AN INFORMATION APPROACH TO PARTS MATING

by

Sergio N. Simunovic S.

Universidad de Chile
(1970)

S.M., Massachusetts Institute of Technology
(1976)

MECH.E., Massachusetts Institute of Technology
(1976)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

April, 1979

Signature of Author.....
Department of Mechanical Engineering, April 26, 1979

Certified by.....
Thesis Supervisor

Accepted by.....
Chairman, Department Committee on Graduate Students

AN INFORMATION APPROACH TO PARTS MATING

by

SERGIO N. SIMUNOVIC

Submitted to the Department of Mechanical Engineering on April 5, 1979 in partial fulfillment of the requirements for the degree of Doctor of Science.

ABSTRACT

This thesis addresses the interaction of mechanical parts during the assembly process, the nature of the information available during this interaction, and the form and use of this information in aiding the parts mating. The work is part of the general work being done in the development of programmable industrial assembly. It describes how to determine the performance characteristics necessary for a positioning device designed for assembly.

Four chapters described: (1) the assembly process as a positioning problem and the study of parts mating as a means of solving for this positioning problem; (2) the transition from transporting the parts to positioning the parts for starting the assembly; (3) the actual parts assembly or mating; and (4) the implications of the parts mating study on the configuration of an assembly system.

Thesis Supervisor: Richard S. Sidell, Associate Professor, Department of Mechanical Engineering

ACKNOWLEDGEMENTS

I wish to thank my thesis advisor Richard Sidell and the members of my thesis committee, Daniel E. Whitney, Karl Hedrick and especially Fred Schweppe, for their encouragement and for allowing me to approach this work in the unorthodox manner I felt would be most appropriate. I thank the IOD group members at the C.S. Draper Laboratory and also Linda Martinez, who cheerfully typed the final manuscript and the many drafts which preceded it. Special thanks to Anne Simunovic for her support and good humor throughout the thesis process, and to Samuel H. Drake for his invaluable insights and constant prodding.

This work was partially supported by the Charles S. Draper Laboratory, Inc. under National Science Foundation Grant Number APR 74-18173 A03.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER I

WORK STATEMENT

1. INTRODUCTION

This work addresses the interaction of mechanical parts during the assembly process, the nature of the information available during this interaction, and the form and use of this information in aiding the parts mating. This work is part of the general work being done at the Charles Stark Draper Laboratory in the development of the programmable industrial assembly system concept. The output of this work is twofold: a technical feasibility study for a particular family of methods for solving the parts mating, and a set of design tools and methods to be used in the implementation of such a system.

This chapter provides a brief overview of the current methods used for mechanical assembly, defines the concept of programmability and adaptability in an industrial assembly system. The third section of this chapter discusses the positioning problem in mechanical assembly and defines the scope of the parts mating study on this subject. Section Four discusses sensory systems in mechanical assembly and how they influence the parts mating study. Finally the last section delineates the content of the successive chapters.

2. BACKGROUND

Present industrial assembly is performed in either of two ways: the intensive use of labor engaged in manual assembly or a highly automated special-purpose system referred to here as fixed automation assembly. The manual assembly method, most commonly used in industry, takes advantage of the great dexterity of humans, of their sensing and judgment capabilities, and of their ability to learn different

tasks. Humans accommodate for variations in the parts, will inspect for defects in parts and assemblies, and can perform fairly complicated assembly tasks. This characteristic of humans makes this method ideal for assembling one or more products with relatively small individual production volume.

Fixed automated assembly is constructed of a series of single task stations to which the parts are shuttled by a transfer device. The very simple task performed by each workstation combined with accurate positioning of the parts is the basis for this method of assembly. Fixed automation equipment is designed to assemble one product throughout its production life. The special engineering of the system, coupled with long debugging and setup times, which often lead to changes in the design of the parts to be assembled, result in high-capital-cost equipment justifiable only for high production and long lifetime products.

A goal of the programmable system is to assemble different products with easy changeovers from one product to the next and allow for modifications in the product design throughout its lifetime. Such a system is economical for products with small-to-medium-size production volumes where the capital investment cost is shared by many products.

Olivetti, Kawasaki Heavy Industries, Unimation, and others have developed programmable assembly systems that use very basic programmable positioning devices (similar in concept to numerical control machines) combined with special-purpose tooling and jigging. This approach suffers from problems characteristic of fixed automation assembly. The inability of the system to overcome small variations in the position of the parts or in their geometry requires individual tuning, special tooling for every task, and special-purpose sensors. The highly structured environment required by this type of system cancels the usefulness of the programmable positioning device.

An analysis of the equipment cost for fixed automation assembly, Reference I-1, shows that special engineering and special tooling can account for 30-50% of the assembly system cost and that setup time and debugging cost anywhere from 15-25% of the fixed automation equipment cost. In manual assembly, these special-

purpose costs are negligible because of the great adaptability of humans. A practical programmable assembly system has to use less special-purpose engineering than the fixed automation system: (1) it has to be able to work without elaborate setups and debugging, especially in model and product changeovers, and (2) it has to have a certain degree of adaptability to allow it to correct for changes in the parts and in the assembly work environment.

Two major problem areas have to be overcome in the conceptual design of a programmable adaptable assembly system, namely the feeding-gross orientation problems and inspection problems, and the actual assembly of the part or parts mating problem. The combination of sensors with some information processing capabilities has been used in an attempt to develop a more adaptable system, capable of correcting for the lack of structure in the industrial working environment. Significant progress has been made in the fields of machine vision for inspection, parts acquisition and gross positioning of the parts in space (Reference I-2, Reference I-3). In the area of parts mating, work has been done through the implementation of force and touch sensing combined with fine positioning devices. Most of this work has concentrated on emulating humans performing the assembly and very little has been done on the actual mechanics of parts mating (Reference I-4, Reference I-5). Exceptions to this have been the work done by Laktionov, Andreev, Gusev, McCallon (Reference I-6, I-7, I-8) that were interested in the assembly of large sized parts where human sensory perception is of little help, and the work done at Draper Lab (Reference I-9) of which the present work is a part. The methods developed in Reference I-4 and Reference I-5 are further discussed in Chapter II.

This work will concentrate on the parts mating issues of the Programmable Assembly system. The approach taken is to analyze the mating of mechanical parts, the information available from this mating, and the nature of the corrections needed for achieving the assembly.

3. POSITIONING ISSUES

Some Definitions: The following definitions are taken from Van Nostrand Scientific Encyclopedia, fifth edition. These concepts, defined in that source in relation to properties of measurements, are expanded in the present work so as to cover properties of controlled actions.

ACCURACY. In terms of instruments and scientific measuring systems, accuracy may be defined as the conformity of an indicated value to an accepted standard value, or true value. Accuracy is usually measured in terms of inaccuracy and expressed as accuracy. As a performance specification, accuracy should be assumed to mean reference accuracy unless otherwise stated. Reference accuracy may be defined as a number or quantity which defines the limit that errors will not exceed when the device is used under reference operating conditions.

RESOLUTION. A term used in a number of specific cases in science to denote the process of separating closely related forms or entities or the degree to which they can be discriminated.

REPEATABILITY. With reference to industrial and scientific instruments, the Scientific Apparatus Makers Association defines repeatability as the closeness of agreement among a number of consecutive measurements of the output for the same value of the input under the same operating conditions, approaching from the same direction for full range traverses. Repeatability is usually measured as a nonrepeatability and expressed as repeatability in percent of span. Repeatability does not include hysteresis.

Mechanical Assembly, A Positioning Problem: The assembly process is strictly a positioning problem. Complete knowledge of the parts and ideal positioning devices would, at least in principle, make the assembly task a trivial matter. The imperfections of the real world are materialized as position errors in the physical assembly systems; these errors translate into an error in the relative position between the parts at mating; the resulting error in the relative position between the parts at mating will cause interference between the geometry of the parts, and

therefore not allow the parts to be assembled. Two approahces are possible for making the assembly possible:

- the errors at their sources, and thus the error in the relative position between the parts, are made small. If so, the geometry of the parts will not interfere and the assembly is possible. This is the approach taken in Fixed Automation Assembly;
- correct for relative position error during the mating process. This is done by humans in manual assembly.

The conceptual approach taken in adaptable assembly will be a combination of the two extreme approaches mentioned above. The system would be capable of correcting for small errors in the relative position between the parts during assembly, thus requiring a less rigidly structured world than that required by Fixed Automation.

Assembly System Configuration: For the purpose of discussion it will be assumed that the Programmable Assembly System will have a configuration of the form shown in Figure I-1. Its main elements are:

- work space, where parts are put together;
- feeders, pallets, sorter, etc., capable of delivering the parts to the assembly work space with a certain degree of gross orientation and retrieving the assemblies from the workspace;
- positioning device(s) capable of transporting and assembling the parts within the workspace ;
- information gathering devices such as: force sensors, position sensors, optical sensor, touch sensors, etc.;
- processor capable of combining a priori information with incoming information from sensors to generate output commands to position devices, feeders and jigs.

Assembly Path: We will denote by assembly path that trajectory required in the space of the relative position between the parts, to drive the parts from their initial feeding position to their assembled state. In the case of two solid bodies, this path represents a six-dimensional trajectory; for deformable bodies, linkages, or more than two bodies simultaneous assembly, this trajectory could involve any number of dimensions above the six dimensions for binary solid bodies assembly. This work will concentrate only on the binary solid bodies assembly.

The assembly path can be divided into gross motion, for the trajectory involved in parts fetching, and fine motion for the actual assembly trajectory.

The fine motion trajectories between the mating parts are characterized by having one of the six generalized rigid body motion degrees of freedom independent; the rest of the degrees of freedom are dependent on the independent one within narrow tolerances. The narrowness of the allowable tolerance for the fine motion path is determined by the geometry of the parts. For parts with axially symmetric mating surfaces one of the degrees of freedom is indifferent. The realization of the fine motion trajectory is equivalent to maintaining the dependent degrees of freedom within their tolerances.

Obstacle avoidance is the only position constraint to the gross motion trajectory and does not constitute an engineering problem in terms of positioning. Problems associated with the kinematics necessary for gross motion execution and time spent in gross motions are serious issues to be considered, but will not be dealt with in this work.

This part mating work will concentrate on the transition from gross to fine motion region. Other problems, such as parts feeding, grasping, jigging, manipulator design, etc.,will be considered only in respect to their contribution to the error in the relative position between the parts at mating.

Errors on Positioning: The difficulty in positioning for assembly is caused by a combination of positioning problems originating in the position device itself and positioning problems inherent in the geometry of the parts.

The positioning device must be capable of both performing the gross motion

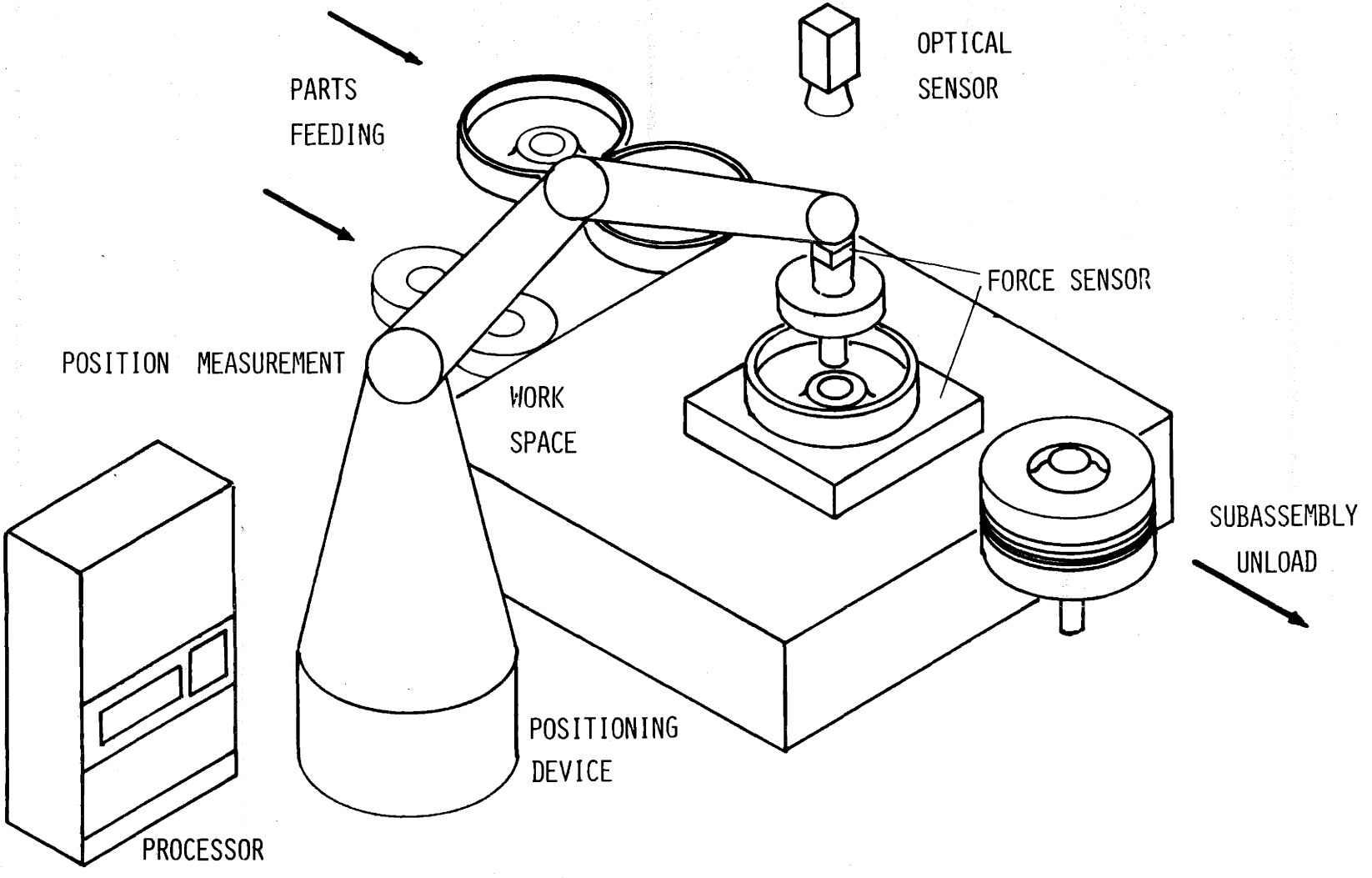


Figure I-1: Schematic of Assembly System Configuration

and fine motion paths of assembly (Figure I-2). Loosely speaking, the gross motion requirements are: (1) it must be able to have a range of motion roughly of two orders of magnitude greater than the characteristic size of the parts to be assembled, so as to accommodate for such things as feeders and jigs in the work-space, and (2) the accuracy requirement for the gross motion movements are of the same order of magnitude as the size of the parts and do not constitute a major problem. The fine motion requirements, on the other hand, are to have a range of movement roughly on the same order of magnitude of the characteristic size of parts but within an accuracy of roughly three orders of magnitude smaller than the size of the parts.

These combined positioning requirements of the individual parts of the assembly path make an overall requirement on the dynamic range of the positioning device. Its dynamic range, according to these rough figures, should be on the order of five orders of magnitude. This is a major engineering requirement for any equipment that has to operate in the manufacturing environment.

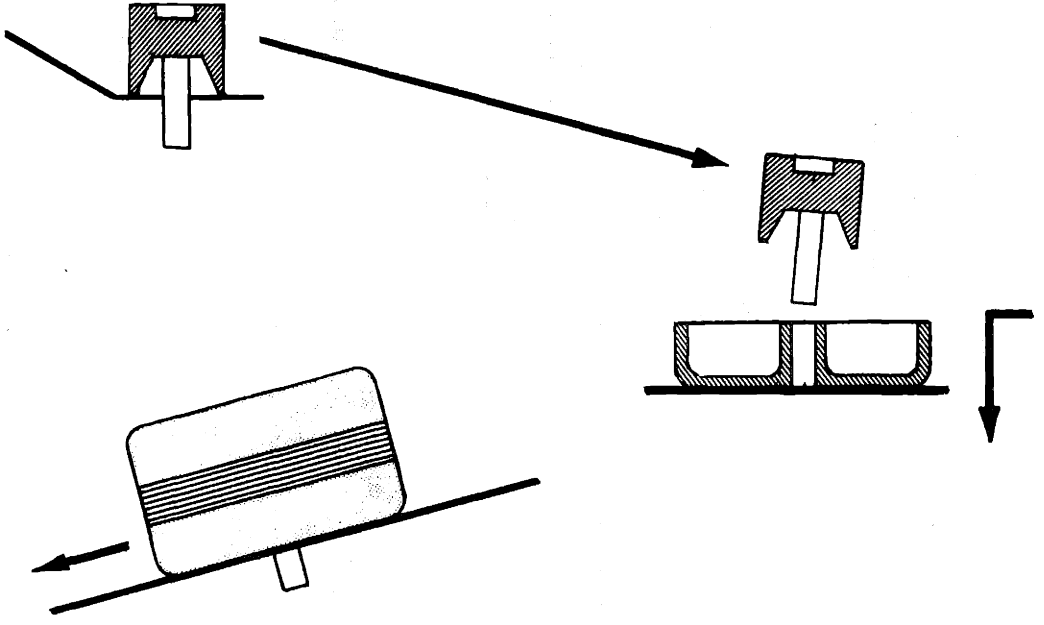
The positioning problem in assembly due to the geometry of parts can be traced to the nature of the surfaces on the parts being assembled. Parts in general have two types of surfaces, those critical to the functions and assembly of the parts and those that are not critical (Figure I-3). All functional surfaces are generally located within tolerances three orders of magnitude smaller than the mating surface's characteristic dimensions; non-functional surfaces are positioned with tolerances that can easily measure up to two orders of magnitude bigger than the tolerances on function surfaces. Non-functional surfaces are generally more accessible for handling and thus their bigger tolerance results in bigger possible errors at the mating surfaces. This problem is generally solved by special tooling or design changes of the parts to accommodate for reference surfaces. However, these solutions are not always possible or economically justifiable.

Parts Mating-Assembly Systems Study: The study of the parts mating has to include the positioning characteristics of the assembly system and also has to reflect back to the design and specifications of the assembly system. The approach

TRANSPORT TRAJECTORY

$$\frac{\text{SIZE PARTS}}{\text{LENGTH PATH}} \sim 10^{-2}$$

$$\frac{\text{SIZE PARTS}}{\text{ACCURACY}} \sim 1$$



MATING TRAJECTORY

$$\frac{\text{SIZE PARTS}}{\text{LENGTH PATH}} \sim 1$$

$$\frac{\text{SIZE PARTS}}{\text{ACCURACY}} \sim 10^{3}$$

Figure I-2: Assembly Trajectories

taken in this parts mating work is based on the work done in Reference I-10; this reference developed a tool, for a general type of assembly system, that transforms positions and positioning errors from the assembly system elements to the relative position between the mating parts and the statistics on the errors of this relative position, independent of how that position and those errors are generated.

The method used in this work is based on the parts mating model schematically shown in Figure I-4. The nominal relative position between parts is fed into a positioning device. This nominal position, distributed by the type of errors discussed in the previous paragraph, makes the parts interact. Forces caused by this interaction will further alter the position through physical impedences present in any real system. The model of Figure I-4 also shows the measurement available at the mating instant, namely force and position measurement shown in the model altered by sensor and computational inaccuracies; these measurements will be discussed in further detail within the next section of this chapter.

Some Remarks: In summary, this section defined the assembly problem as a positioning problem, discussed the nature of the problem of positioning, partially defined the scope of the parts mating study, and introduced a positioning model for analyzing the parts mating. The next section of this chapter will discuss the nature of information available from the parts and the system, and will define more precisely the scope and method of the parts mating study.

4. SENSORY INFORMATION

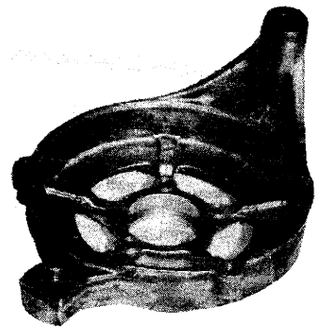
Type of Measurements Available: The preceding section discussed the need for a Flexible Assembly system and also discussed the type of positioning problems that are associated with a system of this nature. The concept of an adaptable system that would correct for the positioning problem through the use of sensory information in the conceptual frame of the Programmable Adaptable Assembly system is discussed in this section.

In general terms, two types of information measurements are possible in relation to the process of assembling parts:

ALTERNATOR CASING

Functional:
- bearing hole
- stator rim locator

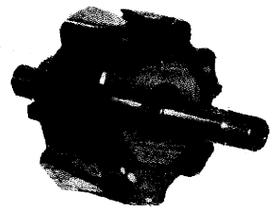
Non-functional:
- casting exterior surfaces
- mounting brackets



ALTERNATOR ROTOR

Functional:
- shaft

Non-functional:
- rotor laminations
- cooling vanes



COOLING FAN

Functional:
- center hole

Non-functional:
- all others

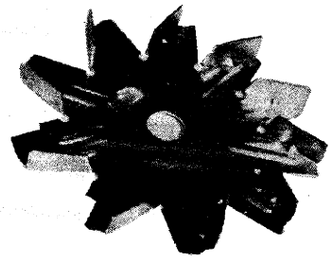


Figure I-3: Example of Functional and Non-Functional Surfaces

- the direct measurement of the relative position between the parts being assembled. This measurement could be obtained by a number of different procedures such as machine vision and scene analysis, direct measurement through optical acoustical or proximity devices, etc., or simply the actual reading of the positioning device's position;
- the measurement of the forces that are generated by the interaction of the geometry of the parts during the assembly. The assembly forces can be measured through a number of different schemes such as gripper and jig sensor or through reactions at the actuators of the positioning device (Figure I-1).

For the purpose of this parts mating work, the position and force will be considered measured at the frames defined on the mating surfaces of the parts being assembled. No reference is made to the particular method used to obtain these measurements--only the actual measurements and their statistics are analyzed here. The tool for transforming all these measurements and their statistics from the sensor's frames to the parts mating was developed in Reference I-10 and expanded upon in Appendix A.

Direct Position Measurement: From the model of Figure I-4, the execution of the assembly task is equivalent to correcting for the disturbance in the relative position between the parts. In principle, and given the appropriate positioning measuring sensors, the relative position error between the parts during assembly could be measured, the positioning device could then correct for any error on the relative position, and the assembly path could be executed within its deviation tolerances. To make feasible this approach two key requirements have to be met: First, the positioning device has to be capable of making the correction required on the path, and second, the position sensors have to be able to measure the relative position between the parts at the mating surfaces within the accuracy required by the assembly path.

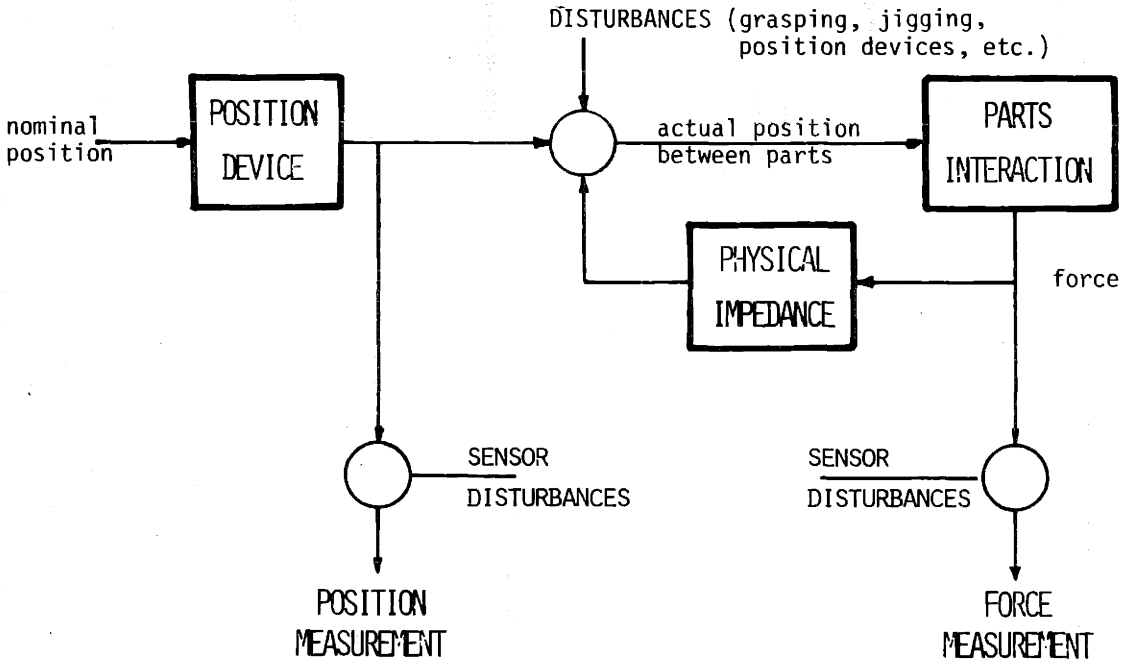


Figure I-4: Assembly System Model

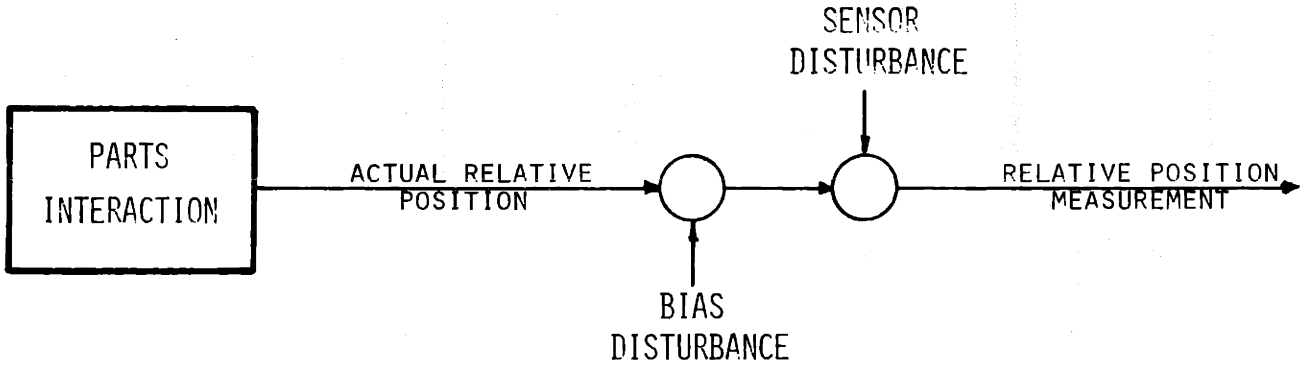
A positioning device capable of making small corrections within the accuracy required by the assembly path is technically feasible. The construction of high resolution devices is a standard procedure in machine tool manufacture. For the purpose of the parts mating study it will be assumed that the positioning device can resolve motions within the tolerances required by the assembly path.

The construction of a position measuring sensor capable of measuring the relative position between the parts at the mating surfaces has many of the problems attributed to the positioning devices that were discussed in the previous section. These problems are derived from dynamic range requirements on the sensors and availability of reference surfaces on the parts themselves.

A universal type of position measuring sensor would have to be located on the workspace and thus should not interfere with the rest of the assembly system elements shown in Figure I-1; this imposes a requirement on the sensor's dynamic range of the same order of magnitude as that imposed on an accuracy positioning device (10^5).

For an accurate absolute position measurement, the actual mating surface should be measured directly or, if these are not accessible for measurement, functional surfaces with known position with respect to the mating surfaces should be used. These surfaces are not generally available for measurement, however, and non-functional surfaces have to be used. Thus, absolute accurate position cannot be measured without the design of special sensors for measuring the relative position between the mating surfaces directly or by providing for reference surfaces on the parts. Such solutions are usually considered undesirable due to their lack of flexibility in terms of the system programmability.

Position Measurement Model: From the previous discussion it is concluded that absolute position measurement, although desirable, is not practically feasible. High resolution measurements of the relative position between the parts can easily be achieved through standard techniques in sensor's design. The special sensor approach is not analyzed here--it is interpreted as a particular case of Fixed Automation Assembly.



MODEL : $\underline{X}(N) + \underline{B}(N) + \underline{V}(N) = \underline{Z}(N)$

Figure I-5. Position Measurement Model.

A position measurement model, of the same nature as the positioning model shown in Figure I-4, is shown in Figure I-5. In this model the lack of positioning accuracy of the sensor is modeled by a bias term in the measurement and the imperfections on the resolution of the measuring device are modelled by a noise term. The model in itself is shown in a simplified form. The function circles in the figure involve transformations of the nature shown in Appendix A. The detail of these operations are shown in later chapters.

Force Measurements: Force information can be used to aid the assembly process by machines. We know that humans use it to a significant degree, although there is no clear understanding of the detailed process by which humans use tactile information in assembly. The approach taken here is to analyze the force information in terms of the parts mating process and not to emulate humans doing assembly.

The contact between the parts being assembled will result in reaction forces between these parts. The resultant forces, expressed in terms of Forces and Moments, can be measured by some sort of force measuring sensor. Reference I-11, Reference I-12, and Reference I-13 show different configurations of force sensors that have been implemented. For the purpose of this parts mating study the resultant of the forces are assumed measured at the frame defined on the parts mating surfaces. The tool for transforming the measurements and their statistics between (to and from) the sensor coordinates and parts coordinates is developed in Appendix A, based on the work done in Reference I-10. Note that only the resultant of the reaction forces can be measured; in other words, detailed information, such as the precise location of individual reaction forces is not directly available.

Some Remarks: In most cases, the positioning device will be controlled through commands generated from the position measurements; for this case the concepts of positioning and position measurement are equivalent. The separation of these concepts was done here for clarity and to stress the nature of these concepts in the framework of a Programmable Assembly System.

In summary, this section analyzed and modeled the position measurements and gave a brief description of the nature of the force measurement.

5. WORK STATEMENT

Conceptual Approach: The argument given in the previous sections of this chapter can be restated in the following three statements:

- a Programmable Assembly System based on the use of special purpose engineering tooling and/or sensors will result in an inherently inflexible system;
- a Programmable Assembly System that does not use special purpose engineering and that works on direct position control (also called open loop assembly) will result in an inherently unreliable system from the point of view of performing the actual assembly;
- in order to design a Programmable Assembly System that is both flexible and reliable, it is possible to make use of the information generated during the assembly process: namely, position and force information.

For the purpose of this work the assembly system will be assumed to have the following characteristics:

- the positioning device cannot position the parts within the accuracy required by the assembly path;
- the positioning device can resolve motions within the accuracy required by the assembly path;
- the system is provided with sensors capable of gathering information from the assembly process, and this information is of the general form described in Section Four of this chapter;
- the system has processing capability so as to generate corrective commands to the positioning device.

The content of this work is stated more specifically as follows:

- develop a general type of approach that can overcome the positioning and sensor characteristics of the system described above through the use of processing;
- construct a first set of engineering design tools for quantifying the system design.

The previous discussions justify the use of the information available at the assembly interface for aiding the mating process in a flexible programmable assembly system. The specific use of this information, its form, and engineering implications of its use are the objects of this work.

In terms of the time frame involved in the information processing, the processing can be divided into micro task and macro task processing (Figure I-6). The micro task processing uses the information of a single task and a priori information for computing the correction needed at that same task. The macro task processing uses the information of successive tasks to improve the a priori information used in the micro task as well as to provide information for other functions such as quality assurance, production, etc. This work addresses the micro task processing. The only involvement with the macro task processing is in the form of the information that is provided to and from the micro task. The purpose of this arbitrary division of the processing was to limit the scope of this work.

Thesis Organization: The thesis is organized into four chapters and six appendices. Chapter I introduces the specific work to be done, Chapter II analyzes the transition from gross to fine motion assembly paths, Chapter III conceptually analyzes the fine motion portion of the assembly path, and Chapter IV gives a summary of the work and outlines future work in this area. The Appendices cover the analytical and implementation work needed to support the concepts described in the chapters.

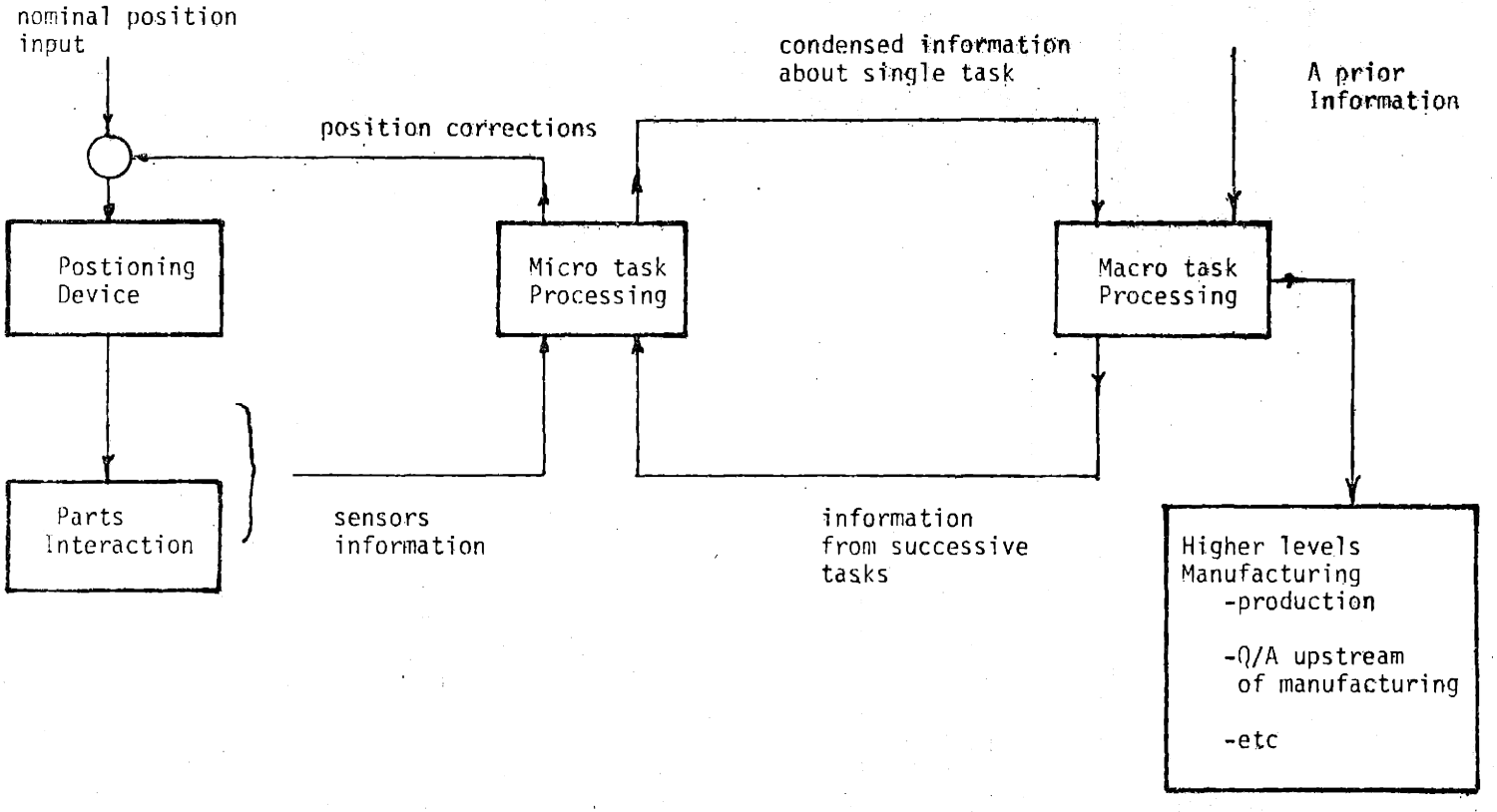


Figure I-6: Micro-task and Macro-task Processing in Assembly

CHAPTER II

TRANSITION FROM GROSS TO FINE MOTION

1. POSITIONING ISSUES IN ASSEMBLY

The assembly problem has been identified as one of accurately positioning the parts with respect to each other. For assembly between two parts, the positioning of the parts with respect to each other is characterized by the relative position between them. The positioning problem is expressed in terms of the error in the relative position allowed by the geometry of the parts during assembly process.

For the study of this positioning problem, Chapter I conceptually divides the assembly process into gross motion--the process of orienting, positioning and transporting the parts in free space--and fine motion--the process of mating the parts within their nested position. This chapter will analyze the use of the sensory information available from the assembly interface for the purpose of aiding the transition from gross to fine motions. The problems of fine motion and the use of the information available at the assembly interface for achieving this portion of the assembly task are analyzed in Chapter III. The problems related to gross motion are not analyzed in this work, except for the implications of gross motion positioning in both the transition from gross to fine motions and in the actual fine motion regions of the assembly task.

The discussion on the nature of the positioning problem for a flexible and programmable assembly system (made in Chapter I) postulated a model for the positioning device. In this model, the expected error on relative position between the parts at mating would not allow the parts to be assembled. In the context of this model of the positioning device, the object of this chapter is to define and evaluate the type of processing to be executed on the information generated by the parts being assembled in order to estimate the error on the relative position,

correct for this error, and proceed with the assembly. If the estimated error of the relative position between the parts and the correction of that error are both made within the margins required by the geometry of the parts, the transition from gross to fine motion can be accomplished.

The sequence of the presentation is as follows:

- the nature of the position information is discussed and a conceptual description of the type of processing required for this information is introduced;
- a theoretical evaluation of the performance of such an approach is conceptually described;
- an actual implementation of this approach is presented, an experiment is described, and the results obtained are detailed; and
- in closing, a brief review of the method as compared to alternative methods is presented.

2. POSITION INFORMATION

Previous discussions on the nature of the positioning measurements (Chapter I, Section Four) postulated that these measurements, in a flexible type of programmable assembly system, will follow the model shown schematically in Figure I-5. In this model, the actual relative position between the parts at an instant n combine with a bias error and a noise term to form the position measurements at this instant n. This is written symbolically as:

$$\underline{z}(n) = \underline{x}(n) + \underline{b} + \underline{v}(n) \qquad \text{II-1}$$

where $\underline{x}(n)$ represents the vector of relative position between the parts, at time n, the constant bias term $\underline{b}$ lumps the uncertainties on this relative position

caused by gross inaccuracies in the positioning device (grasping errors, jigging errors, geometric tolerances on the parts, etc.), and the vector $\underline{v}(n)$ represents the resolution error at time n of the positioning devices (actuator errors, numerical errors, etc.).

NOTE: The form of the right hand side of this expression is only conceptually a summation; the exact expression is obtained through kinematic transformations of the type developed in Appendix A.

From the model postulated in Expression II-1, two fundamental questions emerge that this chapter will try to answer:

(i) Is the form of the model postulated a reasonably accurate model for the position measurements--or, in other words, can the gross uncertainties of the positioning device be modeled as a bias term, fixed in time?

(ii) Can the true relative position between the parts be obtained from a model of the nature given by Expression II-1?

It is more convenient to answer the second question, and then the first. The second question is answered in two steps. First, a theoretical proof of the feasibility of the approach is done. Second, an implementation that obtains the relative position between parts from simulated data is done. The first question is answered through an actual experiment that extends the results of the simulated data to the actual process of assembling the parts.

For the purpose of discussion, let us assume that the measurement is obtained with ideally perfect position sensors. If so, the noise component ($\underline{v}(n)$) in Expression II-1 is zero, and the model becomes:

$$\underline{z}(n) = \underline{x}(n) + \underline{b} \qquad \text{II-2}$$

Recall that the parameter n represents different instances at which measurements are taken. If the relative position at time 1 is to be obtained from only the

measurement ($\underline{z}(1)$) at this time, Expression II-2 becomes:

$$\underline{z}(1) = \underline{x}(1) + \underline{b} \qquad \text{II-3}$$

which results in a dimensionality of the unknown terms ($\underline{x}(1)$ and $\underline{b}$) greater than the dimensionality of the data ($\underline{z}(1)$). This implies that the evaluation of the relative position between the parts at the instant 1($\underline{x}(1)$) in Expression II-3 from direct use of the position measurements at time 1, is algebraically undetermined. The use of multiple position measurements data points ($\underline{z}(1)$, $\underline{z}(2)$,...,$\underline{z}(N)$) does not change this situation since the total dimensionality of the unknown parameters will still exceed the dimensionality of the known parameters. It is thus concluded that the relative position between the parts cannot be obtained directly from the position measurements when these measurements conform to the simplified model in Expression II-2. Consequently, the same conclusion holds if the measurements conform to the model in Expression II-1.

To overcome this indetermination, there are two possible alternatives, namely:

- decrease the total dimension of the unknown parameters in Expression II-2 or Expression II-1; or
- increase the dimension of the data through measurement of different physical variables at the assembly interface.

This chapter develops the approach based on decreasing the dimension of the unknown parameter. The second approach is not analyzed here; however, some discussion is made on this respect in the final section of this chapter.

In kinematic terms, the relative position between two solid bodies in free space has six degrees of freedom. This means that six independent parameters are needed to completely specify this relative position between the bodies. If the solid bodies in consideration are not completely free to move (for example, when the parts form a mechanism, or if the bodies are allowed to move without losing contact), less than six independent parameters would be needed to completely specify

their relative position. Thus, the reduction of the dimension of the unknown parameter (right hand side of Expression II-2) can be achieved by considering only those measurements obtained when the parts are in contact and by formulating the relative position vector ($\underline{x}(n)$, n=1, N) in terms of the parameters needed to completely specify this relative position for the parts in contact.

The unknown parameters in the right hand side of Expression II-2 are thus reduced to the necessary parameters needed to define the relative position between the parts and the components of the bias vector $\underline{b}$.

The operation described above can equivalently be done through the introduction of a set of conditions that guarantee that the relative position vector ($\underline{x}(n)$, n=1, N) represents a state where the parts are touching. The conditions on the relative position between the parts that reflect the parts touching is expressed as a function of this relative position. This is written formally as:

$$\underline{h}(\underline{x}(n)) = 0 \qquad \text{II-4}$$

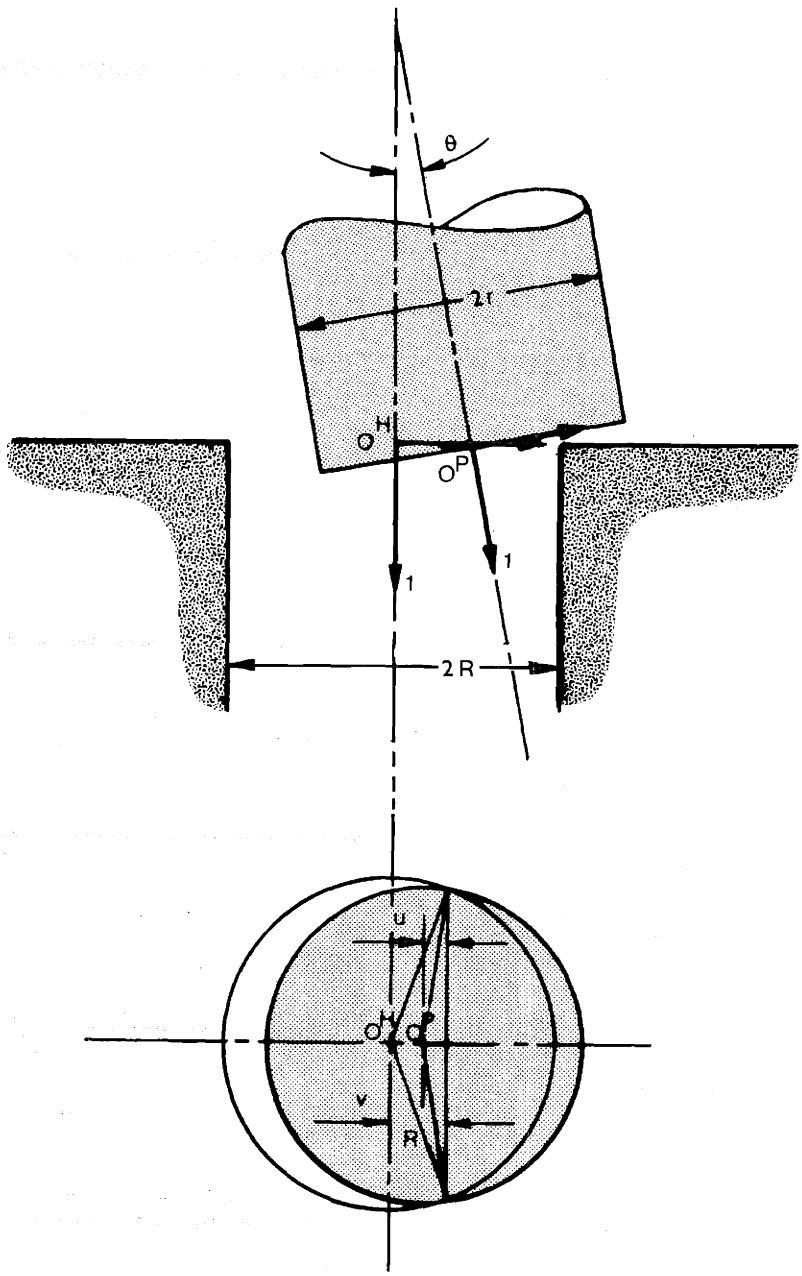
where the components of the vector $\underline{h}$, of dimension k_h, are functions of the relative position vector $\underline{x}$. Equating this vector with zero imposed the touching conditions on the parts. See the two-dimensional example in Figure II-1.

