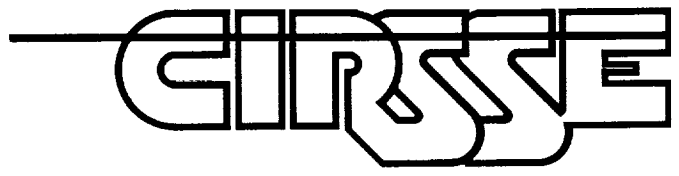
CIRSSE-R-99

NASA-CR-192780

NAGW-1333

GRANT
IN-63-CR
153759
P 166



Center for Intelligent Robotic Systems for Space Exploration

Rensselaer Polytechnic Institute
Troy, New York 12180-3590

TECHNICAL REPORTS

RELIABLE FUSION OF CONTROL AND SENSING IN INTELLIGENT MACHINES

by

John E. McInroy

Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering Department
Troy, New York 12180-3590

August 1991

CIRSSE REPORT #99

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my thanks to George Saridis for the insightful guidance he has provided and for many helpful suggestions. Above all, I would like to thank him for giving me the research freedom to explore many ideas.

I would also like to thank Steve Murphy, who helped me indirectly in many ways throughout my studies, and who directly contributed to this thesis via his PUMA dynamics programs. Deepak Sood has helped me countless times as I fumbled with computers. Kostas Kyriakopoulos has offered several suggestions which have improved the technical content of this thesis. In fact, practically all of the members of CIRSSE and RAL have in some way shared in the creation of this work.

ABSTRACT

Although robotics research has produced a wealth of sophisticated control and sensing algorithms, very little research has been aimed at reliably combining these control and sensing strategies so that a specific task can be executed. To improve the reliability of robotic systems, analytic techniques are developed for calculating the probability that a particular combination of control and sensing algorithms will satisfy the required specifications. The probability can then be used to assess the reliability of the design. An entropy formulation is first used to quickly eliminate designs not capable of meeting the specifications. Next, a framework for analyzing reliability based on the first order second moment methods of structural engineering is proposed. To ensure performance over an interval of time, lower bounds on the reliability of meeting a set of quadratic specifications with a Gaussian discrete time invariant control system are derived. A case study analyzing visual positioning in a robotic system is considered. The reliability of meeting timing and positioning specifications in the presence of camera pixel truncation, forward and inverse kinematic errors, and Gaussian joint measurement noise is determined. This information is used to select a visual sensing strategy, a kinematic algorithm. and a discrete compensator capable of accomplishing the desired task. Simulation results using PUMA 560 kinematic and dynamic characteristics are presented.

1. INTRODUCTION

As the study of robotic systems has progressed throughout the years, a large archive of special purpose algorithms suitable for various control and sensing tasks has been produced. Often several different algorithms are available which are capable of executing the same task with varying levels of performance. Use of redundant measurements, for example, improves accuracy at the cost of increased computational time. Similarly, a vision system may be capable of producing depth measurements using feature matching, point matching, focusing, structured lighting, stadimetry, etc. Control strategies may include either a PD or PID compensator for the same task, thus yielding different accuracies and response times. Classically designed compensators may be well suited for meeting a desired settling time and overshoot, while optimally designed compensators are best at meeting a quadratic cost functional.

1.1 Motivation

Although many robotic control and sensing algorithms have been derived, very little effort has been directed at ensuring and analyzing the reliability of a robotic plan which uses the algorithms to accomplish a specific task. As a consequence, it is quite possible to program an elaborate sequence of events using sophisticated control and sensing algorithms, but no formal methods have been developed for guaranteeing that the sequence will reliably produce the desired effect.

In light of the plethora of algorithms available for treating many common robotic tasks, this work does not derive new control/sensing algorithms, but rather presents a consistent and general framework for analyzing the reliability of these algorithms. As such, it contributes to the final goal of designing "intelligent machines" capable of operating in uncertain environments with minimal supervision or

interaction with a human operator. Other aspects of this subject have already been extensively developed by Saridis, who has proposed the combination of artificial intelligence, control systems, and operations research through the use of information theory [45], [60], [48].

Until recently, reliability efforts were confined for the most part to empirical "common sense engineering" approaches. For instance, to reliably grasp a component, a series of fixtures were fashioned until the component could be held securely in a specified position. After repeated experimental successes, the plan was deemed reliable. This method is time consuming and hard to generalize. Recently, several works have analyzed the reliability of particular sensing algorithms, but have still not suggested methods for combining sets of algorithms and assessing reliability. Unfortunately, increased precision is often gained at the cost of increased time, computations, complexity, etc. Consequently, a sensing algorithm often cannot be fully evaluated in itself, but must be considered in conjunction with the control system which produces actions based on the measured perceptions. Similarly, the control performance depends on the sensing statistics. In short, control and sensing are dual entities: control determines interaction with the environment while sensing observes the effect of these interactions.

1.2 Problem Definition

Given an explicit task to be executed by the intelligent machine and a set of plans, $A = \{A_1, \ldots, A_n\}$, consisting of control and sensing strategies applicable to the task, first select those plans which are potentially capable of attaining performance within the desired specifications (feasible plans). This selection procedure is a coarse and computationally efficient stochastic analysis aimed at greatly reducing the number of plans for which the explicit probability of success must be calculated. For these feasible plans (A_{feas_i}), find the reliabilities associated with the alternative

subsets of control and sensing algorithms such that the task can be accomplished to meet the set of desired specifications, $S_D = \{s_1, \ldots, s_m\}$.

