^ +$$

**where:**

x\_ecc = x\_eccentricity (original radius + x\_delta)

y\_ecc = y\_eccentricity (original radius + y\_delta)

COS\_T = cosine of object rotation angle

SIN\_T = sine of object rotation angle

**Diameter Variation**

$$d 2.0 / \Delta d 2.0 / +$$

**where:**

d = original diameter

delta\_d = change in diameter

**Thermal Expansion**

$$\text{length} \alpha D_{\text{TEMP}} * *$$

**where:**

length = original part length

alpha = coefficient of thermal expansion

D\_TEMP = temperature gradient

**Surface Finish**

$$d 2.0 / \text{surf}_{\text{min}} \text{surf}_{\text{max}} + 2.0 / \text{RAND} * +$$

**where:**

d = original diameter (0 for linear surfaces)

surf\_max = maximum surface finish defect height

surf\_min = minimum surface finish defect height

RAND = random number (0 to 1)

**Straightness Error**

$$\text{straight}_x \text{length} *$$

**where:**

straight\_x = x\_straightness (per unit length)

length = total part length

**Axial Runout**

$$\text{axial}_{\text{max}} \text{axial}_{\text{min}} + 2.0 /$$

**where:**

axial\_max = maximum axial runout

axial\_min = minimum axial runout

Figure B.2: PMDA Error Motion Equations

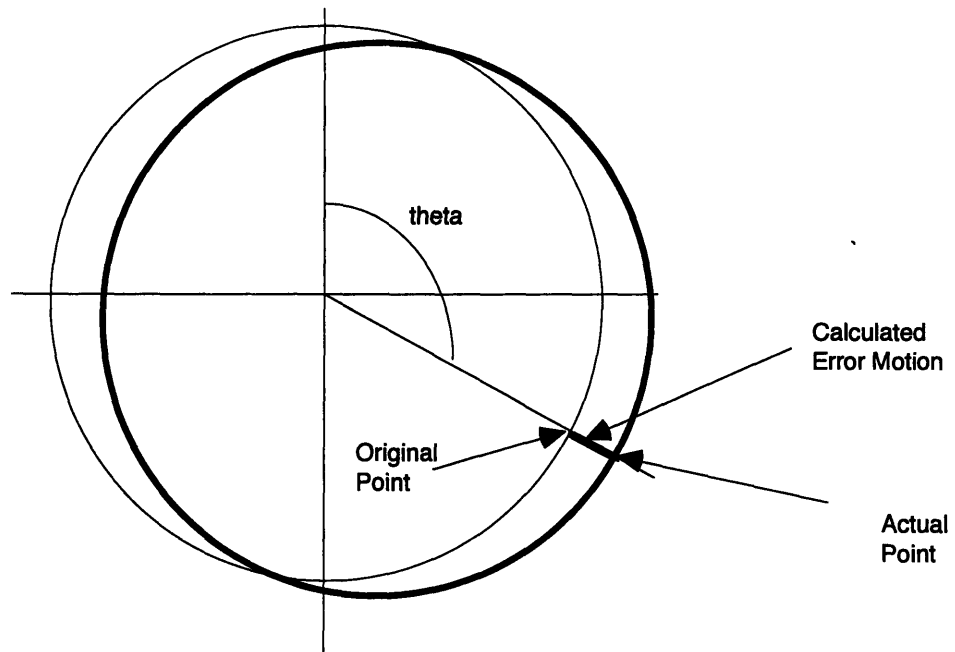


Figure B.3: Sample Error Motion Output

"hard-coded" into the object, or read in from a file, allowing for the modification of equations without recompilation of the code. Once the equation is read in, each term in the equation is replaced by the appropriate variable pointers (see Chapter 4). The equation may then be evaluated to determine the value of the error motion for the given values of the variables included in the equation.

The process for calculating the effect of error motions for a given component object is illustrated in Figure B.4. Each component object containing one or more error motions has a standard function called "add\_errors". Each "add\_errors" function takes a point object as an argument, and contains the necessary calls to member functions inside the given object to calculate the change in the point for the given machine conditions (set by the assembly object) and error motions (determined by the component object). When combined with the HTM-transformation routines (see Appendix C), this method allows for the addition of error motions through a given structural loop, as explained below.

### B.3 Calculation of Error Motions in PMDA Structural Loops

In order to determine the deflected location of a tool, workpiece, or other object in PMDA, the error motions in each component in the structural loop must be calculated into the location of the "ideal" point object (see Chapter 2). As explained above, each PMDA component object contains standard functions for the vector addition of error motions to the

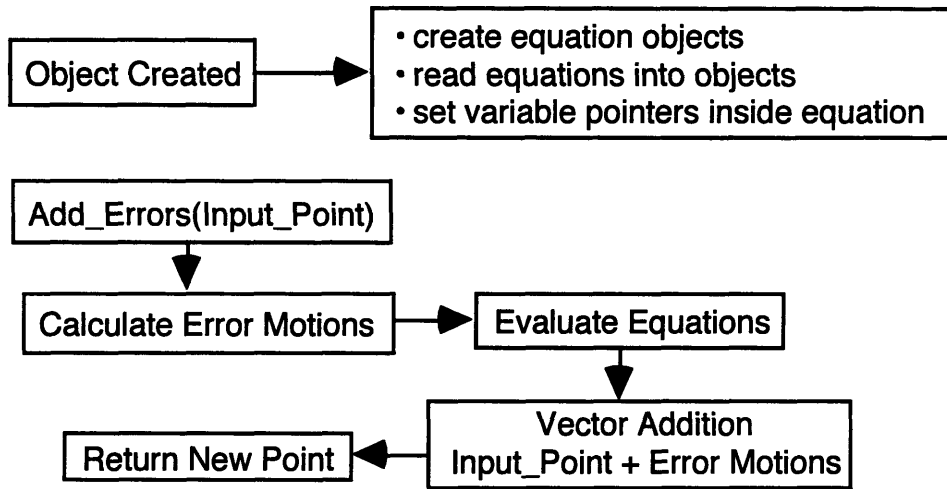


Figure B.4: Process for Calculating Error Motions in PMDA Objects

location of a given point, and for the transformation of any point to the next object in the structural loop. These standard functions allow for the automatic determination of the deflected point location. The procedure is illustrated in Figure B.5.