The following assumptions allow the incorporation of the "parts in contact conditions" to the geometry model:

- the geometry of the parts being assembled is known perfectly;
- the condition necessary for the parts to be touching can be expressed as a function of the relative position between the touching parts.

The use of the geometric touching condition of Expression II-4 with the simplified measurements model in Expression II-2 for the computation of the vectors ($\underline{x}(n)$, n=1, N) and $\underline{b}$ is formally expressed as:

Find the vectors ($\underline{x}(n)$, n=1,...,N) and $\underline{b}$ that satisfy the relations of :



$$(x_1^{PH}/\sin\theta)^2 - (x_2^{PH} + x_1^{PH}\cot\theta)^2 = r^2 - R^2$$

Figure II-1: Geometric Contact Condition,
Two Dimensional Peg and Hole

$$\underline{x}(n) + \underline{b} = \underline{z}(n) \qquad n = 1,\ldots,N$$

$$\underline{h}\ (\ \underline{x}(n)\ ) = 0 \qquad \text{II-5}$$

The incorporation of geometric relation in Expression II-4 into the determination of the relative position vector between the parts has the net effect of reducing the dimensionality of the unknown parameters of Expression II-2 by the amount k_h at every instant a measurement is taken. Formally, if:

k_z is the dimension of the measurement vector $\underline{z}(n)$; should equal six for the three-dimension space;

k_x is the dimension of the vector $\underline{x}(n)$, also equal to six;

k_b is the dimension of the vector $\underline{b}$, also equal to six;

k_h is the dimension of the vector $\underline{h}$;

N is the number of measurements used.

Then the number of unknown parameters in N measurements should be less than the number of data parameters in N measurements. Thus:

$$N(k_x - k_h) + k_b \leq Nk_z$$

which yield the minimum number of data points to make the determination at least physically possible. This is expressed as:

$$N \geq \frac{k_b}{k_z + k_h - k_x} = \frac{6}{k_h} \qquad \text{II-6}$$

This is a necessary condition. Further conditions have to be imposed on $\underline{h}(\underline{x}(n))$ to make a determined problem. The next section deals with some aspects of this

determination problem.

The removal of the ideal sensor assumption results in additional unknown parameters to be determined. If the statisticts of the noise term $\underline{v}(n)$ are assumed known, the position determination is expressed as an estimation problem as follows:

Estimate the vectors $\underline{x}(n)$ and $\underline{b}$ from:

$$\underline{z}(n) = \underline{x}(n) + \underline{b} + \underline{v}(n) \qquad \text{II-7}$$

with $\qquad n = 1,\ldots,N$

$$\underline{h}(\ \underline{x}(n)\ ) = 0$$

The discussion on the minimum number of measurements required in the perfect sensor case (Expression II-6) holds valid for the non-perfect sensor case, formulated as an estimation problem, in Expression II-7. It is understood that these measurements have to be obtained for different relative positions between the parts ($\underline{x}(n)$); different measurements of a single relative position will not contribute information for its estimation.

Methods in Information Theory can be used to solve the estimation problem stated in Expression II-7. The approach used here is based on the exposition of these methods given in Reference II-1.

In summary, this section introduced the concept and consequences of the bias component in the position measurements and introduced the method of using the geometry of the parts in conjunction with the position measurements in order to determine the actual position between the parts being assembled.

3. PERFORMANCE EVALUATION OF THE POSITION ESTIMATION

The previous section discussed the necessary conditions and the procedure for calculating the relative position between the parts under no measurement noise, and postulated a procedure for evaluating this position in the presence of measure-

ment noise. This section discusses a theoretical performance evaluation of the method outlined in the previous section for computing the relative position in the presence of measurement noise.

The performance of the estimation of the position vectors $((\underline{x}(n)), n=1, N)$ is evaluated in detail in Appendix C. This section outlines the procedure and concept used in that evaluation.

The statistics of the noise term in the measurement model is assumed known. Thus the joint probability density of the random process $(\underline{z}(n), n=1, N)$, given the values of the vectors $(\underline{x}(n), n=1, N)$ and $\underline{b}$, can be computed. Let us denote this probability by:

$$p[(\underline{z}(n), n=1, N) \,/\, \underline{b}, (\underline{x}(n), n=1, N)] \qquad \text{II-8}$$

Conceptually, this expression gives the probability of obtaining a set of measurements $(\underline{z}(n), n=1, N)$ when the values of the vectors $\underline{b}$ and $(\underline{x}(n), n=1, N)$ are given. The estimation of the vectors $\underline{b}$ and $(\underline{x}(n), n=1, N)$, on the other hand, involves finding the values of these vectors that yield a set of true measurements denoted here by $(\underline{z}_t(n), n=1, N)$. An intuitively logical estimation procedure for the set of vectors $(\underline{x}(n), n=1, N)$ and $\underline{b}$ is the so-called maximum likelihood estimator. This estimation consists of finding the set of vectors $(\underline{z}(n), n=1, N)$ and $\underline{b}$ that maximizes the conditional joint probability of Expression II-8 evaluated at the actual measurements, $(\underline{z}_t(n), n=1, N)$.

The joint conditional probability, Expression II-8, evaluated at the actual measurement points $(\underline{z}_t(n), n=1, N)$, can be interpreted as a function of the vectors $(\underline{x}(n), n=1, N)$ and $\underline{b}$. This function represents a hypersurface in the space of the variables $(\underline{x}(n), n=1, N)$, $\underline{b}$, and the probability density. The maximization operation consists of finding the crest of this hypersurface. The performance of the estimator can thus be measured by the curvature of this hypersurface at the actual value of the vectors $(\underline{x}(n), n=1, N)$ and $\underline{b}$. In essence, the curvature of the hypersurface measures the sensitivity of the estimation process to the perturbations caused by the measurement noise.

This method of determining the performance of the estimator is based on the ambiguity function concept. This concept is explained in Reference II-2 and the actual calculations are detailed in Appendix C.

The error is expressed in terms of the difference between the estimated values of the vectors ($\underline{x}(n)$, n=1, N) and $\underline{b}$, denoted here by ($\hat{\underline{x}}(n)$, n=1, N) and $\hat{\underline{b}}$, and the true values of these vectors, denoted by ($\underline{x}_t(n)$, n=1, N) and $\underline{b}_t$. The respective errors are written as:

$$\delta\underline{b} = \hat{\underline{b}} - \underline{b}_t \qquad \text{II-9}$$

$$\delta\underline{x}(n) = \hat{\underline{x}}(n) - \underline{x}_t(n) \qquad n=1, N$$

The measure of the performance calculated in Appendix C is given in terms of a hypervolume in the space of the errors ($\delta\underline{x}(n)$, n=1, N) and $\delta\underline{b}$. A reasonable scalar representation of the error is obtained through the longest linear dimension of this hypervolume. This linear measure will be denoted by the scalar ε. Figure II-2 shows the performance of this linear measure of the estimation problem. In it the ratio between the linear performance measure and the noise term of the measurement is plotted against the number of measurement points used for obtaining the estimation. In essence this diagram shows the measure for the estimation error to be directly proportional to the noise term in the measurement, inversely proportional to the square root of the number of measurements used, and inversely proportional to the total displacement used for obtaining the measurements.

In summary, this section discussed the theoretical performance of the determination of the actual position between the parts being assembled when the measurement of these positions is given by the model of Expression II-1. This measure gives a tool for evaluating the trade-offs between the need for good resolution in the measurements, represented by the noise term in Expression II-2, and the time spent in obtaining this estimate, linked here to a number of measurements and the travel used for obtaining these measurements.

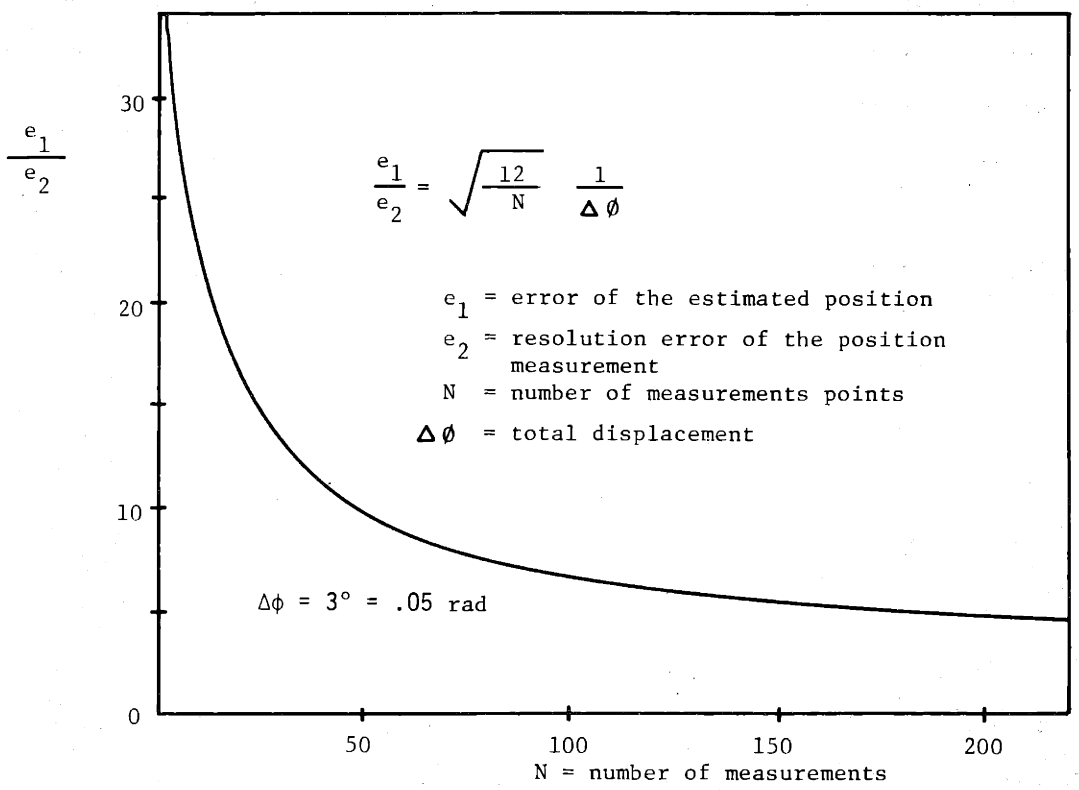


Figure II-2: Theoretical Performance of Position Estimation

4. POSITION ESTIMATORS IMPLEMENTATION

This section explains the concepts behind the implementation of the relative position estimation between the parts assuming the measurement model in Expression II-1. The details of the implementation are given in Appendix F. The purpose of this discussion is as follows:

- conceptually explain the implementation;
- verify the results of the theoretical evaluation detailed in Appendix C and conceptually explained in Section Three of this chapter;
- prove, through an assembly experiment, the validity of the position measurement model of Expression II-1.

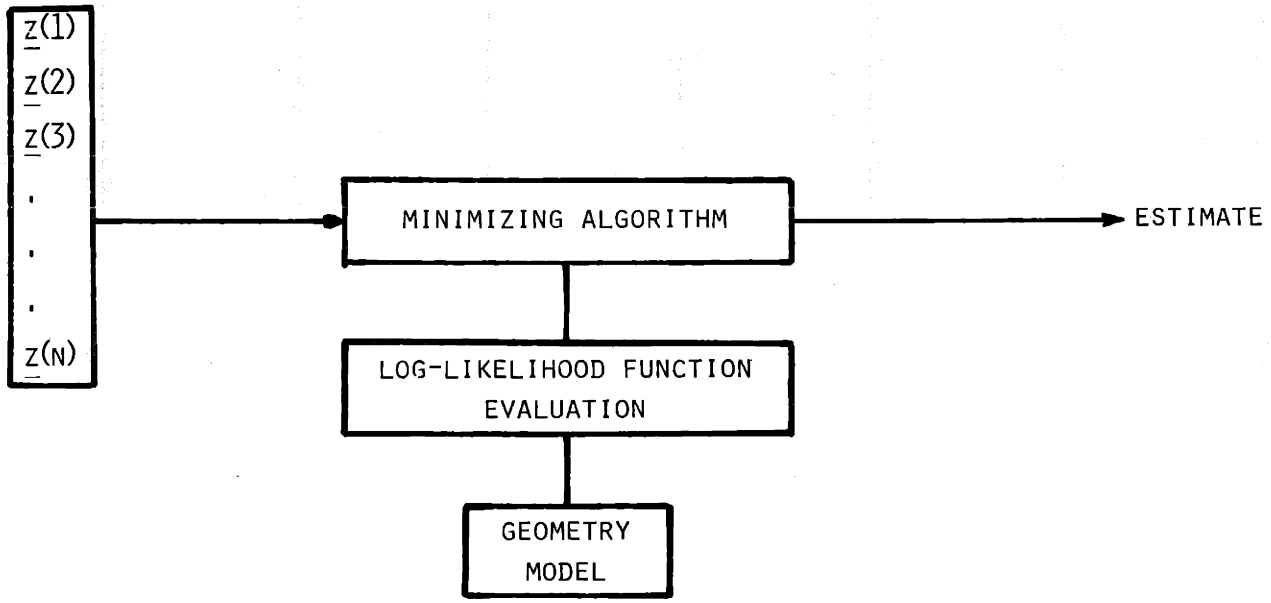


Figure II-3: Smooth Filter Implementation

The estimation of the vectors ($\underline{x}(n)$, n=1, N) and $\underline{b}$ from the measurement data ($\underline{z}(n)$, n=1, N) can be visualized as the search for the vectors that satisfy the geometric conditions of Expression II-4, and best fit the position measurement data. The concept of "best fitting" presumes some sort of performance to measure the fit. The procedure outlined here uses the minimum square error criteria.

Two different approaches were used for implementing the estimator. These are:

- a smooth estimator, that basically uses the whole time series of the position measurements to evaluate the time history of the relative positions and the constant bias vector. This procedure requires that the position data be collected before proceeding with the estimation (See Figure II-3). It is of little practical value for the actual real time assembly, although its application is of potential use in calibration of jigs, fixtures and measuring devices. This procedure was implemented for two basic reasons: first, it is a simpler approach to implement and second, it is a more stable procedure and thus gives a good understanding of the problem at the development stage;
- a recursive implementation of the estimator that gives an estimate of the vectors $\underline{x}(n)$ and $\underline{b}$ at each time step n using the measurements $\underline{z}(n)$ up to time n (See Figure II-4). This procedure makes use of the extended Kalman filter technique. The actual details and code are shown in Appendix F.

Two sets of experiments were performed to test both the recursive and smooth estimation implementations. The first set of experiments used data generated by a two-dimensional simulation of the transition from gross to fine motion of a peg into a hole. The second set of experiments used data generated by a computer-controlled manipulator performing a peg-and-hole assembly.

The objective of the simulated data experiment was to evaluate the validity and performance of the estimators under controlled conditions and to verify the

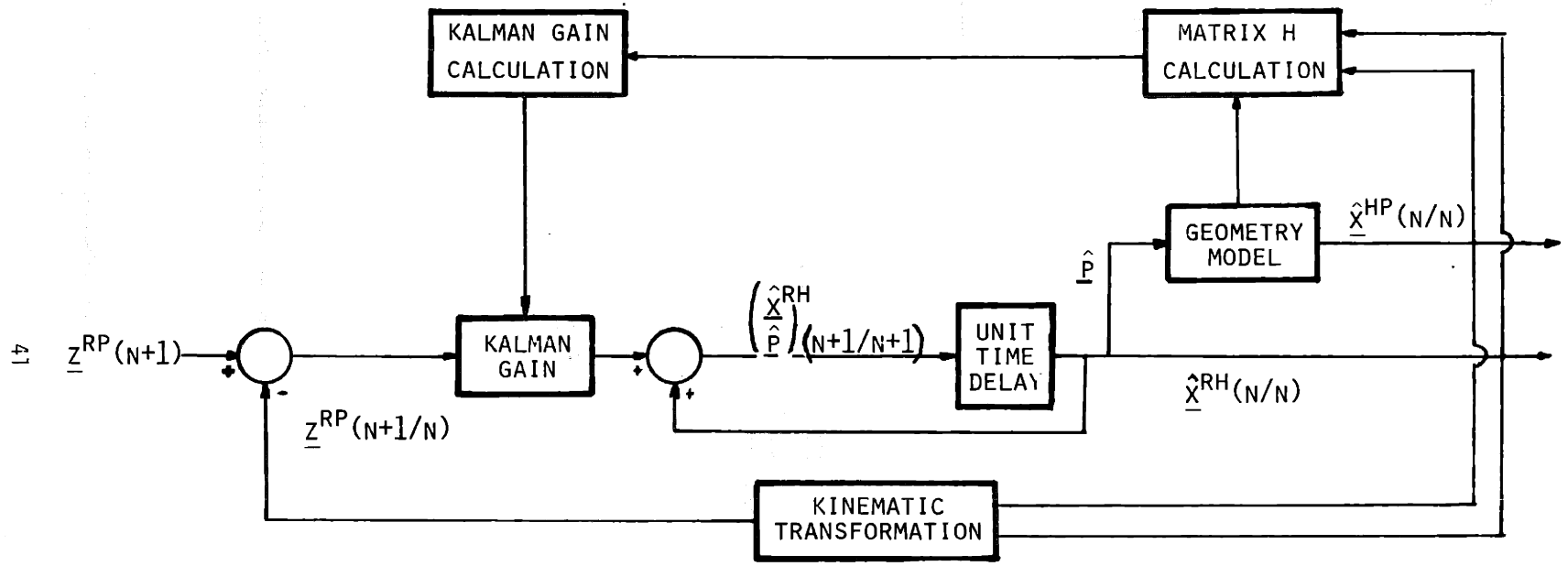


Figure II-4: Schematic of Recursive Filter Implementation

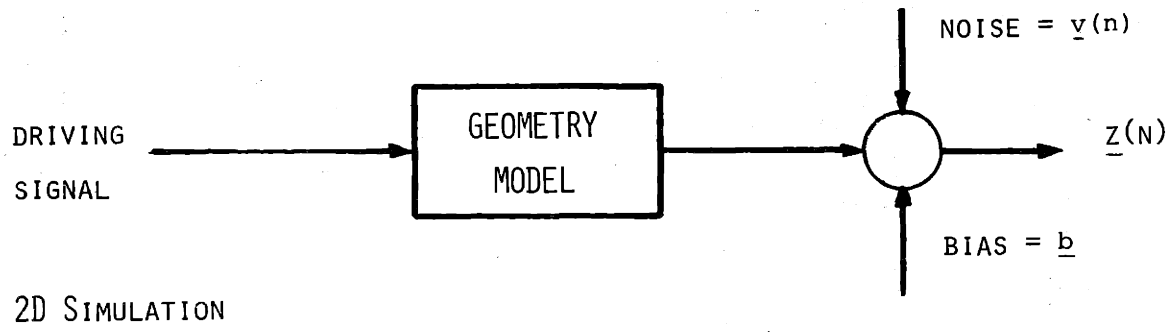


Figure II-5: Schematic of the Generation of Simulated Measurements Data

the performance prediction of the theoretical analysis of the previous section. The objective of the computer-controlled assembly experiment was to evaluate the performance of the three-dimensional implementation of the estimation and to verify the validity of the position measurement model postulated in Expression II-1.

Figure II-5 shows schematically the process involved in the generation of the simulated position measurements. This process involved the following steps:

- the geometric model for the two-dimensional transition, from gross to fine motion, of the insertion of a peg into a hole, shown in Figure II-1, was used to generate a time series of actual position vectors ($\underline{x}(n)$, n=1, N). These vectors will satisfy the geometric condition of parts touching;
- a controlled bias vector, $\underline{b}$, was added to the position vector ($\underline{x}(n)$, n=1, N); the vector $\underline{b}$ simulates the uncertainties introduced in the measurements due to error in jigs, gripper, tolerances in the parts, etc. The resulting vector, ($\underline{y}(n)$, n=1, N), simulates the position measurements that could be obtained with perfect position sensors;
- a white zero mean random vector, ($\underline{v}(n)$, n=1, N), produced by a random number generator was added to the time series of vectors ($\underline{y}(n)$,n=1, N) to create the position measurement vector ($\underline{z}(n)$, n=1, N). The vectors ($\underline{v}(n)$, n=1, N) simulate the imperfections of position measurement sensors. Different resolutions at the position measurement sensor were simulated through changes in the standard deviation of the random vector ($\underline{v}(n)$, n=1, N).

The computer-controlled assembly experiment was executed in the following steps:

- the assembly of a peg into a hole (Figure II-6) was attempted;
- the data generated during the attempted assembly was recorded;
- the implementation of both the smooth and recursive filters were used to process the data off-line.

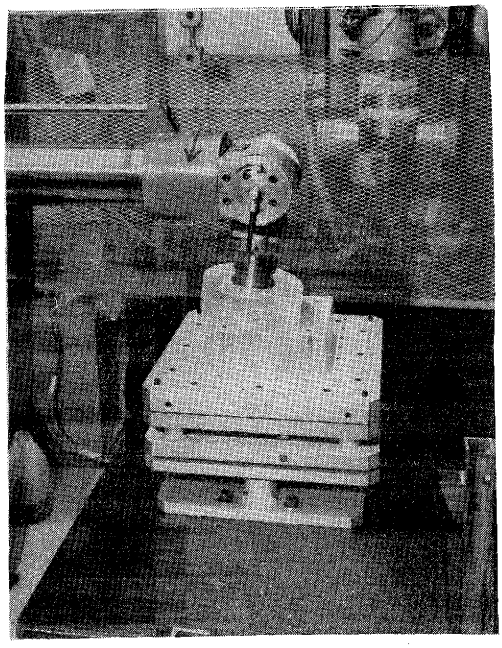


Figure II-6: Big Peg and Hole Being Assembled by Computer Controlled Manipulator

The estimation procedure and the correction of the position error necessary to complete the assembly task were not implemented on-line. The lack of capacity in the processor that controlled the arm and the early dismantling of the experimental test apparatus did not allow this implementation. The validity of the results obtained from the estimation procedure had to be tested through the use of statistical checks on the results.

The position measurements were obtained from the manipulator itself. The set of position values of the axes of the manipulator at each measuring instant were transformed into the relative position measurements between the parts at that instant (Figure II-7). The kinematic procedures used in these transformations are developed in Appendix A; the details of these position measurement calculations are given in Appendix E.

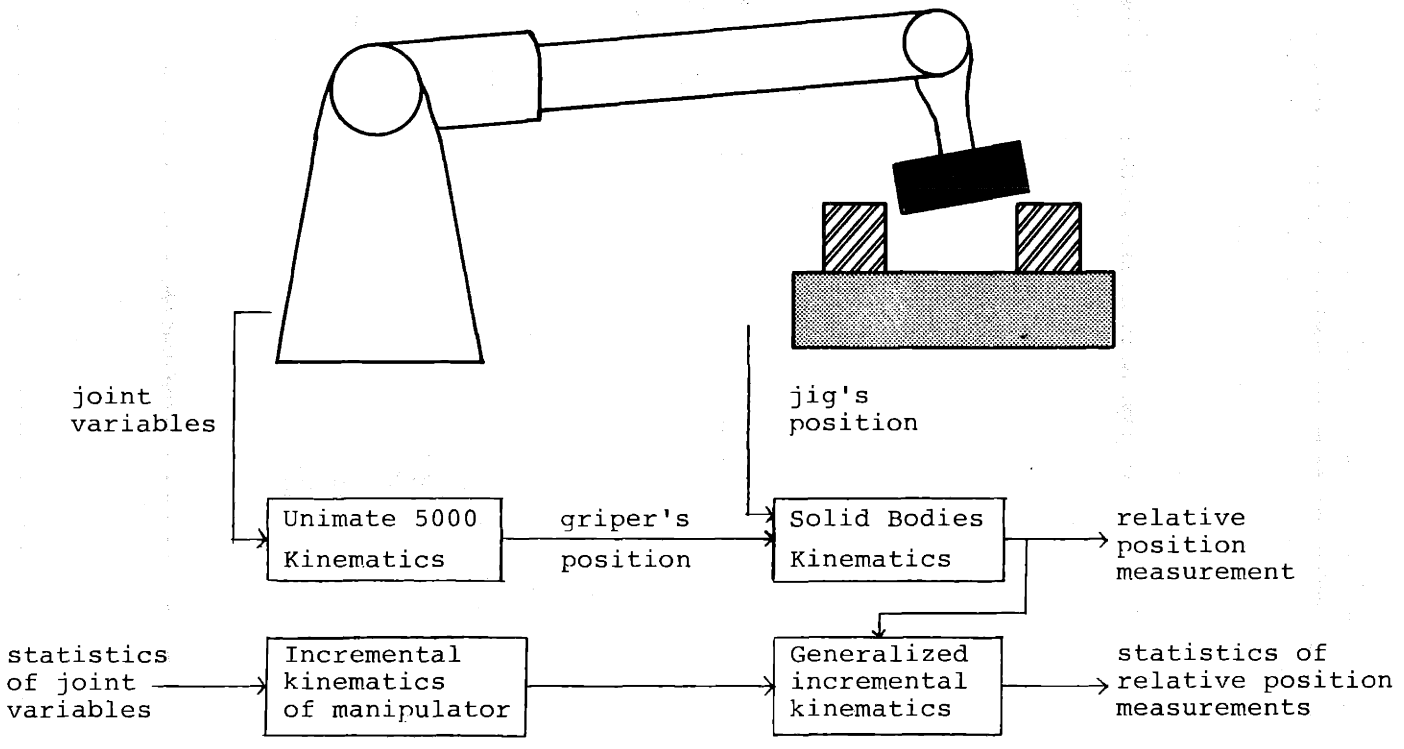


Figure II-7: Schematic of the Experimental Data Gathering

The estimation algorithms call for position measurements while the parts are touching. This was realized by closing a force loop through the computer that guaranteed that the parts would stay touching once they made contact; details about this procedure are also given in Appendix E.

The resolution of the position measurements obtained by this method depends on the resolution of the axes of the manipulator. The calculation of the resolution of the particular manipulator used was done in Reference II-3, and the numerical values taken from this reference are given in Figure E-1.

The result of the performance evaluation of the estimator done in Appendix C, and conceptually explained in Section Two of this chapter, state that, for a reasonable number of measurement points, the expected accuracy of the estimation is at best on the same order of magnitude as the resolution of the position measurements. Parts to be assembled, a peg into a hole in this particular case, have to be located within certain clearances. In typical parts, Reference II-4, these clearances are at least two orders of magnitude smaller than the characteristic size of the parts. Thus, in order to have a representative set of measurements for the implementation of the estimation calculation, oversized parts to had to be used. This artifice simulated a high resolution positioning device. The size of the parts used are:

$$\text{Diameter of peg} = 188.2 \text{ mm}$$

$$\text{Diameter of hole} = 190.5 \text{ mm}$$

$$\text{Clearance ratio} = \frac{\text{difference of diameters}}{\text{diameter of hole}} = .012$$

5. SUMMARY OF RESULTS

The smooth estimator was used only on the data generated by the manipulator and big parts experiment. The recursive estimator was used on both the simulated data generated by the manipulator experiment.

The four figures beginning with Figure II-9 show the results of the recursive implementation on the simulated data. Figure II-9 shows the estimated value of

the lateral off-set ($\underline{x}_2^{PH}$ in Figure II-8) versus the number of measurements; the figure also shows the actual value of lateral off-set. Figure II-10 shows the estimate of the two-component of the bias vector $\underline{b}$ for three different simulated noise values as a function of the number of measurement points. Figure II-11 and Figure II-12 are statistical tests of the behavior of the estimation, the first showing the measurement residue of the estimation for the two-component, and the second showing the result of the filtered implementation of the normalized log likelihood function; both are functions of the number of measured points.

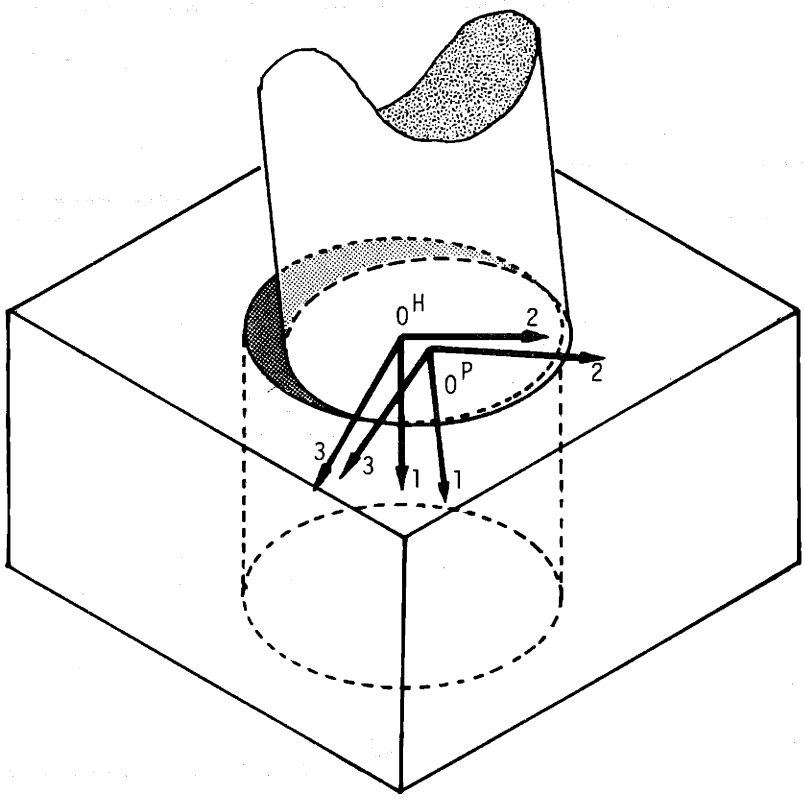


Figure II-8: Peg and Hole Frames, Notation

For a good model of the measurement data, the residues of the measurement (Figure II-11) should be a white random process and normalized log likelihood function (Figure II-12) should converge to a number equal to the degrees of freedom of the model, or, in this case, one.

Figure II-13 shows the result of the implementation of the smooth estimator

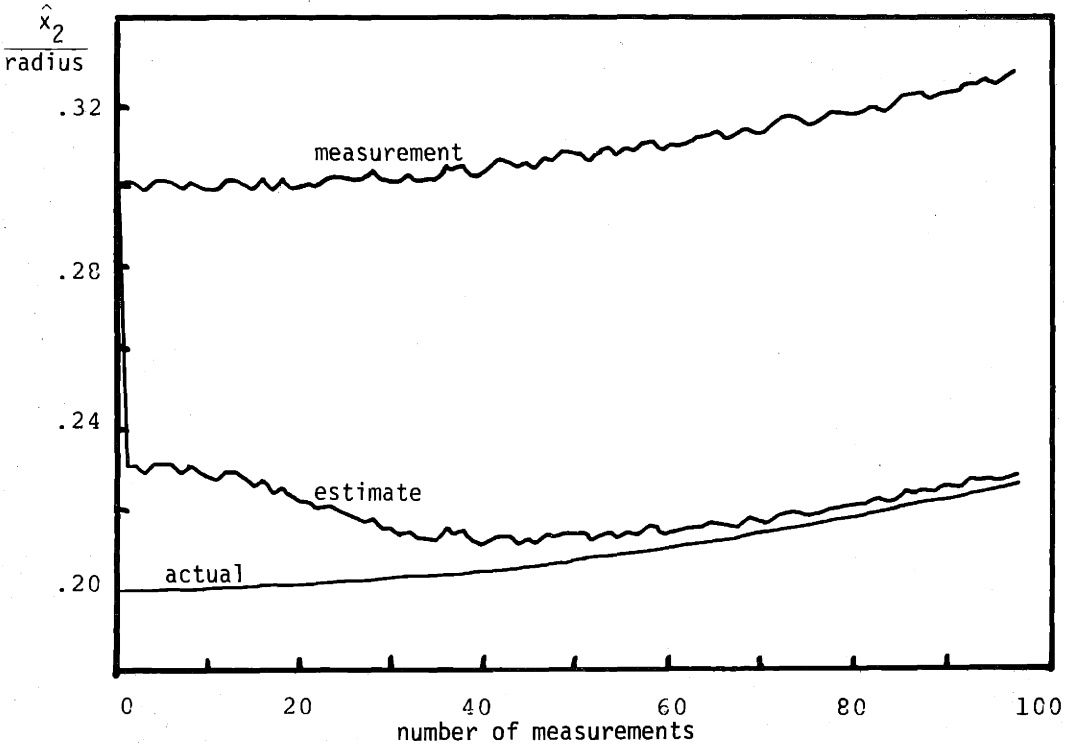


Figure II-9: Actual, measured and estimated values of relative position, two component

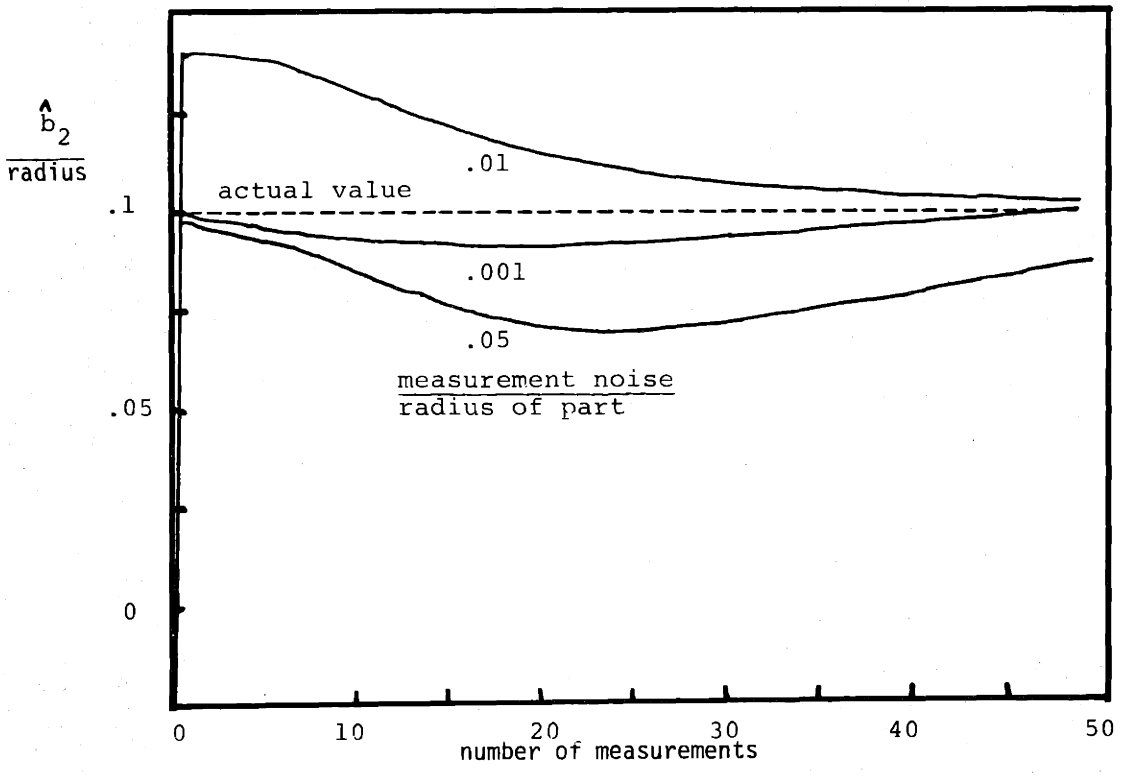


Figure II-10: Actual and estimated values of bias term, two component

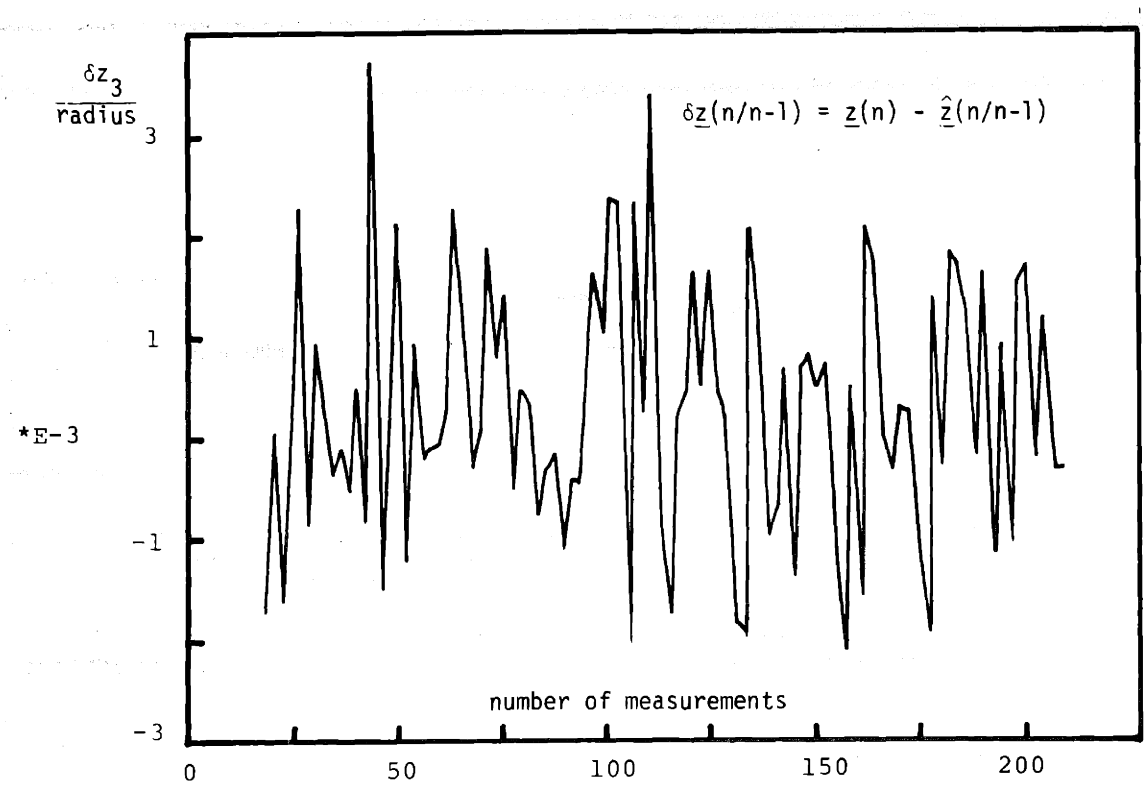


Figure II-11: Time History of Measurement Residues

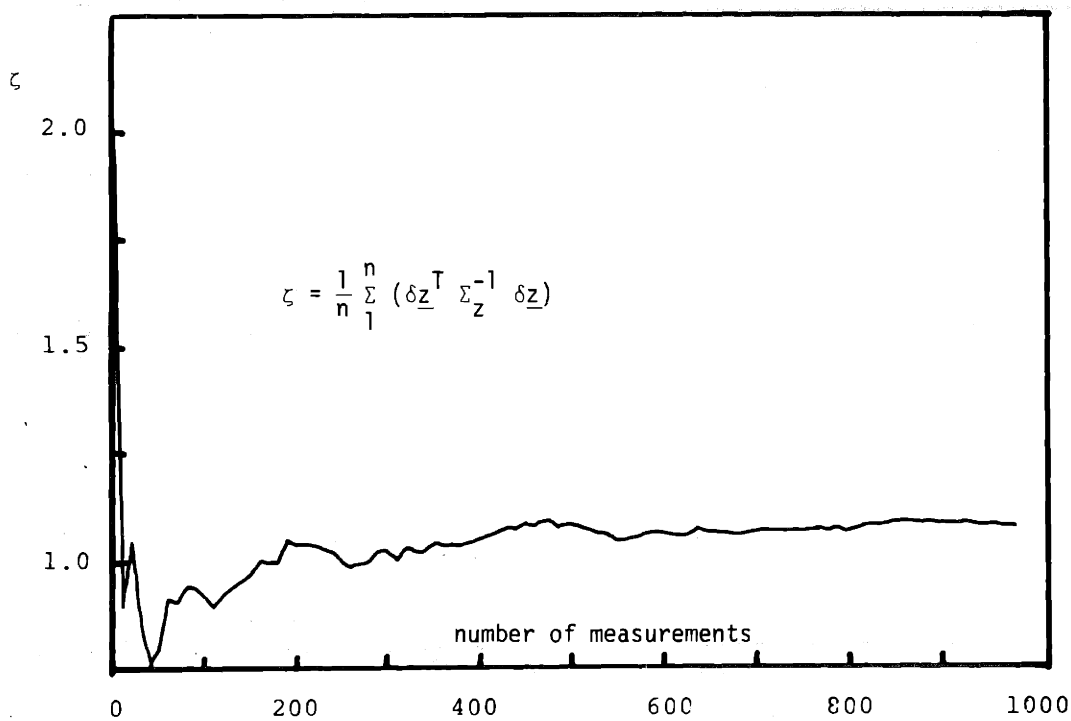


Figure II-12: Time History of Filtered Log-Likelihood Function

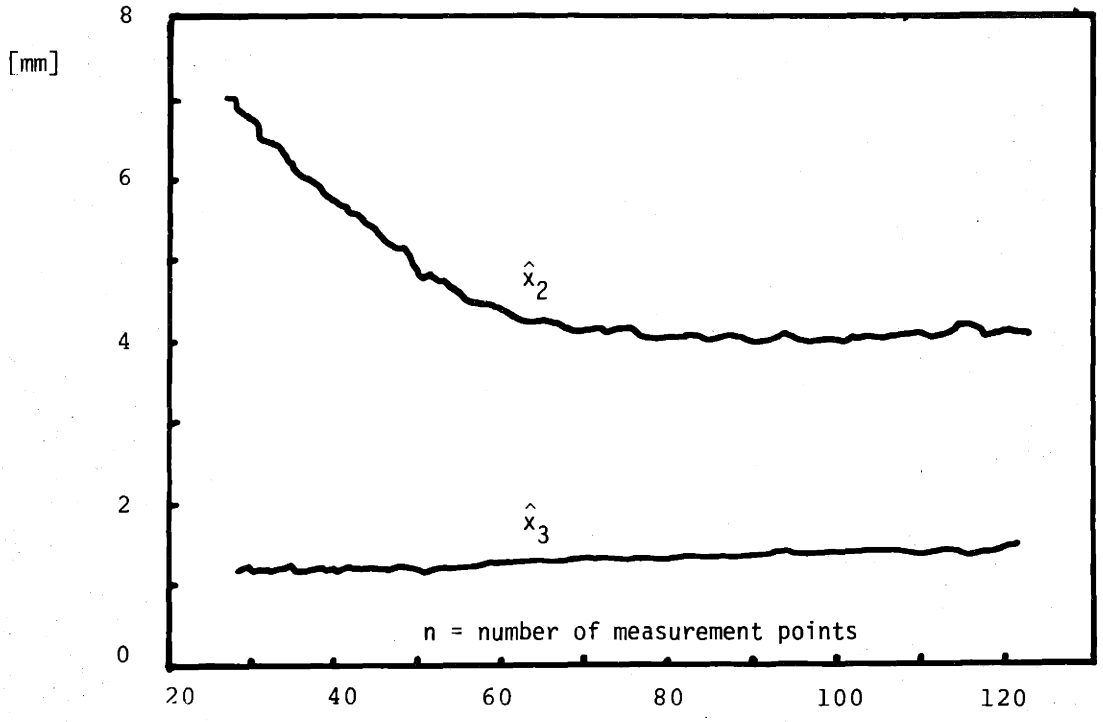


Figure II-13: Position Computed by Smooth Estimator on Experimental Data

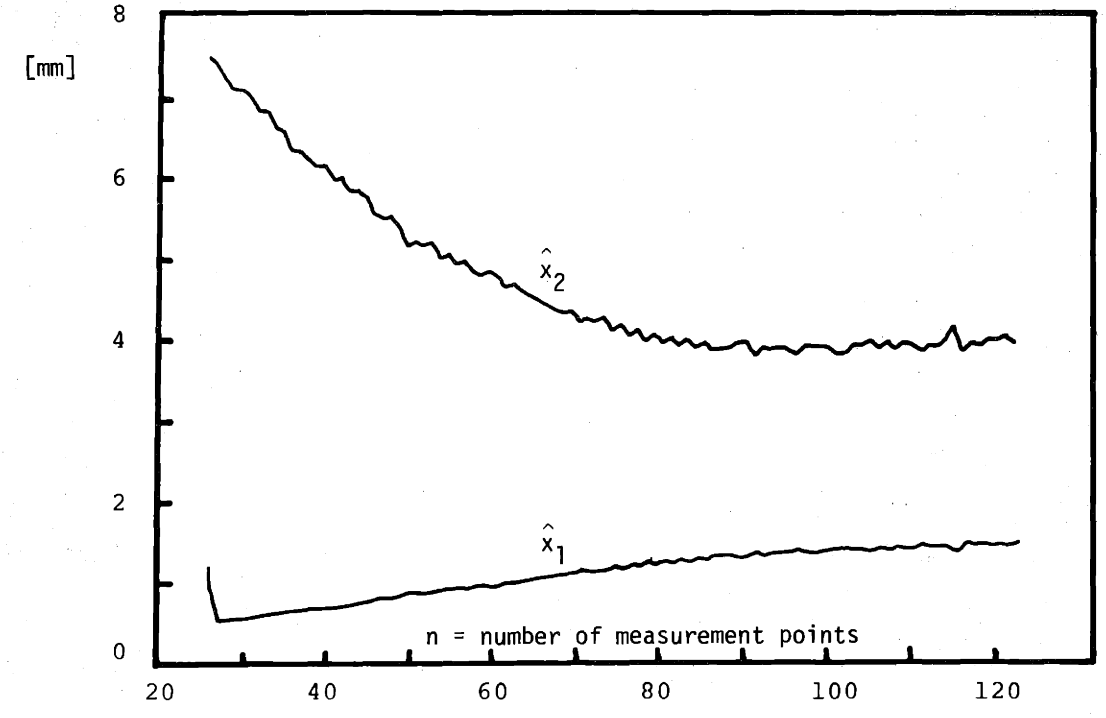


Figure II-14: Position Computed by Recursive Estimator on Experimental Data

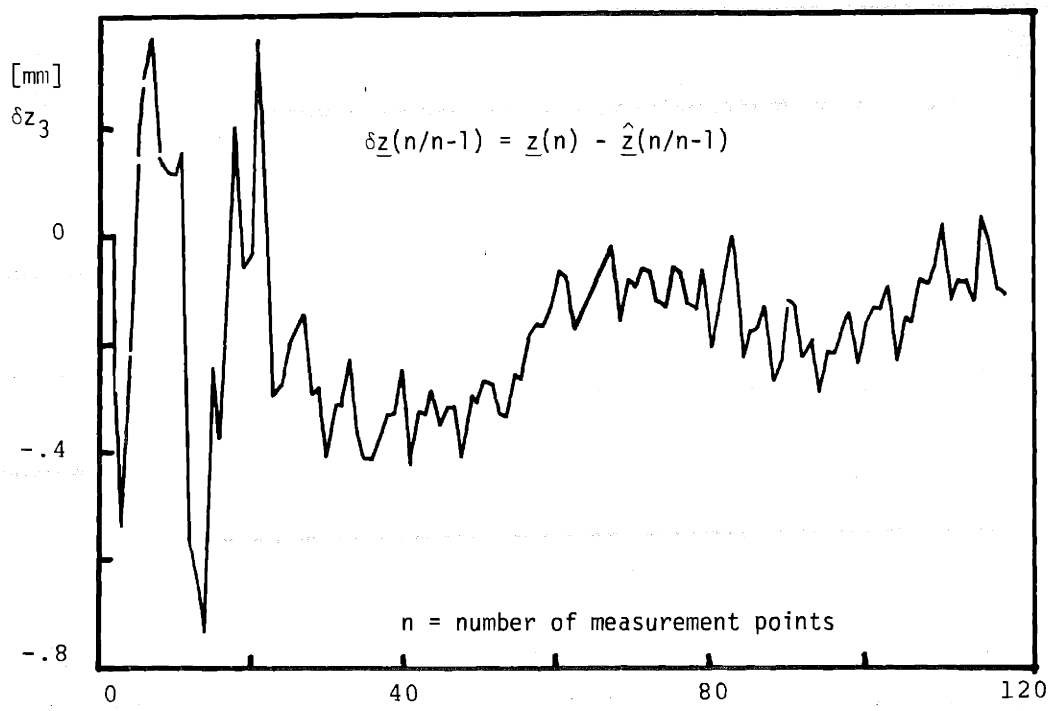


Figure II-15: Time History of Measurement Residues from Smooth Estimator on Experimental Data