At this juncture, the reliability of hardware components (power supplies, processors, sensors, actuators, etc.) is much greater than the reliability of plans to execute tasks. Therefore, hardware failures will be neglected, and the analysis will concentrate on planning faults.

1.3 Overview of the Approach

This work proposes methods for combining control and sensing within the context of robotic systems. Since concepts from reliability theory, control theory, and sensor fusion are used, we have coined the term "reliable control and sensor fusion" [33] to describe this research field. Reliable control and sensor fusion is defined as the unification of sensor information and control strategies such that an acceptable level of reliability in accomplishing a desired task is achieved.

Rather than relying on heuristic simplifications or highly constrained tasks, a general mathematical framework founded on entropy is derived. First, it is shown that reliability specifications can be mapped to a set of entropy constraints which define the precision necessary for the task at hand. Next, an algorithm which makes use of these entropy constraints is presented for selecting feasible sets of control and sensing algorithms for a given task. Since entropy is invariant with respect to homogeneous coordinate frame transformations [28], this approach simplifies the reliability analysis for physically distributed robotic systems. In addition, by using reliability and information theory, techniques are developed to provide an analytical means of evaluating the reliability of a sequence of elementary events. Moreover, by augmenting reliability theory with information theoretic concepts, simple methods of combining the reliabilities of different coordinator sub–systems are obtained. To ensure performance over an interval of time, lower bounds on the reliability of

meeting a set of quadratic specifications with a Gaussian discrete time invariant control system are derived. These concepts are validated via a detailed analysis of an extremely common robotic task–the problem of vision guided positioning to an oriented point.

1.4 Literature Review

Previous research on individual issues of reliability analysis in intelligent machines has contributed greatly to certain segments of this work. In particular, six issues relevant to this analysis are well trodden research areas.

1.4.1 Generation of Reliable Plans

The previous attempts to generate reliable robotic plans can be divided into two categories. The first school of thought attempts production of reliable plans by modeling the environment (using either probabilistic techniques or worst case geometric bounds) and, based on the model, generating a plan robust enough to reliably execute the task without errors. This thesis takes a similar approach from a stochastic viewpoint, but concentrates on analysis which will facilitate planning, rather than considering planning directly. The second school emphasizes error identification and recovery, with the assumption that reliability may be achieved by recovering from most errors. A considerable body of work has been dedicated to this topic, but most works do not pertain to this thesis. Since this thesis uses a stochastic approach, only the stochastic attempts at error identification and recovery will be reviewed.

Brooks [6] models the environment using a geometric analysis of tolerances rather than using stochastic techniques. In particular, geometric bounds on the allowable position error for a peg-in-hole insertion problem are symbolically derived. Since a symbolic expression is available, it is possible to alter geometric parameters

and see the effect on the bounds. The bound information is incorporated into an AI based "plan checker" which checks the bounds against the positioning constraints to ensure that the plan will reliably be executed. The main advantage of the method is that the symbolic expressions allow alternative plans to be investigated. Unfortunately, the symbolic expressions do become very complex even for simple geometric arrangements. In addition, the geometric approach tends to check only the worst case conditions, and thus resulting plans may be quite conservative.

Mazon and Alami [31] are also principally concerned with improving reliability in the presence of positioning uncertainty, but they make use of stochastic methods of modeling the position uncertainty. The uncertainty is modeled as a zero mean Gaussian disturbance to a homogeneous coordinate frame. The disturbance is propagated between homogeneous coordinate systems in the form of covariance statistics. The covariance is transformed between homogeneous coordinate frames using rotation and translation matrices. The uncertainty information is used off–line to build a flexible plan which uses a LISP based program to check the build up of errors and compare them to those allowed by the fixtures. If the error is too large, a search is used to find a strategy which has a greater chance of success.

Havel and Kramosil [15] also use a stochastic approach to robot plan formation for reliability improvement. The planning strategy is rigorously defined using formal language techniques. Based upon reliability estimates, searches for reliable plans are performed. The reliability functions are not obtained through a stochastic analysis, but instead are heuristically defined. For instance, one measure of reliability is defined such that it is inversely proportional to the number of iterations required for success.

Henderson [17] implicitly strives for reliability by attempting to organize the knowledge necessary to map robotic system requirements onto an appropriate assembly of algorithms, processors, sensors, and actuators. Synthesized systems which

are known to operate reliably are packaged as "logical sensors" and are then available as a system resource.

Smith and Gini [53] propose reliability improvement through error analysis and recovery. A set of sensors are used to trace the execution of the robot, and detect when an error has occurred. The nature of the sensors used in this detection are not specified. Once the cause of the error is determined, the plan is reformulated to allow "forward recovery" rather than attempting to return to the state prior to the error ("backward recovery").

Taylor and Taylor [58] also investigate error identification and recovery, but concentrate their work on error identification. The goal is the minimization of sensor processing during identification. All possible errors are enumerated, and the probability of each error is used. The paper does not discuss means of obtaining the error vector or the probabilities. Based on this information, two dual problems are treated. First, estimates of the probability that a particular sensor will detect a particular error source are found. Next, methods of determining the error source given sensor readings are developed. In a later paper [59], Bayes theorem is used to determine the error source.

1.4.2 Reliability Analysis Techniques

Reliability analysis techniques for calculating the probability of success of large systems with algebraic relationships have been developed by civil engineers for use in analyzing structural safety. These techniques have recently been extended for use in manufacturing analysis.

Shinozuka [49] reviews several structural reliability analysis techniques. Three methods are recommended: Monte Carlo simulation, use of the Stoke and Gauss divergence theorem to reduce the dimensionality of the problem, and first order second moment methods. The first order second moment methods are treated in

detail, and it is shown that the reliability index is the point most likely to fail.