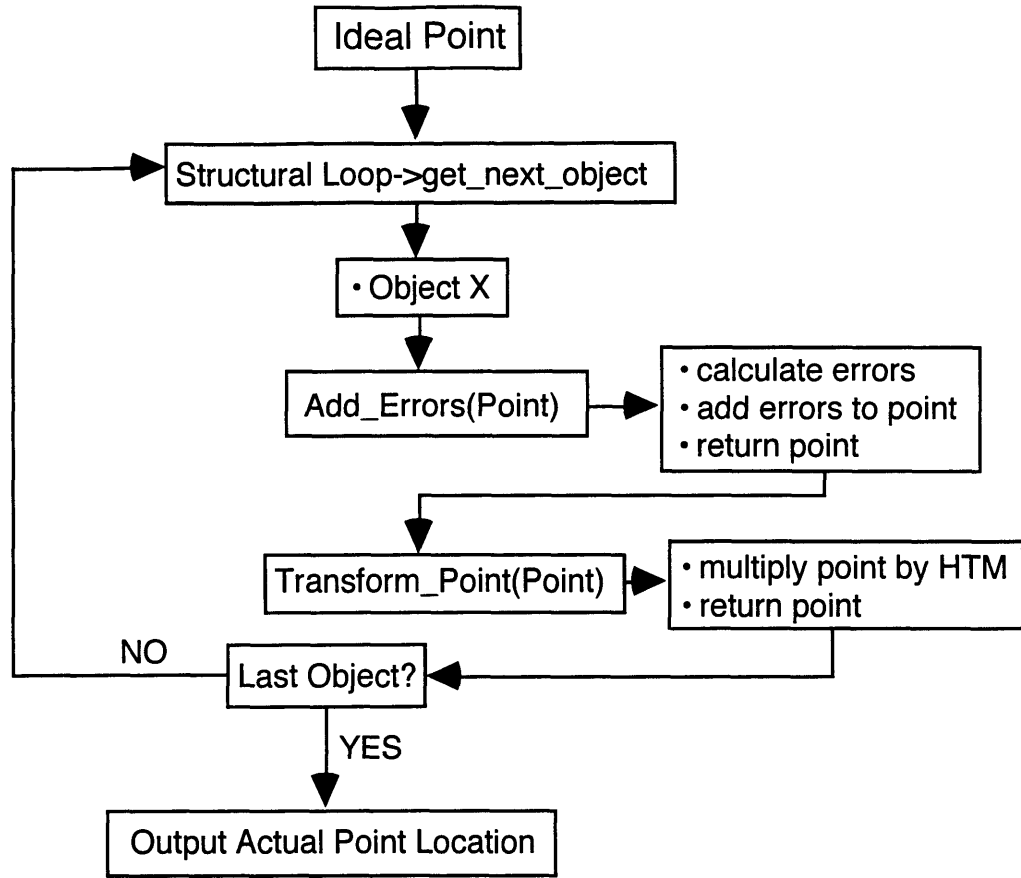


Figure B.5: Error Calculation Procedure

## **C. Kinematic Modeling**

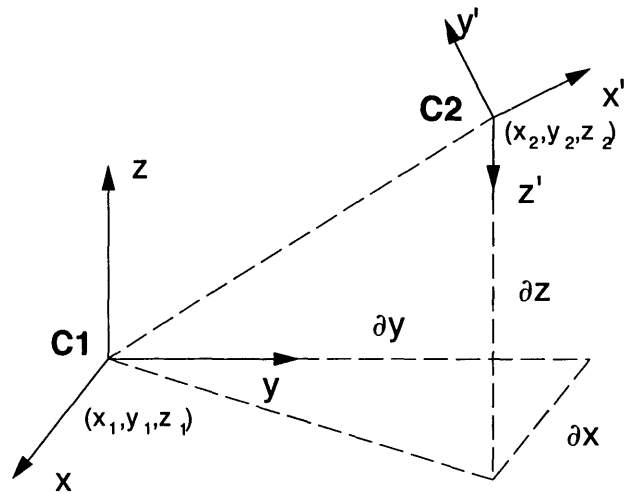
The method of kinematic modeling used in PMDA depends on Homogeneous Transformation Matrices (HTMs) and the specification of structural loops. This appendix presents the matrix methods necessary for kinematic modeling in PMDA, and provides an overview of the methods for storing structural loops in a computational manner.

### **C.1 HTM Calculation**

The calculation of HTMs within PMDA is an automated process requiring the position and orientation of the two coordinate frames of interest. Each of these positions and orientations must be contained within a homogeneous point (a 4X1 matrix described in detail in [Foley, 1990]). The method for calculation of the HTM is shown in Figure C.1. This method is valid for any two coordinate systems, and is incorporated as a function into the Block object described in Chapter 5. This function required knowledge of which block object is the "master" (i.e. which coordinate frame a point will be transformed to); this information is contained within the structural loop described below.

### **C.2 Structural Loops**

A structural loop in PMDA is a series of component objects representing a specific mechanical subassembly. An example of a structural loop is seen in Figure C.2. Currently, PMDA houses structural loops as linked lists in the assembly object (linked lists are described in detail in [Sedgewick, 1992]). Since the objects in the list may be any valid PMDA component object, a second list must be maintained indicating the order and type of components in the list. Figure C.3. provides a detailed description of structural loops in PMDA, as well as an example of the implementation of this structure for the sample loop shown in Figure C.2.



**STEP 1: Place Coordinate Frame Orientations into Matrices**

$$C1 \text{ HTM} = \begin{bmatrix} X1(x) & Y1(x) & Z1(x) & 0.0 \\ X1(y) & Y1(y) & Z1(y) & 0.0 \\ X1(z) & Y1(z) & Z1(z) & 0.0 \\ X1(w) & Y1(w) & Z1(w) & 0.0 \end{bmatrix} \quad C2 \text{ HTM} = \begin{bmatrix} X2(x) & Y2(x) & Z2(x) & 0.0 \\ X2(y) & Y2(y) & Z2(y) & 0.0 \\ X2(z) & Y2(z) & Z2(z) & 0.0 \\ X2(w) & Y2(w) & Z2(w) & 0.0 \end{bmatrix}$$

where:  $Xn(x)$  is the x component of the unit vector for the X axis in terms of the global coordinate frame

**STEP 2: Transpose C2 HTM**

$$C2 \text{ HTM} = \begin{bmatrix} X2(x) & X2(y) & X2(z) & X2(w) \\ Y2(x) & Y2(y) & Y2(z) & Y2(w) \\ Z2(x) & Z2(y) & Z2(z) & Z2(w) \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