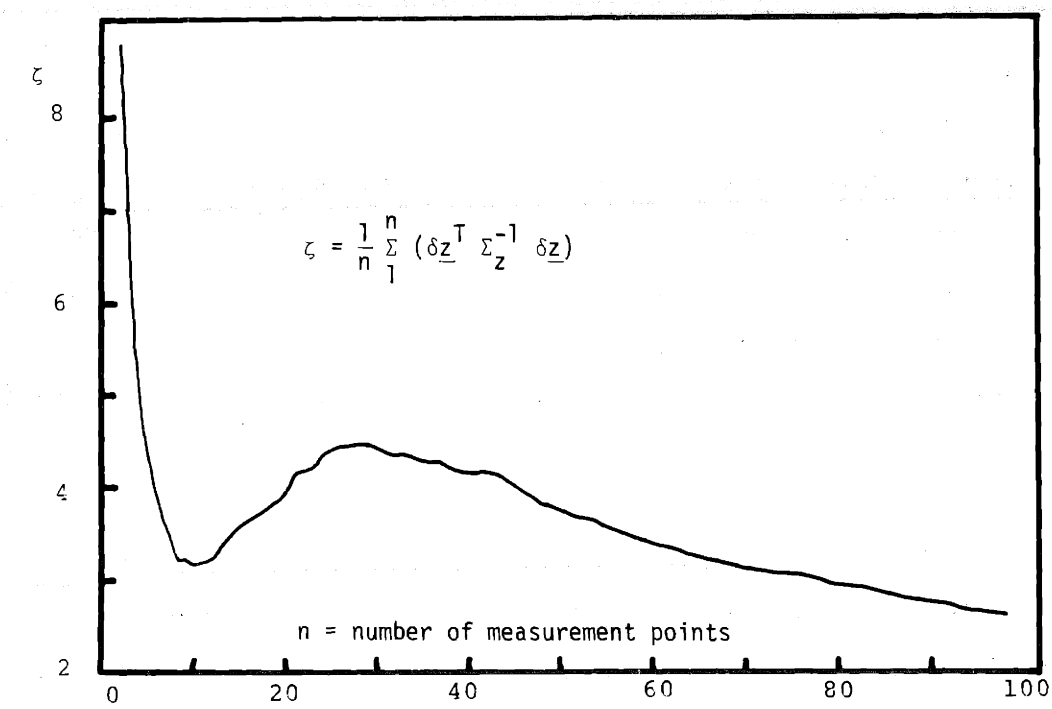


Figure II-16: Normalized Log-Likelihood Function from Experimental Data

on the data generated from the manipulator experiment. The components two and three of the estimated vector for the position between the parts are plotted against the number of measurement points.

Figures II-14 to II-16 show the results of the recursive implementation on the manipulator experiment data. Figure II-14 shows the time history of the one and two components of the estimation of the relative position between the parts; Figure II-15 shows estimation residues of the five-component of the measurement; and Figure II-16 shows the normalized log likelihood function.

The estimator's performance rapidly degraded for large values of measurement noise ($\sqrt{r}$/radius > .1); this was not predicted by the lower bound performance evaluation made in Appendix C. For smaller values of the measurement noise, the predictions of the theoretical performance evaluation were verified by the simulated data experiment (see Figure II-10). On the computer controlled experiment data, the performance of the recursive estimator proved to be better than the performance of the smooth estimator; this was due to the fact that the recursive filter allows for slow time drifts on the bias term to accommodate for incomplete knowledge of the manipulator kinematics.

In summary, the experiments that used simulated data proved that the estimation of the actual position from bias measurements, as generated by the position measurements model, can be realized. The experiments using data generated by the manipulator and big parts gave statistical evidence of the feasibility of the implementation of the estimation procedure in three dimensions, and also gave statistical evidence that the measurements of positions during these experiments conform to the general position model of Expression II-1.

6. SOME REMARKS

The method developed in this chapter uses the touching conditions (Expression II-4) and position measurements while the parts are in contact to determine the actual relative position between parts. The touching conditions were introduced to solve for the bias in the position measurements model postulated in Chapter I. An

alternative method, proposed in Section 2 of Chapter II, consists of an increase in the dimension of the measurement vector through inclusion of different physical measurements, specifically force measurements. This approach should include force-position models that relate the measured forces to the actual relative position between the parts. This modelling is difficult, if not impossible (see Reference II-2).

In Reference II-2, a method was proposed in which the information of two force sensors (one attached to each of the mating parts) in conjunction with position measurement is used to determine the relative position between the force sensors. This method could be used in conjunction with the method developed in this chapter.

Reference I-4 and Reference I-5 approached the transition from gross to fine motion with methods that are conceptually similar to the method developed in this chapter. In all of these methods, position information and force information (for detecting contact) are used in conjunction with geometry information on the mating parts. The main differences between the first two methods and the present method are:

- Where the methods in Reference I-4 and Reference I-5 use a physical search to find the relative position between the parts, the method here processes the information numerically.
- Where the methods in Reference I-4 and Reference I-5 use the geometry information implicitly in the algorithm that does the search, the method here uses the geometry information explicitly.
- Where the methods in Reference I-4 and Reference I-5 use only a discrete number of measurement points, the method here uses the whole time history of measurements.

These differences make the present method:

- More general: only the geometry model has to be changed for a different application instead of changing the search algorithm.
- Potentially more efficient: the use of the whole time history of measurements makes a more powerful "event detection" scheme; the first two methods

rely on forcing events that are easy to detect, while the present method is only used when the transition from gross to fine motion is unsuccessful and corrections are needed.

In summary, the outcome of this chapter is a proof of the technical feasibility for solving the gross to fine motion transition through the use of position and geometry information on the parts. Further refinement of the method developed in this chapter is required in order to prove its engineering feasibility.

The main drawback of the method is its slow speed in its present implementation form: it takes two to three seconds to process every new measurement point. At least one order of magnitude reduction should be achieved before its practical implementation is seriously considered. For the present implementation, the goal was to prove its feasibility and thus no major effort was made to make it speed-efficient.

The satisfactory performance of this position estimation method can be considered as statistical evidence for the validity of the position measurement model postulated in Section 2 of this chapter.

CHAPTER III

FINE MOTION ASSEMBLY

1. INTRODUCTION

In the fine motion assembly trajectory, the parts are nested within their geometries. Changes in the relative position between the parts that deviates from the trajectory needed to assemble the parts will cause them to touch and thus possibly impede further relative motion between them. Under near-perfect position control, fine motion assembly is conceptually straightforward. The use of information generated at the assembly interface to perform fine motion assembly under non-perfect position control is analyzed here.

2. NATURE OF THE INFORMATION

Chapter II concluded that the use of position information for generating the necessary corrections in the assembly trajectory requires the measurement of successive positions. In unsuccessful fine motion assembly, the interaction between the parts will not allow relative motion between them; thus, there are no consecutive relative positions from which to obtain measurements. If successive relative position can be generated, the fine motion assembly is successful and thus corrections are not necessary. Basically, then, position information is not useful in solving fine motion assembly.

In Chapter II, force information was described as being the result of contact between the parts. This chapter will concentrate on using force information to generate the necessary position corrections for successful fine motion assembly.

The study of force information in fine motion assembly is done through the analysis of a two-dimension peg-and-hole insertion. The concepts and results of

the analysis are shown here and the details of the analysis are shown in Appendix B.

3. FINE MOTION ANALYSIS OF A TWO-DIMENSION PEG-AND-HOLE INSERTION

The basis of the analysis is the equilibrium between the force applied on the parts by the positioning device and the reaction forces that are generated between the parts at their contact point. Figure III-1 shows the applied forces as measured on a frame located at the tip of the peg, as well as a particular set of reaction forces at contact points A and B.

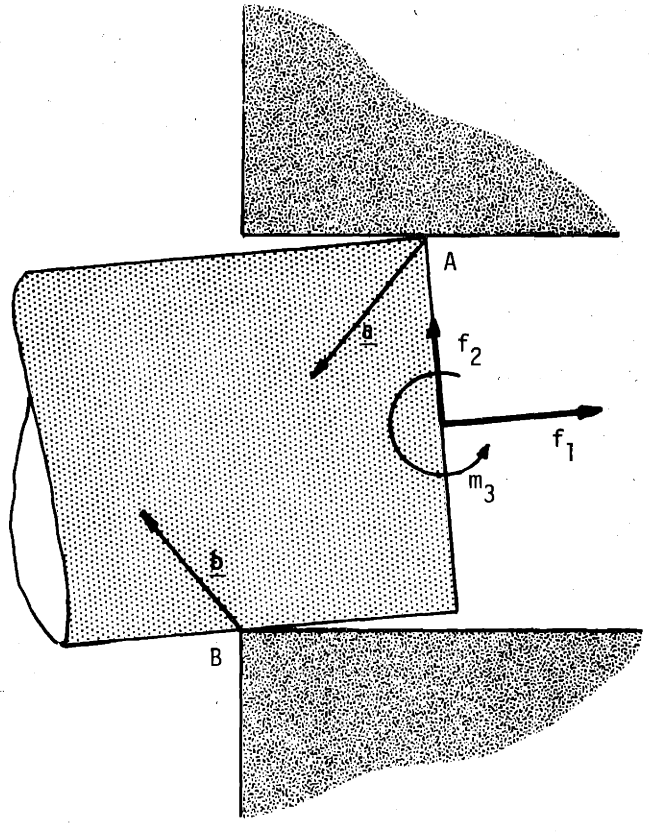


Figure III-1: Applied and Reaction Forces, Notation

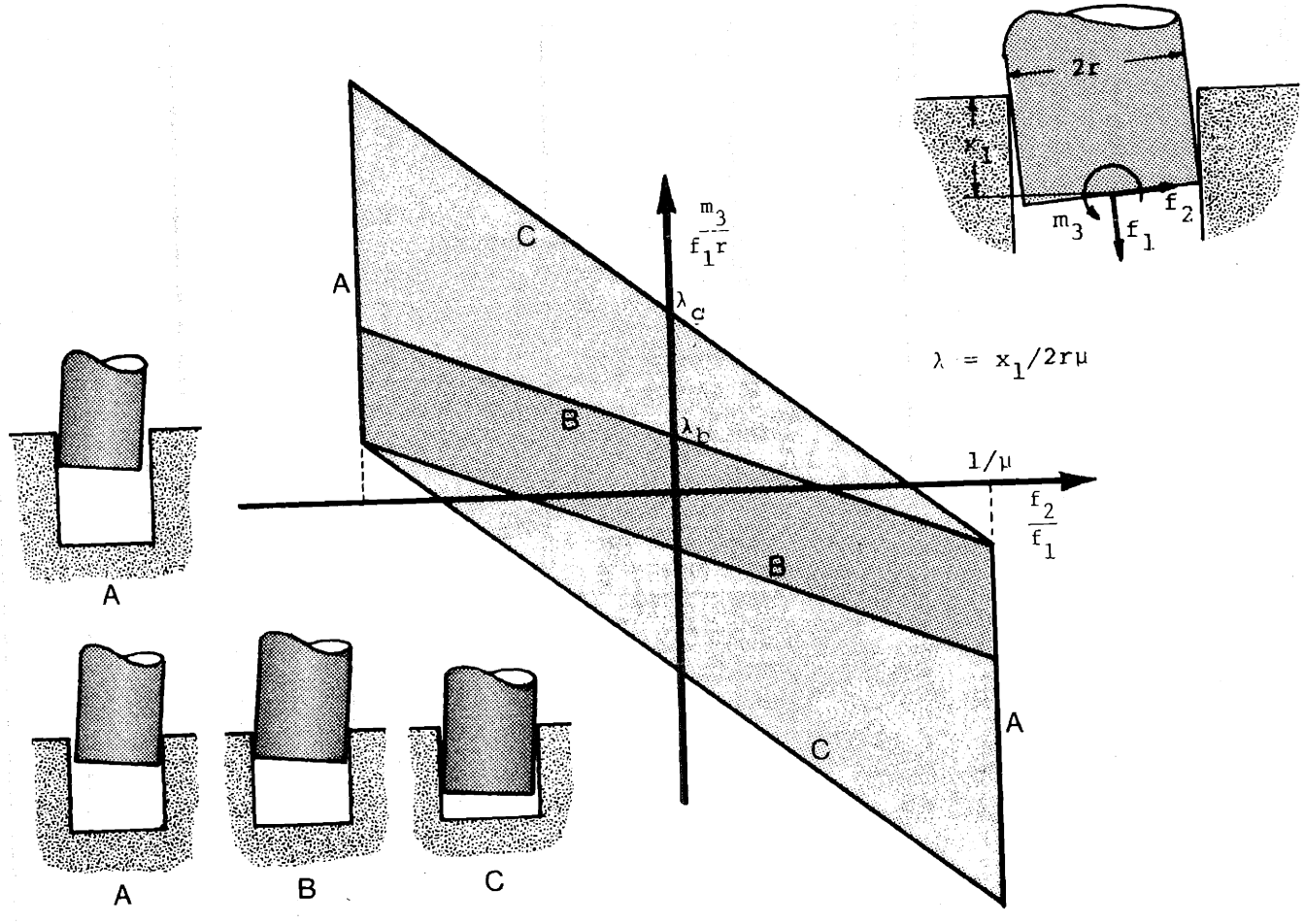


Figure III-2: Sliding Conditions

Proceeding under the assumptions that the process is quasi-static, that the parts are quasi-rigid, and that the contact forces behave according to Coulomb's friction laws, the equilibrium between applied forces and reaction forces, for different configurations yield the conditions on the applied forces that guarantee that the parts will slide. These conditions are summarized in Figure III-2. In essence, these sliding conditions state that the ratio between applied lateral and axial forces and the ratio between applied moment and axial force multiplied by the radius of the hole has to fall inside the region defined by the four lines in the figure. The parameter λ that determines the distance between the lines 1 and 2 and therefore the size of the sliding region is a non-dimensional measure of the penetration of the peg into the hole.

In terms of the fine motion assembly, parts are understood to be quasi-rigid if their deformations under assembly forces are small compared to the trajectory deviations allowed in the fine motion. The deformation of the parts will allow reaction forces in the absence of applied forces due to stored elastic energy in the parts. This phenomenon, shown in Figure III-3, is referred to as wedging, It can only occur for values of λ smaller than unity.

Through the combination of the sliding conditions (Figure III-2) with the wedged condition (Figure III-3), this analysis concludes that:

- the necessary and sufficient conditions to execute the fine motion insertion for values of λ greater than unity is to control the forces to satisfy the sliding conditions of Figure III-2.
- the above condition is only necessary for values of λ smaller than unity. A force level, dependent on the stored elastic energy in the wedged situation, makes that condition also sufficient.

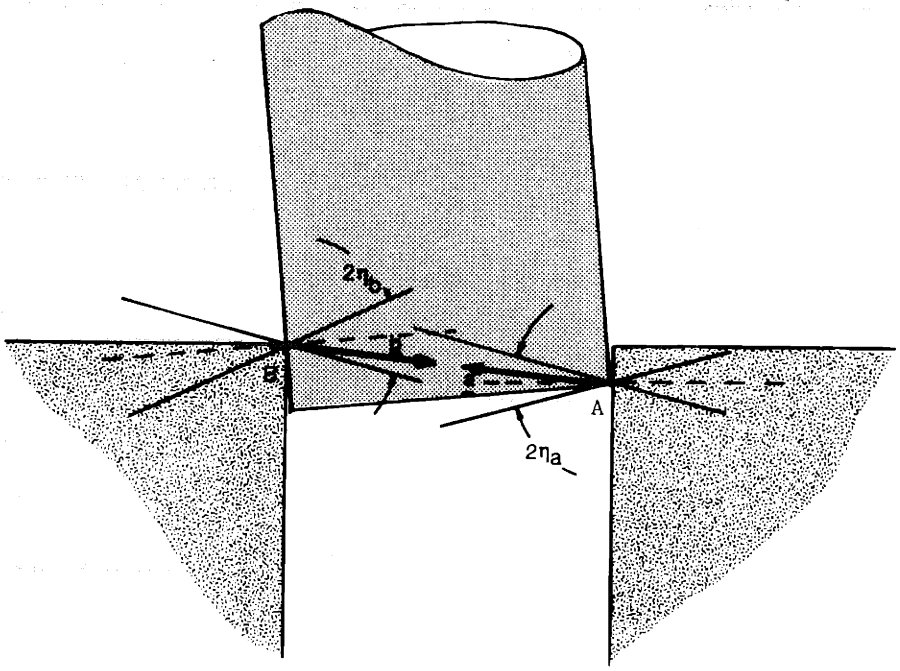


Figure III-3: Reaction Forces in Wedge Configuration

4. IMPLEMENTATION OF THE FINE MOTION CORRECTIONS

The general conclusion of the simple peg-and-hole analysis is that the corrections required for performing fine motion assembly can be generated through the use of only force information. The specific conclusions from that analysis are:

- under perfect force control, a jam-free fine motion insertion will result if the lateral forces and moments are controlled to zero;
- under imperfect control of forces, the acceptable margin of error on the lateral forces and moments becomes wider with large penetration of the peg into the hole (larger λ);
- given near rigid parts, a bigger axial force will widen the margin of acceptable lateral forces and moments.

The correction of position errors between the parts being assembled through the control of forces is conceptually equivalent to the implementation of an impedance between the parts. The implementation of this impedance can be synthesized by all passive elements or by a combination of active and passive means.

The purpose of the passive impedance is to absorb the error in the relative position between the parts, that is, to solve for the difference between the position of the parts and the position where the positioning device requires them to be. This impedance has to be designed so that its deformation will generate forces that satisfy the sliding conditions of Figure III-2.

The active impedance or direct control of assembly forces for the peg-and-hole case will simply consist of controlling the lateral forces and moments to zero while applying a force in the insertion direction. The axial force overcomes the effect of the lateral residual forces and moments due to imperfections in the lateral forces and moments control scheme.

In Appendix B, Section Four and Section Five,a static analysis on the effect of the impedance (a compliance for the static case) on both the active and passive cases concludes that for the peg-and-hole fine motion assembly the best design is achieved by designing the compliance totally symmetric with respect to the tip of the peg. This requires that the compliance will only generate moments when rotated in any direction around the tip of the peg, the moment and the rotation being co-linear, and will only generate forces co-linear with the displacement when deformed in any direction around the tip of the peg. These requirements are not necessary but are very desirable when dealing with difficult assembly problems such as assembly of parts with sharp edges or tight clearance fits. These results can be extended to the interference fit problem by superimposing the force control strategy to the forces caused by the interference.

5. SUMMARY

In summary, this chapter concluded that the fine motion trajectory positioning problem can be solved through the use of force information, at least for the case of peg-and-hole insertion. Position information used in the form described in Chapter II would be useful as a diagnostic and supervisory tool for detecting transitions and problems during the fine motion assembly.

A secondary conclusion, related to the need for controlling the forces and moments, concerns the control through the design of the compliance in between the parts being assembled. This is imperative when dealing with difficult assembly tasks such as assembly of parts with sharp edges, zero clearance assembly, etc.

CHAPTER IV

SUMMARY OF CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

1. CONCLUSIONS

The assumption is made that an assembly system, designed to be flexible and programmable, will have to use positioning means that are inherently inaccurate. The error in the relative position between the parts being assembled that results from these inaccuracies will have to be corrected for in order to achieve the assembly. The use of the information generated during the assembly process to produce the necessary corrections was studied by analyzing the parts mating process.

The parts mating study concluded that both position and force information is needed to correct for the position error between two parts being assembled. High resolution position information is needed to correct for the error in order to perform the transition from gross to fine motion. Force information is needed to maintain contact while gathering position information in the transition from gross to fine motion, and is needed to correct for the error when performing the fine motion assembly trajectory. Information on the geometry of the parts is needed in the processing of the position information during the transition from gross to fine motions and is needed when using the force information in the fine motion trajectory.

2. RECOMMENDATIONS FOR FUTURE WORK

The methods and results developed in Chapter II and Chapter III demonstrate the technical feasibility of the information processing approach to the programmable adaptable assembly problem. Further evaluation of the methods and further development of the necessary system to support the methods are needed to demonstrate the

engineering feasibility of this approach.

Some of the developments in the supporting system needed to implement the methods in this work in a real assembly system are:

- capability of high speed and high resolution position control. Most of the assembly hardware available and in development stages lacks this capability. The development of a limited motion add-on device would make this capability viable on the medium term;
- the parallel processing capability, conceptually described in Reference IV-1, would allow fast real time implementation of control and information processing functions that use the common information generated at the assembly interface;
- full real time implementation of a complete kinematic manipulation package, of the versatility of the one developed in Appendix A;
- dynamic force measurements implementation. These would allow isolation of the effect of inertial forces from the force information generated during the assembly process.

The implementation of the methods developed in this work requires further development in the following areas:

- develop a more generic solution to the geometric modelling of the contact conditions;
- develop the necessary checks and implementations to allow for parallel estimation and identification of alternative touching configurations;
- develop a generic method for generating the nominal assembly path. This would allow a generic way of determining the force control law in the fine motion assembly.

3. SOME REMARKS

The conceptual output of this work is that absolute positioning accuracy in assembly can be replaced, in varying degrees, by high positioning resolution combined with processing of the information generated during the assembly process. Accuracy is measured here in relation to the error allowed by the parts to be assembled. Conceptually, the system could be one of the following:

- an accuracy-based assembly system, in which the expected error between the parts being assembled will allow the mating to proceed without the need for corrections. An example of this is the fixed automation assembly system in widespread use in industry today;
- a medium-accuracy-based assembly system capable of force control, in which the expected position error between the parts being assembled will allow for the transition from gross to fine motion to occur without the need for corrections. The force control capability will allow the system to correct for the position error in the fine motion trajectory. An example of this family of assembly systems is the assembly demonstration of the automotive alternator developed at the C.S. Draper Laboratory (References IV-2 and IV-3);
- a low-accuracy-based assembly system capable of high resolution position control and force control. In this system the expected position error between the parts being assembled will have to be corrected for in order to execute both the transition from gross to fine motion and the fine motion assembly trajectories. Methods of the type developed in this work would have to be implemented in such a system.

In the design of an assembly system for a particular application, the costs associated with the methods based on high resolution positioning and information processing have to be compared with the cost of the position accuracy that they replace and the loss of opportunity cost of the flexibility that they add to the system.

APPENDIX A

KINEMATICS OF RIGID SOLIDS IN SPACE

1. INTRODUCTION

This appendix concentrates on the notations, definitions and relations needed for the manipulator of force and position variables in order to develop a tool to be used in the parts mating study.

In the assembly of rigid parts the following three situations are of special concern:

- The relative position between the two parts being assembled has to be controlled through a controllable positioning device (See Figure A-1);
- The position readings given by a position measuring device have to be translated into relative positions between the parts being assembled;
- The forces and moments generated during the assembly process and measured by a force sensor have to be expressed in terms of the parts being assembled.

The nomenclature and variables needed for the type of manipulations described above are defined in the next section of this appendix; Section Three derives the transformations and Section Four derives the incremental relations for these transformations, both for position variables and force variables. Finally Section Five lists the FORTRAN code for the transformation relations of Sections Three and Four.

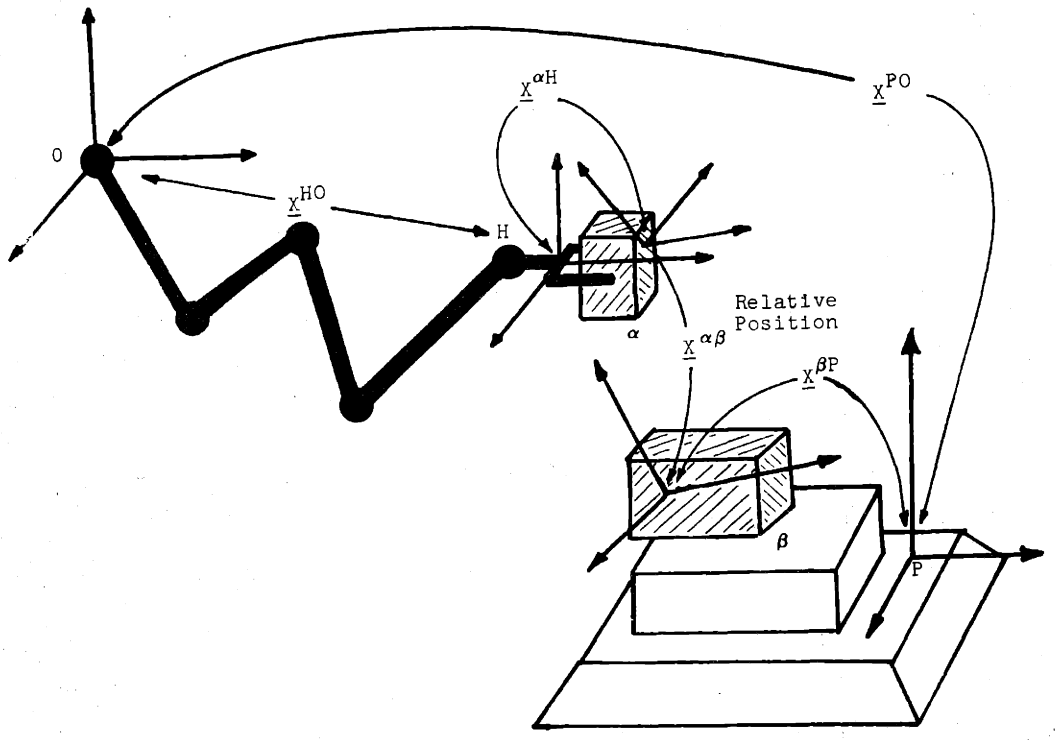


Figure A-1: Schematics of an Assembly System

2. DEFINITIONS, NOTATIONS

The relative position between two rigid bodies in space is expressed in terms of coordinate frames defined in each body; for convenience these frames are chosen tri-orthogonal and right handed. Forces and moments are defined in terms of a co-ordinate frame on the parts or on sensors. In subsection 2-g and 2-h a set of tensor operators are defined; these operators are used in the derivations of Section Three and Four of this appendix.

The upper indexes used on the variables defined in this section make references to the specific frames they relate to. These upper indexes may be dropped in either of the following situations:

- the reference frames in question are self evident;

- the specific reference frames are not important.

(a) Unit coordinate vector "i" on frame "α" (Figure A-2)

$$\underline{e}_i^{\alpha} \qquad i = 1, 2, 3 \qquad \text{A-1}$$

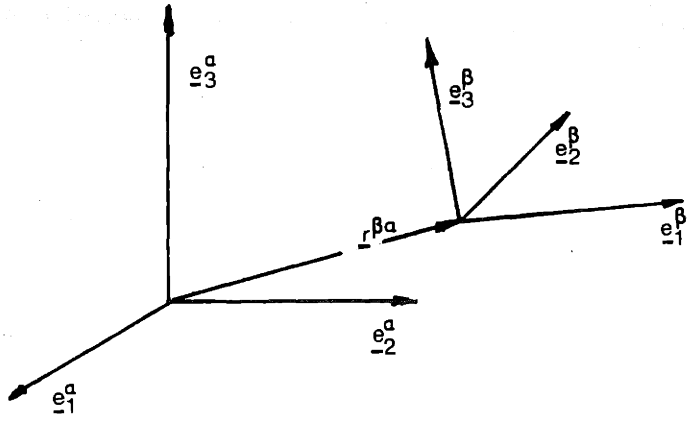


Figure A-2: Vector Notation

(b) Relative displacement vector for frame "β" with respect to frame "α" (Figure A-2)

$$\underline{r}^{\beta\alpha} = r_1^{\beta\alpha}\ \underline{e}_1^{\alpha} + r_2^{\beta\alpha}\ \underline{e}_2^{\alpha} + r_3^{\beta\alpha}\ \underline{e}_3^{\alpha} \qquad \text{A-2}$$

(c) Relative rotation matrix of frame "β" with respect to frame "α" (Figure A-2)

$$A^{\beta\alpha} = \left| a_{ij}^{\beta\alpha} \right| = \left| \underline{e}_i^{\beta} \cdot \underline{e}_j^{\alpha} \right| \qquad \text{A-3}$$

(d) Relative rotation vector of frame "β" with respect to frame "α", in terms of frame "α"

$$\underline{\phi}^{\beta\alpha} = \phi^{\beta\alpha}\, \underline{n}^{\beta\alpha} \qquad \text{A-4}$$

The unit vector $\underline{n}^{\beta\alpha}$ defines the axis around which the frame "α" is rotated onto "β", $\phi^{\beta\alpha}$ is the rotation angle. Also $\underline{n}^{\beta\alpha}$ is the eigenvector of $A^{\beta\alpha}$ for eigenvalue 1 and, $\phi^{\beta\alpha}$ is the argument for the complex pair set of eigenvalues of $A^{\beta\alpha}$.

(e) Generalized position matrix of frame "β" with respect to frame "α"

$$B^{\alpha\beta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \underline{r}^{\alpha\beta} & & A^{\alpha\beta} & \end{bmatrix} \qquad \text{A-5}$$

(f) Generalized relative position vector of frame "β" with respect to frame "α"

$$\underline{x}^{\beta\alpha} = \begin{bmatrix} \underline{r}^{\beta\alpha} \\ \underline{\phi}^{\beta\alpha} \end{bmatrix} \qquad \text{A-6}$$

NOTE: The six dimensional vector $\underline{x}^{\beta\alpha}$ and the four by four matrix $B^{\beta\alpha}$ are two different respresentations of the same information, namely the six dimension relative position between the two frames α and β. The vector $\underline{x}^{\beta\alpha}$ is a compact representation of this information, the matrix $B^{\alpha\beta}$ is used because of its properties in coordinate frames manipulations.

(g) Kronecker delta

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad \text{A-7}$$

(h) Epsilon third order tensor

$$\varepsilon_{ijk} = \begin{cases} 1 \text{ if subindices are in cycle order} \\ 0 \text{ if any subindex is repeated} \\ -1 \text{ if subindices are in inverse cycle order} \end{cases} \qquad \text{A-8}$$

(i) Forces and Moments

$$\underline{f}^{\alpha} = \Sigma_i f_i^{\alpha} \underline{e}_i^{\alpha}$$
$$\underline{m}^{\alpha} = \Sigma_i m_i^{\alpha} \underline{e}_i^{\alpha} \qquad \text{A-9}$$

Forces and moments measured in terms of frame "α"

(j) Generalized force vector

The six dimensional generalized force vector that contains all the information necessary to define the force and moment is defined as follows.

$$\underline{L}^{\alpha} = \begin{bmatrix} \underline{f}^{\alpha} \\ \underline{m}^{\alpha} \end{bmatrix} \qquad \text{A-10}$$

NOTE: In general any vector that is defined only in terms of one frame of reference will have only one upper index.

3. POSITION AND FORCE RELATIONS

The previous section established the notation for force and position variables; this section develops the tool for manipulating these variables. Many of the relations given in this section are listed without derivation or proof; these are generally obtained directly from the definition of the variables involved and/or direct expansion. For more detail the reader is directed to Reference A-1 and Reference A-2.

NOTE: The symbol $\Sigma_{ki\ldots}$ indicates summation over the index k,j, etc. The summation over these indices is done from 1 to 3 unless it is otherwise stated.

(a) Properties of the third order tensor

No proof or explanation of these relations is given here:

$$\Sigma_i \, \epsilon_{ijk} \, \epsilon_{ipq} = \delta_{jp} \, \delta_{kq} - \delta_{jq} \, \delta_{kp} \qquad i = 1,3 \qquad \text{A-11}$$

$$\underline{u} \times \underline{v} = \left\{ \Sigma_{ijk} \, \epsilon_{ijk} \, u_i \, v_j \, \underline{e}_k \right\} \quad i,j,k = 1,2,3 \qquad \text{A-12}$$

$$\Sigma_{jk} \, \epsilon_{ijk} \, a_{jp} \, a_{kq} = \Sigma_r \, \epsilon_{pqr} \, a_{ir} \qquad \text{A-13}$$

$$\left\{ \Sigma_k \, \epsilon_{ijk} \, n_k \right\} = \begin{bmatrix} 0 & n_3 & -n_2 \\ -n_3 & 0 & n_1 \\ n_2 & -n_1 & 0 \end{bmatrix} \qquad \text{A-14}$$

(b) Upper indices permutations

The permutation relations of the upper indices in the rotation vector, rotation matrix, position vector, generalized position matrix, and generalized position vector are direct from the definitions listed in Section Two. Thus:

$$\underline{\phi}^{\alpha\beta} = -\underline{\phi}^{\beta\alpha}$$

$$A^{\alpha\beta} = (A^{\beta\alpha})^T$$

$$\underline{r}^{\alpha\beta} = -A^{\beta\alpha} \, \underline{r}^{\beta\alpha} \qquad \text{A-15}$$

$$B^{\alpha\beta} = (B^{\beta\alpha})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & & & \\ -A^{\alpha\beta} \, \underline{r}^{\alpha\beta} & & (A^{\beta\alpha})^T & \end{bmatrix}$$

$$\underline{x}^{\alpha\beta} = - \begin{bmatrix} A^{\beta\alpha} & 0 \\ 0 & I \end{bmatrix} \underline{x}^{\beta\alpha}$$

These relations are implemented in the subroutines "INVA" that computes the inverse of the matrix rotation A, "INVB" for the inverse of the matrix B and "INVX" that computes the upper indices permutation of the vector $\underline{x}$. These subroutines are shown in Section Five of this appendix.

(c) Multiple frames relations

The necessary relations for doing relative position manipulations are listed below. These relations are direct extensions of the definitions of the position vector, rotation matrix and generalized position matrix.

$$\underline{r}^{\alpha\gamma} = \underline{r}^{\beta\gamma} + A^{\gamma\beta}\ \underline{r}^{\alpha\beta}$$

$$A^{\alpha\gamma} = A^{\alpha\beta} \cdot A^{\beta\gamma} \qquad \text{A-16}$$

$$B^{\alpha\gamma} = B^{\alpha\beta} \cdot B^{\beta\alpha}$$

NOTE: The multiple frame relation for both the matrices A and B is simply done by concantenating the upper indices as shown in Expression A-16. These relations allow us to express the position of any frame in terms of relative position of combination of frames. For example, from Figure A-1 the generalized position matrix between part α and β can be written as follows:

$$B^{\alpha\beta} = B^{\alpha H} \cdot B^{HO} \cdot B^{OP} \cdot B^{P\beta} \qquad \text{A-17}$$

where:

- H is the frame located on the gripper of the positioning device;
- O is the reference frame;
- P is the frame of the jig;
- α and β are the frames located on part α and β respectively.

If B^{HO} is our controllable position, controlled through the positioning device, Expression A-17 gives the relation for the resulting controlled position $B^{\alpha\beta}$. If, on the other hand, B^{HO} is the reading of the position measuring device, then Expression A-17 gives the relative position between α and β.

(d) Rotation vector to rotation matrix transformation

Expression A-17 gives us the tool for concantenating relative position between frames in terms of the generalized position matrix B. It is desirable to perform this operation with the generalized position vector $\underline{x}$ since it and the matrix B, as noted in Section Two of this appendix, are different representations of the same position information. This is done by first computing the transformation between the rotation vector $\underline{\phi}$ and the rotation matrix A. The next subsection finds the inverse of this transformation and the subsequent subsection integrates this transformation into the transformation of the vector $\underline{x}$ to and from the matrix B.

The expression for the transformation of the three-dimensional rotation vector $\underline{\phi}^{\beta\alpha}$ to the three by three rotation matrix $A^{\beta\alpha}$ is found by first computing the rotation of an arbitrary vector $\underline{u}$ around a unit vector $\underline{n}$ on an angle ϕ (Figure A-3).

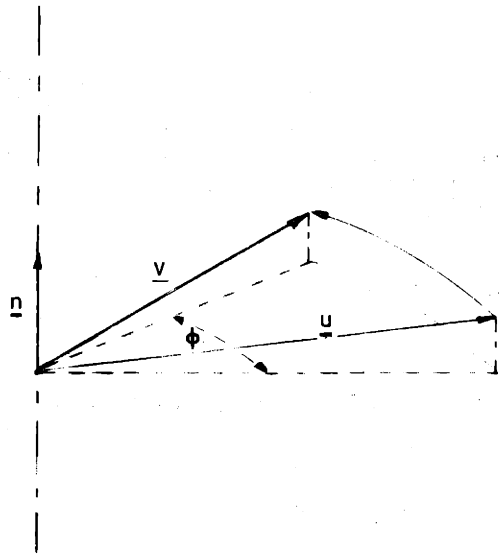


Figure A-3: Rotation of a Vector Around an Axis

This expression used on all three coordinate vectors $\underline{e}_i^{\alpha}$ of the frame α yields the direction cosines of the coordinate vectors of the resulting rotated frame β; these direction cosines constitute the elements for the rotation matrix $A^{\beta\alpha}$ between the two frames.