Parkinson develops the first order second moment methods in a series of papers. First, the technique is derived in terms of correlated Gaussian variates, rather than uncorrelated standard normal variates [36]. In addition, an iterative method of calculating the reliability index is proposed. Next, the first four sample moments are used, along with the Johnson transformation, to obtain a multivariable standard normal distribution corresponding to the sample data [38]. For each measurement, x_i, a corresponding standard normal variate, z_i, is found. These marginally normal variates are assumed to be jointly normal. The covariance between transformed sample data points is found numerically. The statistical uncertainty in the reliability index caused by the sampling is estimated for two special cases–linear and spherical performance functions. A Bayesian based technique is suggested for updating the estimates if additional data is given.

The iterative technique of solving for the reliability index is further developed in [37]. In particular, by numerically evaluating the failure surface at a number of points, an algorithm for categorizing the failure surface as linear or spherical is proposed. Based on the categorization, the reliability may be calculated using either the normal or chi–squared distribution functions.

Parkinson applies the first order second moment method to assembly of manufactured components with given tolerance ranges [39]. In this case, a failure has occurred if the set of components will not assemble correctly. The first four sample moments and the Johnson transform are employed to obtain a set of standard normal variates for use in calculating the reliability index. A numerical example is applied to assembly of a bicycle crank.

Lee and Woo [30] continue Parkinson's application in manufacturing tolerances. Through use of Parkinson's reliability analysis of tolerances, methods of tolerance selection are found. The resulting solution employs integer programming

methods to optimize tolerance choice.

1.4.3 Stochastic Sensor Fusion

Determining the accuracy of robotic sensors holds some unique problems in that the sensor systems are often distributed physically over several different locations which may be changed. To address this issue, Smith and Cheeseman [52] propose a method of transforming the covariance statistics of three degree of freedom position measurements between homogeneous coordinate frames. The method makes use of the 3x6 Jacobian matrix between the frames, and Kalman filtering is employed to combine the measurements once they are transformed to a common coordinate system.

Durrant-Whyte [8] [10] extends the Jacobian based propagation of covariance to the full six degrees of freedom. The analysis demonstrates that orientation errors in an initial frame can cause position errors in the new frame as a function of the distance between frames. In addition, a technique is presented of describing geometric features (i.e. points, lines, planes, circles, etc.) as families of parameterized functions with probability distributions defined on the parameter vectors. These features (represented as parameter vectors) can then be transformed between coordinate frames. The methods are used to develop maximal information sensing strategies.

Durrant-Whyte [9] extends these ideas to develop a theory for integrating and propagating geometric sensor observations throughout a distributed system. The result is a two step process. First, the measurements are integrated into the system representation. Next, the measurements are propagated throughout the system to maintain consistency. Bayesian methods are used to combine measurements, with both the jointly normal and "contaminated Gaussian" distributions treated in detail. Differential homogeneous transforms are used to consistently integrate closed

kinematic chains of measurements.

1.4.4 Reliability Analysis of Individual Algorithms

Several particular algorithms for sensing and control have been analyzed for accuracy and reliability. For instance, Azadivar [3] analyzes the effect of joint positioning errors on hand accuracy using a stochastic approach, rather than bounding volumes. The errors are assumed to be zero mean, and are assumed to arise solely from random joint positioning. In other words, link and joint flexibilities, as well as link parameter errors are neglected. A "success function" is defined which is a function of hand position, desired hand position, and a vector of cost measures. The probability of success is then found by the sample mean of the success function. The method is applied to a pin-in-hole example. A rudimentary form of feedback analysis is incorporated by assuming that the uncertainty can be reduced, through feedback, by a multiplicative factor ρ ($\rho < 1$) each τ seconds.

Considerable research has been focused recently on analysis of stereo vision systems. Blostein and Huang [4] analyze the accuracy of two parallel cameras in measuring the position of a point. The quantization caused by discrete pixels is assumed to be the sole source of error, i.e. calibration and focusing imperfections are neglected. The quantization of the pixels in each image plane produce a volume of uncertainty corresponding to each pair of left and right camera pixels. It is assumed that the point's actual position is uniformly distributed within the volume of uncertainty. Based on this assumption, the corresponding distribution in the image plane is derived. Using this information, the probability that the relative range error is within a tolerance bound is calculated as a function of the disparity.

Rodriguez and Aggarwal [42] continue Blostein and Huang's analysis, but make one different assumption. Rather than assuming that the position of the point is

uniformly distributed in the volume of uncertainty, the analysis is simplified by assuming that the image position of the point is contaminated by uniformly distributed noise. This greatly simplifies the calculations at the cost of decreased model fidelity. The range error probability density is derived in terms of camera parameters.

For tractability, Lee and Kay [29] linearize the error analysis of a stereo vision system with parallel cameras. Two sources of error are considered: errors arising from Gaussian image noise, and positioning errors of the vision system. In contrast to previous papers, the analysis includes all three degrees of freedom for a point (not just range). Moreover, four points arranged like a cube and its vertices are examined simultaneously. As a consequence, both position and orientation errors of an object are determined. A Kalman filtering scheme is employed to combine a sequence of images for use in motion estimation.

Because the correspondence problems in stereo vision pose great difficulties, Grandjean and Robert de Saint Vincent [12] merge data from both a laser range finder and a stereo vision system. An extended Kalman filter is used to fuse low level information into higher level constructs (i.e. points to lines; lines to planes; etc.).