**STEP 3: Multiply Matrices**

$$C2 \text{ HTM} = C1 \text{ HTM} * C2 \text{ HTM}$$

**STEP 4: Replace Final Column with Translations**

$$C2 \text{ HTM} = \begin{bmatrix} C2_{0,0} & C2_{0,1} & C2_{0,2} & \partial x \\ C2_{1,0} & C2_{1,1} & C2_{1,2} & \partial y \\ C2_{2,0} & C2_{2,1} & C2_{2,2} & \partial z \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

This is the HTM to transform points from the C2 system to the C1 system

Figure C.1: Procedure to Calculate HTMs between Arbitrary Systems

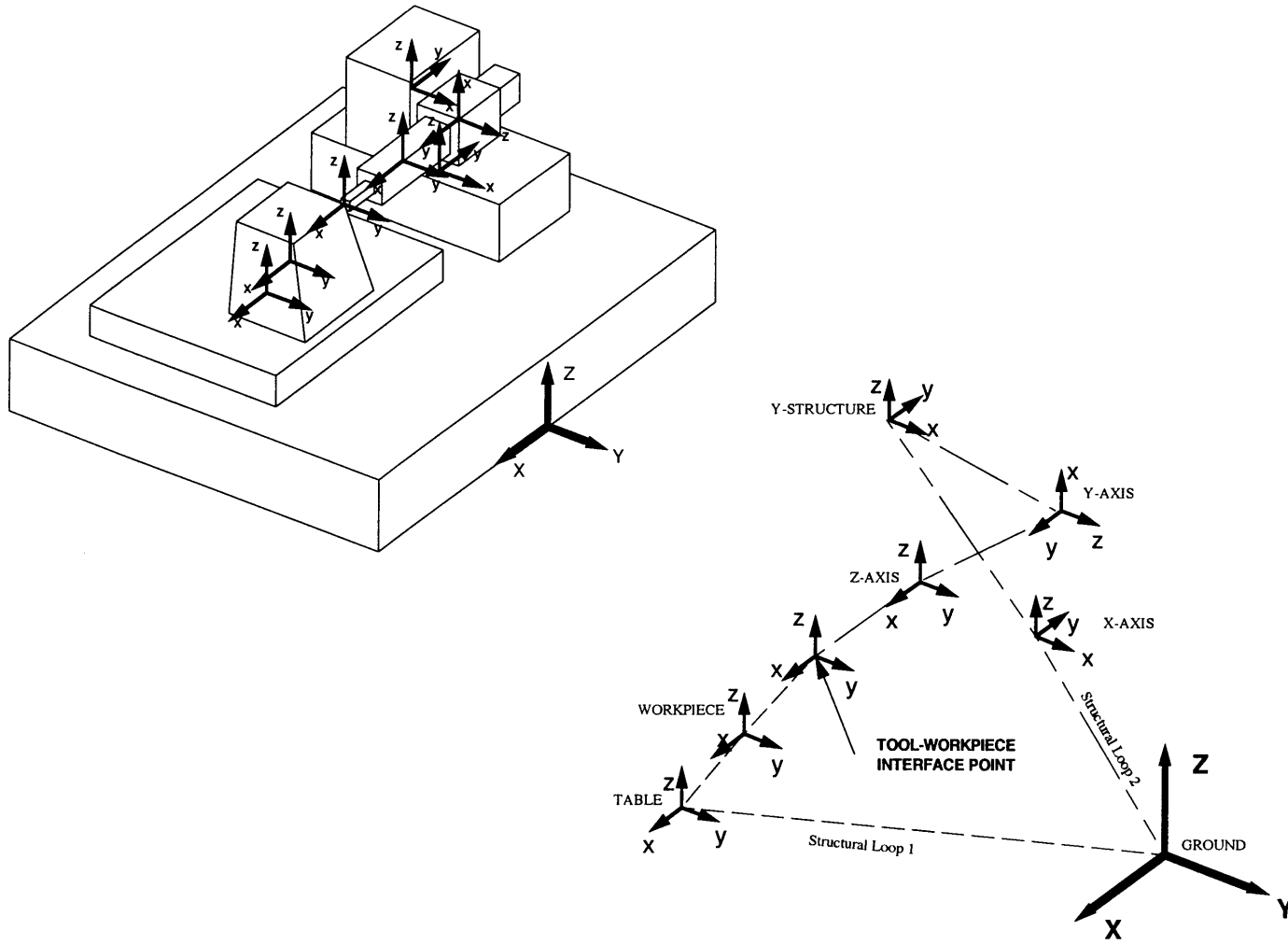
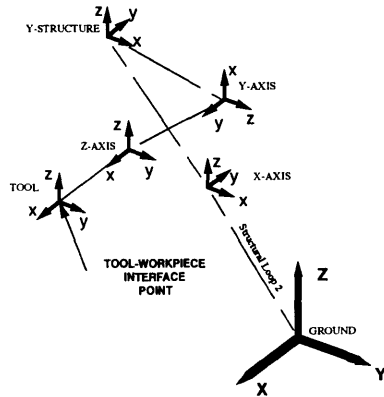


Figure C.2: Example of Structural Loops

**Graphical Representation**

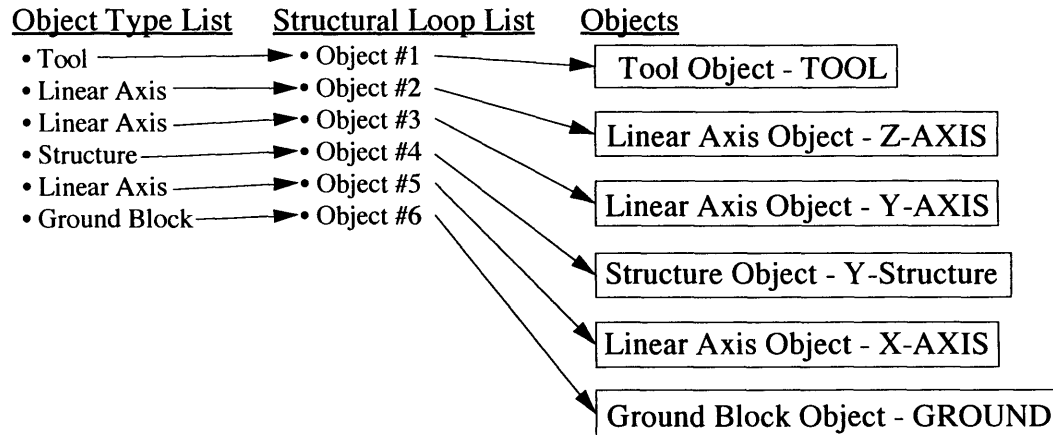


**List Representation**

**Structural Loop 2:**

- Tool
- Z-Axis
- Y-Axis
- Y-Structure
- X-Axis
- Ground

**PMDA Representation**



PMDA structural loop representation consists of two lists: one is a linked list containing pointers to the individual component objects. The second list is an array of character strings containing the type of component object in each position in the structural loop list. This is required by the structure of linked lists in C++.