The increment of the vector $\underline{u}$ product of an incremental rotation $d\phi$ around the unit vector $\underline{n}$ is given by:

$$d\underline{u} = \underline{n} \times \underline{u}\, d\phi = \left\{ \Sigma_k\, \epsilon_{ijk}\, n_k \right\} \underline{u}\, d\phi \qquad \text{A-18}$$

Integrating Expression A-18 and denoting $\underline{v}$, the resulting vector, we have:

$$\underline{v} = \exp \left[\left\{ \Sigma_k\, \epsilon_{ijk}\, n_k \right\} \phi \right] \underline{u} \qquad \text{A-19}$$

Replacing $\underline{u}$ and $\underline{v}$ for $\underline{e}_i^{\alpha}$ and $\underline{e}_i^{\beta}$ respectively, denoting the vector $\underline{n}$ by $\underline{n}^{\beta\alpha}$, denoting the angle ϕ as $\phi^{\beta\alpha}$ and using the Definition 2-e and Definition 2-d for the rotation matrix and vector, respectively (Expression A-3 and Expression A-4), we get:

$$A^{\beta\alpha} = \left\{ \underline{e}_i^{\beta} \cdot \underline{e}_j^{\alpha} \right\} = \exp \left[\left\{ \Sigma_k\, \epsilon_{ijk}\, \phi_k^{\beta\alpha} \right\} \right] \qquad \text{A-20}$$

The individual elements of the matrix $A^{\beta\alpha}$ are obtained through expansion of the matrix exponentiation of the above expression. Thus:

$$a_{ij}^{\beta\alpha} = \cos \phi^{\beta\alpha} \delta_{ij} + (1 - \cos \phi^{\beta\alpha}) n_i^{\beta\alpha}\, n_j^{\beta\alpha} + \sin \phi^{\beta\alpha}\, \Sigma_k\, \epsilon_{ijk}\, n_k^{\beta\alpha} \qquad \text{A-21}$$

In terms of the vector $\underline{\phi}$

A-22

$$a_{ij}^{\beta\alpha} = \cos\phi\ \delta_{ij} + \frac{1 - \cos\phi^{\beta\alpha}}{(\phi^{\beta\alpha})^2}\phi_i^{\beta\alpha}\ \phi_j^{\beta\alpha} + \frac{\sin\phi^{\beta\alpha}}{\phi^{\beta\alpha}}\Sigma_k\ \epsilon_{ijk}\ \phi_k^{\beta\alpha}$$

This transformation will be denoted by:

$$A^{\beta\alpha} = A^{\beta\alpha}(\underline{\phi}^{\beta\alpha}) \qquad \text{A-23}$$

A subroutine that computes this expression is shown in Section Five of this appendix under the subroutine name "FITOA". This subroutine works as follows:

- given the vector $\underline{\phi}^{\beta\alpha}$, compute the angle $\underline{\phi}^{\beta\alpha}$ through the following expression:

$$\phi = \sqrt{\Sigma_i\ \phi_i^{\beta\alpha}\ \phi_i^{\beta\alpha}} \qquad \text{A-24}$$

- compute the unit vector $\underline{n}^{\beta\alpha}$;

$$n_i^{\beta\alpha} = \phi_i^{\beta\alpha} / \phi^{\beta\alpha} \qquad \text{A-25}$$

- compute the element of matrix $A^{\beta\alpha}$ through Expression A-21. This subroutine has a provision that avoids numerical errors when dealing with small angles.

(e) Rotation matrix to rotation vector transformation

The expression for the rotation angle is obtained through the trace of the matrix $A^{\beta\alpha}$; in Expression A-21 adding over the index i when i equals j we get:

$$\Sigma_i \, a_{ii}^{\beta\alpha} = 2 \cos \phi^{\beta\alpha} + 1 \qquad \text{A-26}$$

Thus the angle is given by:

$$\phi^{\beta\alpha} = \arccos \left[1/2 \, (\Sigma_i \, a_{ii}^{\beta\alpha} - 1)\right] \qquad \text{A-27}$$

The components of the vector $\underline{n}^{\beta\alpha}$ are computed by subtracting off-diagonal terms of the matrix A located symmetrically opposed. In detail this is done by multiplying the components of matrix $A^{\beta\alpha}$, given by Expression A-21, by the third order tensor, defined in Expression A-8, and doing a double addition on their indices, using Relation A-11,we have:

A-28

$$\Sigma_{ij} \, \epsilon_{pij} \, a_{ij}^{\beta\alpha} = \sin \phi^{\beta\alpha} \, \Sigma_{ijk} \, \epsilon_{pij} \, \epsilon_{ijk} \, n_k^{\beta\alpha} = 2 n_p^{\beta\alpha} \sin \phi^{\beta\alpha}$$

and

$$n_p^{\beta\alpha} = \frac{1}{2 \sin \phi^{\beta\alpha}} \, \Sigma_{ij} \, \epsilon_{pij} \, a_{ij}^{\beta\alpha} \qquad \text{A-29}$$

and the components of the vector $\underline{\phi}$ are given by:

$$\phi_p^{\beta\alpha} = \frac{\phi^{\beta\alpha}}{2 \sin \phi^{\beta\alpha}} \, \Sigma_{ij} \, \epsilon_{pij} \, a_{ij}^{\beta\alpha} \qquad \text{A-30}$$

This transformation is denoted as:

$$\underline{\phi}^{\beta\alpha} = \underline{\phi}^{\beta\alpha}\,(A^{\beta\alpha}) \qquad \text{A-31}$$

The subroutine "ATOFI" listed in Section Five computes this transformation through the following steps:

- computes the angle ϕ through Expression A-27. This angle is chosen in the range zero to π;
- computes the sine of ϕ and through Expression A-29 calculates the components of the vector $\underline{n}$;
- the rotation vector is computed using Expression A-25. This routine has a provision for small angles rotations.

(f) Transformations between matrix B and vector $\underline{x}$.

The transformation from the vector $\underline{x}^{\alpha\beta}$ to the matrix $B^{\alpha\beta}$ is done in two steps as follows:

- the position partition $\underline{r}^{\alpha\beta}$ on the vector $\underline{x}$ is inverted by using Expression A-15;
- the rotation angle partition of $\underline{x}^{\alpha\beta}$ is transformed into the matrix $A^{\alpha\beta}$ through the use of Expression A-21.

This transformation is denoted by:

$$B^{\alpha\beta} = B^{\alpha\beta}(\underline{x}^{\alpha\beta}) \qquad \text{A-32}$$

This is implemented in the subroutine XTOB listed in Section Five. This subroutine uses subroutine FITOA and INVB. This is denoted by:

$$\underline{x}^{\alpha\beta} = \underline{x}^{\alpha\beta}(B^{\alpha\beta}) \qquad \text{A-33}$$

This transformation is implemented in subroutine BTOX; this subroutine uses subroutines ATOFI and INVB. A listing of this subroutine is given in Section Five.

(g) Force and moment transformation

By directly applying the definition of the rotation matrix $A^{\beta\alpha}$ and the definition of moment on forces, the frame transformation of the generalized force vector L is given by:

$$\underline{L}^{\beta} = \begin{bmatrix} A^{\beta\alpha} & 0 \\ A^{\beta\alpha} \; \Sigma_k \; \varepsilon_{ijk} \; r_k^{\beta\alpha} & A^{\beta\alpha} \end{bmatrix} \underline{L}^{\alpha} \qquad \text{A-34}$$

or, in short term:

$$\underline{L}^{\beta} = P^{\beta\alpha} \, \underline{L}^{\alpha} \qquad \text{A-35}$$

4. KINEMATICS OF INCREMENTAL MOTIONS

A tool for manipulating incremental position variables is required for both generalized positioning error analysis and information processing techniques. This section develops the algebra of incremental position variables. Part(a) of this section formulates the kinematics of incremental rotations and Part(b)formulates the kinematics of generalized incremental positions; Part(c)analyzes the effect of position increments on generalized forces.

Notation

A general tensor V function of a tensor U has been denoted by (e.g. Expression A-31, Expression A-33):

$$V = V(U) \qquad \text{A-36}$$

In general, the relation between the increments of U and V will be denoted by:

$$dV = \frac{D\dot{V}}{DU}\, dU \qquad \text{A-37}$$

Where the tensor $\frac{DV}{DU}$ transforms the increment of U into the increment of V. The order of the tensor $\frac{DV}{DU}$ is equal to the sum of the orders of the tensors U and V.

(a) Incremental Rotations

The incremental relative rotation between two frames can be expressed in terms of:

(i) The incremental change of the relative rotation matrix A between the frames; this increment is denoted here by the matrix dA.

(ii) The incremental change in the relative rotation vector $\underline{\phi}$ between the frames; this incremental vector is denoted here by $d\underline{\phi}$

(iii) The instanteous incremental rotation vector between the frames; the notation used for this increment is $d\underline{\omega}$

The instanteous incremental rotation vector $d\underline{\omega}$ is used for physical interpretations of incremental analysis. The vector $d\underline{\phi}$ and the matrix dA measure the incremental change in the respective representation of rotation information between the frames concerned.

First, the transformation between $d\underline{\omega}$ and $d\underline{\phi}$ is constructed; the transformation between dA and $d\underline{\phi}$ is derived from the other two transformations.

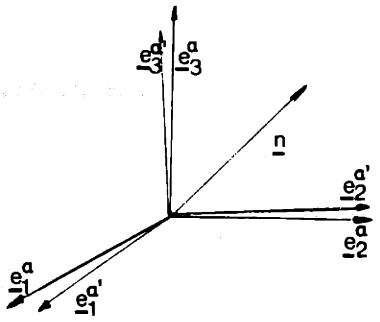


Figure A-4: Incrementally Rotated Frame

In Figure A-4 frame α is rotated to α' by the incremental angle $d\underline{\omega}$ then, using Expression A-16, the rotation matrix between frames α' and β is given by:

$$A^{\alpha'\beta} = A^{\alpha'\alpha}\, A^{\alpha\beta}$$

For an incremental rotation $d\underline{\omega}$, the increment of the rotation matrix $A^{\alpha\beta}$ is defined by:

$$dA^{\alpha\beta} = A^{\alpha'\beta} - A^{\alpha'\alpha} = (A^{\alpha'\alpha} - I)A^{\alpha\beta} \qquad \text{A-38}$$

From Expression A-20, for a small angle $d\underline{\omega}$:

$$A^{\alpha'\alpha} = \exp(\Sigma_k \, \epsilon_{ijk} \, d\omega_k) = I + \left\{ \Sigma_k \, \epsilon_{ijk} \, d\omega_k \right\} \qquad \text{A-39}$$

Thus, combining with A-38:

$$dA^{\alpha\beta} = \left\{ \Sigma_k \, \epsilon_{ijk} \, d\omega_k \right\} \, A^{\alpha\beta} \qquad \text{A-40}$$

The incremental rotation vector $d\underline{\omega}$ will be denoted by $d\underline{\omega}^{\beta\alpha}$; this angle represents the instanteous incremental rotation of the frame α with respect to frame β expressed in terms of the frame α.

Using the above notation in Expression A-40 and writing this expression in terms of its components we have:

$$dA^{\alpha\beta} = \left\{ \Sigma_k \, \epsilon_{ijk} \, d\omega_k^{\beta\alpha} \right\} \, A^{\alpha\beta} \qquad \text{A-41}$$

$$da_{ij}^{\alpha\beta} = \Sigma_{kp} \, \epsilon_{ipk} \, a_{pj}^{\alpha\beta} \, d\omega_k^{\alpha\beta}$$

The expression of $d\underline{\omega}^{\beta\alpha}$ in terms of $dA^{\alpha\beta}$ is obtained by multiplying both sides of Expression A-41 by $a_{qj}^{\alpha\beta}$ and ϵ_{iqt} and adding over the indices q, j and i. This gives:

$$d\omega_t^{\beta\alpha} = 1/2 \, \Sigma_{qji} \, \epsilon_{tiq} \, a_{qj}^{\alpha\beta} \, da_{ij}^{\alpha\beta} \qquad \text{A-42}$$

Repeating the above procedure on the incremental rotation of the frame β with respect to the frame α we get:

$$dA^{\alpha\beta} = A^{\alpha\beta} \left\{ \Sigma_k \; \epsilon_{ijk} \; d\omega_k^{\alpha\beta} \right\} \qquad \text{A-43}$$

thus:

$$d\underline{\omega}^{\alpha\beta} = -A^{\beta\alpha} \; d\omega^{\beta\alpha} \qquad \text{A-44}$$

For notation convenience Expression A-44, Expression A-41 and Expression A-42 will be written as:

$$dA^{\alpha\beta} = \frac{DA^{\alpha\beta}}{D\underline{\omega}^{\beta\alpha}} \; d\underline{\omega}^{\beta\alpha}$$

$$d\underline{\omega}^{\beta\alpha} = \frac{D\omega^{\beta\alpha}}{DA^{\alpha\beta}} \; dA^{\alpha\beta} \qquad \text{A-45}$$

$$d\underline{\omega}^{\alpha\beta} = \frac{D\omega^{\alpha\beta}}{D\omega^{\beta\alpha}} \; d\underline{\omega}^{\beta\alpha}$$

where the third order tensors of the above expressions are given by:

$$\frac{DA^{\alpha\beta}}{D\omega^{\beta\alpha}} = \left[\Sigma_p \; \epsilon_{ipk} \; a_{pj}^{\alpha\beta} \right] \qquad \text{A-46}$$

$$\frac{D\underline{\omega}^{\alpha\beta}}{DA^{\alpha\beta}} = \left[\Sigma_p \ \epsilon_{ijp} \ a^{\alpha\beta}_{pk}\right]$$

A-46

$$\frac{D\underline{\omega}^{\alpha\beta}}{D\underline{\omega}^{\beta\alpha}} = -A^{\beta\alpha}$$

These three tensors are implemented in subroutines DWDA, DADW DWDW,respectively. The listing of these subroutines are given in Section Five of this appendix.

The rotation information between the frames α and β has been expressed in terms of the vector angle $\underline{\phi}^{\alpha\beta}$. The next step is to express the incremental rotation between both frames in terms of the increment of the vector angle $\underline{\phi}^{\alpha\beta}$. The relation is found by first computing the increments of the unit vector $\underline{n}^{\alpha\beta}$, defined in Expression A-4, and the increment of the angle $\underline{\phi}^{\alpha\beta}$.

We know that the unit vector n is the eigenvector of the matrix A for unit eigenvalue. Thus:

$$A\underline{n} = \underline{n} \qquad \text{A-47}$$

differentiating both sides:

$$dA\underline{n} + Ad\underline{n} = d\underline{n}$$

Using Relation A-21 we have:

$$d\underline{\omega} \times \underline{n} = (\cos\phi - 1)d\underline{n} + \sin\phi \ d\underline{n} \times \underline{n} \qquad \text{A-48}$$

Thus, equating the increments $d\underline{n}$ and $d\underline{\omega}$ respectively we have:

$$d\underline{n} = 1/2 \left\{ \frac{\sin \phi}{\cos \phi - 1} (d\underline{\omega} \cdot \underline{n}\, \underline{n} - d\underline{\omega}) - d\underline{\omega} \times \underline{n} \right\} \qquad \text{A-49}$$

and

$$d\underline{\omega} = (\cos \phi - 1)\, \underline{n} \times d\underline{n} + \sin \phi\, d\underline{n} + d\phi\, \underline{n} \qquad \text{A-50}$$

Since

$$\underline{\phi} = \phi\, \underline{n}$$

then

$$d\underline{\phi} = d\phi\, \underline{n} + \phi\, d\underline{n} \qquad \text{A-51}$$

and

$$\phi = \sqrt{\Sigma_i \phi_i \phi_i}$$

then

$$d\phi = \frac{1}{\phi} \Sigma_i \phi_i d\phi_i \qquad \text{A-52}$$

Combining Expression A-49, Expression A-50, Expression A-51 and Expression A-52, we get for the vector $d\underline{\omega}$, in terms of the vector $d\underline{\phi}$:

A-53

$$d\underline{\omega} = \frac{\cos\phi - 1}{\phi^2}\ \underline{\phi} x d\underline{\phi} + \frac{\sin\phi}{\phi}\ d\underline{\phi} + \frac{1}{\phi^2}\left[1 - \frac{\sin\phi}{\phi}\right]\underline{\phi}$$

or in terms of components

A-54

$$d\omega_i = \Sigma_j\left[\frac{\sin\phi}{\phi}\,\delta_{ij} + 1/2\left(1 - \frac{\sin\phi}{\phi}\right)\phi_i\phi_j - \frac{\cos\phi - 1}{\phi^2}\,\Sigma_k\,\epsilon_{ijk}\,\phi_k\right] d\phi_j$$

And, from Expression A-49, Expression A-51 and Expression A-52, the vector $d\underline{\phi}$ in terms of the increment $d\underline{\omega}$ is given by:

$$d\underline{\phi} = \left(1/\phi^2 + \frac{\sin\phi}{2\phi(\cos\phi - 1)}\right)\underline{\phi}\cdot d\underline{\omega}\ \ \underline{\phi} - \frac{\phi\sin\phi}{2(\cos\phi - 1)}\ d\underline{\omega} + 1/2\ \underline{\phi} x d\underline{\omega} \qquad \text{A-55}$$

or in terms of components:

A-56

$$d\phi_i = \Sigma_j\left\{\phi_i\,\phi_j\left(1/\phi^2 + \frac{\sin\phi}{2\phi(\cos\phi - 1)}\right) - \frac{\phi\sin\phi}{2(\cos\phi - 1)}\,\delta_{ij} - 1/2\,\Sigma_k\,\epsilon_{ijk}\,\phi_k\right\} d\omega_j$$

The incremental transformation of Expression A-54 and Expression A-56 are written in short form by:

$$d\underline{\phi}^{\alpha\beta} = \frac{D\underline{\phi}^{\alpha\beta}}{D\underline{\omega}}\ d\underline{\omega}^{\beta\alpha}$$

A-57

$$d\underline{\omega}^{\beta\alpha} = \frac{D\underline{\omega}^{\beta\alpha}}{D\underline{\phi}^{\alpha\beta}}\ d\underline{\phi}^{\alpha\beta}$$

where:

A-58

$$\frac{D\underline{\omega}^{\beta\alpha}}{D\underline{\phi}^{\alpha\beta}} = \left\{ \frac{\sin\phi}{\phi}\,\delta_{ijk} + 1/\phi^2 \left(1 - \frac{\sin\phi}{\phi}\right) \phi_i^{\alpha\beta}\,\phi_j^{\alpha\beta} - \frac{\cos\phi - 1}{\phi^2}\,\Sigma_k\,\varepsilon_{ijk}\,\phi_k^{\alpha\beta} \right\}$$

$$\frac{D\underline{\phi}^{\alpha\beta}}{D\underline{\omega}^{\beta\alpha}} = \left\{ \phi_i^{\alpha\beta}\,\phi_j^{\alpha\beta} \left(1/\phi^2 + \frac{\sin\phi}{2\phi(\cos\phi - 1)}\right) - \frac{\phi\sin\phi}{2(\cos\phi - 1)}\,\delta_{ij} - 1/2\,\Sigma_k\,\varepsilon_{ijk}\,\phi_k^{\alpha\beta} \right\}$$

Remarks:

Combining Expression A-45 and Expression A-57 we have:

$$dA^{\alpha\beta} = \frac{DA^{\alpha\beta}}{D\underline{\phi}^{\alpha\beta}}\; d\underline{\phi}^{\alpha\beta}$$

A-59

$$d\underline{\phi}^{\alpha\beta} = \frac{D\underline{\phi}^{\alpha\beta}}{DA^{\alpha\beta}}\; dA^{\alpha\beta}$$

with

$$\frac{DA^{\alpha\beta}}{D\underline{\phi}^{\alpha\beta}} = \frac{DA^{\alpha\beta}}{D\underline{\omega}^{\beta\alpha}}\;\frac{D\underline{\omega}^{\beta\alpha}}{D\underline{\phi}^{\alpha\beta}}$$

A-60

$$\frac{D\underline{\phi}^{\alpha\beta}}{DA^{\alpha\beta}} = \frac{D\underline{\phi}^{\alpha\beta}}{D\underline{\omega}^{\beta\alpha}}\;\frac{D\underline{\omega}^{\beta\alpha}}{DA^{\alpha\beta}}$$

(b) Generalized position increments

We are interested in analyzing the effect caused by a small variation of the relative position between any two frames on the relative position between the frames of interest. From Relation A-16 we know:

$$B^{\alpha\delta} = B^{\alpha\beta}\, B^{\beta\gamma}\, B^{\gamma\delta}$$

Differentiating at both sides assuming $B^{\alpha\beta}$ and $B^{\gamma\delta}$ fixed, we have:

$$dB^{\alpha\delta} = B^{\alpha\beta}\, dB^{\beta\gamma}\, B^{\alpha\delta} \qquad \text{A-61}$$

Separating the rotation matrix partition of the above expression yields:

$$dA^{\alpha\delta} = A^{\alpha\beta}\, dA^{\beta\alpha}\, A^{\gamma\delta} \qquad \text{A-62}$$

Using the incremental form of the rotation matrix and Property A-13, we get:

$$d\omega_k^{\delta\alpha} = \Sigma_i\, a_{ki}^{\alpha\beta}\, d\omega_i^{\gamma\beta} \qquad \text{A-63}$$

Or, in terms of the matrix notation:

$$d\underline{\omega}^{\delta\alpha} = A^{\alpha\beta}\, d\underline{\omega}^{\gamma\beta} \qquad \text{A-64}$$

Thus, using the tensor transformation notation for incremental motions:

$$\frac{D\underline{\omega}^{\delta\alpha}}{D\underline{\omega}^{\gamma\beta}} = \left\{ a_{ij}^{\alpha\beta} \right\} = A^{\alpha\beta} \qquad \text{A-65}$$

Separating the displacement partition of Expression A-61:

$$d\underline{r}^{\delta\alpha} = A^{\alpha\beta}(d\underline{r}^{\gamma\beta} + dA^{\beta\gamma}\ \underline{r}^{\beta\gamma}) \qquad \text{A-66}$$

Using the expression for the incremental rotation matrix we have:

$$d\underline{r}^{\delta\alpha} = A^{\alpha\beta}\ \ d\underline{r}^{\gamma\beta} - A^{\beta\gamma}\ \ \Sigma_k\ \epsilon_{ijk}\ r_k^{\delta\gamma}\ \ A^{\gamma\beta}\ d\underline{\omega}^{\gamma\beta} \qquad \text{A-67}$$

Thus using the notation defined at Expression A-37, we have

$$d\underline{r}^{\delta\alpha} = \frac{D\underline{r}^{\delta\alpha}}{D\underline{r}^{\gamma\beta}}\ d\underline{r}^{\gamma\beta} \qquad \text{A-68}$$

$$d\underline{r}^{\delta\alpha} = \frac{D\underline{r}^{\delta\alpha}}{D\underline{\omega}^{\gamma\beta}}\ d\underline{\omega}^{\gamma\beta} \qquad \text{A-69}$$

where:

$$\frac{D\underline{r}^{\delta\alpha}}{D\underline{r}^{\gamma\beta}} = A^{\alpha\beta} \qquad \text{A-70}$$

$$\frac{D\underline{r}^{\delta\alpha}}{D\underline{\omega}^{\gamma\beta}} = -A^{\alpha\gamma} \left\{ \Sigma_k\ \epsilon_{ijk}\ r_k^{\delta\gamma} \right\} A^{\gamma\beta} \qquad \text{A-71}$$

Using Relation A-58, we can express all in terms of the increments of the rotation vector $\underline{\phi}$. Thus:

$$d\underline{\phi}^{\delta\alpha} = \frac{D\underline{\phi}^{\delta\alpha}}{D\underline{\phi}^{\gamma\beta}} d\underline{\phi}^{\gamma\beta} \qquad \text{A-72}$$

$$d\underline{r}^{\delta\alpha} = \frac{D\underline{r}^{\delta\alpha}}{D\underline{\phi}^{\gamma\beta}} d\underline{\phi}^{\gamma\beta}$$

where:

$$\frac{D\underline{\phi}^{\delta\alpha}}{D\underline{\phi}^{\gamma\beta}} = \frac{D\underline{\phi}^{\delta\alpha}}{D\underline{\omega}^{\delta\alpha}} \cdot \frac{D\underline{\omega}^{\delta\alpha}}{D\underline{\omega}^{\gamma\beta}} \cdot \frac{D\underline{\omega}^{\gamma\beta}}{D\underline{\phi}^{\gamma\beta}} \qquad \text{A-73}$$

$$\frac{D\underline{r}^{\delta\alpha}}{D\phi^{\gamma\beta}} = \frac{D\underline{r}^{\delta\alpha}}{D\underline{\omega}^{\gamma\beta}} \frac{D\underline{\omega}^{\gamma\beta}}{D\underline{\phi}^{\gamma\beta}}$$

If the increment of the generalized position vector $\underline{x}$ is constructed by the increments of $\underline{r}$ and $\underline{\phi}$ we can write:

$$d\underline{x}^{\delta\alpha} = \begin{bmatrix} d\underline{r}^{\delta\alpha} \\ d\underline{\phi}^{\delta\alpha} \end{bmatrix} = \frac{D\underline{x}^{\delta\alpha}}{D\underline{x}^{\gamma\beta}} d\underline{x}^{\gamma\beta} \qquad \text{A-74}$$

where:

$$\frac{D\underline{x}^{\delta\alpha}}{D\underline{x}^{\gamma\beta}} = \begin{bmatrix} \frac{D\underline{r}^{\delta\alpha}}{D\underline{r}^{\gamma\beta}} & \frac{D\underline{r}^{\delta\alpha}}{D\underline{\phi}^{\gamma\beta}} \\ 0 & \frac{D\phi^{\delta\alpha}}{D\underline{\phi}^{\gamma\beta}} \end{bmatrix} \qquad \text{A-75}$$

The tensor of Expression A-75 is implemented in subroutine DXDX; this subroutine uses subroutines DADW, DWDA, DFIDW and DWDFI that implement Expressions A-46 and A-58. These subroutines are listed in Section Five of this appendix.

(c) Compliance and stiffness matrices

In general, a linear elastic structure will deform when loaded with moments and forces. The relation between the deformations and forces measured in the same reference frame "α" is written:

$$\begin{bmatrix} \delta\underline{r} \\ \delta\underline{\omega} \end{bmatrix}^{\alpha} = C^{\alpha}\, \underline{L}^{\alpha}$$

$$\underline{L}^{\alpha} = K^{\alpha} \begin{bmatrix} \delta\underline{r} \\ \delta\underline{\omega} \end{bmatrix}^{\alpha} \qquad \text{A-76}$$

$$K^{\alpha} = (C^{\alpha})^{-1}$$

where C is the compliance matrix and K is the stiffness matrix.

The transformation of the matrices from α to frame β is given by (See Expression A-34 and Expression A-35):

$$C^{\beta} = (P^{\alpha\beta})^{T}\, C^{\alpha}\, P^{\alpha\beta}$$

$$K^{\beta} = P^{\beta\alpha}\, K^{\alpha}\, (P^{\beta\alpha})^{T} \qquad \text{A-77}$$

For the case of real elastic structures, both the matrix C and the matrix K have to be symmetric and positive definite matrices. The matrix C can be partitioned into its rotational-translation-force moment elements as follows:

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \qquad \text{A-78}$$

The matrix C is said to have a center $\underline{h}$, called center of compliance, if it can be reduced to a block diagonal matrix through a transformation of the form of Expression A-77. The center of compliance $\underline{h}$ is obtained from Expression A-77 when the matrix c_{21} is equated to zero.

$$\left\{ \Sigma_k \, \epsilon_{jki} \, h_k^{\alpha} \right\} = c_{22}^{-1} \, c_{21} \qquad \text{A-79}$$

This requires that the matrix product in Expression A-79 yields a skew matrix for the center of compliance to exist; if so, its expression is given by:

$$\underline{h} = 1/2 \left\{ \Sigma_{ijk} \, \epsilon_{jki} \, (c_{22}^{-1} \; c_{21})_{jk} \right\} \qquad \text{A-80}$$

5. IMPLEMENTATION OF THE KINEMATIC TRANSFORMATIONS

The code of the transformations developed in this appendix is listed in this section.

NOTE: For computational expediency the general position matrix B, defined in Expression A-5, has been reduced to a three by four matrix by eliminating its first row. Thus, all the expressions that use this matrix have been implemented with the following permutation:

$$b_{ij} \rightarrow b_{i-1\,j}$$

```
C********** FITOA.SS ************
C
C
C       THIS PROGRAM GENERATES THE RTT. MATRIX 'A' CORRESPONDING
C       TO A RTT. VECTOR 'FI'
C
C
        SUBROUTINE FITOA(FI,A)
        COMPILER DOUBLE PRECISION
        DIMENSION FI(3),A(3,3),FI(3)
        DATA EPS/7.E-3/
C
        F2=0.
        DO 1 I=1,3
1       F2=F2+FI(I)*FI(I)
        F=SQRT(F2)
        CC=COS(F)
        IF(F .GT. EPS) GO TO 6
        S=1.-F2/6.
        SC=.5-F2/24.
        GO TO 7
6       S=SIN(F)/F
        SC=(1.-CC)/F2
7       DO 5 I=1,3
        A(I,I)=CC
        K=I+1-((I+1)/4)*3
        L=I+2-((I+2)/4)*3
        A(K,L)=FI(I)*S
5       A(L,K)=-A(K,L)
        DO 4 I=1,3
        DO 4 J=1,3
4       A(I,J)=A(I,J)+FI(J)*FI(I)*SC
        RETURN
        END


C********** ATOFI.SS ************
C
C
C       THIS PROGRAM COMPUTES THE RTT. VCTR. 'FI' CORRESPONDING
C        TO A ROTATION MATRIX 'A'
C
        SUBROUTINE ATOFI(A,FI)
        COMPILER DOUBLE PRECISION
        DIMENSION A(3,3),FI(3)
        DATA PI,H,EPS/3.14159,.5,1.E-4/
C
        CC=-1.
        DO 1 I=1,3
1       CC=CC+A(I,I)
        CC=CC*H
        S=1.-CC*CC
        IF(S .GT. EPS) GO TO 4
        F=H*(1.+S/6.)
        GO TO 5
4       S=SQRT(S)
        F=ATAN(S/CC)
```

```
      IF(F.LT.0) F=F+PI
      F=H*F/S
5     DO 2 I=1,3
      K=I+1-((I+1)/4)*3
      L=I+2-((I+2)/4)*3
2     FI(I)=F*(A(K,L)-A(L,K))
      CC=CC+1.
      IF (CC .GT. EPS) GO TO 6
      F=0.
      DO 7 I=1,3
7     F=F+FI(I)*FI(I)
      F=SQRT(F)
      IF (F .EQ. 0) GO TO 9
      F=(PI-F)/F
      DO 8 I=1,3
8     FI(I)=FI(I)*F
      GO TO 6
9     DO 10 I=1,3
      FI(I)= 0.
10    IF(A(I,I) .GT. 0.5) FI(I)= PI
6     CONTINUE
      RETURN
      END


C********************* INVB.SS ***************
C
C
C     THIS SUBRT INVERTS THE MATRIX "B12" INTO MATRIX "B21"
C
C
      SUBROUTINE INVB(B12,B21)
      COMPILER DOUBLE PRECISION
      DIMENSION B(3,4),B12(3,4),B21(3,4)
C
      DO 2 I=1,3
      DO 2 J=2,4
2     B(I,J)=B12(J-1,I+1)
      DO 1 I=1,3
      B(I,1)=0.
      DO 1 L=1,3
1     B(I,1)=B(I,1)-B(I,L+1)*B12(L,1)
      DO 3 I=1,3
      DO 3 J=1,4
3     B21(I,J)=B(I,J)
      RETURN
      END


C********************* XTOB.SS ***************C
C
C
C     THE SIX DIMENSION VECTOR "X" IS COVERTED
C     INTO THE 3X3 POSITION MATRIX "B"
C
C
C
      SUBROUTINE XTOB(X,B)
```

```
      COMPILER DOUBLE PRECISION
      DIMENSION X(6),Y(3),B(3,4),A(3,3)
C
      DO 1 I=1,3
1     Y(I)=X(I+3)
      CALL FITOA(Y,A)
      DO 2 I=1,3
      B(I,1)=0.
      DO 2 J=2,4
      B(I,J)=A(I,J-1)
2     B(I,1)=B(I,1)-A(I,J-1)*X(J-1)
      RETURN
      END


C********** BTOX.SS ***************C
C
C
C     THIS SBRT CONVERTS THE MTRX "B" INTO A
C     SIX DIMENSION POST VECTOR "X"
C
      SUBROUTINE BTOX(B,X)
      COMPILER DOUBLE PRECISION
      DIMENSION B(3,4), X(6), Y(3), A(3,3)
C
      DO 1 I=1,3
      X(I)=0
      DO 1 L=1,3
      X(I)=X(I)-B(L,I+1)*B(L,1)
1     A(I,L)=B(I,L+1)
      CALL ATOFI(A,Y)
      DO 2 I=4,6
2     X(I)=Y(I-3)
      RETURN
      END


C***************** INVX.SS *****************
C
C
C     TIHIS SBRT. INVERTS THE COORDINATES OF VECTOR "X12"
C     INTO THE VECTOR "X21"
C
C
      SUBROUTINE INVX(X12,X21)
      COMPILER DOUBLE PRECISION
      DIMENSION X(3),A(3,3),X12(6),X21(6)
C
      DO 1 I=1,3
1     X(I)=X12(I+3)
      CALL FITOA(X,A)
      DO 2 I=1,3
      X(I)=0
      DO 2 L=1,3
2     X(I)=X(I)-A(I,L)*X12(L)
      DO 3 I=1,3
      K=I+3
      X21(I)=X(I)
```

```
3       X21(K)=-X12(K)
        RETURN
        END


C***************** BBB,SS **********
C
C
C       SBRT COMPUTES THE PRODUCT OF THE 3X4 MATRICES "B12 & B23"
C       AND  GIVES  THE RESULT IN THE 3X4 MATRIX "B13"
C
        SUBROUTINE BBB(B12,B23,B13)
        COMPILER DOUBLE PRECISION
        DIMENSION B(3,4),B12(3,4),B23(3,4),B13(3,4)
C
        DO 1 I=1,3
        DO 1 J=2,4
        B(I,J)=0.
        DO 1 L=1,3
1       B(I,J)=B(I,J)+B12(I,L+1)*B23(L,J)
        DO 2 I=1,3
        B(I,1)=B12(I,1)
        DO 2 L=1,3
2       B(I,1)=B(I,1)+B12(I,L+1)*B23(L,1)
        DO 3 I=1,3
        DO 3 J=1,4
3       B13(I,J)=B(I,J)
        RETURN
        END


C************************ XXX,SS ***********************************
C
C
C
C       COMPUTES THE EXPRESSION X13 = X13( X12 , X23 )
C
C
C
C
C
C
C
        SUBROUTINE XXX(X12,X23,X13)
        COMPILER DOUBLE PRECISION
        DIMENSION X12(6),X13(6),X23(6),B12(3,4),B23(3,4),B13(3,4)
C
        CALL XTOB(X12,B12)
        CALL XTOB(X23,B23)
        CALL BBB(B12,B23,B13)
        CALL BTOX(B13,X13)
C
        RETURN
        END
```

```
C********************** DADW,SS ***************************************
C
C
C
C
C
C         COMPUTES THE THIRD ORDER TENSOR " D = DA12/DW21 (X12) "
C                 USING EXPRESION A-41
C
C
C
C
C
C
C
C
      SUBROUTINE DADW(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),B12(3,4),D(3,3,3)
C
      CALL XTOB(X12,B12)
      DO 1 I3=1,3
      DO 1 I=1,3
      I1=I3+1-((I3+1)/4)*3
      I2=I3+2-((I3+2)/4)*3
      D(I1,I,I1)=0.
      D(I1,I,I3)=B12(I2,I+1)
1     D(I1,I,I2)=-B12(I3,I+1)
C
      RETURN
      END


C****************************** DFIDW,SS ********************************
C
C
C
C
C
C
C         COMPUTES THE SECOND ORDER TENSOR " D = DFI12/DW21 (X12) "
C                 USING EXPRESIONS A-56 AND A-58
C
C
C
C
C
C
C
      SUBROUTINE DFIDW(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),D(3,3)
      DATA EPS,PIMEP/1.E-2,3.13/
C
      FI2= 0.
      DO 1 I=4,6
1     FI2= FI2+X12(I)*X12(I)
      FI=SQRT(FI2)
```

```
      IF ( FI - EPS)10,10,11
10    S=-1.+FI2/12.
      C= (1.+FI2*7./60.)/12.
      GO TO 12
11    IF(FI-PIMEP)14,14,13
13    S= FI*SIN(FI)/(2.*(COS(FI)-1.))
      GO TO 15
14    S=-(1.+COS(FI))*FI/(2*SIN(FI))
15    C= (1.+S)/FI2
12    DO 2 I1=1,3
      I2= I1+1-((I1+1)/4)*3
      I3= I1+2-((I1+2)/4)*3
      D(I1,I1)=S
      D(I1,I2)=-X12(I3+3)/2.
      D(I1,I3)= X12(I2+3)/2.
      DO 2 I=1,3
2     D(I1,I)= D(I1,I)+C*X12(I1+3)*X12(I+3)
C
      RETURN
      END


C***************************** DWDFI.SS ************************
C
C
C
C
C
C     COMPUTES THE SECOND ORDER TENSOR " D = DW21/DFI12 (X12) "
C              USING EXPRESIONS A-54 AND A-58
C
C
C
C
      SUBROUTINE DWDFI(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),D(3,3)
      DATA EPS/7.E-3/
C
      FI2  = 0.
      DO 1 I=4,6
1     FI2=FI2+X12(I)*X12(I)
      FI=SQRT(FI2)
      IF ( FI - EPS) 10,10,11
10    S=1.-FI2/6.
      C=-.5+FI2/24.
      SC= (1.-FI2/20.)/6.
      GO TO 12
11    S=SIN(FI)/FI
      C=(COS(FI)-1)/FI2
      SC=(1-S)/FI2
12    DO 2 I1=1,3
      I2=I1+1-((I1+1)/4)*3
      I3=I1+2-((I1+2)/4)*3
      D(I1,I1)= S
      D(I1,I2)=-X12(I3+3)*C
      D(I1,I3)= X12(I2+3)*C
```

```
      DO 2 J=1,3
2     D(I1,J)= D(I1,J)+SC*X12(I1+3)*X12(J+3)
C
      RETURN
      END


C************************ DWDW.SS *********************************
C
C
C
C
C
C     COMPUTES THE SECOND ORDER TENSOR " D = DW21/DW12 (X12) "
C             USING A=44
C
C
C
C
C
      SUBROUTINE DWDW(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),B12(3,4),D(3,3)
C
      CALL XTOB(X12,B12)
      DO 1 I=1,3
      DO 1 J=1,3
1     D(I,J)=-B12(I,J+1)
C
      RETURN
      END


C********************************* DWDA.SS ****************************
C
C
C
C
C     COMPUTES THE THIRD ORDER TENSOR " D = DW21/DA12 (X12) "
C             USING EXPRESION A=42
C
C
C
C
C
C
      SUBROUTINE DWDA(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),B12(3,4),D(3,3,3)
C
      CALL XTOB(X12,B12)
      DO 1 I1=1,3
      DO 1 J=1,3
      I2=I1+1-((I1+1)/4)*3
      I3=I1+2-((I1+2)/4)*3
      D(I1,I1,J)= 0.
      D(I1,I2,J)= .5*B12(I3,J+1)
```

```
1       D(I1,I3,J)=-.5*B12(I2,J+1)
C
        RETURN
        END


C************************ DFIDA.SS ***********************************
C
C
C
C
C
C
C       COMPUTES THE THIRD ORDER TENSOR " D = DFI12/DA12 (X12) "
C               USING EXPRESION A-60
C
C
C
C
C
        SUBROUTINE DFIDA(X12,D)
        COMPILER DOUBLE PRECISION
        DIMENSION X12(6),D1(3,3),D2(3,3,3),D(3,3,3)
C
        CALL DFIDW(X12,D1)
        CALL DWDA(X12,D2)
        DO 1 I=1,3
        DO 1 J=1,3
        DO 1 K=1,3
        D(I,J,K)= 0.
        DO 1 L=1,3
1       D(I,J,K)= D(I,J,K)+D1(I,L)*D2(L,J,K)
C
        RETURN
        END


C******************************** DADFI.SS ***************************
C
C
C
C
C
C
C       COMPUTES THE THIRD ORDER TENSOR " D = DA12/DFI12 (X12) "
C               USING EXPRESION A-60
C
C
C
C
C
        SUBROUTINE DADFI(X12,D)
        COMPILER DOUBLE PRECISION
        DIMENSION X12(6),D1(3,3,3),D2(3,3),D(3,3,3)
C
        CALL DADW(X12,D1)
        CALL DWDFI(X12,D2)
```

```
      DO 1 I=1,5
      DO 1 J=1,3
      DO 1 K=1,3
      D(I,J,K)= 0.
      DO 1 L=1,3
1     D(I,J,K)= D(I,J,K)+D1(I,J,L)*D2(L,K)
C
      RETURN
      END


C************************ DBDX.SS ************************************
C
C
C
C     COMPUTES THE THIRD ORDER TENSOR "D = DB12 / DX12 (X12)"
C
C
C
      SUBROUTINE DBDX(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),D(3,4,6),D3(3,3,3),D6(6,6)
C
      CALL INVDX(X12,D6)
      CALL DADFI(X12,D3)
      DO 1 I=1,3
      DO 2 K=1,6
2     D(I,1,K)= D6(I,K)
      DO 1 J=2,4
      DO 1 L=1,3
      L3=L+3
      D(I,J,L)=0.
1     D(I,J,L3)=D3(I,J-1,L)
      RETURN
      END


C************************ DXDB.SS ************************************
C
C
C     COMPUTES THE THIRD ORDER TENSOR "D X12 / D B12 (X12)"
C
C
C
      SUBROUTINE DXDB(X12,D)
      COMPILER DOUBLE PRECISION
      DIMENSION X12(6),D(6,3,4)
      DIMENSION D3(3,3,3),B12(3,4)
C
      CALL DFIDA(X12,D3)
      CALL XTOB(X12,B12)
      DO 1 I=1,3
      I3=I+3
      DO 1 J=1,3
      D(I,J,1)=-B12(J,I+1)
      D(I3,J,1)=0.
      DO 1 K=2,4
      D(I3,J,K)=D3(I,J,K-1)
```

```
1       D(I,J,K)=-X12(K-1)*B12(J,I+1)
        RETURN
        END


C******************** INVDX.SS ***************************************
C
C
C       COMPUTES THE TENSOR THAT INVERTS THE VECTOR DX12
C         SUCH THAT "D = DX21 / DX12 (X12)"
C
C
C
C
C
        SUBROUTINE INVDX(X12,D)
        COMPILER DOUBLE PRECISION
        DIMENSION X12(6),B12(3,4),D3(3,3),D(6,6)
C
        CALL XTOB(X12,B12)
        CALL DWDFI(X12,D3)
        DO 1 I=1,3
        I3=I+3
        J=I+1-((I+1)/4)*3
        K=I+2-((I+2)/4)*3
        DO 2 L=1,3
        L3=L+3
        D(I,L)=-B12(I,L+1)
        D(I3,L)=0.
        D(I3,L3)=0.
2       D(I,L3)=B12(J,1)*D3(K,L)-B12(K,1)*D3(J,L)
1       D(I3,I3)=-1.
        RETURN
        END

C***************************** DXDX.SS ********************************
C
C
C
C
C
C
C       COMPUTES THE SECOND ORDER TENSOR " D = DX41/DX32 (X41,X43,X32) "
C               USING EXPRESSION A-75
C
C
C
C
C
        SUBROUTINE DXDX(X41,X43,X32,D)
        COMPILER DOUBLE PRECISION
        DIMENSION X41(6),X43(6),X32(6),B41(3,4),B43(3,4),B32(3,4)
        DIMENSION B14(3,4),B13(3,4),B12(3,4)
        DIMENSION D(6,6),D1(3,3),D2(3,3)
C
        CALL XTOB(X41,B41)
        CALL XTOB(X43,B43)
        CALL XTOB(X32,B32)
```

```
      CALL INVB(B41,B14)
      CALL BBB(B14,B43,B13)
      CALL BBB(B13,B32,B12)
C
C     FIRST COMPUTES DR41/DR32 IN EXPN A=75 , USING EXPN A=70
C
      DO 1 I=1,3
      DO 1 J=1,3
1     D(I,J)=B12(I,J+1)
C
C     COMPUTES DR41/DW23 USING A=71 AND
C     USING DWDFI COMPUTES DR41/DFI32 IN A=75
C     ALSO MAKES   DFI41/DR32 = 0.
C
      DO 2 I2=1,3
      I3= I2+1-((I2+1)/4)*3
      I1= I2+2-((I2+2)/4)*3
      DO 2 J=1,3
2     D1(J,I2)= B13(J,I1+1)*X43(I3)-B13(J,I3+1)*X43(I1)
      CALL DWDFI(X32,D2)
      DO 3 I=1,3
      DO 3 J=4,6
      D(I,J)=0.
      D(J,I)= 0.
      DO 3 L=1,3
3     D(I,J)= D(I,J)+D1(I,L)*D2(L,J-3)
C
C     COMPUTES  DFI41/DFI  USING A=73
C
      CALL DFIDW(X41,D1)
      DO 4 I=4,6
      DO 4 J=4,6
      D(I,J)= 0.
      DO 4 K=1,3
      DO 4 L=1,3
4     D(I,J)= D(I,J)+D1(I-3,L)*B43(L,K+1)*D2(K,J-3)
C
C
      RETURN
      END
```

APPENDIX B

TWO DIMENSION ANALYSIS OF A PEG-IN-HOLE INSERTION

1. INTRODUCTION

The following definitions, which appear elsewhere in this work, are repeated here in order to clearly define the objectives of this appendix:

- Trajectory is the condition under which all degrees of freedom in the space concerned are a continuous function of a single independent parameter.
- Assembly path is the trajectory, defined in the six degrees of freedom space of the relative position between the parts being assembled, that takes the parts from their initial relative position to their final assembled state.
- Fine motion path, segment of the assembly path characterized by small allowable deviations for the nominal trajectory. These deviations from the nominal trajectory are constrained by the geometry of the parts.
- Gross motion path, segment of the assembly path where deviations from the nominal assembly trajectory are not critical to the execution of the assembly task.