Rather than stochastically modeling the sources of error, Han [13] improves the reliability of computation using homogeneous transformations by incorporating fault tolerant linear arithmetic coding. By using various checksums, both error detection and recovery in homogeneous transforms can be realized. The method is especially applicable to parallel processing systems because a series of homogeneous transformations may be performed simultaneously on individual processors, and then transmitted to a central processor. In this manner, the fault tolerant scheme offers one method of compensating for communication errors.

1.4.5 Entropy Based System Analysis and Design

To analyze the information flows in large systems, entropy has been used by several researchers. Conant [7] models hierarchical general systems in this manner. A "partition law of information rates" is derived which shows that requirements on a system for selection of appropriate information, coordination of parts, and throughput are additive. In a related paper, Koomen [25] uses entropy to model the process of design. The model is based on Conant's partition law of information rates.

Saridis and Valavanis [48] make use of Conant's partition law for analytically designing intelligent machines. The intelligent machine is formulated as the mathematical problem of finding the right sequence of internal decisions and controls for a system structured in the order of intelligence and inverse order of precision such that it minimizes its total entropy.

Entropy concepts are used by Sanderson [44] for modeling the assembly of manufacturing parts constrained to a number of discrete positions. The model is useful for assessing product designs and quantifying the complexity of assembly procedures. Sensors are incorporated through use of conditional entropy.

In a series of papers, Kalata and Priemer derive entropy based stochastic approximation algorithms. A minimax error entropy stochastic approximation algorithm is used to estimate the state of a non-linear discrete time system in [22]. In addition, an upper bound formula for the resulting error entropy is presented. The upper bound is further employed in [24] to determine when smoothing should cease. The error entropy method is also used for system identification both with and without uncertainty [21]. The ideas are further clarified in [23].

1.4.6 Reliability Analysis of Control Systems

A large body of research exists within control theory for designing controls which are reliable in responding to uncertain plant parameters or actuator failures. Typically, a "reliable" control is defined to be a control law which remains stable even when the worst case uncertainty occurs. The control development in this thesis, on the other hand, analyzes the ability of a given control law in meeting a set of quadratic specifications when the system is perturbed by zero mean Gaussian noise. Hence the thrust is on analysis, not design, of controls. Moreover, reliability is defined as the probability of meeting a set of specifications.

In order to design control systems which are stable in the presence of failures, Viswanadham's book [62] provides detailed information. In a later work, Viswanadham [61] concentrates on fault detection and diagnosis in automated manufacturing systems. The approach is based on Petri nets.

In a series of papers ([43], [54], [40], [55], [41]) Stengel and Ray develop a procedure for estimating the stochastic robustness of a linear time-invariant system. Given a probability distribution for the uncertainties, a Monte Carlo simulation is performed to find the probability that the system will become unstable. The method can be extended to find probabilities for system characteristics other than stability.

A method for finding the class of all compensators for linear time invariant systems which will produce a desired steady state covariance is developed by Skelton , Hsieh, and Ikeda ([18], [50]). Since methods are developed for designing the stochastic parameters of the system, it may be possible to combine the reliability analysis techniques developed in this work with Skelton's design techniques. Further background information is available in [51].

1.5 Organization of the Thesis

This thesis suggests a two–stage reliability analysis. The first stage allows rapid selection of feasible plans by using entropy constraints and is presented in chapter 2. The second stage, which is outlined in chapter 3, proposes methods for calculating the probabilities of success for each feasible plan. To ensure performance over an interval of time, lower bounds on the reliability of meeting a set of quadratic specifications with a Gaussian discrete time invariant control system are derived in chapter 4. To test these concepts on problems of realistic difficulty, chapter 5 applies the methodology to visual positioning using a stadimetric vision system and a six degree of freedom manipulator. Finally, chapter 6 discusses the work and its contributions.

2. ENTROPY BASED SELECTION OF FEASIBLE PLANS

In order to synthesize control and sensing with reliability, a two step analysis is proposed. First, a set of entropy constraints is found which must be satisfied for reliable operation to occur. Those sets of control and sensing algorithms which satisfy the entropy constraints will form feasible sets of algorithms. For many applications, satisfaction of the entropy constraints alone will yield sufficient reliability for the task at hand. This is especially true when failure imposes only a small penalty. On the other hand, when a very high degree of reliability is required, a second stage of analysis which explicitly calculates the reliability (R_i) corresponding to each feasible subset (A_{feas_i}) is necessary and is explained in the next chapter.

This thesis develops a new entropy based technique which allows efficient selection of feasible plans. The more intelligent activities required to formulate the sets of algorithms are not considered. Rather, the statistical behavior of a given plan is explored and evaluated. Thus (in the context of Intelligent Machines as proposed by Valavanis and Saridis [60]) the Organizer and Coordinators first select, from the library of all possible control and sensing algorithms, subsets of algorithms potentially able to solve the given task. These subsets may well contain multiple control algorithms for applications such as gross positioning followed by fine movement. Moreover, multiple sensing algorithms may be included for utilizing several sensor subsystems. A particular subset is denoted as A_i, and may be regarded as a low level plan for executing the desired task. On the other hand, not all algorithms are included because some obviously do not apply. For instance, if position measurement is required, vision routines for object inventory are not included. For the most part, this portion of the planning utilizes logical predicates which are adroitly handled through the use of Petri net transducers [63]. In contrast, once these plans are formulated, it is still necessary to examine the statistical characteristics of the

plans, A_i, and their relationship to the set of desired specifications, S_D.

The objective of this chapter is: select from the entire space of plans $A = \{A_1, \ldots, A_n\}$ those subsets of algorithms that may reasonably be expected to attain performance within the desired specifications. These plans will be called the feasible plans, A_{feas}. This goal is achieved through an information theoretic interpretation of the desired specifications using Jaynes maximum entropy method (MEM).