Figure C.3: Structural Loop Representation in PMDA

## **D. Input and Output Files**

Currently, all user input to PMDA must be accomplished through input files. The output of PMDA is also to a series of files. This section describes those files, and provides samples of each file required or generated by PMDA.

### **D.1. Input Files**

PMDA currently makes use of a number of input files. These input files are divided into three categories: general input, catalog data, and support data.

#### **General Input**

The main input to PMDA is accomplished through the general input file. The general input file for the sample case presented in Chapter 6 is included in Section D.3. The following information is specified by the general input file:

- Catalog Files (the names of all catalog files containing PMDA objects)
- Material Data Files (see Section D.2)
- Custom PMDA Objects (non-catalog specifications)

The input routines are designed to allow for variations in the input file, and any component object in PMDA may be specified as a custom object through the general input file.

#### **Catalog Data**

The catalog data files are text files containing one or more records of component object information. A sample record for a rotary bearing catalog is shown in Figure D.1. Each record represents the part or all of the information required to specify the values contained within a PMDA component object. Currently, any component object may be specified in a catalog file, but all records in a catalog must be of the same object type. All catalog files must also contain the header information seen in the catalog files in section D.3.

```

NTN_Rot_Bearing:
NTN NU204 CYL_ROLLER
  Bore:      0.5 mm
  Outside_Dia: 0.4 mm
  Width:     .25 mm
  C:         16600 N
  Co:        13900 N
  L.S._Oil:  20000 rpm
  L.S._Grease: 16600 rpm
  Mass:      0.111 kg
  ABEC_Class: 3
  Contact_Angle: 30
  Units:     METRIC
  Tolerances:
  Delta_D_M_P: -0.004 mm
  V_D_P:      0.004 mm
  K_I_A:      0.003 mm
  S_D:        0.004 mm
  S_I_A:      0.008 mm
  S_E_A:      0.008 mm
  Delta_B_S:  -0.120 mm
  V_B_S:      0.0025 mm
#

```

Figure D.1: Sample PMDA Catalog Record

## Support Data

Support data files are text files containing information necessary for the support objects and functions in PMDA. Currently, two types of support files are used: material data files and unit conversion files. The unit conversion file is read as a series of DEFINE statements, and allows for common conversion factors to be stored as defined values within the PMDA assembly. The material data file contains the engineering constants necessary to specify the values of the variables contained in a material object; an example record is shown in Figure D.2. There are no limits to the number of material records that may be specified in the material data file.

## D.2. Output Files

The output files for PMDA contain all of information concerning the make-up and performance of a given assembly. Like the input files described above, PMDA output files are divided into three categories: general, error motion, and constraint.

```

Material: Steel
  Designator:      ASTM-A514
  Density:         7.850E-6
  Elastic_Modulus: 210000
  Shear_Modulus:  80000
  Poissons_Ratio: 0.3
  Yield_Stress:   700
  Ultimate_Stress: 830
  Compressive_Stress: 700
  %_Elongation:   15
  Alpha:          .000014
Material

```

Figure D.2: Sample PMDA Material Data File Record

## General Output Files

The general output files contain all of the information necessary to specify the assembly under investigation. PMDA currently generates two general output files. The first contains the structural loops of the machine, as well as each object contained in the assembly. Other general information, such as the assembly name and a list of components generating error motions, is also listed. If the error motions pertaining to a given structural loop are steady (not time dependent), the deflected point pertaining to this loop is also written to this file (the tool loop from the sample case in Chapter 6 is an example of a loop with steady error motions). The second file contains the HTM information for the given assembly. The HTM for each component in each structural loop is written to this file.

## Error Motion Output Files

PMDA generates error motion output files for assemblies with error motions that vary with the operation of the machine (for example, the error trace for a machine spindle will vary with theta). Currently, this type of output is only supported for rotary motion parts. For that reason, the error motion output file consists of four columns: theta, x-position, y-position, and z-position. These positions are listed with respect to the reference coordinate frame. A sample output file is shown in Section D.3.

## **Constraint Output Files**

Information concerning the constraint specification and satisfaction for a given assembly is output to the constraint output file. This file contains a list of all component and interface objects with active constraints, and a listing of all of the constraints for each object. The constraint report for each object consists of a list of violated constraints and a list of all of the remaining constraints. A sample constraint output file is shown in Section D.3.

### **D.3. Sample Input and Output Files**

This section contains the following sample files, in the order listed and as they appear for use in PMDA:

- General Input file
- Rotary Bearing Catalog file
- Shaft Catalog file
- Material data file
- General Output file
- HTM Output File
- Error Motion Output file
- Constraint Report

All of these files are the actual input and output files used for the sample machine spindle presented in Chapter 6.

## GENERAL INPUT FILE

```
// This is a sample input file for the PMDA. All comments must start with
// a double slash, and there MUST be at least one space between the last
// slash and the text. There are no inline comments. Blocks and Interfaces
// may be mixed, but the blocks referenced in an interface must exist before
// the interface is read in. Capitalization must be as shown. White space
// is ignored
Name: SPINDLE
```

```
// The next lines tell PMDA to read catalog files. "TMP_BEAR_CAT" and
// "TMP_SHAFT_CAT" are bearing and shaft catalogs, while "MATERIAL.DAT"
// is a material object data file.
```

```
Catalog_file: NTN_BEARING_CATALOG
```

```
Catalog_file: BERG_SHAFT_CATALOG
```

```
Catalog_file: MATERIAL.DAT
```

```
// Since we don't have a Chuck, Workpiece, or Tool catalog, we use custom
// objects.
```

```
Chuck: CUSTOM 0001
  Diameter: 160.0 mm
  Jaw_Diameter: 70.0 mm
  Height: 60.0 mm
  Material: 2014-T6
  ID: 3
Chuck
```

```
Workpiece: CUSTOM 0010
  Diameter: 60.0 mm
  Length: 80.0 mm
  Material: 2014-T6
  ID: 6
Workpiece
```

```
Tool: CUSTOM 0100
  Length: 80.0 mm
  Height: 20.0 mm
  Width: 20.0 mm
  Material: ASTM-A514
  ID: 9
Tool
```

```
// End of file - all other details of the assembly are dealt with in
// the code and other input files
```