The purpose of this appendix is to analyze exclusively the fine motion path under the following set of conditions:

- The positioning device used for executing the assembly path has a positioning error envelope bigger than the allowable deviation for the fine motion path of concern.
- The parts will be assumed initially within their fine motion region of the assembly path.

Without the first condition the problem is, at least in principle, straightforward. The transition from gross to fine motion, avoided here by the second condition, is dealt with in detail in Chapter II.

The fine motion analysis is done on the two dimension insertion of a round peg into a round hole, Figure B-1. The peg-and-hole case and the planar assumption were chosen for the simplicity in the geometry configuration. The goal of this analysis is to determine the usefulness of the force information, generated by the parts touching, in the execution of the fine motion assembly. The following assumptions are introduced in the analysis:

- Low speed. Quasi static analysis, inertia forces are considered to be small.
- Static and dynamic coefficient of friction are assumed equal. This restriction can easily be removed.
- Rigid peg and hole. The deformation of their geometry is assumed small.
- Perfect control of applied forces. This requires a perfect reading of forces and moments, measured at the tip of the peg, and an infinite resolution in the position control. These constraints are lifted when the design criteria are analyzed.
- Two dimensional analysis, all forces and position displacements are assumed contained in one plane, moments and angular positions are considered only perpendicular to the plane.

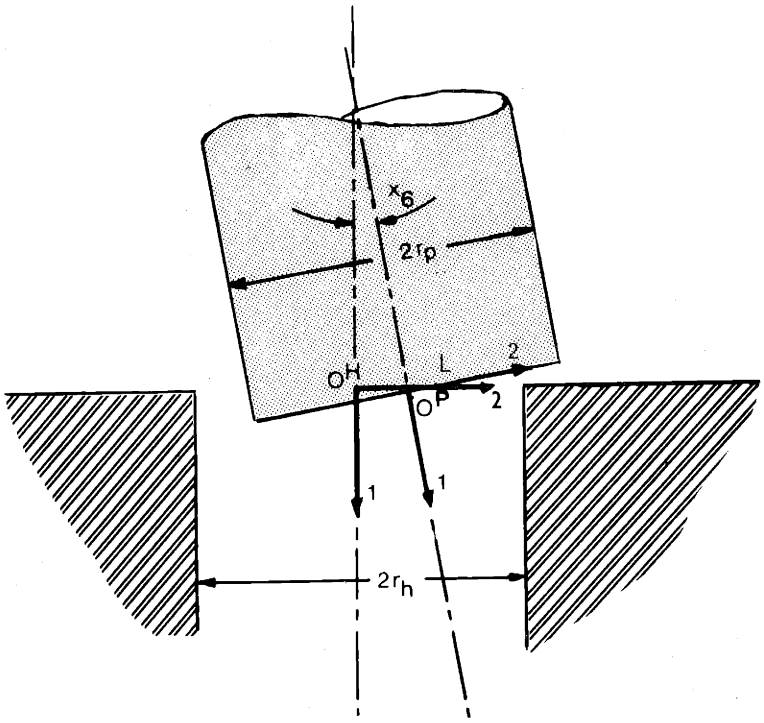


Figure B-1: Notation of Two Dimension
Peg and Hole Insertion

The relative position between peg and hole is expressed in terms of the relative position between the frame P, defined at the tip of the peg, and the frame H, defined at the mouth of the hole (Figure B-1). The "one" axes are chosen to coincide with the respective symmetry axis of the peg and the hole, the "three" axes are chosen perpendicular to the plane of displacements and forces, and the "two" axes form right handed triorthogonal frames with the respective other two axes. For this configuration of reference frames the planar assumption results in:

$$r_3^{PH} = x_3^{PH} = 0$$

$$\phi_1^{PH} = x_4^{PH} = 0$$

$$\phi_2^{PH} = x_5^{PH} = 0$$

In the next section the different systems of forces acting on the parts are discussed; Section Three analyses the different equilibrium situations and Sections Four and Five discuss some of the conclusions of this analysis.

2. FORCE ANALYSIS

For the analysis the peg is considered under two sets of force systems: applied forces or measured forces, which can be controlled; and reaction forces or contact forces, caused by the interaction of the parts at assembly.

(a) Applied Forces

Let $\underline{f}^P$ and $\underline{m}^P$ be the applied force and moment on the peg. Due to the planar assumption, the components f_3, m_1 and m_2 are identically zero (See Figure B-2). The system of applied forces can be equivalently represented by the force $\underline{f}$ acting along the line of action given by the vector equation

$$\underline{r}^P \times \underline{f}^P = \underline{m}^P \qquad \text{B-1}$$

This equation is written in terms of the components of the vectors, for the planar assumption as:

$$r_1 f_2 - r_2 f_1 = m_3 \qquad \text{B-1}$$

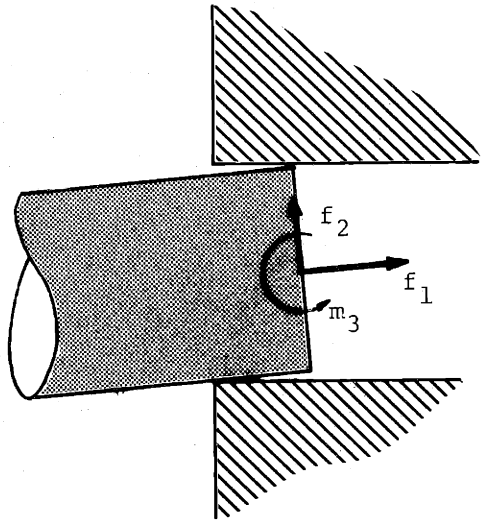


Figure B-2: Notation of Applied Forces and Moments

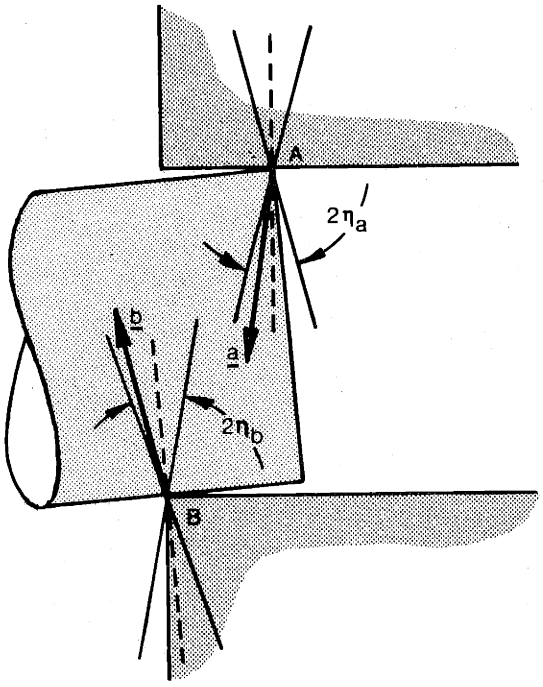


Figure B-3: Notation of Reaction Forces

(b) Reaction Forces

Let A be the contact point between the tip of the peg and the hole's wall and B the contact point between the mouth of the hole and the peg's wall. Figure B-3 shows one possible configuration for these contact points. The vector $\underline{a}$ denotes the force acting through A and $\underline{b}$ the force through B. The vectors and $\underline{a}$ and $\underline{b}$ are decomposed into their normal and tangential components to the wall of the hole and to the wall of the peg respectively as follows:

$$a_n = -\underline{a} \cdot \underline{e}_2^H \text{ sign } (\phi_3^{PH}); \quad a_t = \underline{a} \cdot \underline{e}_1^H$$
$$b_n = \underline{b} \cdot \underline{e}_2^P \text{ sign } (\phi_3^{PH}); \quad b_t = \underline{b} \cdot \underline{e}_1^P \qquad \text{B-2}$$

If we denoted by μ_a and μ_b the coefficient of friction and η_a and η_b the friction angles at A and B, then Coulomb's friction law is simply written as:

$$\frac{a_t}{a_n} \leq \mu_a = \tan \eta_a \qquad \text{B-3}$$

$$\frac{b_t}{b_n} \leq \mu_b = \tan \eta_b$$

Point A and B are contact points between the parts. This implies that the normal forces, as defined in Expression B-2, have to be positive; negative contact forces require adhesion between the parts.

$$a_n \geq 0; \qquad b_n \geq 0 \qquad \text{B-4}$$

For the case of only one point contact, $\underline{a}$ or $\underline{b}$ is zero. For zero ϕ_3^{PH}, the contact between the peg and hole is along a line; the reaction force could be anywhere along that contact line and within the friction cone as defined by Expression B-3.

3. FORCE EQUILIBRIUM

Since it has been assumed that the positioning device cannot be controlled within the position margins required for the fine motion path, the peg will be touching the hole's wall during the mating process. The method used in this section is to find the conditions on the forces acting on the peg that will guarantee the sliding of the peg into the hole.

The quasi-static assumption, that considers the inertia forces to be negligible, guarantees that the forces on the peg are either zero or must be in an equilibrium condition. Under this quasi-static assumption, a non equilibrium situation between the forces will instantaneously, through movements of parts, evolve to an equilibrium condition. Thus, if the applied force can be controlled so as to force a non equilibrium configuration, the parts will slide in order to maintain a marginal equilibrium situation.

The equilibrium or non equilibrium conditions of the forces on the peg is analyzed for three separate cases: first, for the case of zero applied forces; second, for the case of one reacting force and lastly, for the case of both reacting forces and a non-zero applied force.

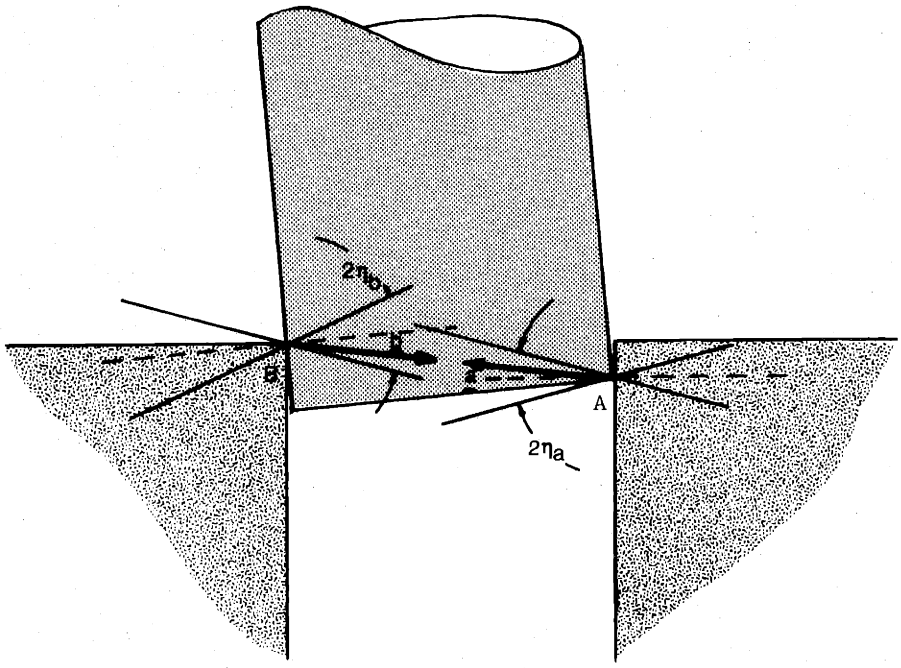


Figure B-4: Reaction Forces with Zero Applied Forces

Equilibrium between reaction forces with zero applied can only be attained when both reaction forces are opposed on the same line of action. Both reaction forces, as expressed on Relations B-3 and Relation B-4, are restricted to their respective friction cones. Thus an equilibrium condition is only possible when contact point B is inside the friction cone at point A and vice versa, when point A is inside the friction cone at point B. Figure B-4 shows a case in which both reaction forces are in equilibrium with no applied forces. Through geometric manipulations, this condition is expressed as a restriction on the penetration of the peg into the hole by:

$$\frac{x_1^{PH}}{2R} \leq \mu_b(1 - c)\ (\cos x_6^{PH} + \frac{\sin x_6^{PH}}{2\mu_b} \qquad \text{B-5}$$

and

$$\frac{x_1^{PH}}{2R} \leq \mu_a(1 + \frac{(1 - c)\sin x_6^{PH}}{\mu_a})$$

where:

- x_1^{PH} and x_6^{PH} are the penetration and misalignment angle between peg and hole measured in the frames defined in Section One of this appendix.
- c is the clearance ratio between radius of hole and peg. See Figure B-1.

$$c = \frac{R - r}{R} \qquad \text{B-6}$$

- μ_a and μ_b are the coefficient of friction at points A and B respectively.

Considering that the clearance ratio c and the angle x_6^{PH} are small in relation to unity, and assuming that both coefficients of friction are numerically equal, the equilibrium conditions in Expression B-5 become simply:

$$\lambda = \frac{x_1^{PH}}{2R\mu} \leq 1 \qquad \text{B-7}$$

The condition just described, where in the absence of applied forces reaction forces are not zero, will be referred to here as "wedging". Thus wedging can only occur when Relation B-5, or simply Relation B-7, is satisfied.

In practice, real parts are not infinitely rigid. The presence of reaction forces implies some elastic deformation of the parts and thus stored elastic energy. For the case of wedged parts, the stored energy has to have been supplied by a previously applied force or some other phenomena, such as inertia forces.

For the case of an applied force and one reaction force, equilibrium can only occur when both these forces are on the same line of action with opposed directions. Reaction forces are restricted to their friction cone. Thus equilibrium for this case (Figure B-5) can only occur if the line of action of the applied force is fully contained inside the friction cone at the reaction force. Considering that the misalignment between the peg and the hole is small, the non-equilibrium condition between the applied force and one reaction force is given simply by:

$$-\mu < \frac{f_2}{f_1} < \mu \qquad \text{B-8}$$

Equilibrium of three forces in space can only be obtained when these forces are coplanar and their lines of action intersect at a single point in space. Figure B-6 shows five possible equilibrium situations between an applied force and two reaction forces. Configurations labeled zero and four in the figure are impossible equilibrium solutions because the reaction forces violate the friction cone restriction, Relation B-3, or the positive normal force restriction, Relation B-4; any configuration of reaction forces between configurations labeled one and three in the figure are possible equilibrium solutions.

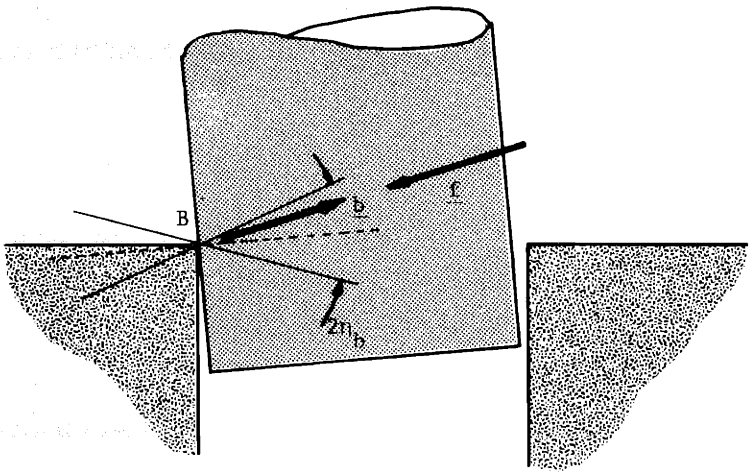


Figure B-5: Equilibrium Between Applied Force and One Reaction Force

Since parts are not infinitely rigid, the reaction forces will take the configuration that minimizes the stored elastic energy. For the case where the reaction forces are only products of the applied force, solution one in Figure B-6 represents the equilibrium configuration; if the reaction forces are caused by the applied force superimposed by a wedging situation, when the geometry Relation B-7 is satisfied, the equilibrium solution could be anywhere between the configurations labeled one and three in Figure B-6.

Thus, from the above argument we conclude that:

(i) Equilibrium is only possible if the applied force goes through the region defined by the intersection of both friction cones.

(ii) If equilibrium is present without wedging, the stable equilibrium configuration will be given by the intersection of the applied force line of action with the boundary of the region defined by both friction cones.

(iii) If wedging is superimposed over the applied forces, the equilibrium configuration does not necessarily lie at the boundary of the region defined by the intersection of the friction cones. The equilibrium point can be driven to the boundary of the friction cones region by increasing the magnitude of the applied force.

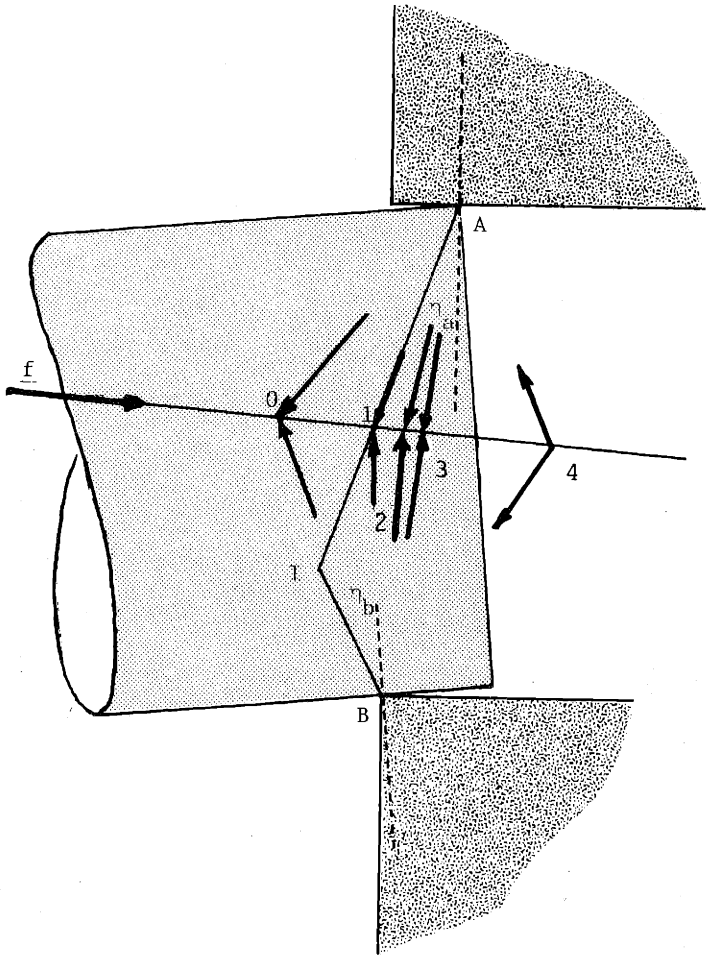


Figure B-6: Equilibrium Between Applied Force and Two Reaction Forces

If the equilibrium point is at the boundary of the region defined by the intersection of the friction cones, at least one of the points of contact is in marginal sliding condition. From Figure B-3, only point A can slide; point B is physically constrained by the geometry of the parts.

Thus we conclude that the peg will slide only when the moment of the applied force around the point I, defined by the intersection of the boundaries of the friction cones, Figure B-6, has opposed sign to the misalignment angle between peg and hole. Assuming that both coefficient of friction are numerically equal and that the misalignment angle is small compared to the friction angle, the components of the point I in the peg's frame are

$$x_1/r = (\lambda + 1)\mu$$
$$x_2/r = -\text{sign}\,(\phi_3^{PH})\lambda \qquad \text{B-9}$$

where $\lambda = x_1^{PH}/2r\mu$

And the moment condition around point I is given by:

$$-1 < \frac{\frac{m_3}{f_1 r}}{\lambda} + \frac{\frac{f_2}{f_1}}{(\lambda + 1)\mu} < 1 \qquad \text{B-10}$$

Figure B-7 shows the sliding and jammed regions generated by the conditions in Expression B-8 and Expression B-10.

4. CONTROL OVER THE APPLIED FORCES

A force control scheme designed to satisfy the sliding conditions (derived from the peg-and-hole analysis) will automatically allow the insertion of the peg into the hole. Control can be accomplished directly through the control of the complete force vector, indirectly through the use of a passive impedance, or through the control over some of the components of the force vector in combination with passive impedance.

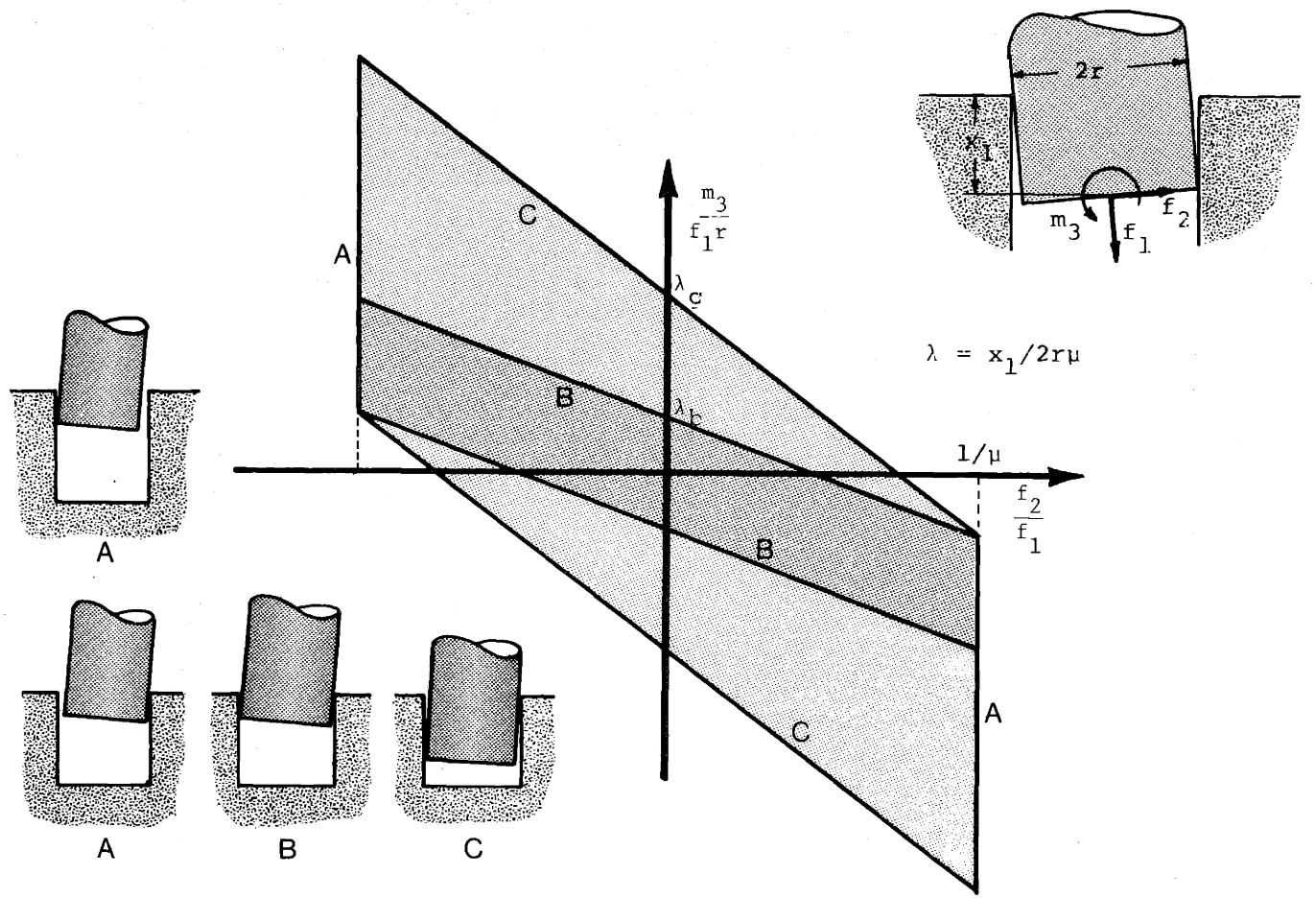


Figure B-7: Sliding Conditions

Under direct control of the forces, the objective is to make the lateral force and moments (f_2 and m_3 in the two-dimensional case) equal to zero and to increase the axial force (f_1) until the parts slide. The magnitude of the axial force required to make the peg go in will be determined by the residual forces and moments on the lateral direction. For the two-dimensional case, the axial force needed (from Expression B-10) to make the insertion is given by:

$$f_1 = \left| \frac{f_2}{\frac{\lambda}{(\lambda + 1)\mu}} + \frac{m_3}{\lambda} \right| \qquad \text{B-11}$$

NOTE: For the direct control of forces a compliance is needed to stabilize the force loop (Reference B-1).

Under indirect control, the contact forces will generate the necessary corrections through a passive impedance constructed in between the parts being assembled. For the two-dimensional quasi-static case, the impedance will take the form of a compliance. From Expression A-76, this compliance is written as:

$$\begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta \omega_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{16} \\ c_{21} & c_{22} & c_{26} \\ c_{61} & c_{62} & c_{66} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ m_3 \end{bmatrix} = K^{-1} \begin{bmatrix} f_1 \\ f_2 \\ m_3 \end{bmatrix} \qquad \text{B-12}$$

In order to analyze the best form for this compliance, a peg in a jammed configuration is assumed (point P in Figure B-8). Letting p be the distance from the sliding condition line to the point P, the Expression for this distance is given by:

$$p = \sqrt{\frac{1}{\lambda^2} + \frac{\mu(\lambda + 1)}{\lambda}}^2 \left[\frac{\left(\frac{m_3}{f_1 r}\right)_P}{\lambda v} + \frac{\left(\frac{f_2}{f_1}\right)_P}{\frac{\lambda v}{(\lambda + 1)}} \right] \qquad \text{B-13}$$

where $v = \text{sign}\,(x_6^{PH})$.

The compliance should correct for the jamming during the insertion. As the jammed peg is driven into the hole the compliance will absorb the movements in the insertion direction (increase of $-\delta x_1$ in Expression B-12). The change in force pattern generated by the deformation of the compliance should drive the point P, in Figure B-8, into the sliding region. This is measured by the change of the distance p due to changes in δx_1. Introducing Expression B-12 and differentiating on δx_1 results in:

$$dp = \left(A_1 v \frac{k_{61}}{f_1} + A_2 v \frac{k_{21}}{f_1} - A_3 \frac{k_{11}}{f_1}\right) d(\delta x_1) \qquad \text{B-14}$$

where

$$A_1 = \sqrt{\frac{1}{\lambda^2} + \left(\frac{\mu(\lambda + 1)}{\lambda}\right)^2} \cdot \frac{1}{r\lambda} > 0$$

$$A_2 = \sqrt{\frac{1}{\lambda^2} + \left(\frac{\mu(\lambda + 1)}{\lambda}\right)^2} \cdot \frac{\mu(\lambda + 1)}{\lambda} > 0$$

$$A_3 = p + \sqrt{\frac{1}{\lambda^2} + \left(\frac{(\lambda + 1)}{\lambda}\right)^2} > 0$$

The compliance will force the sliding conditions when the coefficient of $d(\delta x_1)$ in Expression B-14 is positive. This is better achieved by setting

$$k_{21} = k_{61} = 0 \qquad \text{B-15}$$

In terms of the compliance matrix

$$c_{21} = c_{61} = 0 \qquad \text{B-15}$$

This implies that the compliance is symmetrical with respect to the axis of the peg.

For the case where only some of the components of the force vector are directly controlled, the analysis and results for the totally passive approach

are also applicable.

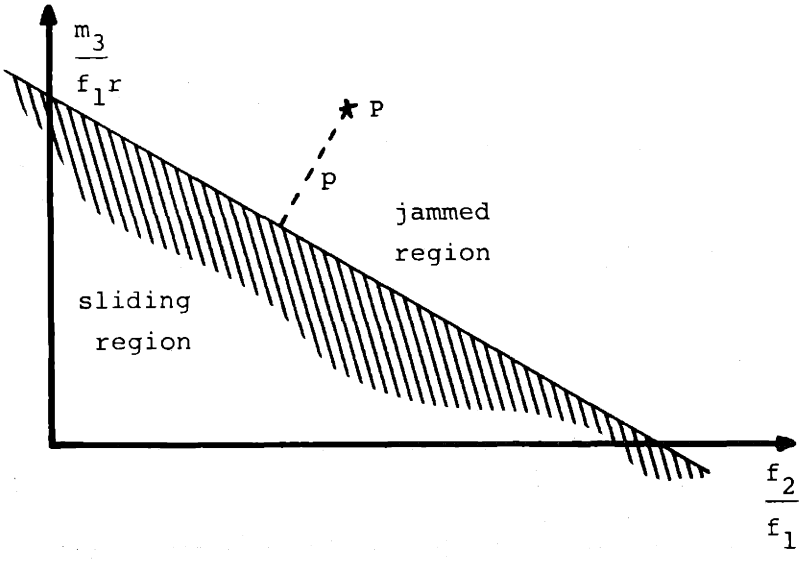


Figure B-8: Peg in Jammed Condition

5. COMPLIANCE EFFECT ON THE FORCE CONTROL SCHEME

The analysis done on the fine motion trajectory of simple insertion tasks suggests that the problem can be reduced to a basic force control problem. A fact, well known by instrument designers, is the second order effect of the compliant elements in a force measuring configuration. The purpose of this section is to relate these second order effects to the design of the compliant element to be used. The results are expressed in terms of relations suited for three different strategies, namely: totally active, totally passive, and partially active approaches.

One fundamental assumption used in the analysis is that the compliant elements are assumed to be completely linear throughout the range of their deformations. This restrictive assumption could be removed given a particular configuration of compliant elements. For a configuration in which the displacement deformation is small compared to the size of the elements and the angular deformations are also small, this constitutes a reasonable assumption. In the course of the analysis, depending on the particular case studied, other assumptions are made, such as: symmetry of the compliance along a particular axis, accuracy in locating the compliance center, two-dimensional approximation, etc.

Appendix A deals with displacement and rotation transformations as well as their small increments approximations. From Expression A-34 the transformation of a force measured in frame "1" to frame "2" is given by:

$$\underline{L}^2 = T^{21}\,\underline{L}^1 = \begin{bmatrix} A^{21} & 0 \\ A^{21}\{\Sigma_k \epsilon_{ijk}\, r_k^{21}\} & A^{21} \end{bmatrix} \begin{bmatrix} \underline{f}^1 \\ \underline{m}^1 \end{bmatrix} \qquad \text{B-16}$$

and for small deformations

$$A^{21} = I + \{\Sigma_k\, \epsilon_{ijk}\, d\omega_k^{12}\}$$

$$A^{21}\{\Sigma_k\, \epsilon_{ijk}\, r_k^{21}\} = \{\epsilon_{ijk}\, dr_k^{21}\} \qquad \text{B-17}$$

The second order influence in frame "2" as a function of forces applied at frame "1" is then given by:

$$df_i^2 = \Sigma_{jk}\, \epsilon_{ijk}\, d\omega_k^{12}\, f_j^1 \qquad \text{B-18}$$

$$dm_i^2 = \Sigma_{jk} \left[dr_k^{21} \, f_j^1 + d\omega_k^{12} \, m_j^1 \right] \varepsilon_{ijk} \qquad \text{B-18}$$

NOTE: The upper indexes are omitted in the rest of the analysis. It is understood that the incremental terms refer to forces and moments measured at the displaced frame and the finite terms correspond to the undisplaced frame.

The use of the linear compliant assumption gives:

$$\begin{bmatrix} d\,\underline{r} \\ d\,\underline{\omega} \end{bmatrix} = C\,\underline{L} \qquad \text{B-19}$$

Let's assume a compliance matrix of the form:

$$C = \begin{bmatrix} D & H \\ H^T & G \end{bmatrix} \qquad \text{B-20}$$

The use of the notation from Expression B-20 in Expression B-19 yields:

$$\begin{aligned} dr_i &= \Sigma_j \left[d_{ij} \, f_j + h_{ij} \, m_j \right] \\ d\omega_i &= \Sigma_j \left[h_{ji} \, f_j + g_{ij} \, m_j \right] \end{aligned} \qquad \text{B-21}$$

Introducing Expression B-21 into Expression B-18 yields a general expression for the second order effect of the deformations of the compliant element:

$$\begin{aligned} df_i &= \Sigma_{jkv} \, \varepsilon_{ijk} \left[h_{vk} \, f_v f_j + g_{kv} \, m_v \, f_j \right] \\ dm_i &= \Sigma_{jkv} \, \varepsilon_{ijk} \left[d_{kv} \, f_v f_j + h_{kv} \, m_y f_j + h_{vk} \, f_v m_j + g_{kv} \, m_v m_j \right] \end{aligned} \qquad \text{B-22}$$

Since the fine motion sliding has been formulated for the two-point contact planar case, the analysis will focus on this problem. It is assumed that the contact points are in the (1,2) plane; the analysis is done over the 2 force and 3 moment. The 1 and 2 moments and the 3 force are assumed to be zero. Thus:

$$m_1 = m_2 = f_3 = 0$$

$$df_2 = -h_{13}f_1f_1 - g_{33}f_1m_3 \qquad \text{B-23}$$

$$dm_3 = d_{21}f_1f_1 + d_{22}f_1f_2 - d_{11}f_1f_2 - d_{11}f_2f_2 + h_{23}f_1m_3 - h_{13}f_2m_3$$

From the compliance transformations in Expression A-34 and the definition of the center of compliance in Expression A-80 , the expression for the compliance submatrices in terms of the center of compliance $\underline{h}$ is given by:

$$D = \begin{bmatrix} c_1 + h_3^2c_5 + h_2^2c_6 & -h_1h_2c_6 & -h_1h_3c_5 \\ -h_1h_2c_6 & c_2 + h_3^2c_4 + h_1^2c_6 & -h_2h_3c_4 \\ -h_1h_3c_5 & -h_2h_3c_4 & c_3 + h_2^2c_4 + h_1^2c_5 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & -h_3c_5 & h_2c_6 \\ h_3c_4 & 0 & -h_1c_6 \\ -h_2c_4 & h_1c_5 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} c_4 & 0 & 0 \\ 0 & c_5 & 0 \\ 0 & 0 & c_6 \end{bmatrix} \qquad \text{B-24}$$

From the results of the previous section the compliance will be assumed symmetrical with respect to the axis of the peg and perfectly constructed. Thus:

$$h_2 = 0 \text{ and } h_3 = 0 \qquad \text{B-25}$$

The effect of the compliance in the direct force control scheme is to introduce terms to the force and moment ratios (Expression B-10) that do not vanish for large magnitudes of the axial force f_1. These terms, obtained from Expression B-22 and Expression B-24, are:

$$\frac{df_2}{f_1} = c_6(h_1 f_2 - m_3)$$

$$\frac{dm_3}{f_1 r} = \left(\frac{c_2 - c_1}{r} + \frac{h_1^2}{r} c_6\right) f_2 - \frac{h_1}{r} c_6 m_3 \qquad \text{B-26}$$

where f_2 and m_3 constitute the marginal errors in the control of the lateral force and moment. This suggests locating the compliance center at the tip of the peg ($h_1 = 0$) and making the compliance equally compliant in the axial as well as the lateral direction ($c_1 = c_2$).

The compliance effect on the passive control scheme is obtained from writing Expression B-26 in terms of the deformations of the compliance.

$$\frac{df_2}{f_1} = -\delta\omega_3$$

$$\frac{dm_3}{f_1 r} = \frac{\delta r_2}{r}\left(1 - \frac{c_1}{c_2}\right) - \frac{c_1}{c_2}\frac{h_1}{r}\delta\omega_3 \qquad \text{B-27}$$

This expression also suggests designing the compliance with its center at the tip of the peg and equally compliant in axial and lateral directions.

These results can be extended to the case where only some components of the force vector and directly controlled.

NOTE: Expression B-26 and B-27 do not include terms from variation of the axial force due to the deformation of the compliance. These terms vanish for large values of the axial force (f_1) and thus were omitted.

APPENDIX C

THEORETICAL PERFORMANCE EVALUATION

1. INTRODUCTION

The purpose of this appendix is to obtain a measure for the performance of the class of position methods described in Chapter II. The approach used for obtaining this measure is to compute the lower bound of the error in the estimate of the relative position between the parts being assembled. The analysis is done on the planar insertion of a peg into a hole; this analysis uses the three dimensional, axial and lateral displacements and angular alignment, versions of the position measurement model and the geometric relations for the peg and hole mating.

The independent variables in this performance analysis are: the resolution of the position measurements, the number of measurements used in the estimation and the total displacement of the movements measured. The statistics for the error on the position estimation is expressed in terms of those variables.

2. GEOMETRY MODEL

The realtive position between two parts is represented by the relative position between reference frames placed on each part. For the peg and hole case analyzed in this appendix, the locations of the frames and the components of the vector of relative positions $\underline{x}$, are shown in Figure C-1.

NOTE: The upper indexes used in the notation definitions of Appendix A have been dropped.

For the conditions of the two bodies touching, shown in the figure, the following condition has to be met:

$$(x_2^2 - x_1^2)\ \sin x_6 + 2\ x_1\ x_2\ \cos x_6 = (r_h^2 - r_p^2)\ \sin x_6 \qquad \text{C-1}$$

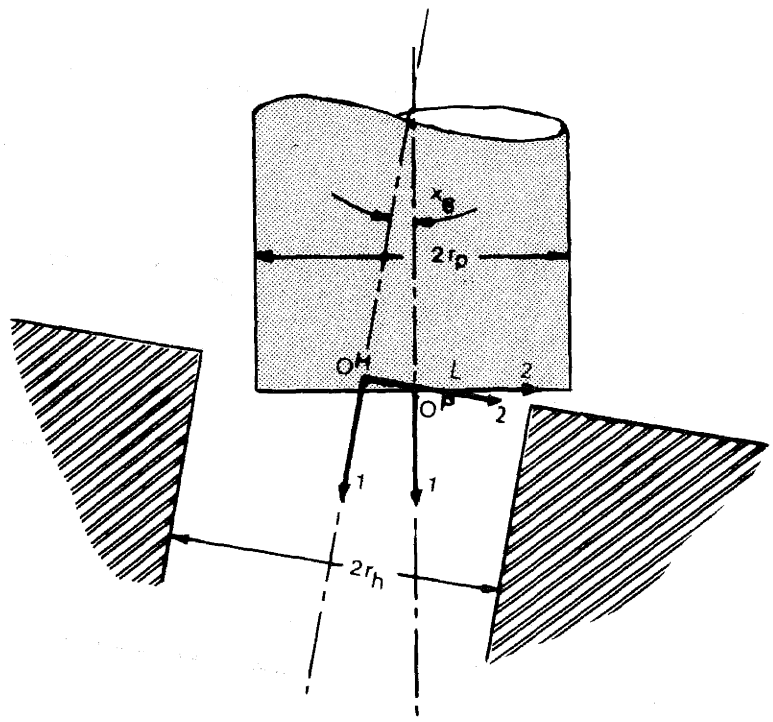


Figure C-1: Two Dimension Peg and Hole, Notation

where r_h and r_p are respectively the hole and peg radii.

Normalizing the displacement component with respect to the hole's radius and using the clearance ratio defined as:

$$c = \frac{r_h - r_p}{r_h} \qquad \text{C-2}$$

Expression C-1 becomes:

$$(x_2^2 - x_1^2)\ \sin x_6 + 2\ x_1\ x_2\ \cos x_6 - (2 - c)\ c\ \sin x_6 = 0 \qquad \text{C-3}$$

For the purpose of this simplified analysis it will be assumed that the peg does not slide into the hole but only rotates around point L in Figure C-1.

3. MEASUREMENT MODEL

The form of the three dimensional measurement model used in this appendix is much simpler than the general six dimensional version. The position measurement, that in the general six dimensional case is a product of geometric transformations between a bias term, the actual position between the parts and a measurement resolution noise term, becomes the result of simple addition between these terms. Thus, the measurement model is written as follows:

$$\underline{z}(n) = \underline{x}(n) + \underline{b} + \underline{v}(n) \qquad \text{C-4}$$

where:

n denotes the time dependence of the variables;

$\underline{z}(n)$ denotes the position measurement at time n;

$\underline{x}(n)$ is the actual relative position between the parts at mating at time n. The first and second components of the vector are the displacement components, the sixth component

is the angular rotation between the parts;

$\underline{b}$ is the bias term, fixed in time. For the purpose of the analysis this bias will be assumed totally unknown;

$\underline{v}(n)$ represents the noise resolution term. In this analysis this will be assumed to be a white random Gaussian vector with covariance matrix given by:

$$E\left[\underline{v}(n)\ \underline{v}^{T}(m)\right] = \delta_{nm}\ R \qquad \text{C-5}$$

The measurement vector $\underline{z}$, the position vector $\underline{x}$, the bias term $\underline{b}$ and the noise $\underline{v}$ are assumed normalized with respect to a characteristic dimension of the parts, in this case the radius of the hole. The algebraic manipulation is greatly simplified if the covariance matrix R is assumed diagonal with equal terms, thus:

$$R = r\ I \qquad \text{C-6}$$

Where I is the three by three identity matrix. This assumption means that the angular resolution noise is numerically equal to the ratio of the position resolution to radius of the hole; this is a reasonable assumption that does not limit the generality of the results.

4. ESTIMATOR LOWER BOUND

The concept behind the position measurement method is simply to fit the time series of measurements $\underline{z}(n)$ to both the geometry and measurement models. This fit yields the estimate of the time series of actual positions $\underline{x}(n)$ and the estimate of the bias vector $\underline{b}$.

A reasonable criteria to use for this fit is the maximum likelihood criteria; the maximum likelihood estimate for the vectors $\underline{x}(n)$ and $\underline{b}$ is defined in the following paragraphs.