2.1 An Entropy Formulation of Desired Specifications

Jaynes maximum entropy method is a technique for determining the least biased probability distribution of a probability space subject to given constraints. The least biased distribution is that distribution which maximizes the entropy subject to the given constraints [19], [20]. The MEM has been applied to many statistical inference questions, and several classes of problems have useful and tractable solutions. For instance, suppose we wish to find the density, $f(x)$, of a random variable x subject to the condition that the expected values, μ_i, of n known functions $g_i(x)$ are given. MEM analysis yields a density

$$f(x) = A \exp[-\lambda_1 g_1(x) - \cdots - \lambda_n g_n(x)] \tag{2.1}$$

where $A, \lambda_1, \cdots, \lambda_n$ are constants derived from the constraints [35]. The maximum entropy corresponding to the constraints is

$$H(x) = \lambda_1 \mu_1 + \cdots + \lambda_n \mu_n - \ln A \tag{2.2}$$

Consequently, if constraints involving expected values (i.e. moments, etc.) are known, then the MEM, through (2.1) and (2.2), provides the least biased distribution and entropy.

MEM distributions and entropies for several common sets of constraints are presented in [14] and [23]. For convenience, Table 2.1 summarizes those results.

Constraints	Distribution	Density	Entropy
Bounds, (a, b)	Uniform	$\frac{1}{b-a}$	$\ln[b-a]$
Mean (μ), Variance (σ^2)	Gaussian	$\frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{(x-\mu)^2}{2\sigma^2}]$	$\frac{1}{2}\ln[2\pi e\sigma^2]$
Bounds (a, b), Mean μ	Truncated Exponential	$\frac{\lambda}{e^{-\lambda a}-e^{-\lambda b}}e^{-\lambda x}$, $\mu = \frac{1}{\lambda} - \frac{be^{-\lambda b}-ae^{-\lambda a}}{e^{-\lambda a}-e^{-\lambda b}}$	$\lambda\mu$ $-\ln[\frac{\lambda}{e^{-\lambda a}-e^{-\lambda b}}]$

Table 2.1: Maximum Entropy distributions and entropies for several common sets of constraints

The key to unifying the reliability and information theoretic approaches lies in the MEM of Jaynes. Given the set of desired specifications, S_D, it is possible using the MEM to generate the least biased probability density function given the specifications. Moreover, the maximum entropy corresponding to the specifications is also provided by the method. For instance, if the specification is a safe velocity range, $(-v_{max}, v_{max})$, then the MEM yields a uniform distribution as the maximum entropy distribution, with $H(\text{specification}) = \ln 2v_{max}$ (Table 2.1). Similarly, if the specification is a positioning error with mean of zero and variance $\sigma^2_{e_p}$, then the maximum entropy distribution is Gaussian with $H(\text{specification}) = \ln\sqrt{2\pi e\sigma^2_{e_p}}$. The method is especially adept at modeling hard bounds due to sensor or actuator saturation. For instance, a desired force mean subject to force sensor saturation can be modeled with the truncated exponential distribution.

In effect, the set of specifications (S_D) include into the design an allowable level of uncertainty. The MEM facilitates the expression of this design uncertainty as a Shannon entropy. Once formulated in this manner, several important concepts from information theory may be invoked. For instance, information theorists have long noted that entropy is analogous to information. Consequently, the specifications needed for reliable operation may, through MEM analysis, be incorporated into information flows within an Intelligent Machine. Moreover, Saridis [48] defines *knowledge* as a form of structured information. As a result, the knowledge embodied

in the design specifications can be mathematically represented through the information theoretic approach. A method for utilizing this knowledge as a criteria for reliable control and sensor fusion is presented in the next section.