## Rotary Bearing Catalog File

NUM\_RECS: 3  
NTN\_Rot\_Bearing:  
NTN NU204 CYL\_ROLLER  
Bore: 0.5 mm  
Outside\_Dia: 0.4 mm  
Width: .25 mm  
C: 16600 N  
Co: 13900 N  
L.S.\_Oil: 20000 rpm  
L.S.\_Grease: 16600 rpm  
Mass: 0.111 kg  
ABEC\_Class: 3  
Contact\_Angle: 30  
Units: METRIC  
Tolerances:  
Delta\_D\_M\_P: -0.004 mm  
V\_D\_P: 0.004 mm  
K\_I\_A: 0.003 mm  
S\_D: 0.004 mm  
S\_I\_A: 0.008 mm  
S\_E\_A: 0.008 mm  
Delta\_B\_S: -0.120 mm  
V\_B\_S: 0.0025 mm  
#

NTN NU7204 Ang\_Contact\_Ball  
Bore: 20.0 mm  
Outside\_Dia: 47.0 mm  
Width: 14.0 mm  
C: 14500 N  
Co: 8400 N  
L.S.\_Oil: 23000 rpm  
L.S.\_Grease: 17000 rpm  
Mass: 0.100 kg  
ABEC\_Class: 3  
Contact\_Angle: 30  
Units: METRIC  
Tolerances:  
Delta\_D\_M\_P: -0.004 mm  
V\_D\_P: 0.004 mm  
K\_I\_A: 0.003 mm  
S\_D: 0.004 mm  
S\_I\_A: 0.008 mm  
S\_E\_A: 0.008 mm  
Delta\_B\_S: -0.120 mm  
V\_B\_S: 0.0025 mm  
#

NTN NU2307 CYL\_ROLLER  
Bore: 2.0 mm  
Outside\_Dia: 2.4 mm  
Width: .5 mm  
C: 64500 N  
Co: 65500 N  
L.S.\_Oil: 18000 rpm  
L.S.\_Grease: 15000 rpm  
Mass: 0.122 kg  
Units: METRIC  
Special:  
#

## Shaft Catalog File

NUM\_RECS: 5

Shaft:

Shaft: BERG LMS-46-12H

Diameter: 1.0

Length: 12.0

Material: ASTM-A514

EccX: .55

EccY: .44

Offset: .78

DiamError: .01

Units: ENGLISH

Shaft

Shaft: BERG LMS-46-12M

Diameter: 1.0

Length: 12.0

Material: ASTM-A514

EccX: .55

EccY: .45

Offset: 1.0

DiamError: .02

Units: ENGLISH

Shaft

Shaft: BERG LMS-46-12L

Diameter: 40.0

Length: 240.0

Material: ASTM-A514

EccX: .50

EccY: .40

Offset: 1.2

DiamError: .04

Units: METRIC

Shaft

Shaft: BERG LMS-56-12

Diameter: .5

Length: 12.0

Material: ASTM-A514

Units: ENGLISH

Shaft

Shaft: BERG LMS-72-36

Diameter: 2.0

Length: 36.0

Material: ASTM-A514

Units: ENGLISH

Shaft

## Material Data File

NUM\_RECS: 2

Material:

Material: Aluminum

Designator: 2014-T6  
Density: 2.800E-6  
Elastic\_Modulus: 73000  
Shear\_Modulus: 28000  
Poissons\_Ratio: 0.33  
Yield\_Stress: 410  
Ultimate\_Stress: 480  
Compressive\_Stress: 410  
%\_Elongation: 13  
Alpha: .000023

Material

Material: Steel

Designator: ASTM-A514  
Density: 7.850E-6  
Elastic\_Modulus: 210000  
Shear\_Modulus: 80000  
Poissons\_Ratio: 0.3  
Yield\_Stress: 700  
Ultimate\_Stress: 830  
Compressive\_Stress: 700  
%\_Elongation: 15  
Alpha: .000014

Material

## General Output File

```
*****  
** The Precision Machine Design Assistant **  
** Version 1.0 - Copyright 1997 M.I.T. **  
** Code written by: Brad Homann **  
** Anna Thornton **  
*****  
*****
```

Assembly Name: SPINDLE

\*\*\*\*Structural Loops\*\*  
\*\*Tool Loop:

Tool: CUSTOM 0100  
Length: 80 mm  
Width: 20 mm  
Height: 20 mm  
Material: ASTM-A514  
ID: 9  
Tool  
Location: (-240,110,80,1)

\*\*Workpiece Loop:

Workpiece: CUSTOM 0010  
Diameter: 60 mm  
Length: 80 mm  
Material: 2014-T6  
ID: 6  
Workpiece  
Location: (-220,180,80,1)

Chuck: CUSTOM 0001  
Diameter: 160 mm  
Jaw\_Diameter: 70 mm  
Height: 60 mm  
Material: 2014-T6  
ID: 3  
Chuck  
Location: (-150,180,80,1)

Shaft: BERG LMS-46-12L  
Diameter: 40  
Length: 240  
Material: Steel  
Units: METRIC  
ID: 2  
Shaft  
Location: (0,180,80,1)

NTN NU7204 Ang\_Contact\_Ball  
Bore: 20 mm  
Outside\_Dia: 47 mm  
Width: 14 mm  
C: 14500 N  
Co: 8400 N  
L.S.\_Oil: 23000 rpm  
L.S.\_Grease: 17000 rpm  
Mass: 0.1 kg  
ABEC\_Class: 3 Units: METRIC

Special:  
ID: 1  
# Location: (-90,180,80,1)

\*\*\*\*Active Error Motions\*\*\*\*

- \* Chuck
- \* Shaft
- \* Bearing
- \* Tool

IDEAL Tool-Workpiece Interaction Location: (-240,150,80,1)  
Actual ToolPoint Location: (-240.00756,150.00914,80,1)  
Actual WorkpiecePoint Locations written to file: PMDA\_ERRORS.OUT  
Constraint Information written to file: PMDA\_CONSTRAINTS.OUT

## HTM Output File

```
*****  
**           The Precision Machine Design Assistant           **  
**           Homogeneous Transformation Matrices             **  
**                   for                                     **  
**                   **                                     **  
**           ASSEMBLY: SPINDLE                               **  
*****
```