For convenience let's define the following expanded vectors:

$$\underline{Z} = \begin{bmatrix} \underline{z}(1) \\ \underline{z}(2) \\ \cdot \\ \cdot \\ \cdot \\ \underline{z}(N) \end{bmatrix} \quad ; \quad \underline{X} = \begin{bmatrix} \underline{b} \\ \underline{x}(1) \\ \underline{x}(2) \\ \cdot \\ \cdot \\ \underline{x}(N) \end{bmatrix} \quad ; \quad \underline{V} = \begin{bmatrix} \underline{v}(1) \\ \underline{v}(2) \\ \\ \cdot \\ \cdot \\ \underline{v}(N) \end{bmatrix} \qquad \text{C-7}$$

In terms of $\underline{X}$, $\underline{Z}$, and $\underline{V}$ the measurement equation becomes:

$$\underline{Z} = \Omega \, \underline{X} + \underline{V} \qquad \text{C-8}$$

And the covariance of the noise term is written as:

$$\ell = E\left[\underline{V} \, \underline{V}^T \right] \qquad \text{C-9}$$

Given a set of measurements $\underline{Z}_t$, the maximum likelihood estimate of $\underline{X}$ is the value of this vector that maximizes the probability density of $\underline{Z}$, evaluated at the actual measurement $\underline{Z}_t$ and, at the same time, satisfies the geometric model at all times considered. The maximization just described can also be performed on a monotonic function of the probability density function, the logorithm of the probability density function is defined as the log likelihood function, denoted here as:

$$\xi(N : \underline{X}) = \ln(p(\underline{Z} : \underline{X})) \qquad \text{C-10}$$

The log likelihood function is a function of the variable $\underline{Z}$ and thus is random, its average is called the ambiguity function and is formally defined as:

$$\gamma(\underline{X}_t : \underline{X}) = E\left[\ln(p(\underline{Z} : \underline{X}))\right]$$

$$\gamma(\underline{X}_t : \underline{X}) = \int \ln(p(\underline{Z} : \underline{X}))\, p(\underline{Z} : \underline{X}_t)\, d\underline{Z} \qquad \text{C-11}$$

where $\underline{X}_t$ denotes the actual true value of the vector $\underline{X}$.

If $\hat{\underline{X}}$ denotes the estimate of the vector $\underline{X}$, that will be assumed to be an unbiased estimate for the purpose of this analysis, and measuring Σ the covariance on the error of this estimate we have:

$$E\left[\hat{\underline{X}}\right] = \underline{X}_t \qquad \text{C-12}$$

$$\Sigma = E\left[(\underline{X}_t - \hat{\underline{X}})\,(\underline{X}_t - \hat{\underline{X}})^T\right] \qquad \text{C-13}$$

From Reference C-1, a lower bound for this estimated error is given by:

$$\Sigma \geq B^{-1} \qquad \text{C-14}$$

where B is:

$$B = \left[\frac{\partial^2}{\partial \underline{X}^2}\, 2\gamma(\underline{X}_t : \underline{X})\right] \qquad \text{C-15}$$

The matrix B^{-1} is the performance measure used in this appendix for the estimation of the vector $\underline{X}$. This matrix represents a lower bound to the error on the estimate, this means that a good performance index does not necessarily guarantee a good performance of the estimator, it only implies that a good performance estimator exists.

5. PERFORMANCE MEASUREMENT EVALUATION

For clarity the construction of the matrix B^{-1} is done in three steps. First,

through the measurement model, the ambiguity function is evaluated. Second, the expression of the matrix B is obtained by imposing the constraints given by the geometry model, Section Two, and computing the second derivatives of the ambiguity function, Expression C-15. Finally partitions of the matrix B^{-1} are computed and analyzed in terms of the performance of the estimation problem.

Ambiguity function evaluation: It is assumed that the noise term in Equation C-4 is Guassian the probability density and the log likelihood function respectively becomes:

$$p(\underline{Z} : \underline{X}) = \left[(2\pi)^{3N} \, |\ell| \right]^{-1} \exp\left[-1/2(\underline{Z} - \Omega\underline{X})^T \, \ell^{-1}(\underline{Z} - \Omega\underline{X}) \right] \qquad \text{C-16}$$

$$2\xi(\underline{Z} : \underline{X}) = -3N \log(2\pi) - \log|\ell| - (Z - \Omega X)^T \, \ell^{-1}(\underline{Z} - \Omega\underline{X}) \qquad \text{C-17}$$

Using Expression C-9 and Expression C-6 we have:

$$|\ell| = r^{3N} \qquad \text{C-18}$$

$$(\underline{Z} - \Omega\underline{X})^T \, \ell^{-1} \, (\underline{Z} - \Omega X) = 1/r \sum_{1}^{N} (\underline{z}(n) - \underline{x}(n) - \underline{b})^T (\underline{z}(n) - \underline{x}(n) - \underline{b})) \qquad \text{C-19}$$

$$\gamma(\underline{X},\underline{X}_t) = \gamma_{bias}(\underline{X},\underline{X}_t) + \gamma_{obs}(\underline{X},\underline{X}_t) \qquad \text{C-20}$$

where:

$$2 \, \gamma_{bias}(\underline{X},\underline{X}_t) = -3N \, \ln(2\pi r) \qquad \text{C-21}$$

$$2 \, \gamma_{obs}(\underline{X},\underline{X}_t) = 1/r \sum_{1}^{N} E\left[(\underline{z}(n) - \underline{x}(N) - \underline{b})^T \, (\underline{z}(n) - \underline{x}(n) - \underline{b}) \right] \qquad \text{C-22}$$

denoting:

$$\delta\underline{x}(n) = \underline{x}(n) - \underline{x}_t(n) \qquad \text{C-23}$$

(cont)

$$\delta \underline{b} = \underline{b} - \underline{b}_t \qquad \text{C-23}$$

and we know that:

$$\underline{v}(n) = \underline{z}(n) - \underline{x}(n) - \underline{b}$$

$$E\left[\underline{v}^T(n)\ \underline{v}(n)\right] = \text{trace } E\left[\underline{v}(n)\ \underline{v}^T(n)\right] = 3r \qquad \text{C-24}$$

then:

$$E\left[(\underline{z}(n) - \underline{x}(n) - \underline{b})^T(\underline{z}(x) - \underline{x}(n) - \underline{b})\right] = E\left[(\underline{v}(n) - \delta\underline{x}(n) - \delta\underline{b})^T(\underline{v}(n) - \delta\underline{x}(n) - \delta\underline{b})\right]$$

$$= 3r + E\left[(\delta\underline{x}(n) + \delta\underline{b})^T(\delta\underline{x}(n) + \delta\underline{b})\right]$$

and:

$$-\left[2\ \gamma_{obs}(\underline{X} : \underline{X}_t) + 3N\right] r = \sum_1^N (\delta\underline{b} + \delta\underline{x}(n))^T(\delta\underline{b} + \delta\underline{x}(n)) \qquad \text{C-25}$$

Matrix B evaluation: From the expressions of the ambiguity function Expression C-20, Expression C-21 and Expression C-25, only γ_{obs} yields terms to the derivatives of the ambiguity function. The differentiation defined in Expression C-15 has to satisfy the geometric constraint imposed by Expression C-3, this is done by introducing the following change in variables:

$$x_1(n) = \rho(n) \sin x_6(n)$$

$$x_2(n) = q(n) - \rho(n) \cos x_6(n) \qquad \text{C-26}$$

with:

$$q^2(n) = c(2 - c) + \rho^2(n)$$

The parameter c is the clearance ratio and was defined previously in Section Two. This change of variables satisfies Expression C-3 automatically. The simplifying assumption, stated in Section Two, that the point L (Figure C-1) is fixed, makes both q(n) and p(n) independent of the parameter n. If the, now constant, parameter q is added to the component b_2 of the vector $\underline{b}$ a new generalized vector $\underset{\sim}{\underline{X}}$ can be defined as follows:

$$\underset{\sim}{\underline{X}} = \begin{bmatrix} b_1 \\ b_2 + q \\ b_6 \\ \rho \\ x_6(1) \\ \cdot \\ \cdot \\ x_6(N) \end{bmatrix} \qquad \text{C-27}$$

The upper "similar to ($\sim$)" in the vector $\underset{\sim}{\underline{X}}$, defined in Expression C-27, will be omitted herein. The differentiation is performed with respect to the components of this newly defined vector, thus the geometric constraints are always satisfied. From the expression of the ambiguity function (Expression C-20, Expression C-21 and Expression C-25) and the change of variables given in Expression C-26, the second derivatives of the ambiguity function with respect to the components of the vector $\underset{\sim}{\underline{X}}$, defined in Expression C-27, are given by:

$$\frac{\delta^2}{\delta \underline{b}^2}\left[-r\gamma(\underline{x}_t : \underline{x})\right] = N\ I$$

$$\frac{\delta^2}{\delta \rho^2}\left[-r\gamma(\underline{x}_t : \underline{x})\right] = N$$

$$\frac{\delta^2}{\delta x_6^2(n)}\left[-r\gamma(\underline{x}_t : \underline{x})\right] = 1 + \rho^2 + (\delta\underline{b} + \delta\underline{x}(n))^T \begin{bmatrix} \sin x_6(n) \\ \cos x_6(n) \\ 0 \end{bmatrix} \cdot \rho \qquad \text{C-28}$$

$$\frac{\delta^2}{\delta \underline{b}\delta\rho}\left[-r\gamma(\underline{x}_t : \underline{x})\right] = \Sigma_n \begin{bmatrix} \sin x_6(n) \\ -\cos x_6(n) \\ 0 \end{bmatrix}$$

$$\frac{\delta^2}{\delta \underline{b}\delta x_6(n)}\left[-r\gamma(\underline{x}_t : \underline{x})\right] = \begin{bmatrix} \rho \cos x_6(n) \\ \rho \sin x_6(n) \\ 1 \end{bmatrix}$$

$$\frac{\delta^2}{\delta\rho\delta x_6(n)}\left[-r\gamma(\underline{x}_t : \underline{x})\right] = (\delta\underline{b} + \delta\underline{x})^T \begin{bmatrix} \cos x_6(n) \\ -\sin x_6(n) \\ 0 \end{bmatrix}$$

To form the matrix B defined in C-15, the derivatives have to be evaluated at $\underline{X} = \underline{X}_t$, this cancels all the terms containing $\delta\underline{b}$ and $\delta\underline{x}(n)$ as a factor. The matrix B is then given by:

$$B=\begin{bmatrix}
N & 0 & 0 & \Sigma_n \sin x_6(n) & \rho\cos x_6(1) & \rho\cos x_6(2) & \cdots & \rho\cos x_6(N) \\
0 & N & 0 & \Sigma_n \cos x_6(n) & \rho\sin x_6(1) & \rho\sin x_6(2) & \cdots & \rho\sin x_6(N) \\
0 & 0 & N & 1 & 1 & 1 & \cdots & 1 \\
\Sigma_n \sin x_6(n) & -\Sigma_n \cos x_6(n) & 1 & N & 0 & 0 & \cdots & 0 \\
\rho\cos x_6(1) & \rho\sin x_6(1) & 1 & 0 & \rho^2+1 & 0 & \cdots & 0 \\
\rho\cos x_6(2) & \rho\sin x_6(2) & 1 & 0 & 0 & \rho^2+1 & \cdots & - \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & - \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & - \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & - \\
\rho\cos x_6(N) & \rho\sin x_6(N) & 1 & 0 & 0 & 0 & \cdots & \rho^2+1
\end{bmatrix}$$

Performance evaluation: The inverse of matrix B has to be evaluated to obtain the performance of the estimator, as defined in Section Four. Only the left top four by four partition of the matrix B^{-1} will be computed, for doing so we use the formula for the inverse of a partitioned matrix (See Reference C-2). Thus:

$$(B_{11}^{-1})^{-1} = B_{11} - B_{12}(B_{22})^{-1} B_{21} \qquad \text{C-30}$$

Where B_{ij} and B_{ij}^{-1} are the matrices on the i^{th} row and j^{th} column resulting from partitioning the respective matrices B and B^{-1}.

The matrix B_{22} is diagonal and thus has a direct inverse. The matrix B_{11}^{-1} is the lower bound to the error covariance of the vector formed by the three components of the bias vector $\underline{b}$ and the parameter ρ:

$$E\left[\left[\frac{b}{\rho}\right]\left[\underline{b}^T\rho\right]\right] \geq B_{11}^{-1} \qquad \text{C-31}$$

For notation convenience the following change in variable $X_6(n)$ is introduced:

$$x_6(i) \equiv \phi_i \quad i = 1,N \qquad \text{C-32}$$

The expression of $(B_{11}^{-1})^{-1}$ is then:

$$(B_{11}^{-1})^{-1} = \frac{1}{r}\begin{bmatrix} N - \frac{\rho^2}{\rho^2+1}\Sigma_i\cos^2\phi_i & \frac{-\rho^2}{\rho^2+1}\Sigma_i\sin\phi_i\cos\phi_i & \frac{-\rho}{\rho^2+1}\Sigma_i\cos\phi_i & \Sigma_i\sin\phi_i \\ \frac{-\rho^2}{\rho^2+1}\Sigma_i\sin\phi_i\cos\phi_i & N - \frac{\rho^2}{\rho^2+1}\Sigma_i\sin^2\phi_i & \frac{-\rho}{\rho^2+1}\Sigma_i\sin\phi_i & -\Sigma_i\cos\phi_i \\ \frac{-\rho}{\rho^2+1}\Sigma_i\cos\phi_i & \frac{-\rho}{\rho^2+1}\Sigma_i\sin\phi_i & N\frac{\rho^2}{\rho^2+1} & -\Sigma_i\cos\phi_i \\ \Sigma_i\sin\phi_i & -\Sigma_i\cos\phi_i & 0 & N \end{bmatrix}$$

The inverse of this matrix is computed by partitioning it in four two by two matrices. The two by two left top partition of the matrix B_{11}^{-1} represents the lower bound to the error on the estimate to the position bias (b_1,b_2), the eigenvalues of this matrix give a good measure of the error on the estimate of the bias position. The other two diagonal elements to the matrix B_{11}^{-1} (third and fourth) give the lower bounds on the errors to the estimates of the bias angle (b_6) and the parameter ρ respectively. The following notation is introduced to avoid proliferation of sub indexes:

$$B_{11}^{-1} = C \qquad \text{C-34}$$

Then, using the same formula as in Relation C-30 for the partitioned matrix inversion we get:

$$(C_{11})^{-1} = C_{11}^{-1} - C_{12}^{-1}\,(C_{22}^{-1})^{-1}\,C_{21}^{-1} \qquad \text{C-35}$$

$$(C_{11})^{-1} = \frac{1}{Nr(\rho^2+1)} \cdot W \qquad \text{C-36}$$

where:

$$W = \begin{bmatrix} N^2 - \Sigma_{ij}\cos(\phi_i - \phi_j) + & \rho^2\Sigma_{ij}(\sin\phi_i\cos\phi_j - \sin\phi_i\cos\phi_i) \\ + \rho^2\Sigma_{ijk}(\sin^2\phi_i - \sin\phi_i\sin\phi_j) & \\ & N^2 - \Sigma_{ij}\cos(\phi_i - \phi_j) + \\ \rho^2\Sigma_{ij}(\sin\phi_i\cos\phi_j - \sin\phi_i\cos\phi_i) & \\ & + \rho^2\Sigma_{ij}(\cos^2\phi_i - \cos\phi_i\cos\phi_j) \end{bmatrix}$$

If λ is an eigenvalue of $(C_{11})^{-1}$ and making:

$$\lambda N(p^2+1)r - (N^2 - \Sigma_{ij}\cos(\phi_i - \phi_j)) = p^2\mu \qquad \text{C-37}$$

then:

$$\det\left[\lambda I - (C_{11})^{-1}\right] = \left(\frac{p^2}{N(p^2+1)r}\right)^2\left[\mu^2 - L_1\mu + L_2\right] \qquad \text{C-38}$$

where

$$L_1 = \Sigma_{ij}(\cos^2 \phi_i + \sin^2 \phi_i - \cos \phi_i \cos \phi_j - \sin \phi_i \sin \phi_j) =$$

$$= \Sigma_{ij}\left[1 - \cos (\phi_i - \phi_j)\right] \qquad \text{C-39}$$

$$L_2 = N^2\Sigma_{ij} (\cos^2 \phi_i \sin^2 \phi_j - \sin \phi_i \cos \phi_i \sin \phi_j \cos \phi_j) + \qquad \text{C-40}$$

$$+ N\Sigma_{ijk}(2 \sin \phi_i \cos \phi_i \sin \phi_j \cos \phi_k - \cos^2 \phi_i \sin \phi_j \sin \phi_k - \sin^2 \phi_i \cos_j \cos \phi_k)$$

$$= N/2\ \Sigma_{ijk}\left[\sin^2(\phi_i-\phi_j) + \cos(2\ \phi_i-\phi_j-\phi_k) - \cos(\phi_i-\phi_j)\right]$$

Note that the expression for μ and λ will not depend on absolute values of the ϕ_i angles but on the difference between these angles. To obtain a close form solution for the eigenvalues, the following assumptions are made:

(i) the true points that generate the data are evenly spaced, thus:

$$\phi_i - \phi_j = (i - j)\varepsilon \qquad \text{C-41}$$

(ii) the angles on the preceding expressions are small, the trigonometric expressions can be approximated by Taylor expansions.

Expression C-37 then becomes:

$$\lambda = \frac{\rho^2 \mu}{Nr(\rho^2+1)} + \frac{1}{Nr(\rho^2+1)} \left[\Sigma_{ij}\, i^2 - (\Sigma_i i)^2 \right] \epsilon^2 \qquad \text{C-42}$$

$$\lambda = \frac{\rho^2 \mu}{Nr(\rho^2+1)} + \frac{N^2(N^2-1)}{12Nr(\rho^2+1)} \epsilon^2$$

If we define the total displacement angle as:

$$\Delta\phi = N\, \epsilon \qquad \text{C-43}$$

and if we consider N to be large thus:

$$N^2 - 1 \simeq N^2$$

Expression C-42 becomes:

$$\lambda \simeq \frac{\rho^2 \mu}{Nr(\rho^2+1)} + \frac{\Delta\phi^2 N}{12r(\rho^2+1)} \qquad \text{C-44}$$

Expression C-39 is written:

$$L_1 \simeq \frac{N^2 \Delta\phi}{12} \qquad \text{C-45}$$

In the expression for L_2 (Expression C-40), the use of the assumption in Expression C-41 and the Taylor expansion for the trigonometric functions results in cancellation of the terms with order on ϵ smaller than six thus we get:

$$L_2 \simeq \frac{N\,\varepsilon^6}{2\ 6!}\ \Sigma_{ijk}\left[33(i-j)^6 - (2i-j-k)^6\right]$$

$$= \frac{N\,\varepsilon^6}{4}\ \Sigma_{ijk}(i^4j^2 - i^3j^3 - i^4jk + 2i^3j^2k - i^2j^2k^2) \qquad \text{C-46}$$

$$= \frac{N\varepsilon^6}{12\ 6!}\ N^3(N^2-1)^2\ (N^2-4) \simeq \frac{N^4\Delta\phi^6}{12\ 6!}$$

Equating Expression C-38 to zero and computing the roots of the polynomial μ_1 and μ_2 we have:

$$\mu = \frac{N^2\Delta\phi^2}{24}\left(1 \pm \sqrt{1 - \frac{\Delta\phi^2}{15}}\right) \qquad \text{C-47}$$

For small values of $\Delta\phi$ we get:

$$\mu_1 \simeq \frac{N^2\Delta\phi^2}{12} \qquad \text{C-48}$$

$$\mu_2 \simeq \frac{N^2\Delta\phi^4}{720}$$

Thus:

$$\lambda_1 = \frac{N\Delta\phi^2}{12r}\ ;\quad \lambda_2 = \frac{N\Delta\phi^2}{12(\rho^2+1)r}\left[\frac{\rho^2\Delta\phi^2}{60} + 1\right] \qquad \text{C-49}$$

The eigenvalues for the matrix C_{11} are the inverse of the eigenvalues of $(C_{11})^{-1}$, thus the performance index for the estimation of the position bias is given by the square root of the inverses of λ_1 and λ_2.

$$\sqrt{1/\lambda_1} = \sqrt{\frac{12r}{N}} \quad \frac{1}{\Delta\phi} \qquad \text{C-50}$$

$$\sqrt{1/\lambda_2} \simeq \sqrt{\frac{12r(\rho^2+1)}{N}} \quad \frac{1}{\Delta\phi} \left[1 - \frac{\rho^2\Delta\phi^2}{120}\right]$$

The lower bound for the errors on the estimate of the bias angle and on the variable ρ are given by the square root of the third and fourth diagonal elements of the matrix C.

$$C_{22} = (C_{22}^{-1})^{-1} + (C_{22}^{-1})^{-1} C_{21}^{-1} C_{11} C_{12}^{-1} (C_{22}^{-1})^{-1} \qquad \text{C-51}$$

Using the same assumptions and algebraic manipulations on the calculations of the diagonal terms of C_{22} and neglecting the higher order terms on ε, the following lower bounds are obtained:

$$\sigma(\delta b_3) = E\left[b_3^2\right] \geq \sqrt{\frac{12r}{N\rho}} \quad \frac{1}{\Delta\phi} \qquad \text{C-52}$$

$$\sigma(\delta\rho) = \sqrt{E\left[\rho^2\right]} \geq \sqrt{\frac{12(\rho^2+1)r}{N}} \quad \frac{1}{\Delta\phi}$$

6. SOME REMARKS

The measures used for the performance of the bias estimator are given by Expression C-50 and C-52. For the position estimate performance the largest of the

two expressions is chosen; the square root of the inverse of the smallest eigenvalue is a reasonable index for the standard deviation of the position estimate error. The lower bound for $\sigma(b_6)$ is given in Expression C-52; this is a reasonable performance measure for the standard deviation of the error on the estimate of the bias angle.

If we assume that the resolution of the measurement ($\sqrt{r}$) is small, a crude but reasonable estimate of the vector $\underline{x}$ is given by:

$$\hat{\underline{x}}(n) \simeq \underline{z}(n) - \underline{b} \qquad \text{C-53}$$

thus

$$E\left[\delta\underline{x}(n)\,\delta\underline{x}^T(n)\right] = E\left[\delta\underline{b}\;\delta\underline{b}^T\right] + rI \qquad \text{C-54}$$

And then, the performance measures for the bias ($\underline{b}$) and the state $\underline{x}(n)$ are, for practical purposes, equal.

The role of each of the parameters involved in the estimation of the state of the assembly is given by Expression C-52 and Expression C-54. In summary the standard deviation on the error of the estimate of the state of the assembly is:

- directly proportional to the standard deviation of the sensor measurement noise ($\sqrt{r}$);
- inversely proportional to the square root of the number of measurements (N);
- inversely proportional to the length of the trajectory used for obtaining the measurements ($\Delta\phi$).

APPENDIX D

GEOMETRIC INFORMATION

1. INTRODUCTION

The position method described in Chapter III postulates that the gross uncertainties in the relative position measurements between the parts being assembled can be reduced if the following conditions are met:

- the position measurement between the parts being assembled follows the form postulated in Chapter I. Specifically this means that the uncertainties on these measurements can be traced down to two main components: a time dependent noise component, of small magnitude compared to the size of the parts, generated by the resolution uncertainties of the position measurements device, and a bias component, fixed in time, of magnitude comparable to the size of the parts;
- parts are kept in contact throughout the gathering of the position measurements;
- a representation of geometric conditions of the two parts in contact, in terms of the relative position between the parts, is available.

This appendix formulates the relations for the geometric constraints of two parts touching. The code derived from these relations, and listed in Section Three of this appendix, was used in the implementation of the position measuring technique described in Chapter II.

The derivation of the geometric relations in this appendix was done for the particular case of a round peg and round hole touching. This procedure, realized

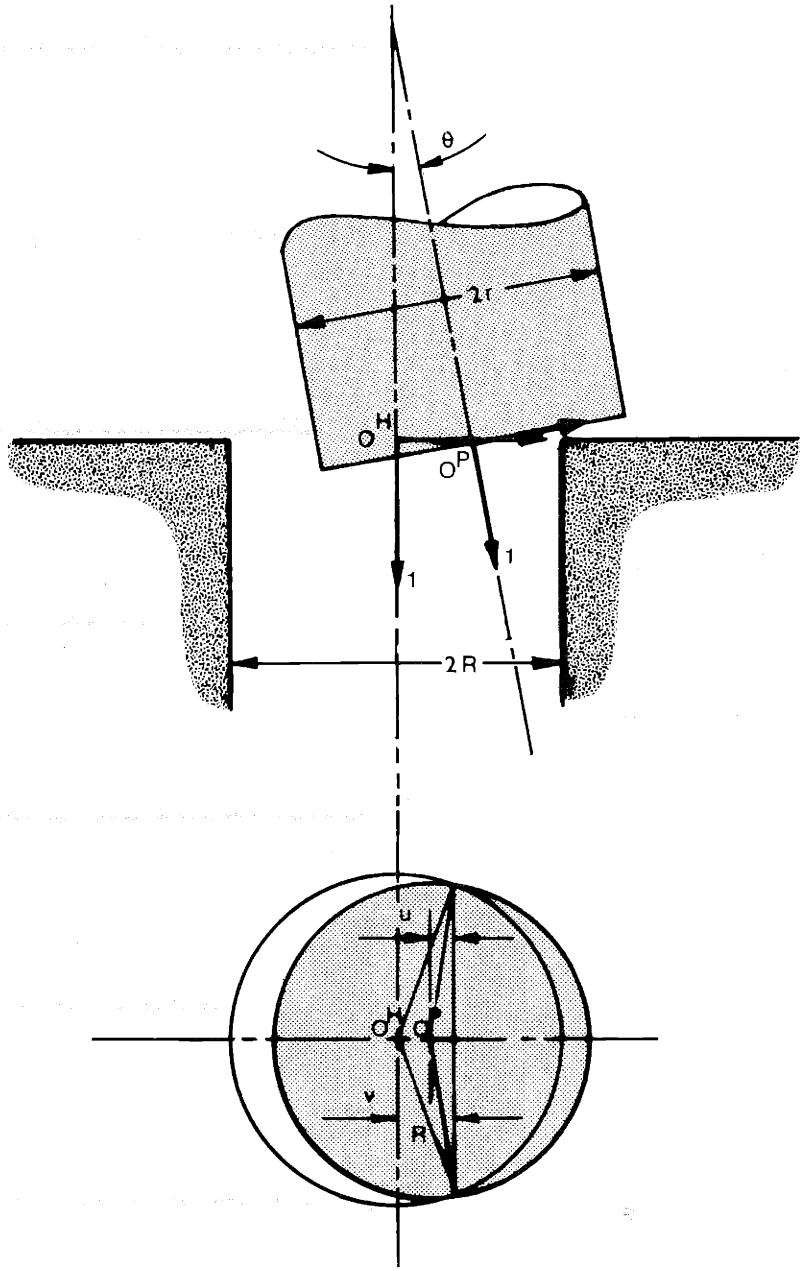


Figure D-1: Peg and Hole Touching Condition, Notation

here for a particular set of geometric parts, could be expanded for general types of mating surfaces; a development of this nature would require developments in the areas of geometric descriptions of solids and topology of touching solid geometries. Extensive work has been done on this first area for applications such as Automatic Drafting systems, Numerical Control Machine tape generation, etc. The development of a general technique for formulating the geometry constaints was not pursued in this work.

The next section of this appendix derives the relations for the geometric constraints of a peg and hole touching. The last section lists the Fortran code for this geometric implementation.

2. PEG AND HOLE TOUCHING RELATIONS

Figure D-1 shows the coordinate systems used for defining the relative position between the two parts and shows the configuration analyzed. From Figure D-1:

$$R^2 - v^2 = r^2 - u^2$$

$$x_1^{PH} = u \sin\theta \qquad \text{D-1}$$

$$\rho = v - u\cos\theta \qquad \rho = (x_2^{PH})^2 + (x_3^{PH})^2$$

where: R, r are the radii of hole and peg respectively;

u,v are shown in Figure D-2;

ρ is the distance from the peg frame to the axis of the hole;

x_i^{PH} i = 1,6 is the vector of the relative position of the peg with respect to the hole.

Introducing the translation components of the relative position vector $\underline{x}^{PH}$ and the misalignment angle θ, shown in Figure D-1, Expression D-1 is reduced to:

$$\left(\sqrt{(x_2^{PH})^2 + (x_3^{PH})^2}\,\sin\theta + x_1^{PH}\cos\theta\right)^2 - (x_1^{PH})^2 - (R^2 - r^2)\sin\theta = 0 \qquad \text{D-2}$$

Defining the clearance ratio :

$$C = \frac{R - r}{r} \qquad \text{D-3}$$

And introducing in Expression D-2 the general position matrix B^{HP} defined in Appendix A:

D-4

$$\left[(b_{31}^{HP})^2 + (b_{41}^{HP})^2 - (b_{21}^{HP})^2 - R^2(2 - c)\right]\sqrt{1-(b_{22}^{HP})^2} + 2b_{21}^{HP}\,b_{22}^{HP}\,\sqrt{(b_{31}^{HP})^2 + (b_{41}^{HP})^2} = 0$$

This expression is satisfied for the case where the vectors $\underline{e}_1^H$ and $\underline{e}_1^P$ are coplanar. the relation for this condition is written:

$$\underline{r}^{PH} \cdot (\underline{e}_1^P \times \underline{e}_2^H) = 0 \qquad \text{D-5}$$

Using the definitions of the rotation matrix A^{HP} (Expression A-3), the generalized position matrix B^{HP} (Expression A-5) and the expressions for the dot and cross products (Expression A-2 and Expression A-12) we have:

$$b_{41}^{HP} \cdot b_{32}^{HP} - b_{42}^{HP}\,b_{31}^{HP} = 0 \qquad \text{D-6}$$

The geometric constraint, as defined in Chapter III, is made by making the two components of the vector $\underline{h}$ equal to the Expression D-4 and Expression D-6 respectively. Thus:

$$\underline{h} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \qquad \text{D-7}$$

where:

$$h_1 = \left[(b_{31}^{HP})^2+(b_{41}^{HP})^2-(b_{21}^{HP})^2-Rc(2-c) \right] \sqrt{1-(b_{22}^{HP})^2} + 2b_{21}^{HP} b_{22}^{HP} \sqrt{(b_{31}^{HP})^2+(b_{41}^{HP})^2}$$

$$h_2 = b_{41}^{HP} \cdot b_{32}^{HP} - b_{42}^{HP} b_{31}^{HP}$$

The computation of the matrix derivative of the vector $\underline{h}$ with respect to the vector position $\underline{x}^{HP}$ is done through the following two steps:

- compute the tensor derivative of the vector $\underline{h}$ with respect to the components of the matrix B^{HP} by direct differentiation of Expression D-7, thus obtaining $\frac{\partial \underline{h}}{\partial B^{HP}}$;
- through the use of the transformation tensor $\frac{DB^{HP}}{D\underline{x}^{HP}}$ developed in Appendix A, the expression of $\frac{\partial \underline{h}}{\partial \underline{B}^{HP}}$ is transformed to $\frac{\partial \underline{h}}{\partial \underline{x}^{HP}}$.

The expression for the derivatives is not given here but can be easily obtained from Expression D-7.

3. GEOMETRIC CONSTRAINT IMPLEMENTATION

The expression for the geometric constraint of a peg and hole touching is given in Relation D-7. That expression and the expression of its derivatives with respect to the position vector $\underline{x}^{HP}$ are implemented in the routine called "TPC6D".

This routine uses routines XTOB and DBDX developed and listed in Appendix A.

NOTE: For computational expediency the general position matrix B, defined in A-5, has been reduced to a 3 by 4 matrix by eliminating its first row; thus all terms in Expression D-7 and its derivatives have been implemented with the following permutation:

$$b_{ij} \rightarrow b_{i-1\ j}$$

```
C********************** TPC6D,SS ***************************
C
C
C
C.......THIS SUBROUTINE COMPUTES THE GEOMETRY VECTOR H FOR THE TWO
C.......POINT CONTACT PEG ON HOLE AND THE DERIVATIVES OF THE VECTOR
C.......H ("DH") WRT.THE POSITION VECTOR XPH
C
C.......INPUT : BHP= 3X4 MTX OF HOLE W.R.T. PEG
C     ...OUTPUT: N=DIMENSION OF THE OUTPUT VECTOR
C.......          H= GEOMETRY VECTOR (=0 WHEN SATISFIED)
C                 DH=DERIVATIVE MATRIX OF THE VECTOR H
C
C        (REFERENCE : A-73)
C
C
C
C
        SUBROUTINE TPC6D(XHP,H,DH,N)
        COMPILER DOUBLE PRECISION
        COMMON /GEOMETRY / RDS,CTRR
        DIMENSION XHP(6),BHP(3,4),H(2),DH(2,6),DD(3,4,6)
C
        N=2
        CALL XTOB(XHP,BHP)
C
C....SCALE THE IMPUT POSITION (EXTENSION VRBLS) BY THE RADIUS OF HOLE
        HP1=BHP(1,1)/RDS
        HP2=BHP(2,1)/RDS
        HP3=BHP(3,1)/RDS
C
C....EITHER OF THE TWO FOLLOWING EXPRESIONS ARE EQUIVALENT
CC      Z=(1-BHP(1,2)**N)
        Z=BHP(2,2)**N+BHP(3,2)**N
        Z=SQRT(Z)
        V=HP3*HP3+HP2*HP2
        Y=SQRT(V)
        U=V-HP1*HP1-CTRR*(N-CTRR)
C
C....THE NEXT SET OF STATEMENTS AVOID INTERFEARENCE OF PEG & HOLE VOLUMES
        H1=BHP(3,2)*HP3+BHP(2,2)*HP2
        IF(H1 .GT. 0.) GO TO 1
        H(1)= H1
        D132= HP3
        D131= BHP(3,2)/RDS
        D122= HP2
```

```
      D121= BHP(2,2)/RDS
      D111= 0.
      D122= 0.
      MM=1
      GO TO 3
C
C     THE NEXT SET OF STATEMENTS AVOID THE PEG FROM GOING INTO
C     THE HOLE
C
1     CONTINUE
      H1= Y - 1.+(1.-CTRR)*BHP(1,2)
      IF(H1 .GT. 0.) GO TO 2
      H(1)=H1
      D131= HP3/(Y*RDS)
      D121= HP2/(Y*RDS)
      D112= 1.-CTRR
      D132= 0.
      D122= 0.
      D111= 0.
      MM=2
      GO TO 3
C
C...H1=0 IS THE CONDITION FOR TWO POINT CONTACT AT MOUTH OF HOLE
C
2     CONTINUE
      H(1)=Z*U+2*HP1*BHP(1,2)*Y
C
C...DH=DERIVATIVES, THE ONES DIVIDED BY RDS ARE THE DERIVATIVES
C       OF EXTENSION VARIABLES; FOR DETAILS SEE C-2
C
      D111=2*(BHP(1,2)*Y-Z*HP1)/RDS
      D131=2*(Z*HP1*BHP(1,2)/Y)/RDS
      D121=D131*HP2
      D131=D131*HP3
      D112=2*HP1*Y-BHP(1,2)*U/Z
      D132=0.
      D122=0.
      MM=0
3     CONTINUE
C
C...H2=0 IS SATISFIED WHEN THE AXES OF PEG AND HOLE ARE COPLANAR
C
      H(2)=-BHP(3,2)*HP2+BHP(2,2)*HP3
C
C     NOW COMPLETE THE DERIVATIES
C
      CALL DBDX(XHP,DD)
      DO 10 I=1,6
      DH(1,I)=D111*DD(1,1,I)+D121*DD(2,1,I)+D131*DD(3,1,I)+D112*DD(1,2,I)
     #  + D122*DD(2,2,I) + D132*DD(3,2,I)
10    DH(2,I)=-BHP(3,2)*DD(2,1,I)/RDS+BHP(2,2)*DD(3,1,1)/RDS+HP3*DD(2,2,I)
     #  -HP2*DD(3,2,I)
C
      RETURN
      END
```

APPENDIX E

EXPERIMENT DESCRIPTION

1. INTRODUCTION

An experiment was designed to test the validity of the measuring model, described conceptually in Chapter I. It was also designed to evaluate the technical feasibility of the position measuring method, described in Chapter II and analytically evaluated in Appendix C. The experiment was performed as follows:

- the assembly of a round peg into a round hole, Figure II-6, was attempted using a general type seven degrees of freedom positioning device. The positioning device used was implemented with a force loop control that prevented the parts from destroying each other and kept the parts in contact once they touched;
- the time history of positions of the positioning device, used also as a position measuring device, were recorded and stored;
- the data generated by the experiment was processed off-line with the purpose of determining the actual position of the parts being assembled. A series of statistical tests were also performed in this processing for determining the validity of the results.

This appendix will describe the experimental apparatus and the mechanization of the data collection; Appendix F together with Appendix D and Chapter II detail the results of the experiment.

It is to be noted that the scheme used throughout the description of this positioning measurement method was designed to be implemented on line. This could not be done in this work because of the lack of time and unavailability of software to support the implementation.

Section Two of this appendix describes the apparatus used in the experiment and Section Three describes the data gathering and storing process.

2. DESCRIPTION OF THE APPARATUS

The elements used in the implementation are shown schematically in Figure E-1.

- a Unimate 5000 seven axis machine; Figure E-1 shows a schematic representation of this machine and gives the technical specifications of such a machine;
- a six axis pedestal force sensor shown schematically in Figure E-1 together with its sensitivity and specifications;
- a Nova 2 32K minicomputer and interphase electronics used for controlling and driving the arm, as well as gathering the information. This computer was also used to reduce the data from the experiment;
- an oversized peg and matching hole with the following dimensions:

  Radius hole R = 95.25 mm

  Radius Peg r = 94.10 mm

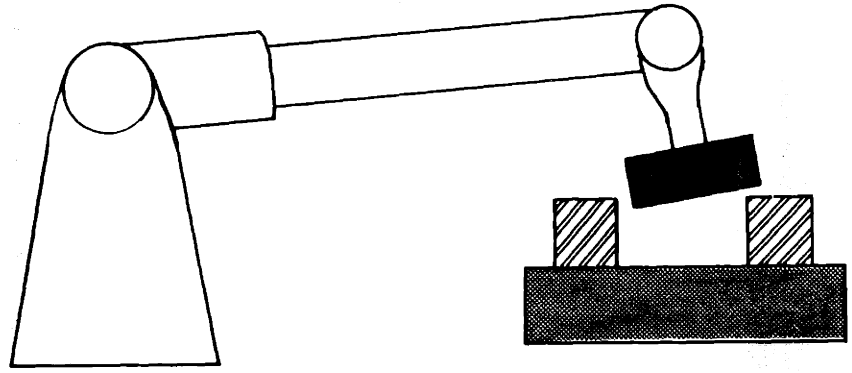
  Clearance ratio $C = \frac{R - r}{R} = .012$

The choice of oversized peg and hole was done to simulate a high resolution positioning device. The Unimate's position resolution would not have been acceptable, according to the conclusions of Appendix C, for regular sized parts ($R \simeq 1/2" = 5$ mm); this concept is explained in more detail in Chapter II.

Figure E-2 shows schematically the implementation of the control scheme. The system takes commands in the form of velocities expressed in terms of the coordinate frame located at the gripper.

- the gripper velocities are transformed to joint variables velocities through the following transformation:

$$\underline{v}^P = J\,\dot{\underline{\theta}} \qquad \text{E-1}$$



Encoder	# Counts	Range	Quantization Unit
1	8573	200°	$3.96 \cdot 10^{-3}$ rad
2	2981	12 in	$4.07 \cdot 10^{-4}$ in
3	16138	60°	$6.48 \cdot 10^{-5}$ rad
4	5962	24 in	$3.96 \cdot 10^{-3}$ in
5	6429	360°	$9.77 \cdot 10^{-4}$ rad
6	11430	260°	$3.97 \cdot 10^{-4}$ rad
7	32335	200°	$1.08 \cdot 10^{-4}$ rad

Position Resolution:	Position resolution	Angular resolution
	$20.7 \cdot 10^{-3}$ in	$0.98 \cdot 10^{-3}$ rad
	0.525 mm	$56.0 \cdot 10^{-3}$ °

Force Sensor

Performance Specifications

Condition	Force Per Channel (Newtons)	Bridge Output mV.	Amplifier Output V.
Full Scale	+222.4 (50 lbs)	30.0	+10.0
-Full Scale	-222.4 (50 lbs)	-30.0	-10.0
1 Bit	0.11 (1/40 lb)	0.0149	.005

Figure E-1: Schematic Configuration of Experimental Apparatus

where:

$\underline{v}^P$ is the velocity of the P frame expressed in terms of the P frame.

$\underline{\theta} = \{\theta_i : i = 1,7\}$ is the vector formed by the readings at each axis of the positioning device's actuator; the vector $\dot{\underline{\theta}}$ stands for the vector whose components are the rates of these readings.

J, the Jacobian matrix of the transformation in Expression E-1 is formally defined through the notation defined in Appendix A, by:

$$J = \begin{bmatrix} -\frac{\partial \underline{r}^{OP}}{\partial \underline{\theta}} \\ \\ \frac{\partial \omega^{OP}}{\partial \theta} \end{bmatrix} \qquad \text{E-2}$$

- the commands in the joint coordinates of the positioning device result in its movement through space and hence performing the assembly;
- when parts touch, forces between them get generated and are detected by the force sensor;
- force readings are transformed into velocity command through the following transformation:

$$\Delta \underline{v}^H = D \, \underline{L}^H \qquad \text{E-3}$$

where D is the force feedback matrix.

3. POSITION METHOD DATA COLLECTION

The relative position measurement between the frames P and R in Figure E-1 has to be obtained from the joint variables readings $\underline{\theta}(n)$ available from the positioning device during the experiment, as shown in Figure II-7. It is also desirable to obtain a measure on the accuracy of these measurements; this is done through the use of the following notation:

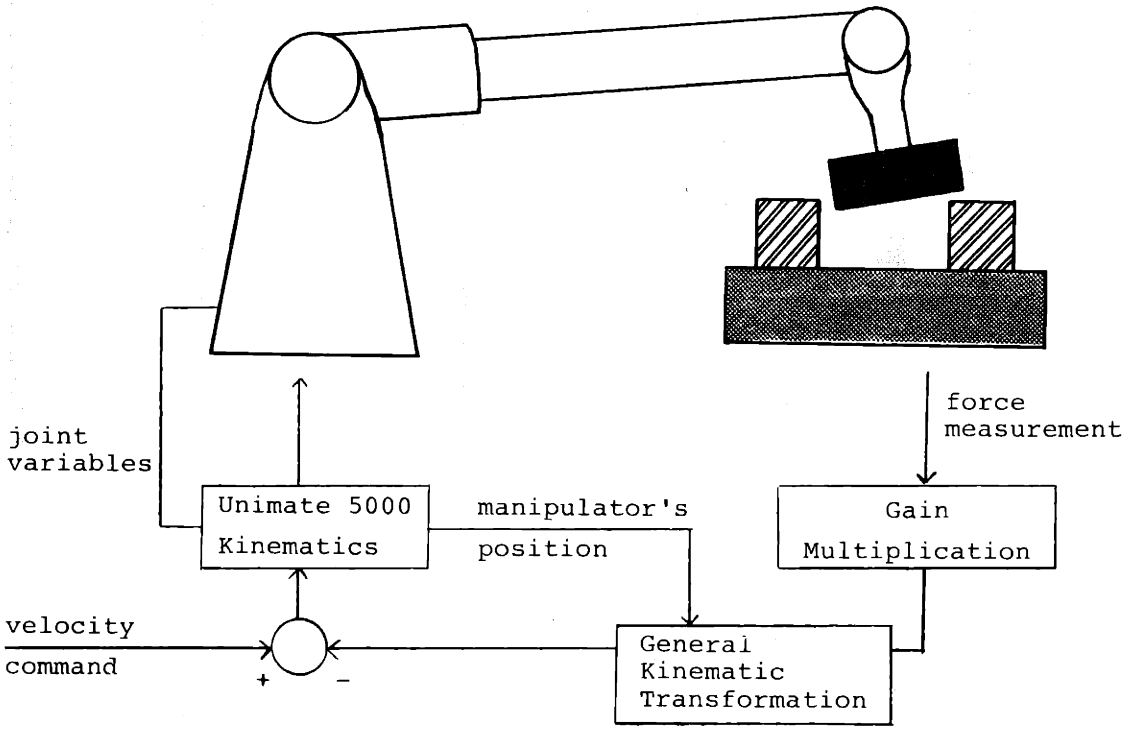


Figure E-2: Schematic of Force Loop Implementation

$\underline{x}^{RP}(n)$ will represent the generalized actual relative position between frames R and P at time n;

$\underline{z}^{RP}(n)$ will represent the actual measurement of the relative position between frames R and P obtained from the joint coordinates at time n;

$\underline{v}(n)$ will represent the error on the measurement at time n;

Thus:

$$\underline{z}^{RP}(n) = \underline{x}^{RP}(n) + \underline{v}(n) \qquad \text{E-4}$$

The measure used for the error on the measurement is the matrix covariance of the vector $\underline{v}$ defined formally as:

$$R(n) = E\left[\underline{v}\,(n)\;\underline{v}^{T}\,(n)\right] \qquad \text{E-5}$$

The computation of the vector $\underline{z}^{PR}(n)$ is done through the following sequence of steps:

- computation of the relative position between the 0 reference frame of the positioning device and its gripper frame from the values of the joint coordinates; this is formally written as:

$$B^{0P} = B^{0P}(\underline{\theta}) \qquad \text{E-6}$$

  This is done through the subroutined KAWB07 listed in Section Four. This subroutine was automatically generated by a program called TOAD, developed in Reference E-1;
- computation of the relative position between gripper frame (made to coincide with the pegs frame) and the jig frame R; this is done through the following operations:

$$B^{RP} = B^{RO} \; B^{OP}$$

$$\underline{x}^{RP} = \underline{x}^{RP}(B^{RP}) \qquad \text{E-7}$$

These operations are defined in Appendix A, Expression A-17 and Expression A-33 respectively. The relative position B^{HO} is known from a priori.