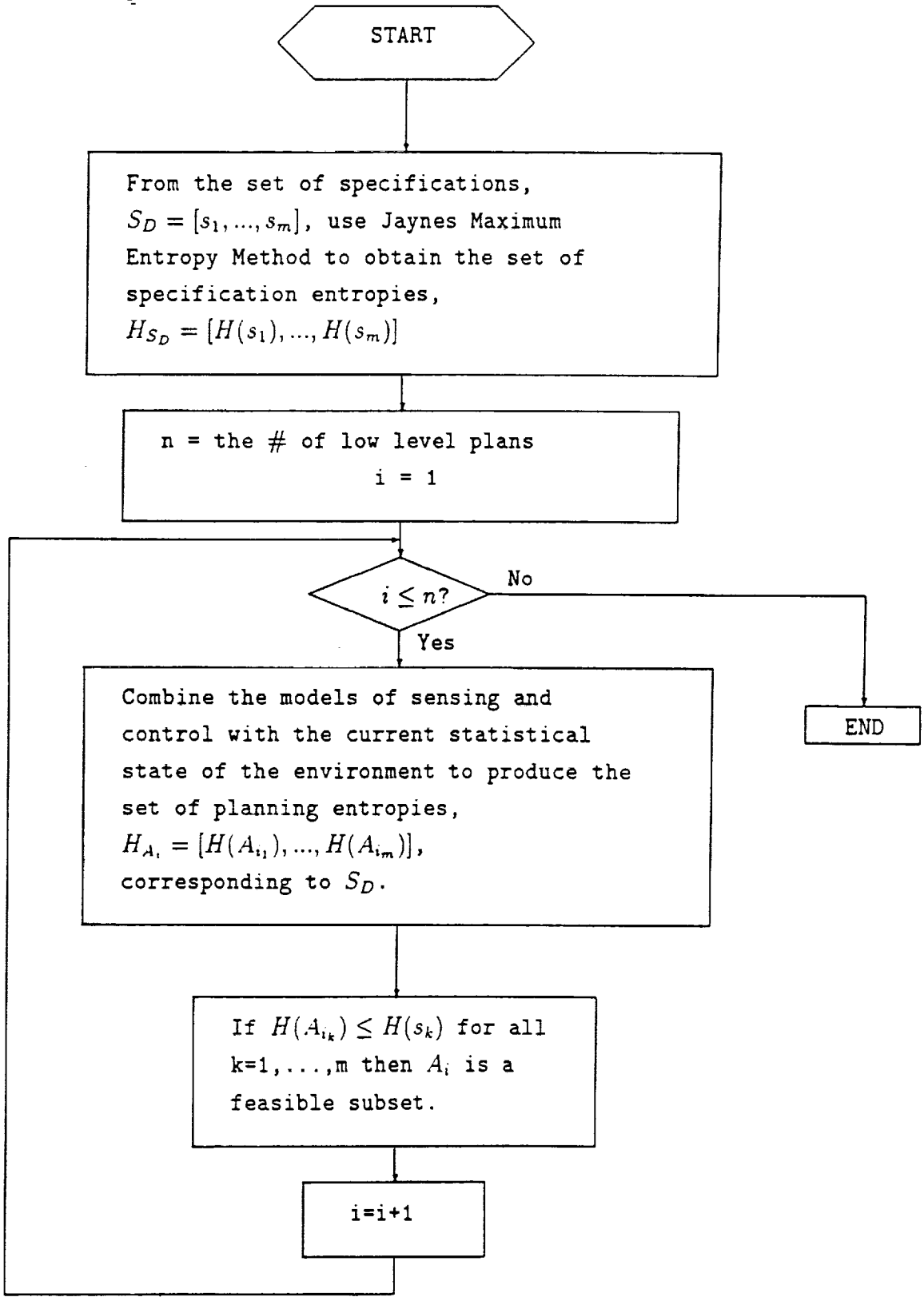
2.2 Selection of Feasible Plans

The entropy formulation of the specifications allows well developed concepts from information theory and intelligent control theory to be combined and utilized in the plan selection. By propagating the entropies of distributed sensor readings to the control coordinate frame and fusing these entropies with that of the controller, the total entropy H(control, sensing) can be compared to the maximum entropy allowed by the specifications, H(specification). If H(control, sensing) exceeds H(specification), then that set of sensing and control may not be capable of reliably meeting the specification, and will therefore not be included in the feasible algorithms, $A_{feas} = \{A_{feas_1}, A_{feas_2}, \ldots, A_{feas_l}\}$. This induces a formal definition for feasible plans:

> *Definition:* Given an explicit task to be executed, and a subset of control and sensing algorithms, $A_i = \{C_i, S_i\}$, corresponding to that task, A_i is a *feasible plan* (denoted by A_{feas_i}) if the entropy constraints
>
> $$H(A_{i_k}) \leq H(s_k) \tag{2.3}$$
>
> are satisfied for all of the specifications, s_k, $k = 1, \cdots, m$. $H(A_{i_k})$ is the entropy of A_i in responding to the k^{th} specification, and $H(s_k)$ is the uncertainty embodied in the k^{th} specification.

This definition yields a set of entropy constraints which must be satisfied to ensure reliable operation. Thus once all entropies have been determined, finding those plans which are feasible can be accomplished in a straightforward manner from the entropy constraints as depicted in Figure 2.1.

Figure 2.1: A Flowchart for Selecting Feasible Plans.



Intuitively, equation (2.3) implies that the uncertainty regarding a feasible subset's ability to meet each design criteria must be less than the permissible level of uncertainty embodied in each design specification. To be more concrete, consider what is perhaps the most common design scenario- a specification consisting of a tolerance range (a, b) and a feasible plan whose response is $A \sim N([a+b]/2, \sigma^2)$. This may represent a positioning tolerance, a desired force and safe bounds, velocity constraints, etc. For feasibility, (2.3) and Table 2.1 imply that

$$\ln \sqrt{2\pi e \sigma^2} \leq \ln[b-a] \tag{2.4}$$

or, equivalently

$$4.13\sigma \leq b-a \tag{2.5}$$

For this problem, the reliability is the probability that A stays within the tolerance range, i.e.

$$R = Pr\{a \leq A \leq b\} \tag{2.6}$$

Since the response distribution is Gaussian, the reliability is easily found:

$$R = 2\Phi\left(\frac{b-a}{2\sigma}\right) - 1 \tag{2.7}$$

where $\Phi(.)$ is the standard normal cumulative distribution function. The worst case reliability allowed by (2.5) is $4.13\sigma = b-a$ which produces $R \geq 0.96$. Thus the entropy constraints (2.3) yield at least 0.96 reliability for Gaussian responses subject to tolerance specifications. Consequently, the implicit reliability contained in the entropy constraints is sufficient to ensure satisfactory performance for this example. Lower bounds of the reliability implied by satisfaction of (2.3) for a class of distributions subject to tolerance specifications are derived in Appendix A.

Satisfying the set of entropy inequalities puts the plans through a thresholding process. As long as the thresholds are met, then a lower bound reliability is guaranteed for tolerance (Appendix A) or quadratic (Section 4.3) specifications. In

addition to the thresholding process, it is possible to order the plans according to their entropy levels. The plans with the least entropy can then be analyzed first by the reliability analysis stage. The analysis is then considered complete when a plan of sufficient reliability is found. This approach is very similar to the entropy based planning in Saridis and Valavanis's Hierarchically Intelligent Machine [46].