TOOL HTM:

```
Matrix :  
| 0 -1 0 -240 |  
| 1 0 0 110  |  
| 0 0 1 80   |  
| 0 0 0 1    |  
END TOOL HTM
```

WORKPIECE HTM:

```
Matrix :  
| 1 0 0 70 |  
| 0 1 0 0 |  
| 0 0 1 0 |  
| 0 0 0 1 |  
END WORKPIECE HTM
```

CHUCK HTM:

```
Matrix :  
| 1 0 0 150 |  
| 0 1 0 0   |  
| 0 0 1 0   |  
| 0 0 0 1   |  
END CHUCK HTM
```

SHAFT HTM:

```
Matrix :  
| 1 0 0 -90 |  
| 0 1 0 0   |  
| 0 0 1 0   |  
| 0 0 0 1   |  
END SHAFT HTM
```

ROT\_BEAR HTM:

```
Matrix :  
| -1 0 0 -90 |  
| 0 -1 0 180 |  
| 0 0 1 80   |  
| 0 0 0 1   |  
END ROT_BEAR HTM
```

## Constraint Output File

```
*****
**          PMDA Constraint Violation Report          **
**                                                    **
**          for Assembly: SPINDLE                    **
*****
```

```
###Tool      :
  Catalog Number: 0100
  ID Number    : 9
```

```
*****
****V I O L A T I O N S****
*****
```

```
Constraint: MAX_Y_STRAIT
            SYM strait_y # <= SYM max_y_strait #
            0.000671 <= 0.0006
```

```
-----
-- Complete Constraint Set --
-----
```

```
Constraint: MAX_MASS
  Parent:   TOOL
  Symbolic: SYM mass # <= SYM max_m #
  Numeric  : 0.23864 <= 0.3
  Value:    1
```

```
Constraint: MIN_MASS
  Parent:   TOOL
  Symbolic: SYM mass # >= SYM min_m #
  Numeric  : 0.23864 >= 0.00123
  Value:    1
```

```
Constraint: MAX_LENGTH
  Parent:   TOOL
  Symbolic: SYM length # <= SYM max_l #
  Numeric  : 80 <= 100
  Value:    1
```

```
Constraint: MIN_LENGTH
  Parent:   TOOL
  Symbolic: SYM length # >= SYM min_l #
  Numeric  : 80 >= 0.00123
  Value:    1
```

```
Constraint: MAX_WIDTH
  Parent:   TOOL
  Symbolic: SYM width # <= SYM max_w #
  Numeric  : 20 <= 30
  Value:    1
```

```
--More--(39%)
```

```
Constraint: MIN_WIDTH
  Parent:   TOOL
  Symbolic: SYM width # >= SYM min_w #
  Numeric  : 20 >= 0.00123
  Value:    1
```

```
Constraint: MAX_HEIGHT
  Parent:   TOOL
```

```

Symbolic:  SYM height # <= SYM max_h #
Numeric :  20 <= 30
Value:     1

Constraint: MIN_HEIGHT
Parent:    TOOL
Symbolic:  SYM height # >= SYM min_h #
Numeric :  20 >= 0.00123
Value:     1

Constraint: MAX_XC_LOADING
Parent:    TOOL
Symbolic:  SYM app_l_x # <= SYM height width sigma_c * * #
Numeric :  0 <= 280000
Value:     1

Constraint: MAX_XT_LOADING
Parent:    TOOL
Symbolic:  SYM app_l_x # <= SYM height width * sigma_y * #
Numeric :  0 <= 280000
Value:     1

Constraint: MAX_X_STRAIT
Parent:    TOOL
Symbolic:  SYM strait_x # <= SYM max_x_strait #
Numeric :  0.000189 <= 0.0005
Value:     1

Constraint: MAX_Y_STRAIT
Parent:    TOOL
Symbolic:  SYM strait_y # <= SYM max_y_strait #
Numeric :  0.000671 <= 0.0006
Value:     0

Constraint: MAX_Z_STRAIT
Parent:    TOOL
Symbolic:  SYM strait_z # <= SYM max_z_strait #
Numeric :  0.000243 <= 0.0007
Value:     1

```

##### End Tool Constraint Report #####

```

###Rotational Bearing:
Catalog Number: NU7204
ID Number: 1

```

```

*****
****V I O L A T I O N S****
*****
Constraint: MAX_L10
SYM L_n_tmp # <= SYM max_L10 #
3.04862e+09 <= 1e+06
Constraint: COST
SYM cost # <= SYM max_cost #
30 <= 20

```

--More--(66%)

```

-----
-- Complete Constraint Set --
-----

```

```

Constraint:  MAX_MASS

```

```

Parent:      ROT_BEARING
Symbolic:    SYM m # <= SYM max_m #
Numeric :    0.1 <= 1
Value:       1

Constraint:  MIN_MASS
Parent:      ROT_BEARING
Symbolic:    SYM m # >= SYM min_m #
Numeric :    0.1 >= 0
Value:       1

Constraint:  MAX_OD
Parent:      ROT_BEARING
Symbolic:    SYM D # <= SYM max_od #
Numeric :    47 <= 50
Value:       1

Constraint:  MIN_OD
Parent:      ROT_BEARING
Symbolic:    SYM D # >= SYM min_od #
Numeric :    47 >= 0
Value:       1

Constraint:  MAX_BD
Parent:      ROT_BEARING
Symbolic:    SYM b # <= SYM max_bd #
Numeric :    20 <= 40
Value:       1

Constraint:  MIN_BD
Parent:      ROT_BEARING
Symbolic:    SYM b # >= SYM min_bd #
Numeric :    20 >= 0
Value:       1

Constraint:  MAX_WIDTH
Parent:      ROT_BEARING
Symbolic:    SYM w # <= SYM max_w #
Numeric :    14 <= 30
Value:       1

Constraint:  MIN_WIDTH
Parent:      ROT_BEARING
Symbolic:    SYM w # >= SYM min_w #
Numeric :    14 >= 0
Value:       1

Constraint:  MIN_ABEC
Parent:      ROT_BEARING
Symbolic:    SYM ABEC_class # >= SYM min_Abec #
Numeric :    3 >= 1
Value:       1

Constraint:  LOADING
Parent:      ROT_BEARING
Symbolic:    SYM Pr # <= SYM Cr #
Numeric :    1000 <= 14500
Value:       1

--More--(93%)
Constraint:  TEMPERATURE
Parent:      ROT_BEARING
Symbolic:    SYM Top # <= SYM Tmax #
Numeric :    73 <= 180