For the computation of the matrix R(n), Expression E-1 is written for incremental vectors as:

$$\begin{bmatrix} -d\underline{r}^{OP} \\ d\underline{\omega}^{OP} \end{bmatrix} = J \; d\underline{\theta} \qquad \text{E-8}$$

Through the use of Expression A-57 in Appendix A the instantaneous incremental vector $d\underline{\omega}^{OP}$ is transformed into the incremental rotation vector $d\underline{\phi}^{OP}$ by:

$$d\underline{\phi}^{OP} = \frac{D\underline{\phi}^{OP}}{D\underline{\omega}^{OP}} \; d\underline{\omega}^{OP} \qquad \text{E-9}$$

Thus the incremental vector $d\underline{x}^{OP}$ is written in terms of the incremental joint variables by:

$$d\underline{x}^{OP} = S \; J \; d\underline{\theta} \qquad \text{E-10}$$

where:

$$S = \begin{bmatrix} -I & 0 \\ 0 & \dfrac{D\underline{\phi}^{OP}}{D\underline{\omega}^{OP}} \end{bmatrix} \qquad \text{E-11}$$

Equating the vector $d\underline{x}^{RP}$ with the noise vector $d\underline{v}$ and forming the covariance matrix R defined in Expression E-5, gives:

$$R(n) = \frac{D\underline{x}^{RP}}{D\underline{x}^{OP}} S J \; E\left[d\underline{\theta}\, d\underline{\theta}^T\right] S^T J^T \left(\frac{D\underline{x}^{RP}}{D\underline{x}^{OP}}\right)^T \qquad \text{E-12}$$

The covariance matrix $E\left[d\underline{\theta}(n) d\underline{\theta}^T(n)\right]$ is established from the manufacturer's specifications given in Figure E-1; the matrices $\frac{D\underline{x}^{RP}}{D\underline{x}^{OP}}$ and S are computed from the vector $\underline{x}^{RP}$, $\underline{x}^{OP}$ and $\underline{x}^{RO}$; the matrix J is computed by the subroutine DRWDE produced automatically by TOAD (Reference E-1) and listed in Section Four.

The program LISTING, listed in Section Four, implements the calculations of the vector $\underline{x}^{RP}(n)$ and the covariance matrix R(n).

4. POSITION MEASUREMENT IMPLEMENTATION

The procedure schematically shown in Figure II-7 is implemented in one main program named LISTING and two subroutines named KAWB07 and DRWDE. The two subroutines were produced automatically by TOAD (Reference E-1).

```
C******************* LISTING.SS  ****************************
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       ---READS THE JOINT ANGLES AND FORCES
C       ---DETECT CONTACT-POINT EVENT
C       ---ELIMINATES THOSE POINTS NOT NEEDED
C       ---COMPUTES B0R AND ITS INVERSE BR0, TAKEN AS REFERENCE, FROM
C          THE 20TH DATA POINT
C       ---COMPUTES B07 FOR THE JOINT ANGLES READ
C       ---COMPUTES BR7, XR7, THEN COMPUTES X7R
C          AND STORES X7R IN A FILE
C       --- COMPUTES DX07/DE FROM ENCODER READINGS
C       ---TRANSFORMS THE TENSOR DX07/DE INTO DX7R/DE
C       ---MULTIPLIES DX7R/DE BY ITS TRANSPOSE TO FORM
C          THE ERROR COVARINCE MTX, OF THE VECTOR POSITION X7R
C          AND STORES IT IN THE SAME FILE AFTER X7R
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
        COMPILER DOUBLE PRECISION
        DIMENSION IE(7),XFILE(79),ITH(220,8),F(220), YFILE(79),B07(3,4)
```

```
      DIMENSION B07D(3,4),BR0D(3,4),B0RD(3,4),BR7D(3,4)
      DIMENSION DJ(6,7),D(6,7),D1(6,6),D3(3,3)
      DIMENSION XZERO(6),XR0(6),X07(6),XR7(6),X7R(6),X70(6)
C
C
C
C  READ:
      DATA ISIX,ISEV/6,7/
      DATA (XZERO(I),I=1,6)/6*0.0/
      TYPE "INPUT NAME OF FILE WHERE DATA IS: "
      READ(11,101) XFILE(1)
101   FORMAT (S79)
      TYPE "INPUT THE NAME OF FILE WHERE DATA GOES: "
      READ(11,101)YFILE(1)
      OPEN 1,XFILE
      ACCEPT "# OF DATA POINTS YOU WANT FROM FILE: ",MK
      DO 9 J=1,MK
9     READ (1,10) ((ITH(J,I),I=1,8),F(J))
      CLOSE 1
C
C  DETECT AND ELIMINATE:
      KK=0
      A=-50.
      DO 1 J=1,220
      IF (F(J).GT.A)GO TO 1
      A=-5
      KK=KK+1
      F(KK)=F(J)
      DO 2 I=1,7
2     ITH(KK,I)=ITH(J,I+1)
1     CONTINUE
      IF(KK.GT.150) KK=150
      TYPE "USEFUL # OF POINTS FOR PROCESING = ",KK
C
C  COMPUTES B0R, BR0 AT THE 20TH DATA POINT
C
      OPEN 7,YFILE
      WRITE(7,113)
113   FORMAT("95.6055                    .0154                ",/)
      DO 5 J=1,7
5     IE(J)=ITH(20,J)
      CALL DB07(IE,B0RD)
      CALL INVB(B0RD,BR0D)
      CALL BTOX(BR0D,XR0)
C
C  COMPUTES B07, TRANSFORMS IT TO X7R AND STORES IT IN FILE
C  OTHER VECTORS ARE COMPUTED FOR THE PURPOSE OF TRANSFORMATIONS
C
      DO 3 II=1,KK
      DO 4 J=1,7
4     IE(J)=ITH(II,J)
      CALL DB07(IE,B07D)
      CALL BTOX(B07D,X07)
      CALL BBB(BR0D,B07D,BR7D)
      CALL BTOX(BR7D,XR7)
      CALL INVX(XR7,X7R)
      WRITE(7,6)(X7R(I),I=1,6)
6     FORMAT(2( 3(E24.14),/))
```

```
C
C  CALLS THE JACOBIAN CALCULATION (MTX DJ)
C  THEN TRANSFORMS THE LOWER 3*7 PARTITION OF DJ FROM
C                DW07/DE  TO  DFI07/DE
C
        CALL DRWDE(IE,DJ)
        CALL INVX(X07,X70)
        CALL DFIDW(X70,D3)
        DO 51 J=1,7
        DO 52 I=4,6
        D(I,J)=0.
        DO 52 L=1,3
52      D(I,J)=D(I,J)-D3(I-3,L)*DJ(L+3,J)
        DO 51 I=4,6
51      DJ(I,J)=D(I,J)
C
C  FIRST: DJ IS TRANSFORMED FROM DX07/DE TO DXR7/DE
C  SECND: DJ IS TRANSFORMED FROM DXR7/DE TO DX7R/DE
C
        DO 53 KJ=1,2
        IF( KJ .EQ. 2) GO TO 62
        CALL DXDX(XR7,XZERO,XR0,D1)
        GO TO 63
62      CALL INVDX(XR7,D1)
63      DO 53 J=1,7
        DO 54 I=1,6
        D(I,J)=0.
        DO 54 L=1,6
54      D(I,J)=D(I,J)+D1(I,L)*DJ(L,J)
        DO 53 I=1,6
53      DJ(I,J)=D(I,J)
C
C  MULTIPLIES THE MTX DJ BY ITS TRANSPOSE AND FORMS
C  THE MATRIX COVARIANCE OF ERROS IN THE VECTOR X7R
C  DUE TO DISCRETISATION OF ENCODER READINGS
C
        DO 55 I=1,6
        DO 55 J=1,6
        D(I,J)=0.
        DO 55 L=1,7
55      D(I,J)=D(I,J)+DJ(I,L)*DJ(J,L)
        WRITE (7,20)((D(III,JJ),JJ=1,6),III=1,6)
20      FORMAT(6(6(E13.5),/))
3       CONTINUE
        CLOSE 7
10      FORMAT (/,I3,7I7,F7.1)
        END



C***************** KAWB07.SS *****************
C
C     HAND-TO-SHOULDER TRANSFORMATION FOR THE
C        UNIMATE/KAWASAKI MODEL 5000
C          HYDRAULIC ARM
C
C     REVISION 01     SEPTEMBER 27, 1974
C
```

```
C      AUTHORS: TOAD USING BOB STURGES' PARAMETERS
C               CONVERSION TO FORTRAN BY MLI ON NOVA
C               USING D.KILLORAN'S MACROS
C
C
C
C
C    B07(I,J) IS THE "3X4" TRANSFORMATION FROM "HAND COORDINATES"
C     TO "SHOULDER COORDINATES".
C        "B07" OR "B07" GIVES THE RELATIVE POSITION W.R.T. "HAND FRAME"
C        OF "REFERENCE" OR "SHOULDER FRAME"
C
C
C
C    Q(I) IS A 7-DIMENSIONAL VECTOR GIVING THE CURRENT
C        JOINT ANGLES.
C
C
C
         SUBROUTINE SB07(Q,B07)
         DIMENSION B07(3,4),Q(7)
         DATA A3/120.65/
         DATA S7/203.2/
         DATA A7/0/
         DATA ALF7/0/
         DATA S6/15.88/
         S001=SIN(Q(3))
         S002=SIN(Q(5))
         S003=SIN(Q(2))
         S004=SIN(Q(6))
         S005=SIN(Q(7))
         S006=SIN(ALF7)
         C001=COS(Q(2))
         C002=COS(Q(3))
         C003=COS(Q(5))
         C004=COS(Q(6))
         C005=COS(Q(7))
         C006=COS(ALF7)
         P001=Q(4)*S001
         P002=S001*C001
         P003=S001*S002
         P004=S001*S003
         P005=S001*C003
         P006=C001*S6
         P007=C001*S002
         P008=C001*C002
         P009=C001*C003
         P010=S6*S003
         P011=S002*C002
         P012=S002*S003
         P013=C002*S003
         P014=C002*C004
         P015=C002*S004
         P016=C002*C005
         P017=C002*S005
         P018=S003*C003
```

```
      P019=S003*C005
      P020=S003*S005
      P021=C003*S7
      P022=C003*C004
      P023=C003*S004
      P024=S7*C004
      P025=S7*S004
      P026=C004*A7
      P027=C004*C005
      P028=C004*S005
      P029=C004*C006
      P030=C004*S006
      P031=S004*A7
      P032=S004*C005
      P033=S004*S005
      P034=S004*C006
      P035=S004*S006
      P036=A7*C005
      P037=A7*S005
      P038=C005*C006
      P039=C005*S006
      P040=S005*C006
      P041=S005*S006
      P042=S004*P021
      P043=C005*P022
      P044=C005*P026
      P045=C005*P031
      P046=C006*P007
      P047=C006*P023
      P048=C006*P028
      P049=C006*P033
      P050=S006*P007
      P051=S006*P023
      P052=S006*P028
      P053=S006*P033
      P054=P007*P017
      P055=P008*P022
      P056=P011*P019
      P057=P011*P020
      P058=P013*P022
      B07(1,1)= P055*P056 + P012*(P025-P044)-A7*P054-P018*P037-P001*C001
     # + P006*P011 + C003*P010 + P008*(-P042 + A3) + P002*(-P045-P024)
      B07(1,2)= -P054-P012*P027-P018*S005 + P008*P043-P002*P032
      B07(1,3)= -P055*P040 + P012*(P035 + P048)-P018*P038-P008*P051 + P002
     # *(P049-P030)-P016*P046
      B07(1,4)= P055*P041 + P012*(P034-P052) + P018*P039-P008*P047 + P002*(
     # -P053-P029) + P050*P016
      B07(2,1)= -A7*P057 + P037*P009 + P036*P058-P001*S003-P006*C003
     # +P011*P010 + P013*(-P042 + A3) + P004*(-P045-P024) + P007*(-P025
     # +P044)
      B07(2,2)= -P057 + P013*P043-P004*P032 + P007*P027 + P009*S005
      B07(2,3)= -P013*P051 + P004*(P049-P030)-P007*P035-P058*P040 + P009
     # *P038-C006*P056-P028*P046
      B07(2,4)= -P013*P047 + P004*(-P053-P029)-P007*P034 + P058*P041-P009
     # *P039 + P056*S006 + P028*P050
      B07(3,1)= Q(1) + P036*P015 + A3*S001 + Q(4)*C002 + P003*(-P037 +
     # S6) + P005*(-P025 + P044) + S7*P014
      B07(3,2)= P015*C005-P003*S005 + P005*P027
```

```
      B07(3,3)= -P015*P040-P003*P038 + P005*(-P035-P048) + P014*S006
      B07(3,4)= P015*P041 + P003*P039 + P005*(-P034 + P052) + P014*C006
      RETURN
      END

C****************** DRWDE,SS ******************************
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C       JACOBIAN MATRIX FOR THE
C       UNIMATE/KAWASAKI MODEL 5000
C         HYDRAULIC ARM
C
C     REVISION 01     SEPTEMBER 27, 1974
C
C      AUTHORS: TOAD USING BOB STURGES' PARAMETERS
C               CONVERSION TO FORTRAN BY MLI ON NOVA
C               USING D.KILLORAN'S MACROS
C
C
C    DJ(M,N)  IS A 6X7 MATRIX GIVING THE JACOBIAN
C
C       DJ * VJ = VX
C
C   WHERE J IS THE JACOBIAN, VJ IS THE ANGULAR VELOCITY
C     OF THE JOINTS, AND VX IS THE 6-DIMENSIONAL
C     VELOCITY (TRANSLATIONAL AND ANGULAR) IN THE
C     COORDINATES ATTACHED TO THE HAND.
C
C
C   THIS VERSION IS A MODIFICATION ON THE ORIGINAL JAC,SS, THE
C    MODIFICATIONS CONSISTS OF:
C                                - CHANGING THE SIGN OF ALL TERMS
C    IN THE UPPER 3*7 PARTINTION OF THE ORIGINAL JACOBIAN MATRIX
C                                - CHANGING THE FRAME DEFINED BY
C    BOB STURGES INTO FRAMES MORE SUITABLE WITH THE ONES I USE
C    FOR DEFINING THE GEOMETRY. THE STATEMENTS THAT HAVE BEEN MODIFIED ARE
C    HEADED BY "CC". REFERENCE C-1 GIVES THE ANALYSIS THAT IS PERTINENT.
C
C
C   THE FINAL OUTPUTED MTX IS PARTITIONED AS FOLLOWS:
C        - TOP    3*7 PARTITION =    DR07/DE
C        - BOTTOM 3*7 PARTITION =    DW07/DE
C           (E=VECTOR OF ENCODER READINGS )
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE DRWDE(IE,DJ)
      COMPILER DOUBLE PRECISION
      DIMENSION IE(7),DJ(6,7),Q(7)
      DATA A3/120.65/
```

```
      DATA S6/15.875/
      DATA A7,S7,ALF7/0.,203.2,0./
      DATA R1,R2,R3,R5/.1,-1.15192E-4,-6.45772E-5,4.06662E-4/
      DATA R6,R7,R65,R75/-3.9619E-4,9.75639E-4,9.02789E-5,-1.10612E-4/
      DATA R76,Q01,Q02,Q03/6.43808E-4,-.9,5.02935,2.07340/
      DATA Q04,Q05,Q06,Q07/1280.23,-1.75963,1.83635,-6.39984/
      Q(1)=R1*IE(1)+Q01
      Q(2)=R2*IE(2)+Q02
      Q(3)=R3*IE(3)+Q03
      Q(4)=-R1*IE(4)+Q04
      Q(5)=R5*IE(5)+Q05
      Q(6)=R65*IE(5)+R6*IE(6)+Q06
      Q(7)=R75*IE(5)+R76*IE(6)+R7*IE(7)+Q07
     S001=SIN(Q(3))
     S002=SIN(Q(5))
     S003=SIN(Q(7))
     S004=SIN(Q(6))
     S005=SIN(ALF7)
     C001=COS(Q(5))
     C002=COS(Q(6))
     C003=COS(Q(7))
     C004=COS(Q(3))
     C005=COS(ALF7)
     P001=S001*S002
     P002=S001*S003
     P003=S001*C001
     P004=S002*S003
     P005=S002*C002
     P006=S002*C003
     P007=S002*S004
     P008=S003*C001
     P009=S003*C002
     P010=S003*S004
     P011=S003*C005
     P012=S003*S005
     P013=C001*C002
     P014=C001*C003
     P015=C001*S004
     P016=C002*C003
     P017=C002*C004
     P018=C002*C005
     P019=C002*S005
     P020=C003*S004
     P021=C003*C005
     P022=C003*S005
     P023=C003*S7
     P024=S004*C004
     P025=S004*C005
     P026=S004*S005
     P027=C004*C005
     P028=C004*S005
     P029=C004*A3
     P030=C005*Q(4)
     P031=C005*S6
     P032=C005*A3
     P033=C005*S7
     P034=C005*A7
     P035=S005*Q(4)
```

```
      P036=S005*S6
      P037=S005*A3
      P038=S005*S7
      P039=S005*A7
      P040=S005*P001
      P041=C003*P005
      P042=C004*P020
      P043=C005*P002
      P044=C005*P010
      P045=C005*P017
      P046=S005*P002
      P047=S005*P010
      P048=S005*P017
      P049=Q(4)*P001
      P050=Q(4)*P003
      P051=Q(4)*P008
      P052=S6*P003
      P053=S6*P004
      P054=S6*P009
      P055=S6*P016
      P056=A3*P007
      P057=A3*P014
      P058=S7*P001
      P059=S7*P005
      P060=S7*P010
      P061=S7*P020
      P062=A7*P001
      P063=A7*P003
      P064=A7*P004
      P065=A7*P010
      P066=P001*P009
      P067=P001*P021
      P068=P001*P022
      P069=P002*P013
      P070=P002*P015
      P071=P003*P016
      P072=P003*P025
      P073=P003*P026
      P074=P004*P017
      P075=P010*P027
      P076=P010*P028
      P077=P013*P043
      P078=P013*P046
CC      DJ(1,1)=
       A= -P040 +P071 +P042
       DJ(3,1)=-A
      DJ(2,1)= -P067-P073-P077-P075 +P048
CC      DJ(3,1)=
       DJ(1,1)= P068-P072 +P078 +P076 +P045
CC      DJ(4,1)=
       DJ(6,1)=- 0
      DJ(5,1)= 0
CC      DJ(6,1)=
       DJ(4,1)= 0
CC      DJ(1,2)=
       A= -P001*P023-S7*P069-P049*P016-C001*Q(4)*P002-P020*P052
     # +C004*(-P060 +P055) +P029*(P041 +P008)
       DJ(3,2)=-A
```

```
      DJ(2,2)= -P053*P071 +P022*P062-P025*P063 +P039*P069 +P017*P034
     # -P036*P024 +P027*(-P061-P054 +P057) +P028*(P065-P056)-P032*P074
     # +P011*P058 +P026*P049 +P030*P066-P021*P050 +P031*P070-P019*P052
CC       DJ(3,2)=
      DJ(1,2)= -P035*P066-P018*P052 +P037*P074-P012*P058 +P038*P071
     # +P022*P050 +P062*P021 +P025*P049 +P063*P026-P039*P017 +P034*P069
     # -P036*P070 +P027*(P065-P056) +P028*(P061 +P054-P057)-P031*P024
CC       DJ(4,2)=
      A= -P040 +P071 +P042
      DJ(6,2)=-A
     DJ(5,2)= -P067-P073-P077-P075 +P048
CC       DJ(6,2)=
      DJ(4,2)= P068-P072 +P078 +P076 +P045
CC       DJ(1,3)=
      A= P006*S004*S6 +A3*P020-P014*S7 +P004*(Q(4)
     # +C002*S7 )-C003*P013*Q(4)
      DJ(3,3)=-A
     DJ(2,3)= P059*P021 +P007*P034 +P006*P030 +P014*P039 +P015*P035
     # +P051*P018-P053*P025 +P005*P036-P010*P032 +P033*P008 +P019*(-P064
     # +A3)
CC       DJ(3,3)=
      DJ(1,3)= -P059*P022-P007*P039-P006*P035 +P014*P034 +P015*P030-P051
     # *P019 +P053*P026 +P005*P031 +P010*P037 +P018*(-P064 +A3)-P038
     # *P008
CC       DJ(4,3)=
      A= -P041-P008
      DJ(6,3)=-A
     DJ(5,3)= P007*S005-C005*P014 +P018*P004
CC       DJ(6,3)=
      DJ(4,3)= P007*C005 +S005*P014-P004*P019
CC       DJ(1,4)=
      A= P020
      DJ(3,4)=-A
     DJ(2,4)= -P044 +P019
CC       DJ(3,4)=
      DJ(1,4)= P047 +P018
CC       DJ(4,4)=
      A= 0
      DJ(6,4)=-A
     DJ(5,4)= 0
CC       DJ(6,4)=
      DJ(4,4)=0
CC       DJ(1,5)=
      A= -P060 +P055
      DJ(3,5)=-A
     DJ(2,5)= -P009*P031 +A7*P018 +P010*P039-S6*P026-P033*P020
CC       DJ(3,5)=
      DJ(1,5)= P009*P036-A7*P019 +P010*P034 +P038*P020-S6*P025
CC       DJ(4,5)=
      A= P020
      DJ(6,5)=-A
     DJ(5,5)= -P044 +P019
CC       DJ(6,5)=
      DJ(4,5)= P047 +P018
CC       DJ(1,6)=
      A= -P023
      DJ(3,6)=-A
     DJ(2,6)= A7*P022 +P011*S7
```

```
CC        DJ(3,6)=
          DJ(1,6)= A7*P021-P012*S7
CC        DJ(4,6)=
          A= -S003
          DJ(6,6)=-A
         DJ(5,6)= -P021
CC        DJ(6,6)=
          DJ(4,6)= P022
CC        DJ(1,7)=
          A= 0
          DJ(3,7)=-A
         DJ(2,7)= P034
CC        DJ(3,7)=
          DJ(1,7)= -P039
CC        DJ(4,7)=
          A= 0
          DJ(6,7)=-A
         DJ(5,7)= S005
CC        DJ(6,7)=
          DJ(4,7)= C005
C
C    HERE THE JACOBIAN WRT. KINNEMATICS ANGLES IS CHANGED TO
C    A JACOBIAN WRT. ENCODERS READINGS
          DO 20 I=1,6
          DJ(I,1)=DJ(I,1)*R1
          DJ(I,2)=DJ(I,2)*R2
          DJ(I,3)=DJ(I,3)*R3
          DJ(I,4)=-DJ(I,4)*R1
          DJ(I,5)=DJ(I,5)*R5+DJ(I,6)*R65+DJ(I,7)*R75
          DJ(I,6)=DJ(I,6)*R6+DJ(I,7)*R76
20        DJ(I,7)=DJ(I,7)*R7
C
          RETURN
          END
```

APPENDIX F

POSITION ESTIMATION IMPLEMENTATION

1. INTRODUCTION

The position measurement method, described conceptually in Chapter II, can be thought of in terms of the following sequence of steps:

- the nominal assembly path is fed into the positioning device;
- implementation of a force control scheme, of the form described in Chapter II , will be assumed. It should allow for the parts to touch without extreme forces and ensure that they maintain contact;
- position measurements between the parts are gathered;
- through the use of the position information from the measurements and the knowledge of the geometric characteristics of the parts, the actual relative position between the parts is estimated;
- the difference between the desired nominal position and the estimate of the actual position is computed and fed back to the positioning device. If the estimate is close to the actual relative position of the parts, the parts should go together.

This appendix will cover the implementation of the estimation schemes used. The data used for evaluating the estimator and the modelling of the measurements are covered conceptually in Chapter I, and in detail in Appendix E. The geometric modelling used for this implementation is covered conceptually in Chapter II and in detail in Appendix D.

Two different approaches are used for implementing the position estimator:

- a smooth filter approach by which the whole time series of measurements is used in a one step computation to obtain the time history of position estimation;
- a recursive filter approach that computes the estimate of the actual relative position by means of successive estimation at every new measurement.

The smooth estimate has little practical applicability and was used in this work mainly as a proof of concept. The method used in the recursive estimator is based on the Extended Kalman Filter exposition made in Reference F-1.

Sections Two and Three of this appendix describe the implementation of the smooth and recursive filters respectively; Section Four shows the program listings for the smooth filter implementation and Section Five shows the code for the recursive filter implementation.

2. SMOOTH POSITION ESTIMATOR

The model for the position measurements was described conceptually in Chapter I and detailed in Appendix E. It is formally written as:

$$\underline{z}^{RP}(n) = \underline{x}^{RP}(\underline{x}^{HP}(n), \underline{x}^{RH}) + \underline{v}(n) \qquad \text{F-1}$$

where:

$\underline{z}^{RP}(n)$ represents the position measurement vector at time, n, its dimension is k_z

$\underline{x}^{HP}(n)$ represents the k_x dimension vector of the relative position between the parts at time n

$\underline{x}^{RH}$ represents the bias term in the measuring model

$\underline{v}(n)$ represents the resolution noise of the measurement at time n

$\underline{x}^{PR}(\cdot)$ represents the kinematic transformations developed in Appendix A

The measurement noise $\underline{v}(n)$ is assumed to be a zero mean white random vector with covariance matrix:

$$E\left[\underline{v}(n)\ \underline{v}^T(m)\right] = R(n)\ \delta_{nm} \qquad \text{F-2}$$

For the purpose of this filter implementation the noise will also be assumed Gaussian. Thus, for a given set of position and bias vector $(\underline{x}^{HP}(n), \underline{x}^{HR})$, the measurement $\underline{z}^{RP}(n)$ is a white Gaussian process of mean $\underline{x}^{RP}(n)$ and covariance matrix $R(n)$; this is formally written as:

$$\underline{z}^{RP}(n) = N(\underline{x}^{RP}(\underline{x}^{HP}(n), \underline{x}^{RH}), R(n)) \qquad \text{F-3}$$

The estimator chosen for this particular implementation is the maximum likelihood estimate of the set $(\underline{x}^{HP}(n), \underline{x}^{RH})$. This estimate is defined as the set of values that maximizes the joint probability density of the random variables $\underline{z}^{RP}(n)$, evaluated at the actual values of the measurements $\underline{z}_t^{RP}(n)$. The maximization is made with respect to the vectors $\underline{x}^{HP}(n)$ and $\underline{x}^{RH}$, and subject to the condition that the vectors $\underline{x}^{HP}(n)$ satisfy the geometric constant at every time n. This is formally written as:

$$(\hat{\underline{x}}^{HP}(n), \hat{\underline{x}}^{RH}): \text{ the value that maximizes } (p(\underline{z}_t^{RP}(n):\underline{x}^{HP}(n), \underline{x}^{RH}))$$
$$\text{such that: } \underline{h}(\underline{x}^{HP}(n)) = 0 \qquad \text{F-4}$$

Where $p(\cdot)$ denotes the joint probability density of the random process $\underline{z}^{RH}(n)$ for given values of $(\underline{x}^{HP}(n), \underline{x}^{RP})$.

$h(\cdot)$ represents the expression for the geometric constrain on the relative position between the parts.

The extreme operation on the joint probability density can equivalently be done on a monotonic function of this probability. A standard procedure in this respect is to do the extreme operation on the logarithm of the probability density function; this function is called the log likelihood function and is formally

written as:

$$\xi(\underline{z}^{RP}(n):\ \underline{x}^{HP}(n),\ \underline{x}^{RH}) = -\ell n\ (p(\underline{z}^{RP}(n):\underline{x}^{HP}(n),\ \underline{x}^{PH}) \qquad \text{F-5}$$

Since the random process $\underline{z}^{RP}(n)$ is white and Gaussian the log likelihood function takes the form:

$$\xi(\underline{z}^{RP}(n):\underline{x}^{HP}(n),\ \underline{x}^{RH}) = -\ln C + 1/2\ L \qquad \text{F-6}$$

where:

$$C = (2\pi)^{-\frac{1}{2}}\ (\pi_n \det\ (R(n)))^{-\frac{1}{2NK_z}}$$

$$k_z = \text{dimension of the vector } \underline{z}^{RP}(n) \qquad \text{F-7}$$

$$L(\underline{z}^{RP}(n),\ \underline{x}^{HP}(n),\ \underline{x}^{RH}) = \Sigma_n\ \underline{u}^T(n)\ R^{-1}(n)\ \underline{u}(n)$$

and

$$\underline{u}(n) = \underline{z}^{RP}(n) - \underline{x}^{RP}(\underline{x}^{HP}(n),\ \underline{x}^{RH}) \qquad \text{F-8}$$

Thus, the maximum likelihood estimate of $\underline{x}^{HP}(n)$, $\underline{x}^{RH}$ can be obtained by finding the vectors $((\underline{x}^{HP}(n),\ n = 1,\ N), \underline{x}^{RH})$ that minimizes the expression of L in F-7, subject to $(\underline{x}^{HP}(n),\ n = 1,\ n)$ satisfying the geometric constraint; formally:

$$\hat{\underline{x}}^{HP}(n),\ \hat{\underline{x}}^{RH} = \min_{\underline{x}^{HP}(n),\ \underline{x}^{RH}} L(\underline{z}^{RP}_a(n),\ \underline{x}^{HP}(n),\ \underline{x}^{RH}) \qquad \text{F-9}$$

subject to:

$$\underline{h}(\hat{\underline{x}}^{HP}(n)) = \underline{0}$$

This simple formulation for the estimation presents the difficult problem of having to perform the extreme operation over all components of the vector $\underline{x}^{RH}$ and the vectors $\underline{x}^{HP}(n)$ for all times n; this amounts to optimizing over $k_z(N + 1)$ scalar components and subject to $k_h N$ scalar constraints. This problem was solved by making the following approximation to the optimization problem stated in Expression F-9.

Let's minimize the following unconstrained expression.

$$L' = \sum_n \left[\underline{u}^T(n)\, R^{-1}(n)\, \underline{u}(n) + \lambda \underline{h}^T(x(n))\underline{h}(x(n)) \right] \qquad \text{F-10}$$

The choice of a comparatively large value for the scalar parameter λ will force the minimization to make $\underline{h}\,(x(n))$ very close to zero and thus approximate to the minimization of L subject to the constraint $(\underline{h}(x(n))$ in Expression F-9.

Making the gradient of L' with respect to $\underline{x}^{HP}(n)$ equal to zero gives:

$$\frac{\partial L'}{\partial \underline{x}^{HP}(n)} = -2\, \underline{u}^T(n) R^{-1}(n) \frac{\partial \underline{x}^{RP}(n)}{\partial \underline{x}^{HP}(n)} + 2\lambda \underline{h}^T(n) \frac{\partial \underline{h}}{\partial \underline{x}^{HP}(n)} = 0 \qquad \text{F-11}$$

Since both R^{-1} and the differentiation tensor on the $\underline{x}^{RP}$ to $\underline{x}^{HP}$ transformation are non-singular we have:

$$\hat{\underline{u}}^T(n) = \lambda \underline{h}^T(n) \frac{\partial \underline{h}(n)}{\partial \underline{x}^{HP}(n)} \frac{\partial \underline{x}^{HP}(n)}{\partial \underline{x}^{RP}(n)} R(n) \qquad \text{F-12}$$

Introducing the expression of $\hat{\underline{u}}(n)$ into the L' we have:

$$L' = \lambda \left[L'' + 1/\lambda \sum_n \underline{h}^T \underline{h} \right] \qquad \text{F-13}$$

where:

$$L'' = \sum_n \left[\underline{h}^T(n) \frac{\partial \underline{h}(n)}{\partial \underline{x}^{HP}(n)} \frac{\partial \underline{x}^{HP}(n)}{\partial \underline{x}^{RP}(n)} R \left(h^T \frac{\partial \underline{h}(n)}{\partial \underline{x}^{HP}(n)} \frac{\partial \underline{x}^{HP}(n)}{\partial \underline{x}^{RP}(n)} \right)^T \right] \qquad \text{F-14}$$

And minimizing L' for large values of λ is approximately equivalent to the minimization of L".

Furthermore, from the assumption that the measurement noise $\underline{v}(n)$ (resolution) is comparatively smaller we make:

$$\hat{\underline{x}}^{HP}(n) \simeq \underline{x}^{HP}(\underline{z}^{RP}(n), \hat{\underline{x}}^{HR}) \qquad \text{F-15}$$

Thus the maximum likelihood estimate of the time series $\underline{x}^{HP}(n)$ and $\underline{x}^{HR}$ is reduced to minimizing the expression of L" (Expression F-14) over the bias vector $\underline{x}^{HR}$, using the expression of $\underline{x}^{HP}(n)$ given by Expression F-15.

Section Four of this appendix shows the program listings of the actual implementation of this smooth estimator.

3. RECURSIVE POSITION ESTIMATOR

The recursive filter approach used for estimating the position between the parts being assembled is based on the Extended Kalman filter developed in Reference F-1. A brief description of the notation and expressions from this reference are shown here.

Extended Kalman Filter

Let's assume a system whose state at time n can be fully described by the state vector $\underline{x}(n)$ and whose time behavior can be described by the following dynamic equation.

$$\underline{x}(n+1) = \underline{X}(\underline{x}(n)) + G(n)\underline{w}(n) \qquad \text{F-16}$$

where:

$\underline{w}(n)$ is a zero mean with random process with covariance matrix:

$$E\left[\underline{w}(n)\underline{w}^T(m)\right] = Q(n)\delta_{nm}$$

G(n) is a time dependent matrix, $\underline{X}(\bullet)$ represents a vector operation on the vector $\underline{x}(n)$.

Let's also assume that the measurement $\underline{z}(n)$ is available from the system at time n and has the form:

$$\underline{z}(n) = \underline{Z}(\underline{x}(n)) + \underline{v}(n) \qquad \text{F-17}$$

where $\underline{v}(n)$ is a white zero mean random process of covariance matrix:

$$E\left[\underline{v}(n)\underline{v}^T(m)\right] = R(n)\delta_{nm}$$

$\underline{Z}(\bullet)$ represents a vector operation on $\underline{x}(n)$.

The goal is to estimate the state vector $\underline{x}(n)$ from the available measurements $\underline{z}(n)$ and the knowledge of the system. The Kalman Filter, shown schematically in Figure F-1, will give a minimum error covariance recursive estimate of the state vector when dealing with linear systems ($\underline{X}(\cdot)$ and $\underline{Z}(\cdot)$ linear functions on the vector $\underline{x}$). For a non-linear system the Kalman Filter has been expanded to what has been called the Extended Kalman Filter; the output of this filter does not necessarily represent an optimal type of estimate. The Extended Kalman filter has the same structure shown in Figure F-1; the expression for estimate is:

$$\hat{\underline{x}}(n + 1/n + 1) = \hat{\underline{x}}(n + 1/n) + K(n + 1)\ (\underline{z}(n + 1) - \underline{Z}(x\ (n + 1/n))) \qquad \text{F-18}$$

where $\hat{\underline{x}}(n/m)$ is the estimate of the state at time n given the measurements up to time m.

$$\hat{\underline{x}}(n + 1/n) = \underline{X}(\underline{x}(n/n))$$

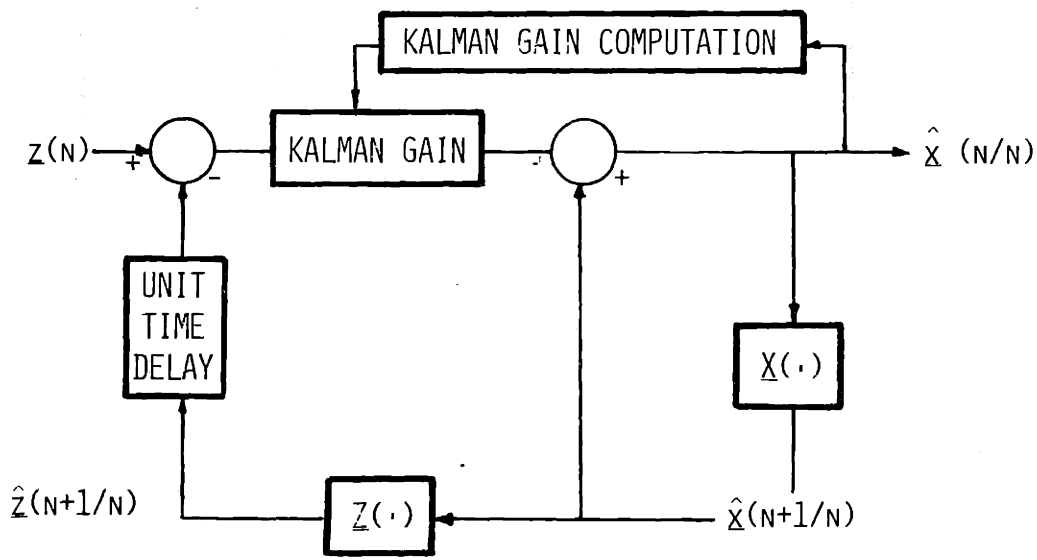


Figure F-1: Schematic Representation of a Kalman Filter

$K(n)$ is the Kalman gain computed through the following series of relations:

$$K(n + 1) = \Sigma\ (n + 1/n + 1)\ H(n + 1)\ R^{-1}\ (n + 1) \qquad \text{F-19}$$

where:

$H(n + 1)$ is the matrix of the derivatives of the vector function $\underline{Z}(\cdot)$ with respect to the state vector $\underline{x}$ and evaluated at the one step estimate of this vector $\underline{x}(n + 1/n)$; formally:

$$H(n + 1) = \left(\frac{\partial Z}{\partial \underline{x}}\right)_{\underline{x} = \hat{\underline{x}}(n + 1/n)}$$

$\Sigma(n/m)$ is the matrix covariance of the error in estimating $\underline{x}(n)$ using the measurements up to $\underline{z}(m)$; formally:

$$\Sigma(n/m) = E\left[(\hat{\underline{x}}(n/m) - \underline{x}(n))\ (\hat{\underline{x}}(n/m) - \underline{x}(n))^T\right]$$

The matrix covariance of the estimate error covariance is computed through the following set of recursive relations:

$$\Sigma^{-1}(n + 1/n + 1) = H^T(n + 1)R^{-1}(n + 1)H(n + 1) + \Sigma^{-1}(n + 1/n) \qquad \text{F-20}$$

$$\Sigma^{-1}(n + 1/n) = \left[D(n)\Sigma(n/n)D^T(n)\right]^{-1} - \left[D(n)\Sigma(n/n)D^T(n)\right]^{-1} G(n)\Big[Q^{-1}(n) + G^T(n)\left\{D(n)\Sigma(n/n)D^T(n)\right\}^{-1} G(n)\Big]^{-1} G^T(n)\left[D(n)\Sigma(n/n)D^T(n)\right]^{-1}$$

where $D(n)$ is the matrix derivative of the function $\underline{X}(\ )$ with respect to the vector $\underline{x}$ and evaluated at $\underline{x} = \hat{\underline{x}}(n/n)$.

Given an initial value for the estimate of the state, $\underline{x}(0/0)$, and an initial value for the matrix covariance of the estimated error, $\Sigma(0/0)$, the recursive

estimate of the state $\underline{x}(n)$ can be computed from Expression F-18 and Expression F-20 provided that all the operations in Expression F-18 and Expression F-20 are well defined. The implementation described in the rest of this section is designed to avoid ill-defined operations.

In order to use the Extended Kalman Filter technique in the Position Measuring approach, the part mating process has to be modelled in terms of the generalized system form given by Expression F-16 and Expression F-17. This model has to include the following characteristics of the mating process.

- the presence of a bias term ($\underline{x}^{RH}$), constant in time, that disturbs the measurement of the position of the parts; the value of this bias is not known with certainty and has to be estimated from the available measurements;
- the parts, during the mating process, will maintain contact and thus their relative position ($\underline{x}^{HP}$) has to satisfy the geometric conditions; the relative position between the parts is not known with certainty and has to be estimated;
- the position measurements available ($\underline{z}^{RP}(n)$) are altered by resolution noise; the statistics of this noise can be modelled from the physics of the particular measuring technique used.

From the above conditions for the model it is concluded that the state vector of the system should contain information of both the bias vector ($\underline{x}^{RH}$) and the relative position vector between the parts ($\underline{x}^{HP}$). The bias vector is constant in time; thus its dynamic equation can simply be written as:

$$\underline{x}^{RH}(n + 1) = \underline{x}^{RH}(n) \qquad \text{F-21}$$

As shown in Figure I-4, the relative position vector between the mating parts ($\underline{x}^{HP}$) changes in time according to combination of nominal position and force control loop commands on the positioning device and physical impedance displacement due to reaction forces. This displacement has been assumed such that the parts will

always maintain contact; this is expressed in terms of the k_h dimensional vector relation:

$$\underline{h}(\underline{x}^{HP}(n)) = \underline{0} \qquad \text{F-22}$$

This relation has to be satisfied at every time n. This implies that the dimensionality of the relative movement between the mating parts is $k_x - k_h$ (k_x is the dimension of the vector $\underline{x}^{HP}$). The vector $\underline{x}^{HP}(n)$ can be partitioned into two vectors $\underline{p}(n)$ and $\underline{q}(n)$ of dimensions $k_x - k_h$ and k_h respectively. If this partition is made such that the vector $\underline{q}(n)$ can be calculated from the value of the vector $\underline{p}(n)$ and the geometric Relation F-22, the the vector $\underline{p}(n)$ can vary freely within the range of definition of the vector function $\underline{h}()$ and the vector $\underline{x}^{HP}$ calculated from it will still satisfy the geometric constraint. This is modelled by the following dynamic equation on the vector $\underline{p}(n)$.

$$\underline{p}(n + 1) = \underline{p}(n) + \underline{w}(n) \qquad \text{F-23}$$

If we want to avoid the very complicated, detailed modelling of the contact reaction forces between the parts we have to assume very little knowledge about the driving vector $\underline{w}(n)$. This is modelled by making its covariance matrix (Q in Expression F-16) very big. The dynamic equation for the parts mating process is formed by combining the dynamic Expression F-21 and Expression F-23 in one state vector denoted here by $\underline{x}(n)$ and defined by:

$$\underline{x}(n) = \begin{bmatrix} \underline{x}^{RH}(n) \\ \underline{p}(n) \end{bmatrix} \qquad \text{F-24}$$

and the dynamic equation is given, in terms of the new vector $\underline{x}(n)$ by:

$$\underline{x}(n + 1) = \underline{x}(n) + G\ \underline{w}(n) \qquad \text{F-25}$$

where:

$$G(n) = \begin{bmatrix} 0 \\ I \end{bmatrix} \qquad \text{F-26}$$

of dimension k_h, $k_x + k_h$ and the matrix covariance $Q(n)$ of the vector $\underline{w}(n)$ with a value chosen to be comparatively large.