Note that the mean of the plan and that of the specification are identical for the examples considered. This ensures that the plan uncertainty and the specification uncertainty pertain to the same physical parameters; without this property, the unmodified entropy constraints can allow less reliable plans to be regarded as feasible. Typically, the desired specifications require that the mean value of the plan's response is equal to a particular value. In this case, if the mean of the plan is not equal to the mean of the specification, then the desired specification is not satisfied. Thus the plan is immediately excluded from the feasible set. One exception to this rule is the specification consisting only of bounds, $[a, b]$. Since a plan's response is acceptable if it occurs anywhere within the bounds, the plan's mean could conceivably be anywhere within the bounds while maintaining reliable operation. Reliable entropy constraints can be found by mapping the desired specification to a new specification which has a mean equal to the plan mean. The new specification can be found by retaining the specification bound which the response mean is closest to and matching means. For instance, suppose the i^{th} plan's mean in responding to the j^{th} specification (consisting of bounds $[a, b]$) is $\mu_{A_{ij}}$. Since the MEM indicates that the uniform distribution corresponds to this specification (Table 2.1), the mean of the specification is $(a + b)/2$. If $\mu_{A_{ij}} > (a + b)/2$, then the upper bound of the specification, b, is preserved. Next, the mean of the new specification is made equal to the mean of the plan. If $[a', b']$ denotes the new specification, then this procedure results in two equations

$$b' = b \tag{2.8}$$

$$\frac{a' + b'}{2} = \mu_{A_{ij}} \tag{2.9}$$

These two equations are solved simultaneously to produce the new specification, $[a', b'] = [2\mu_{A_{ij}} - b, b]$. Similarly, if $\mu_{A_{ij}} < (a + b)/2$, then the new specification is $[a', b'] = [a, 2\mu_{A_{ij}} - a]$. The technique is applied in the case study to the timing specification.

2.3 Advantages of the Information Theoretic Approach

The information theoretic approach to reliable control and sensor fusion holds several advantages over other methods of analysis. These advantages can be divided into three categories. First, the entropy associated with many sensed features useful in robotics is invariant with respect to coordinate frame transformations. Second, since the MEM is a well developed tool of statistical inference, it provides exact methods of handling uncertainty for any distribution. Third, because entropy can be interpreted as information, it provides a consistent representation throughout all levels in a hierarchically Intelligent Machine.

First, consider entropy in the presence of coordinate transformation. The joint entropy of many diverse types of measurements are invariant with respect to coordinate frame transformations [28]. This is easily shown by noting that the joint entropy of a random vector $x_j = g(x_i)$ is found by [35]:

$$H(x_j) = H(x_i) + E\{\ln |det(J)|\} \tag{2.10}$$

where J is the Jacobian of the uniquely invertible transformation $g(\cdot)$, $J = \nabla_x g(x)$. For coordinate transformations, $x_j = g(x_i) = T_i^j x_i$ ([11]) where x_j is a point represented in the j^{th} coordinate frame, x_i is a point represented in the i^{th} frame, and

$$T_i^j = \begin{bmatrix} n & o & a & q \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & q \\ 0 & 1 \end{bmatrix} \tag{2.11}$$

where n, o, a, and q are vectors, and R is an orthonormal rotation matrix. The Jacobian of this transformation is simply $J = T_i^j$. Since R is orthonormal, $det(J) = det(R) * 1 = 1$. Substitution into (2.10) then yields $H(x_j) = H(x_i)$. This fact was first noted by Kyriakopoulos [28]. Kyriakopoulos also found that the result could be generalized far beyond mere 3-D points to include parameter vectors of several common geometric features: lines, planes, and spheres. In this formulation, a geometric feature such as a line is represented as a point (parameter vector) in parameter space following Durrant-Whyte [8]. The joint entropy of the entire parameter vector is then invariant with respect to coordinate frame transformations. The transformations for several useful entropy invariant parameter vectors are listed in Table 2.2.

The method can also be extended to include six dimensional quantities such as oriented points in $\Re^3$ or force vectors. If $p_i = \Gamma_j^i(p_j)$ represents the transformation of the oriented point p_j in frame j to frame i, then J has a highly structured form [8]:

$$J = J_j^i = \begin{bmatrix} R^T & 0 \\ M & R^T \end{bmatrix} \tag{2.12}$$

where R is the rotation matrix previously defined, the oriented points are represented as 6x1 vectors with the orientation stacked on top of the position, and

$$M = \begin{bmatrix} qxn & qxo & qxa \end{bmatrix}^T \tag{2.13}$$

This implies that $det(J) = det^2(R) = 1$. Thus $E\{\ln|det(J)|\} = 0$ and from (2.10) the joint entropy is again invariant with respect to coordinate frame transformations.

This invariance with respect to coordinate frame transformations is very important when comparing and combining sensor readings distributed over many different coordinate frames– a very common situation in advanced robotic systems. To illustrate, cameras are often mounted on gimbals or on a link of the manipulator.