```

```

Value: 1

Constraint: RPM_GREASE
Parent: ROT_BEARING
Symbolic: SYM rpm # <= SYM L_S_grease #
Numeric : 10000 <= 17000
Value: 1

Constraint: RPM_OIL
Parent: ROT_BEARING
Symbolic: SYM rpm # <= SYM L_S_oil #
Numeric : 10000 <= 23000
Value: 1

Constraint: MAX_L10
Parent: ROT_BEARING
Symbolic: SYM L_n_tmp # <= SYM max_L10 #
Numeric : 3.04862e+09 <= 1e+06
Value: 0

Constraint: MIN_L10
Parent: ROT_BEARING
Symbolic: SYM L_n_tmp # >= SYM min_L10 #
Numeric : 3.04862e+09 >= 5000
Value: 1

Constraint: MAX_L10_h
Parent: ROT_BEARING
Symbolic: SYM Lnh # <= SYM max_L10_h #
Numeric : 5075.96 <= 1e+06
Value: 1

Constraint: MIN_L10_h
Parent: ROT_BEARING
Symbolic: SYM Lnh # >= SYM min_L10_h #
Numeric : 5075.96 >= 3000
Value: 1

Constraint: COST
Parent: ROT_BEARING
Symbolic: SYM cost # <= SYM max_cost #
Numeric : 30 <= 20
Value: 0

##### End Rot_Bearing Constraint Report #####