From the dynamic Equation F-16, the vector function $\underline{X}(\cdot)$ and the matrix of the derivatives are simply given by:

$$\underline{X}(\underline{x}(n)) = x(n)$$

$$\Phi(n) = I \qquad \text{F-27}$$

The matrix equation in Expression F-20 for the computation of the error covariance matrix of the $(n + 1)^{th}$ estimate given the n^{th} measurement requires the value of the inverse of the matrix Q; an arbitrary large value of Q implies a small value of Q^{-1} that can be made equal to zero. The equation of $\Sigma(n + 1/n)$ thus becomes:

$$\Sigma^{-1}(n + 1/n) = \begin{bmatrix} S_{11} - S_{12}\ S_{22}^{-1}\ S_{21} & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{F-28}$$

where S_{ij} is the i^{th} j^{th} submatrix result of the partition of $\Sigma^{-1}(n/n)$ at the k_x^{th} row and column.

The position measurement equation is in the general form of Expression F-17 but this vector expression has to be written in terms of the state vector $\underline{x}(n)$ or, in other words, the vector $\underline{x}^{RP}$ in Expression F-1 has to be computed from the vector $\underline{x}(n)$ defined in Expression F-24. This computation is done through the following series of steps:

- the vector $\underline{x}^{RH}$ is separated from the top k_x elements of the vector $\underline{x}(n)$ defined in Expression F-24;
- the partition $\underline{p}$ of the vector $\underline{x}^{HP}$ is separated from the bottom $k_x - k_h$ elements of the vector $\underline{x}(n)$;
- the partition $\underline{q}$ of the vector $\underline{x}^{HP}$ is computed by iterating over this partition such that the geometric constraints ($\underline{h}(x^{HP}) = 0$) are satisfied; the vector $\underline{x}^{HP}$ is reconstructed for the values of $\underline{p}$ and $\underline{q}$;
- the vector $\underline{x}^{RP}$ is computed from the vectors $\underline{x}^{RH}$ and $\underline{x}^{HP}$ making use of the transformations developed in Appendix A.

The matrix H, defined in Expression F-19, needs to be evaluated for implementing the recursive equation in Expression F-20 and the Kalman Gain term in Expression F-19; this is done by partitioning the matrix H as follows:

$$H = \frac{\partial \underline{Z}}{\partial \underline{x}} = \left[\frac{\partial \underline{Z}}{\partial \underline{x}^{RH}} \quad \frac{\partial \underline{Z}}{\partial \underline{p}} \right] \qquad \text{F-29}$$

The left partition of the matrix H corresponds to the term $\frac{\partial \underline{Z}}{\partial \underline{x}^{RH}}$ and is computed through the difference transformations developed in Appendix A. The right partition of the matrix H in Expression F-29 has to consider that the geometric constraint of Expression F-22 has to be satisfied thus:

$$d\underline{h} = 0 = \frac{\partial \underline{h}}{\partial \underline{p}} d\underline{p} + \frac{\partial \underline{h}}{\partial \underline{q}} d\underline{q} \qquad \text{F-30}$$

or

$$\frac{\partial \underline{q}}{\partial \underline{p}} = \left(- \frac{\partial \underline{h}}{\partial \underline{q}} \right)^{-1} \frac{\partial \underline{h}}{\partial \underline{p}} \qquad \text{F-30}$$

The right hand side partition of the matrix H is thus computed through the following series of steps:

- the matrix difference $\frac{\partial \underline{x}^{RP}}{\partial \underline{x}^{HP}}$ is computed through the difference of transformation developed in Appendix A; this difference matrix is then partitioned into:

$$\frac{\partial \underline{x}^{RP}}{\partial \underline{p}} \quad \text{and} \quad \frac{\partial \underline{x}^{RP}}{\partial \underline{q}} \qquad \text{F-31}$$

- through the geometric constraint relation the Expression F-30 is evaluated, the right hand side partition of the matrix H becomes:

$$\frac{\partial Z}{\partial \underline{p}} = \frac{\partial \underline{x}^{RP}}{\partial \underline{p}} + \frac{\partial \underline{x}^{RP}}{\partial \underline{q}} \frac{\partial \underline{q}}{\partial \underline{p}} \qquad \text{F-32}$$

With the expression of $\underline{Z}(\underline{x}(n))$ and $H(n)$ evaluated at $\hat{\underline{x}}(n/n)$ the Kalman Filter is implemented as shown schematically in Figure F-2.

The computations of the matrix H and the vector $\underline{Z}(\ )$ have to be well defined. This implies that both the iteration that finds the partition $\underline{q}$ of the vector $\underline{x}^{HP}$ and the matrix inversion in Expression F-30 have to be possible. This is done, in this particular implementation, by selecting the partition $\underline{q}$ of the vector $\underline{x}^{HP}$ in which the determinant of the derivative of the geometric constraint with respect to the vector $\underline{q}$ has the biggest absolute value. This procedure is done at every measured point.

The implementation of this recursive filter is listed in Section Five of this appendix.

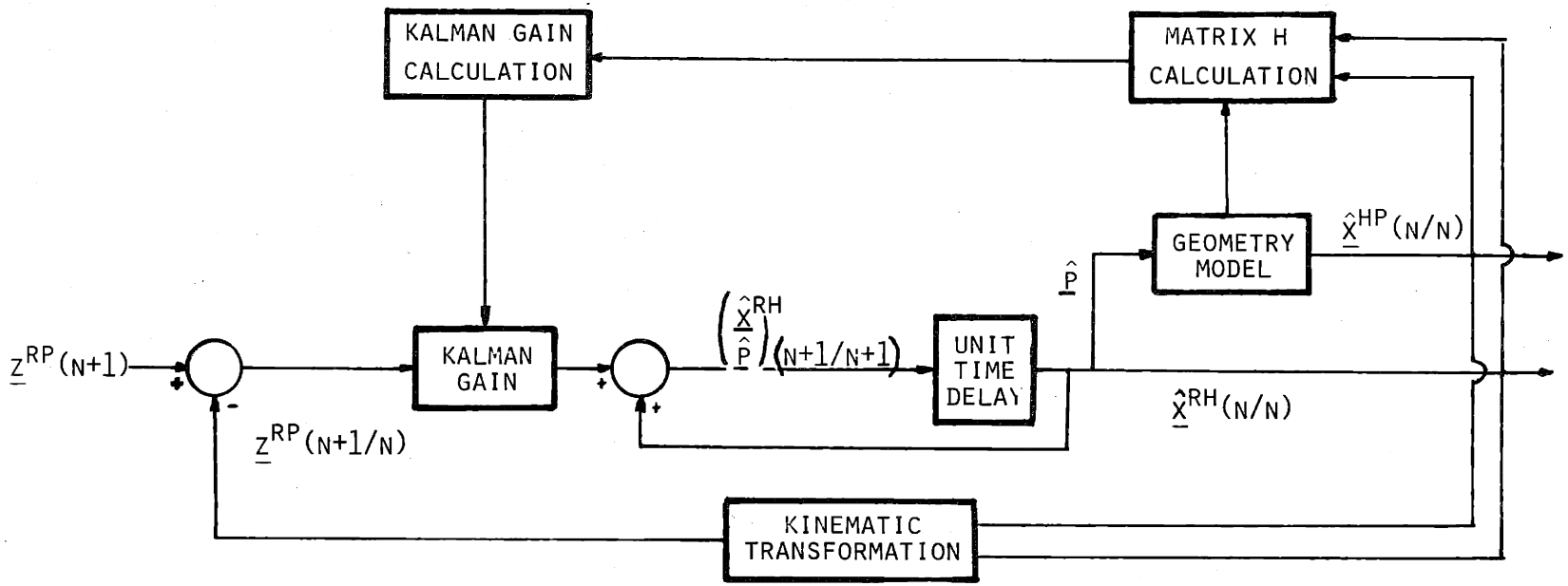


Figure F-2: Schematic Representation of the Recursive Filter Implementation

4. LISTING FOR SMOOTH ESTIMATOR

The implementation of the smooth estimator described in Section Two is realized through the following set of programs:

- a main program, denoted by STT6D. It inputs the measurements, initializes the optimization, calls for the minimization program and calls for the program that outputs the estimated data;
- a minimization routine, called PATSH, provided by Reference F-2;
- a routine that evaluated L" in Expression F-14. This is denoted by LKFCT;
- a routine that evaluates the geometry constants vectors and their derivatives. This program is called TPC6D and is listed in Appendix D;
- a set of routines for operating the different kinematic transformations. These routines are developed and listed in Appendix A;
- a routine, called WRSTT, that outputs the data at the estimated points.

```
C****************************** STT6D.SS  ******************************
C
C
C       ---READS THE RELATIVE POSITIONS MEASUREMENTS AND THE ERROR
C          COVARIANCE MATRICES GIVEN BY TABLES "Z#.SS",..
C          (THIS TABLES WERE PRODUCED BY LISTING.SV FROM
C          FILES "DATA-.SS" OF SEQUENCES OF ENCODER READINGS)
C       ---CALLS PATSH AND STARTS TO DO A SESARCH FOR FINDING
C          THE TRUE BIAS IN POSITION BETWEEN PEG AND HOLE
C          FOR COMPUTING THE ESTIMATE OF POSITION AND BIASES VECTOR
C          THIS METHOD USES THE MAXIMUM LIKELIHOOD ESTIMATE OF THE
C          THE POSITION THEN COMPUTES THE LOGLIKELIHOOD FCT, AND
C          MINIMIZES IT; QUE TAL??
C       ---PRINTS POINTS SELECTED AND GEOMETRY FUNCTIONS AT THEM
C
C
C
        INCLUDE "STT6D.SP"
        INTEGER XFILE(7),YFILE(7),IMM1,IMM2,IMM3,M,ILIM,ICK
        INTEGER K1,K2,K3,K4,J,I,KKK,LS
        REAL V,DEL,DLMIN, XHR(6)
        DOUBLE PRECISION RDS,CTRR,ZRPN,RN,X(6)
C
C  DATA:
        DATA (XHR(I),I=1,6)/6*0.1/
C  READ:
        TYPE "INPUT NAME OF FILE WHERE DATA IS: "
        READ(11,101) XFILE(1)
```

```
101     FORMAT (S13)
        OPEN 1,XFILE
        ACCEPT "YOU WANT TO START AT: ",IMM1
        ACCEPT "END AT               : ",IMM2
        ACCEPT "STEPPING EVERY       : ",IMM3
        READ(1,111) RDS,CTRR
111     FORMAT (2F20.6,/)
        K4=1
        DO 9 J=1,IMM2
        READ (1,10) (ZRPN(K4,I),I=1,6)
        READ (1,100) ((RN(K4,KKK,I),I=1,6),KKK=1,6)
100     FORMAT(6(6(E13.5),/))
10      FORMAT(2(3(E24.14),/))
        IF(J.LT.IMM1) GO TO 9
        L=((J-IMM1)/IMM3)*IMM3-(J-IMM1)
        IF(L.EQ.0) K4=K4+1
9       TYPE J,K4
        CONTINUE
        K2=IMM1
        K3=IMM3
        CLOSE 1
C
C   ACCEPTS DATA AND ASKS FOR REFERENCE TO LOCATION OF RESULTS
C
        TYPE " WHERE YOU WANT THE RESULTS:  "
        READ(11,101) YFILE(1)
        ACCEPT "YOU WANT INITIAL JIG ERROR (XHR) CHANGED?  0=NO; 1=YES
        IF(LS.LE.0) GO TO 1000                                    : ",LS
        DO 1001 I=1,6
1001    ACCEPT ":  ",XHR(I)
1000    M=6
        ACCEPT"INCREMENTS *: INITIAL= ",DELS
        ACCEPT"                FINAL= ",DLMIN
        ACCEPT"MAX # OF ITERATIONS = ? ",ITLIM
        TYPE"CHOOSE A #:  <0 FOR NO PRINTOUT"
        TYPE"             =0 FOR SOME PRINTOUT"
        TYPE"             >0 FOR FULLY DETAILED"
        ACCEPT"SO, WHAT'S YOUR CHOICE? : ",ICK
C
C  CALL SEARCH ROUTINE AND PUT #'S IN DIFFERENT FILES
C
        OPEN 2,"CHK1.RS"
        OPEN 3,YFILE
        K1=1
        CALL PATSH(XHR,V,M,DELS,DLMIN,ITLIM,ICK)
        DO 19 I=1,6
19      X(I)=XHR(I)
        CALL WRSTT(X)
        WRITE(2,209) M
        WRITE(3,209) M
209     FORMAT ("  DIMENSION OF THE SEARCH= ",I4)
        WRITE(3,702)IMM1,IMM2,IMM3
        WRITE(2,702) IMM1,IMM2,IMM3
        IX=(IMM2-IMM1)/IMM3
        WRITE(3,105) IX
105     FORMAT(/,/," NUMBER OF POINTS PROCESSED= ",I3,/)
702     FORMAT ("START AT: ",I3,";  THROUGH: ",I3,";  STEP: ",I3)
        K1=1
```

```
      WRITE(3,757)V,ICK
      WRITE(2,757)V,ICK
      WRITE(2,102) XFILE(1)
      WRITE(3,102) XFILE(1)
      WRITE(3,103) YFILE(1)
      CLOSE 3
      CLOSE 2
102    FORMAT("  STATIC RUN ON FILE: ",S13)
103   FORMAT(//,"       RESULT FILE:   ",S13)
757   FORMAT("VALUE= ",1PE12.5,"       # OF ITERATIONS= ",I3)
      TYPE"                                              "
      END


C******************** VALUE.SS ************************
C
C
C
C     THIS SUBROUTINE ACTS AS A DUMMY SUBROUTINE BETWEEN
C     PATSH (SINGLE PRECISION) AND LKFCT (DOUBLE PRECISION)
C
C
C
C
C
      SUBROUTINE VALUE(X,V)
      REAL X(12),V
      DOUBLE PRECISION XD(12),VD
C
      DO 1 I=1,12
1     XD(I)=X(I)
      CALL LKFCT(XD,VD)
      V = VD
      TYPE V
C
      RETURN
      END

C*************** LKFCT.SS ***************
C
C  THIS SUB IS USED BY PATSR FOR EVALUATING A COST FCT
C  TO ESTIMATING A BAD POINT FOR THE ACTUAL POSITION
C  PATSR MINIMIZES THIS COST
C
      SUBROUTINE LKFCT(XHR,V)
      COMPILER DOUBLE PRECISION
      INCLUDE "STT6D.SP"
      DIMENSION XRP(6),XHR(6),XHP(6),BRP(3,4),BHR(3,4),BHP(3,4)
      DIMENSION H(2),DH(2,6),DHPRP(6,6),VV(6),VVV(6)
      V=0.
      DO 1 N=K1,K4
      DO 2 I=1,6
2     XRP(I)=ZRPN(N,I)
      CALL XTOB(XHR,BHR)
      CALL XTOB(XRP,BRP)
      CALL BBB(BHR,BRP,BHP)
      CALL BTOX(BHP,XHP)
      CALL TPC6D(XHP,H,DH,KH)
      CALL DXDX(XHP,XHR,XRP,DHPRP)
```

```
      DO 3 I=1,6
      VV(I)=0.
      DO 3 J=1,KH
3     VV(I)=VV(I)+H(J)*DH(J,I)
      DO  5 I=1,6
      VVV(I)=0.
      DO 5 J=1,6
5     VVV(I)=VVV(I)+VV(J)*DHPRP(J,I)
      DO 4 I=1,6
      VV(I)=0.
      DO 4 J=1,6
4     VV(I)=VV(I)+VVV(J)*RN(N,I,J)
      DO 6 I=1,6
6     V=V+VV(I)*VVV(I)
1     CONTINUE
100   FORMAT ("XRP",/,6(1PE12.4),//)
101   FORMAT ("XHR",/,6(1PE12.4),//)
114   FORMAT ("V",/,1PE12.4,//)
110   FORMAT ("H",/,2(1PE12.4),//)
111   FORMAT("DH",/,2(6(1PE12.4),/),/)
112   FORMAT ("DXHP/DXRP",/,6(6(1PE12.4),/),/)
113   FORMAT ("VV",/,6(1PE12.4),//)
102   FORMAT ("XHP",/,6(1PE12.4),//)
109   FORMAT ("R(N)",/,6(6(1PE12.4),/),/)
115   FORMAT ("VVV",/,6(1PE12.4),//)
C
      RETURN
      END

C*************** WRSTT.SS ***************
C
C  THIS SUB COMPUTES AND PRINTS THE VALUES AT OPTIMUN
C
C
C
      SUBROUTINE WRSTT(XHR)
      COMPILER DOUBLE PRECISION
      INCLUDE "STT6D.SP"
      DIMENSION XRP(6),XHR(6),XHP(6),BRP(3,4),BHR(3,4),BHP(3,4)
      DIMENSION XPH(6),H(2),DH(2,6),DHPRP(6,6),VV(6)
C
      V=0.
      DO 1 N=K1,K4
      DO 2 I=1,6
2     XRP(I)=ZRPN(N,I)
      CALL XTOB(XHR,BHR)
      CALL XTOB(XRP,BRP)
      CALL BBB(BHR,BRP,BHP)
      CALL BTOX(BHP,XHP)
      CALL INVX(XHP,XPH)
      CALL TPC6D(XHP,H,DH,KH)
      CALL DXDX(XHP,XHR,XRP,DHPRP)
      DO 3 I=1,6
      VV(I)=0.
      DO 3 J=1,KH
      DO 3 L=1,KH
3     VV(I)=VV(I)+H(J)*DH(J,L)*DHPRP(L,I)
      V=0.
```

```
      DO 4 I=1,6
      DO 4 J=1,6
4     V=V+VV(I)*VV(J)*RN(N,I,J)
      IX=K2+(N-1)*K3
      WRITE(3,109)IX,(XRP(IX),IX=1,6),(XHP(IX),IX=1,6),(XPH(IX),IX=1,6),
     #  (H(IX),IX=1,2),V
109   FORMAT(I3,2X,6(1PE12.5),5(/,5X,6(1PE12.5)))
1     CONTINUE
57    WRITE(3,108)
108   FORMAT(/,"INDX",5X,"ZRP(1)",6X,"ZRP(2)",6X,"ZRP(3)",6X,"ZRP(4)",6X,
     # "ZRP(5)",6X,"ZRP(6)",/,9X,"XHP(1)",6X,"XHP(2)",6X,"XHP(3)",6X,"XHP(4)",
     # 6X, "XHP(5)",6X,"XHP(6)",/,9X,"XPH(1)",6X,"XPH(2)",6X,"XPH(3)",6X,
     # "XPH(4)",6X,"XPH(5)",6X,"XPH(6)",6X)
      WRITE(3,107)
107   FORMAT(8X,"  H(1)  ",4X,"  H(2)  ",4X,"COST  ")
      WRITE(3,111) (XHR(I),I=1,6)
      WRITE(3,110)CTRR,RDS
110   FORMAT(/,"CLEARENCE RATIO: ",F15.6,/,"RADIUS :     ",F15.6,/)
111   FORMAT("XHR = ",/,6(1PE12.4),/)
      RETURN
      END
```

5. RECURSIVE FILTER IMPLEMENTATION

The Extended Kalman Filter described in Section Three of this appendix is implemented through the following programs:

- a main program denoted RCSV6 that initializes the recursion, takes the reading of the measurements and outputs the estimate at each step;
- a subroutine SETX to set the initial value of $\underline{x}$ for the Recursion;
- a subroutine named KF. It is called by the main program and computes the new estimate given the old one and the new measurement;
- a subroutine denoted ZHPOFX that computes the expression of the estimated measurement explained in Section Three, and the matrix H defined in Expression F-19 and Expression F-29;
- a geometry routine named TPC6D and explained in Appendix D.

```
C************************** RCSV6,SS ***************************
C
C
C
C
C       COMPUTES THE RECURSV ESTIMATE OF THE POSITION BETWEEN PEG
C
C
C
C
C
C
C
        COMPILER DOUBLE PRECISION
        COMMON/ GEOMETRY / RDS,CTRR
        COMMON K4,KX,KX1,K2X,KZ,KP,KXX,SNN(10,10),SINN(10,10),XHP(6),XRH(6)
        COMMON SHP(6,6),SRHHP(6,6),IX(6)
        INTEGER XFILE(7), YFILE(7), UFILE(7)
        REAL LGLKD
        DIMENSION X(10),XPH(6),ZRP(6),DLTZ(6),ZERO(6),XRP(6)
        DIMENSION R(6,6),R1(6,6),H(6,10),T(6,10),TT(10,6),T0(6,6),U(6)
C
        DATA KX,KZ,KX1,K2X / 2*6,7,12 /
        DATA (ZERO(I),I=1,6) / 6*0.0 /
C  READ:
        TYPE "INPUT NAME OF FILE WHERE DATA IS: "
        READ(11,101) XFILE(1)
101     FORMAT (S13)
        OPEN 1,XFILE
        ACCEPT "YOU WANT TO START AT: ",IMM1
        ACCEPT "END AT              : ",IMM2
        ACCEPT "STEPPING EVERY      : ",IMM3
        READ(1,111) RDS,CTRR
111     FORMAT (2F20.6,/)
```

```
C
C    ACCEPTS DATA AND ASKS FOR REFERENCE TO LOCATION OF RESULTS
C
         TYPE " WHERE YOU WANT THE RESULTS:  "
         READ(11,101) YFILE(1)
         TYPE "WHERE YOU WANT THE GARBAGE"
         READ(11,101) UFILE(1)
         OPEN 3 , YFILE
         OPEN 2, UFILE
C
C        SETS THE INITIAL VALUES FOR THE RECURSION
C
         DO 1 N=1,IMM1
         READ (1,102) (ZRP(I),I=1,KZ)
1        READ (1,100) ((R(K,I),I=1,KZ),K=1,KZ)
100      FORMAT(6(6(E13.5),/))
102      FORMAT(2(3(E24.14),/))
C
C        MAKES INITIAL XHP=0 , AND COMPUTES XRH BY MAKING XHP(XRH,ZRP)=0
C        FOR THE FIRST MEASUREMENT
C
         CALL XEQY(ZERO,XHP,KX)
         CALL XEQY(ZRP,XRH,KX)
         CALL XEQY(ZERO,DLTZ,KZ)
         KXX=2*KX
         ACCEPT "YOU WANT TO IMPUT AN INITIAL CONDITION? ( 0=NO ; 1=YES )
         IF(LS .EQ. 0) GO TO 2                                          :",LS
         DO 3 I=1,KX
3        ACCEPT"       XPH= ",XPH(I)
         CALL INVX(XPH,XHP)
         CALL XXX(ZRP,XPH,XRH)
2        DO 5 I=1,KX
         DO 5 J=1,KX
         SNN(I,J)=R(I,J)*10.
         SRHHP(I,J)=0.
5        SHP(I,J)=SNN(I,J)
         DO 33 IZ=1,KX
33       X(IZ)=XRH(IZ)
         CALL SETX(X)
         LGLKD=0.
         K4=0
         WRITE(3,200) N,(ZRP(I),I=1,KZ),K4,(X(I),I=1,KX),(XHP(I),I=1,KX),
     #   (XPH(I),I=1,KX),(SNN(I,I),I=1,KX),(SHP(I,I),I=1,KX),(DLTZ(I),I=1
     #   LGLKD                                                          ,KZ),
C
C        START THE RECURSION
C
         DO 6 N=IMM1+1,IMM2
         READ (1,102) (ZRP(I),I=1,KZ)
         READ (1,100) ((R(K,I),I=1,KZ),K=1,KZ)
         L=((N-IMM1)/IMM3)*IMM3-(N-IMM1)
         IF(L .LT. 0) GO TO 6
         K4=K4+1
         CALL INVAS(R,R1,KZ,NN)
         IF(NN .LT. KZ)WRITE(2,588)K4,NN,KZ
588      FORMAT("RCSV MATX R SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
         CALL ZHOFX(X,XRP,H)
         CALL XMNSY(ZRP,XRP,DLTZ,KZ)
```

```
      CALL KF6D(X,DLTZ,H,R1)
CCCC  WRITE(2,304)(X(IS),IS=1,KXX)
304   FORMAT("X= ",2(6(1PE12.3),/))
      CALL ABC(R1,H,T,KZ,KZ,KXX)
      CALL ATB(T,TT,KZ,KXX)
      CALL ABC(T,SNN,T,KZ,KXX,KXX)
      CALL ABC(T,TT,T0,KZ,KXX,KZ)
      CALL AMNSB(R1,T0,T0,KZ,KZ)
      CALL AXY(T0,DLTZ,U,KZ,KZ)
      CALL XPTY(U,DLTZ,Q1,KZ)
      Q4=FLOAT(K4)
      LGLKD=LGLKD+(Q1-LGLKD)/Q4
      CALL INVX(XHP,XPH)
      WRITE(3,200) N,(ZRP(I),I=1,KZ),K4,(X(I),I=1,KZ),(XHP(I),I=1,KX),
     #  (XPH(I),I=1,KX),(SNN(I,I),I=1,KX),(SHP(I,I),I=1,KX),
     #   (DLTZ(I),I=1,KZ),LGLKD
200   FORMAT(2(I3,2X,6(1PE12.5),/),10(5X,6(1PE12.5),/))
6     CONTINUE
      WRITE (3,201)
201   FORMAT("ITT 3   ZRP(1)       ZRP(2)       ZRP(3)       ZRP(4)
     #       ZRP(5)       ZRP(6)",/,"FL  #    XRH(1)       XRH(2)       XRH(3)
     #       XRH(4)       XRH(5)       XRH(6)",/,8X,"XHP(1)",6X,"XHP(2)",6X,"XHP(3
     #)",6X,"XHP(4)",6X,"XHP(5)       XHP(6)",/,7X,"SRH(1,1)     SRH(2,2)
     # SRH(3,3)     SRH(4,4)     SRH(5,5)     SRH(6,6)",/,7X,"SHP(1,1)
     #  SHP(2,2)     SHP(3,3)     SHP(4,4)     SHP(5,5)     SHP(6,6)",/,8X,
     #  "DLTZ(1)     DLTZ(2)      DLTZ(3)      DLTZ(4)      DLTZ(5)
     #DLTZ(6)",/,6X,"LOGLKHD FCT")
      WRITE(3,202) XFILE(1),UFILE(1),YFILE(1)
202   FORMAT("READ FROM :",S13,//,"GARBAGE IN :",S13,//,"FILE NAME :",S13)
C
      END

C************************ SETX,SS ***********************************
C
C
C
C     COMPUTE THE VECTOR Z GIVEN A VECTO X(N)
C     ALSO COMPUTES THE MATRIX H(N)=DZ(N)/DX(N) AT X(N)
C
C             XHP[ IX(I) ] = X( KX + I )
C
C
C
C
      SUBROUTINE SETX(X)
      COMPILER DOUBLE PRECISION
      COMMON K4,KX,KX1,K2X,KZ,KP,KXX,SNN(10,10),SINN(10,10),XHP(6),XRH(6)
      COMMON SHP(6,6),SRHHP(6,6),IX(6)
      DIMENSION X(10),D(2,2),H(2),DH(2,6),IN(2)
C
      DATA EPS/1E-6/
C
      MN=1
50    CALL TPC6D(XHP,H,DH,KH)
      KP=KX-KH
      KXX=KX+KP
      A=0.
      DO 6 I=1,KX-1
      DO 6 J=I+1,KX
```

```
      AA=ABS(DH(1,I)*DH(2,J)-DH(1,J)*DH(2,I))
      IF( AA .LE. A) GO TO 6
      A=AA
      IN(1)=I
      IN(2)=J
      TYPE "AA=",AA,"   A=",A,"   I=",IN(1),"   J=",IN(2)
6     CONTINUE
      DO 11 I=1,KH
      II=KP+I
11    IX(II)= IN(I)
      K=0
      DO 7 I=1,KX
      DO 17 J=KP+1,KX
17    IF(I .EQ. IX(J)) GO TO 7
      K=K+1
      IX(K)=I
7     CONTINUE
CCCC  WRITE(2,103) ((IX(I),I=1,KX),A)
103   FORMAT("IX                    DET D",/,6I2,1PE12.3,/)
C
C     ITERATE FOR FINDING THE NEW Q
C
      DO 2 I=1,KH
      DO 2 J=1,KH
2     D(I,J)=DH(I,IX(KP+J))
C*C
      CALL INVA(D,D,KH,NN)
      IF(NN .LT. KH)WRITE(2,588)K4,NN,KH
588   FORMAT("SX MATX D SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
      MN=2
      DO 3 I=1,KH
      A=0
      J=IX(KP+I)
      DO 4 L=1,KH
4     A=A-D(I,L)*H(L)
      XHP(J)=XHP(J)+A
      A=ABS(A/XHP(J))
3     IF(A .GT. EPS) MN=1
      GO TO (50,52),MN
C
C     IF A NEW VECTOR Q IS CHOSEN, RECONTTRUCTS THE NEW VECTOR
C     X AND COVARIANCE MATRIX SNN
C
52    DO 18 I=KX1,KXX
      II=IX(I-KX)
18    X(I)=XHP(II)
      DO 8 I=KX1,KXX
      II=IX(I-KX)
      DO 9 J=KX1,KXX
      JJ=IX(J-KX)
9     SNN(I,J)=SHP(II,JJ)
      DO 8 J=1,KX
      SNN(J,I)=SRHHP(J,II)
8     SNN(I,J)=SNN(J,I)
      CALL INVAS(SNN,SINN,KXX,NN)
```

```
      IF(NN .LT. KXX)WRITE(2,589)K4,NN,KXX
589   FORMAT("SX MATX SNN SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
C
      RETURN
      END


C************************ KF6D.SS ****************************
C
C
C
C
C     COMPUTES THE ESTIMATE OF THE VECTOR X(N) GIVEN
C     THE MEASUREMENT RESIDUE DLTZ(N)
C     THE VECTOR X(N) IS PARTITIONED AS FOLLOWS :
C                     B(N) (DIMSN: KX)
C        X(N)=        P(N) (DIMSN: KP)       KP+KX= KXX
C                     Q(N) (DIMSN: KH)
C
C     THE MEASUREMENT IS : Z(N)  (DIMSN: KZ)
C
C
C
C
C
C
      SUBROUTINE KF6D(X,DLTZ,H,R1)
      COMPILER DOUBLE PRECISION
      COMMON K4,KX,KX1,K2X,KZ,KP,KXX,SNN(10,10),S1NN(10,10),XHP(6),XRH(6)
      COMMON SHP(6,6),SRHHP(6,6),IX(6)
      DIMENSION X(10),HT(10,6),H(6,10),S(4,4),R1(6,6)
      DIMENSION GK(10,6),DLTZ(6),DLTX(10),U(6,10),UU(10,10),UT(10,6)
C
      DO 1 I=1,KP
      DO 1 J=1,KP
1     S(I,J)=S1NN(KX+I,KX+J)
      CALL INVAS(S,S,KP,NN)
      IF(NN .LT. KP)WRITE(2,588)K4,NN,KP
588   FORMAT("KF MATX S SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
      CALL ABC(R1,H,U,KZ,KZ,KXX)
      CALL ATB(H,HT,KZ,KXX)
      CALL ABC(HT,U,UU,KXX,KZ,KXX)
      DO 2 I=1,KX
      DO 2 J=1,KX
      S1NN(I,J)=S1NN(I,J)+UU(I,J)
      DO 2 L=KX1,KXX
      DO 2 M=KX1,KXX
2     S1NN(I,J)=S1NN(I,J)-S1NN(I,L)*S(L-KX,M-KX)*S1NN(M,J)
      DO 3 I=KX1,KXX
      DO 4 J=1,KX
      S1NN(I,J)=UU(I,J)
4     S1NN(J,I)=UU(J,I)
      DO 3 J=KX1,KXX
3     S1NN(I,J)=UU(I,J)
CCCC  WRITE(2,363)((S1NN(IS,JS),JS=1,KXX),IS=1,KXX)
363   FORMAT(" S1NN: ",/,10(6(1PE12.4),/,4(1PE12.4),/))
      CALL INVAS(S1NN,SNN,KXX,NN)
      IF(NN .LT. KXX)WRITE(2,589)K4,NN,KXX
```

```
589     FORMAT("KF MATX SINN SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
        CALL ABC(SNN,HT,UT,KXX,KXX,KZ)
        CALL ABC(UT,R1,GK,KXX,KZ,KZ)
        CALL AXY(GK,DLTZ,DLTX,KXX,KZ)
        CALL XPLSY(X,DLTX,X,KXX)
C
        RETURN
        END

C************************* ZHOFX,SS **********************************
C
C
C
C       COMPUTE THE VECTOR Z GIVEN A VECTO X(N)
C       ALSO COMPUTES THE MATRIX H(N)=DZ(N)/DX(N) AT X(N)
C
C                XHP[ IX(I) ] = X( KX + I )
C
C
C
C
        SUBROUTINE ZHOFX(X,XRP,HMTX)
        COMPILER DOUBLE PRECISION
        COMMON K4,KX,KX1,K2X,KZ,KP,KXX,SNN(10,10),S1NN(10,10),XHP(6),XRH(6)
        COMMON SHP(6,6),SRHHP(6,6),IX(6)
        DIMENSION X(10),XRP(6),HMTX(6,10)
        DIMENSION H(2),DH(2,6),D(2,2),DHDP(2,4),DQDP(2,4),IN(2)
        DIMENSION XZERO(6),DRPRH(6,6),DRPHP(6,6)
C
        DATA EPS,(XZERO(I),I=1,6)/1.E-6,6*0.0/
C
C       FIRST RECONTRUCT THE VECTORS XRH AND XHP
C
        DO 1 I=1,KX
1       XRH(I)=X(I)
        DO 12 I=1,KP
        J=IX(I)
12      XHP(J)=X(KX+I)
C
C       USE GEOMETRIC CONSTRAINT AND ITERATE FOR FINDING THE NEW Q
C
        MN=1
50      CALL TPC6D(XHP,H,DH,KH)
        KP=KX-KH
        DO 2 I=1,KH
        DO 2 J=1,KH
2       D(I,J)=DH(I,IX(KP+J))
        CALL INVA(D,D,KH,NN)
        IF(NN .LT. KH)WRITE(2,588)K4,NN,KH
588     FORMAT("ZHOFX MATX D SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
        GO TO (51,52),MN
51      MN=2
        DO 3 I=1,KH
        A=0
        J=IX(KP+I)
        DO 4 L=1,KH
4       A=A-D(I,L)*H(L)
        XHP(J)=XHP(J)+A
```

```
      A=ABS(A/XHP(J))
3     IF(A .GT. EPS) MN=1
      GO TO 50
C
C     ESTABLISH THE DIMENSION OF THE DIFERENT PARTITIONS OF X
C     AND COMPUTES THE MATRIX DQ/DP =DQDP FOR H=0.
C
52    KXX=KX+KP
      DO 19 I=1,KH
      DO 19 J=1,KP
19    DHDP(I,J)=DH(I,IX(J))
      CALL ABC(D,DHDP,DQDP,KH,KH,KP)
C
C     COMPLETE THE MATRIX SNN FOR THE
C     VECTOR Q (NOTE THAT THIS MATRIX IS NOW SINGULAR)
C
      DO 20 I=KX1,KXX
      II=IX(I-KX)
      DO 21 J=1,KX
21    SRHHP(J,II)=SNN(J,I)
      DO 20 J=I,KXX
      JJ=IX(J-KX)
      SHP(II,JJ)=SNN(I,J)
20    SHP(JJ,II)=SNN(J,I)
      DO 22 I=1,KH
      II=IX(I+KP)
      DO 23 J=1,KX
      SRHHP(J,II)=0.
      DO 23 L=1,KP
23    SRHHP(J,II)=SRHHP(J,II)-SNN(J,L+KX)*DQDP(I,L)
      DO 25 J=KX1,KXX
      JJ=IX(J-KX)
      SHP(JJ,II)=0.
      DO 24 L=1,KP
24    SHP(JJ,II)=SHP(JJ,II)-SNN(J,L+KX)*DQDP(I,L)
25    SHP(II,JJ)=SHP(JJ,II)
      DO 22 J=1,KH
      JJ=IX(J+KP)
      SHP(II,JJ)=0.
      DO 22 L=1,KP
22    SHP(II,JJ)=SHP(II,JJ)-SHP(II,IX(L))*DQDP(J,L)
C
C     SEARCH FOR THE KH*KH PARTITION OF DH THAT YIELDS THE
C     DETERMINANT WITH MAXIMUM ABSOLUTE VALUE THIS PARTITION
C     DEFINES THE PARTITION Q ON XHP
C
      A=0.
      DO 6 I=1,KX-1
      DO 6 J=I+1,KX
      AA=ABS(DH(1,I)*DH(2,J)-DH(1,J)*DH(2,I))
      IF( AA .LE. A) GO TO 6
      A=AA
      IN(1)=I
      IN(2)=J
6     CONTINUE
      DO 29 I=1,KH
29    IF ( IN(I) .NE. IX(KP+I) ) GO TO 10
      DO 13 I=KX1,KXX
```

```
13      X(I)=XHP(IX(I-KX))
        GO TO 15
C
C       IF A NEW VECTOR Q IS CHOSEN, RECONTTRUCTS THE NEW VECTOR
C       X AND COVARIANCE MATRIX SNN
C
10      DO 11 I=1,KH
        II=KP+I
11      IX(II)= IN(I)
        K=0
        DO 7 I=1,KX
        DO 17 J=KP+1,KX
17      IF(I .EQ. IX(J)) GO TO 7
        K=K+1
        IX(K)=I
7       CONTINUE
        DO 18 I=KX1,KXX
        II=IX(I-KX)
18      X(I)=XHP(II)
        DO 8 I=KX1,KXX
        II=IX(I-KX)
        DO 9 J=KX1,KXX
        JJ=IX(J-KX)
9       SNN(I,J)=SHP(II,JJ)
        DO 8 J=1,KX
        SNN(J,I)=SRHHP(J,II)
8       SNN(I,J)=SNN(J,I)
        CALL INVAS(SNN,S1NN,KXX,NN)
        IF(NN .LT. KXX)WRITE(2,589)K4,NN,KXX
589     FORMAT("ZHOFX MATX SNN SING AT K4= ",I3,"  RANK=",I2,"  ORDER= ",I2)
C
C       COMPUTES THE VECTOR XRP
C
15      CONTINUE
        CALL XXX(XRH,XHP,XRP)
C
C       AND NOW COMPUTE THE MATRIX HMTX
C
        CALL DXDX(XRP,XZERO,XRH,DRPRH)
        CALL DXDX(XRP,XRH,XHP,DRPHP)
        DO 30 I=1,KZ
        DO 31 J=1,KX
31      HMTX(I,J)=DRPRH(I,J)
        DO 30 J=KX1,KXX
        HMTX(I,J)=DRPHP(I,IX(J-KX))
        DO 30 L=1,KH
30      HMTX(I,J)=HMTX(I,J)-DRPHP(I,IX(KP+L))*DQDP(L,J-KX)
CCCC    WRITE(2,440) ((HMTX(IS,JS),JS=1,KXX),IS=1,KZ)
440     FORMAT(" ZHOFX MATX HMTX: ",/,6(6(1PE12.4),/),4(1PE12.4),/))
C
        RETURN
        END
```

LIST OF REFERENCES

I-1. Lynch, P.M., Economical-Technological Modeling and Design Criteria for Programmable Assembly Machine, Ph.D. Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA., June 1976.

I-2. Rosen, C., et al., Exploratory Research in Advance Automation, Project Reports 1 through 6, NSF Grant GI-38100X, GI-38100X1 and APP75-13074, Stanford Research Institute, December 1973 through November 1976.

I-3. Birk, J. et al., General Method to Enable Robtos with Vision to Acquire, Orient and Transport Workpieces, Third Report, University of Rhode Island, August 1977.

I-4. Bolles, R.C. and R. Paul, The Use of Sensory Feedback in a Programmable Assembly System, Stanford University, Artificial Intelligence Laboratory Memo AIM-220, October 1973.

I-5. Inoue, H., Force Feedback in Precise Assembly Tasks, MIT, Artificial Intelligence Laboratory, Memo 308, August 1974.

I-6. McCallion, H. and P.C. Wang, Some Thoughts on the Automation Assembly of a Peg and Hole, pp 141-146, The Industrial Robot, Vol 2, No. 4, December 1975.

I-7. Laktionov, N.M. and C.Ya. Andreev, The Automatic Assembly of Large Parts, Vol. XLVI, No. 8, pp 40-43 and Problems in the Assembly of Large Parts, Vol. XLVI, No. 1, pp. 60-61, Russian Engineering Journal, 1966.

I-8. Gusev, A.S., Automatic Assembly of Cylindrical Shaped Components, Russian Engineering Journal, Vol. XLIX, No. 1, pp 53-57, 1969.

I-9. Nevins, J.L., et al., Exploratory Research in Industrial Modular Assembly, C.S. Draper Laboratory Reports R-800, R-850, R-921 and R-996, 1974-1976.

I-10. Simunovic, S., Task Descriptors for Automated Assembly, SM Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA, January 1976.

I-11. Groome, R.C., Force Feedback Steering of a Teleoperator System, SM Thesis, Department of Astronautics, MIT, Cambridge, MA., August 1972.

I-12. Watson, P.C. and S.H. Drake, Pedestal and Wrist Force Sensors for Automatic Assembly, C.S. Draper Laboratory Report P-176, Cambridge, MA., June 1975.

I-13. Nevins, J.L., D.E. Whitney, A.E. Woodin, S.H. Drake, M. Lynch, D. Seltzer, R. Sturges and P. Watson, A Scientific Approach to the Design of Computer Controlled Manipulators, C.S. Draper Laboratory, Report R-837, Cambridge, MA., August 1974.

II-1. Schweppe, F.C., Uncertain Dynamic Systems, Prentice-Hall, Inc., 1973.

II-2. Schweepe, F.C., Uncertain Dynamic Systems, Section R.4, pp 376, Prentice-Hall, Inc., 1973.

II-3. Simunovic, S., Task Descriptors for Automated Assembly, Appendix II, SM Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA., February 1976.

IV-1. Spencer, R.M., High Level Control of Reprogrammable Automatic Assembly Machines, SM Thesis, Department of Electric Engineering, MIT, Cambridge, MA., September 1975.

IV-2. Drake, S.H., Using Compliance in Lieu of Sensory Feedback for Automatic Assembly, ScD Thesis, Department of Mechanical Engineering, September 1977.

IV-3. Whitney, D.E., Computer Controlled Robot Assembly, C.S. Draper Laboratory 16 mm movie, Cambridge, MA., February 1978.

A-1. Sokolnikoff, I.S., Tensor Analysis, J. Wiley & Sons, 1967.

A-2. Simunovic, S., Task Descriptors for Automated Assembly, Chapter II and Chapter III, SM Thesis, Department of Mechanical Engineering, MIT, Cambridge, Ma., February 1976.

B-1. Whitney, D.E., Force Feedback Control of Manipulator Fine Motions, JACC, 1977.

C-1. Schweppe, F.C., Uncertain Dynamic Systems, Section 12.4, pp 376, Prentice-Hall, Inc., 1973.

C-2. Schweppe, F.C., Uncertain Dynamic Systems, Appendix A, pp 495, Prentice-Hall Inc., 1973.

E-1. Sturges, R., Teleoperator Arm Design Program (TOAD), C.S. Draper Laboratory, MIT, Cambridge, MA., February 1973.

F-1. Schweppe, F.C., Uncertain Dynamic Systems, Section 13.10, pp 402, Prentice-Hall, Inc., 1973.

F-2. Hooke and Jeeves, Direct Search...., journal of the A.C.M., Vol. 8, No. 2, pp 212-229, April 1971.