```



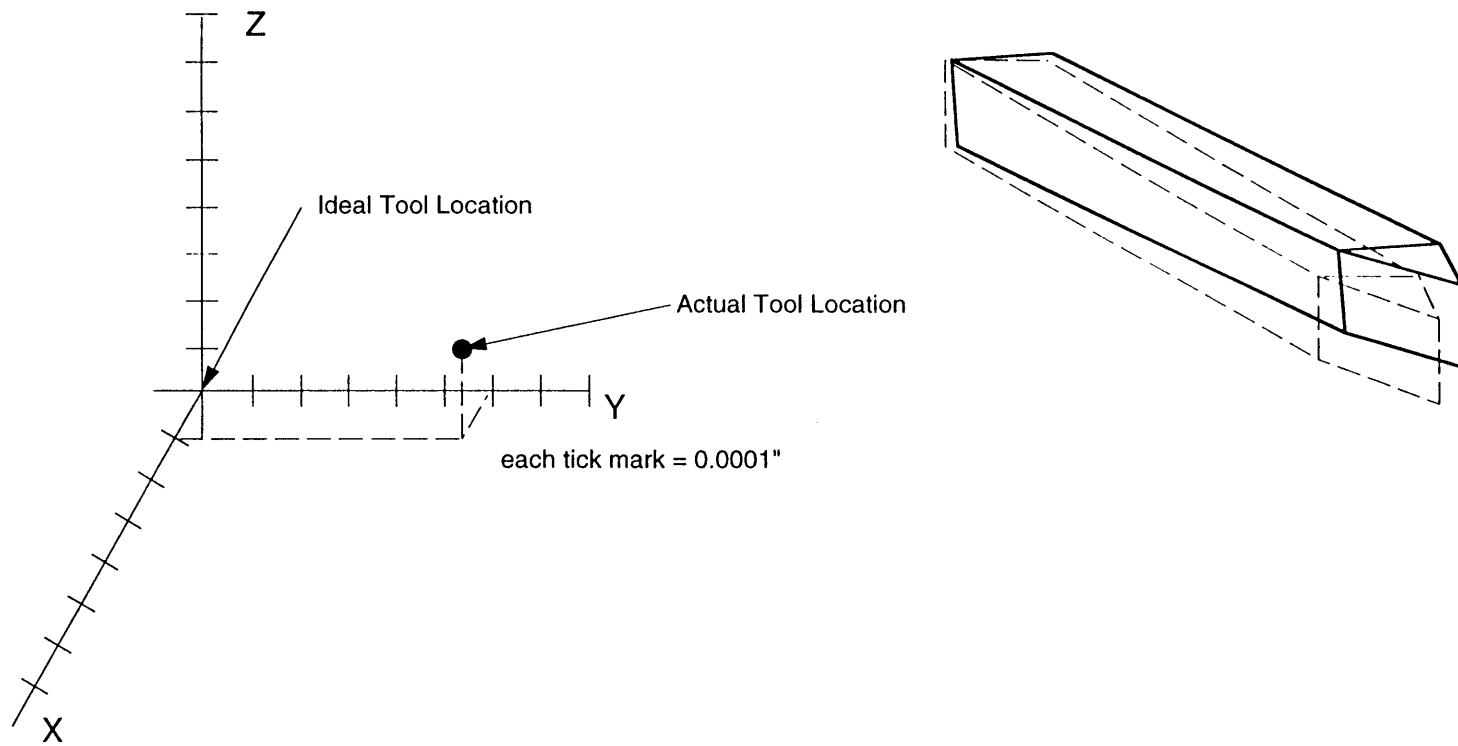


Figure D.3a: Tool point Displacement Errors for the Simple Machine Spindle

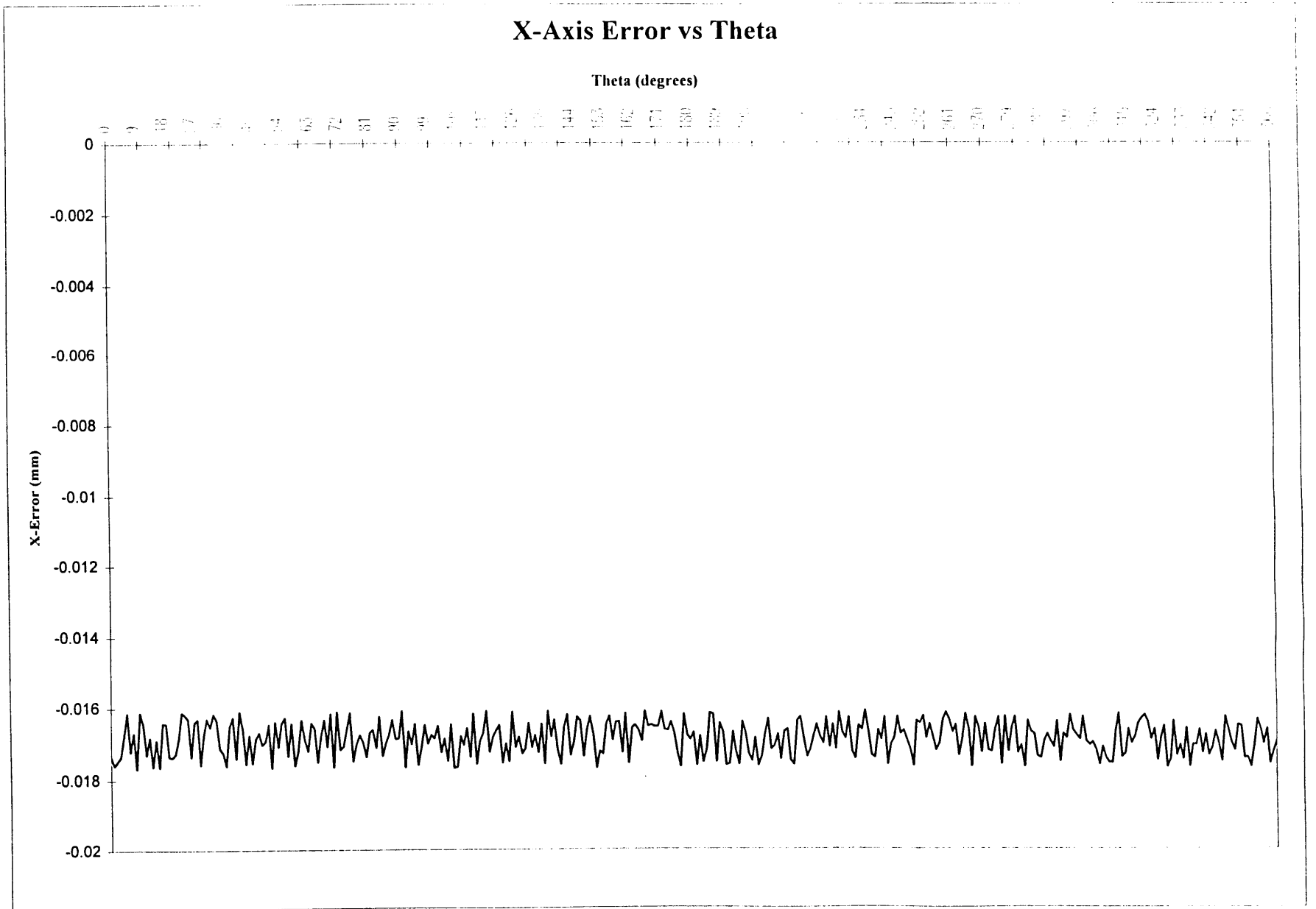


Figure D.3b: Axial Displacement Errors for the Simple Machine Spindle

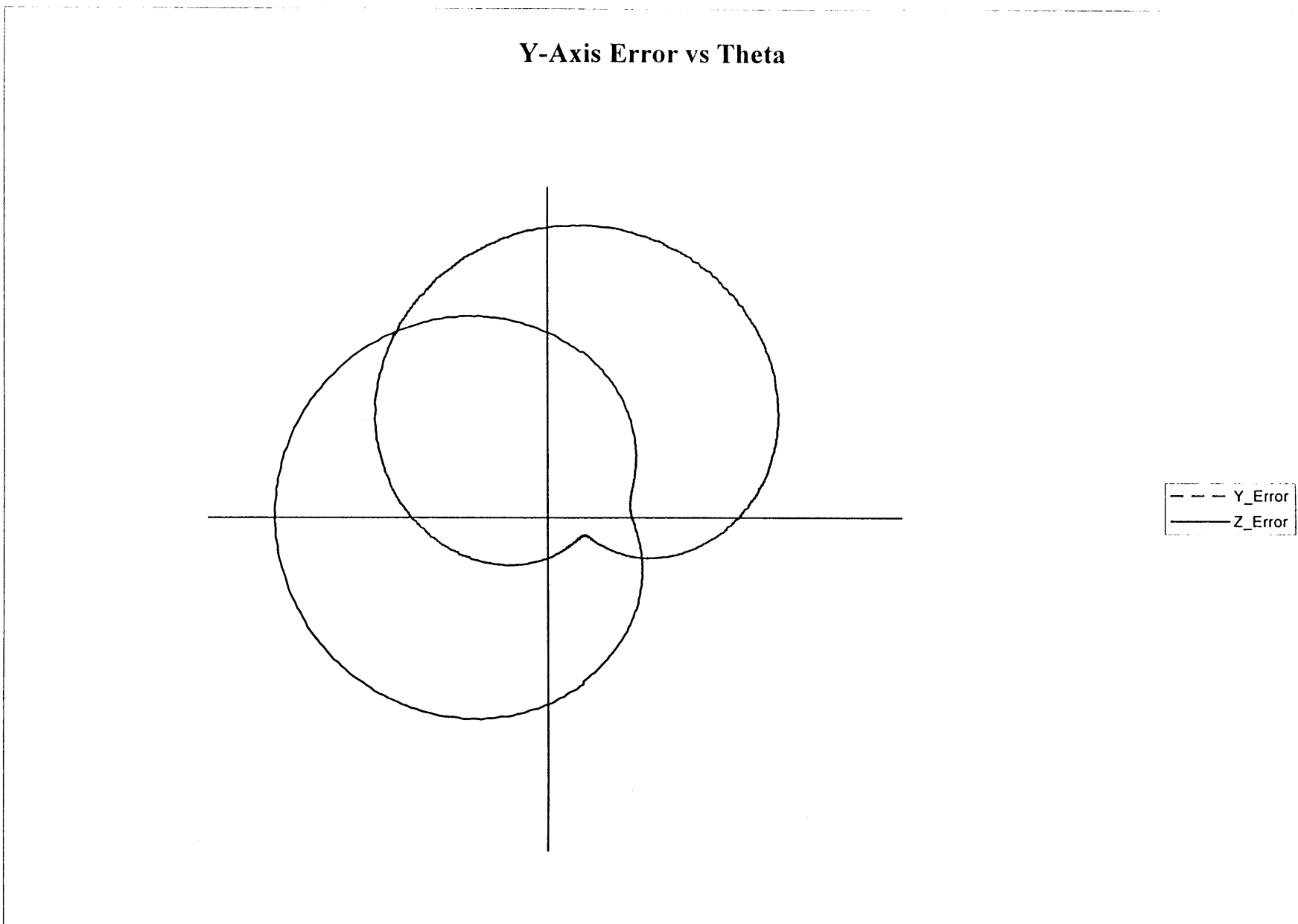


Figure D.3c: Radial Displacement Errors for the Simple Machine Spindle