Feature	Equation	Parameters	Transformation
Point		$p = [\text{x y z } 1]^T$	$p_j = T_i^j p_i$
Line	$r(t) = r_0 + td$	$p = [\ d^T\ r_0^T\ 1]^T$	$p_j = \begin{bmatrix} R_i^j & 0 \\ 0 & T_i^j \end{bmatrix} p_i$
Plane	$n^T(r - r_0) = 1$	$p = [r_0^T\ 1\ n^T]^T$	$p_j = \begin{bmatrix} T_i^j & 0 \\ 0 & R_i^j \end{bmatrix} p_i$
Sphere	$\| r - c \| = d$	$p = [c^T\ 1\ d]^T$	$p_j = \begin{bmatrix} T_i^j & 0 \\ 0 & 1 \end{bmatrix} p_i$
Circle	$n^T(r - r_0) = 0$ $\cap \| r - c \| = d$	$p = [n^T\ r_0^T\ 1\ c^T\ 1\ d]^T$	$p_j = \begin{bmatrix} R_i^j & 0 & 0 & 0 \\ 0 & T_i^j & 0 & 0 \\ 0 & 0 & T_i^j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} p_i$
Oriented Point		$p = [\theta_x\ \theta_y\ \theta_z\ \text{x y z}\ ]^T$	$p_j = \Gamma_i^j(p_i)$ $J = J_i^j$
Force		$p = [\tau\ F]^T$	$p_j = J_j^{iT} p_i$

Table 2.2: Features which display coordinate frame entropy invariance

Similarly, mobile robots perceive the environment from constantly changing positions. Even if the sensor itself is immobile, the relationship of the sensor with respect to the end effector's coordinate frame will vary as the manipulator position changes. In addition, often several different sensors are distributed throughout the workcell. For instance, a laser range finder often complements a stereo vision system. The efficient fusion of such measurements is of great importance.

Since modern robotic systems contain many sensors distributed over a variety of different physical locations which may change with time, transforming these diverse sensor readings to a common coordinate frame has posed one of the fundamental issues of sensor fusion research. Recently, attention has been focused on transforming both the sensed measurement and the uncertainty regarding that measurement, as the uncertainty may be used in the fusion of sensor readings. Most authors have expressed this uncertainty through the covariance matrix [8], [52]. A more compact and useful representation of the uncertainty is Shannon's entropy.

Shannon's definition of entropy has gained wide acceptance because it complies with a heuristic understanding of uncertainty and provides a mathematical framework to express this uncertainty [35]. Moreover, entropy can be used in estimation to replace covariance statistics [23], [22].

In contrast, the covariance matrix varies under coordinate frame transformations. Propagation of the covariance matrix can be approximately calculated as [8]:

$$C_j = J_i^j C_i J_i^{j^T} \tag{2.14}$$

where C_j is the covariance in the j^{th} frame, J_i^j is the Jacobian matrix of the coordinate transformation defined in equation (2.12), and C_i is the covariance in the i^{th} frame. As a result, in order to evaluate the performance of a distributed sensor in the control frame using covariance analysis, it is first necessary to calculate the Jacobian between those frames, and then find the propagated covariance using (2.14). In contrast, since the joint entropy is invariant with respect to coordinate frame transformations, it is not necessary to find the Jacobian when using entropy based analysis. Because many different measurements are often available at a variety of coordinate frames, this simplification makes comparison between alternative sensing strategies much more computationally tractable. In all fairness, it must be stated that the complete covariance matrix is sometimes a useful quantity to be propagated in its own right. This is especially true when the desired specifications concern individual elements of the feature's parameter vector. However, even in these cases, the information theoretic approach still holds several important advantages.

To be explicit, the information theoretic approach is advantageous because it is a well developed statistical inference technique. As a result, the wealth of previous knowledge contained in information theory literature may be immediately drawn upon. For instance, the entropy formulation expresses *uncertainty* in a meaningful

fashion for any probability distribution. In contrast, propagating merely the covariance is sufficient in expressing the uncertainty only for certain distributions such as those listed in [23]. Other distributions require higher order moments for complete analysis of the uncertainty. In addition, Kalata [23] notes that

> A significant result obtained from the characterization of communication systems using information theory is that performance bounds can be established independently of coding procedures. A similar result is possible when information theory is applied to the general estimation problem: independent of the estimation procedure, a performance bound can be specified.

Thus, when fusing sensor measurements, the entropy scheme allows determination of the estimation performance bounds. This knowledge is useful for determining sensing strategies of sufficient accuracy.

The final advantage which the unification of reliability and entropy concepts presents is the consistent representation of uncertainty throughout all levels of a hierarchically Intelligent Machine. Since entropy can be regarded as information, it is a sufficient analytic measure that unifies the treatment of all the levels of an intelligent machine [48]. This implies, for instance, that, in an impasse, the higher level Organizer may re-design the problem, thus generating new specifications for the lower level reliable control and sensor fusion. An iterative design procedure is then possible in the spirit of Koomen [26]. In addition, since entropies are produced for each feasible subset and each specification, the techniques have great potential for integration with other well established results of Intelligent Control. For instance, if a positioning specification is phrased in terms of a maximum integral error, then it may be possible to make use of entropy formulations of optimal control [45].

3. RELIABILITY ANALYSIS OF FEASIBLE PLANS

When a high degree of reliability must be assured, a second stage of analysis which explicitly calculates the reliability (R_i) corresponding to each feasible subset (A_{feas_i}) is necessary in addition to the entropy based elimination procedure. Just as the specifications play an integral role in the determination of feasible plans, S_D also greatly influences R_i. The set of specifications may include constraints on total execution time, positioning accuracy, maximum overshoot, robustness, tracking errors etc. Each element of the specifications, s_k, represents a desired characteristic to be achieved during task execution, while R_i incorporates all of these criteria into a single term and measures the probability that they will all be satisfied. This measurement is accomplished by first calculating the reliability of the plan in responding to each separate specification (R_{ik}, $k = 1, \ldots, m$). Next, the individual reliability terms are combined to form R_i.

3.1 Calculation of Reliability Terms

The individual reliability terms (R_{ik}) can be found by first defining reliability performance functions (RPF), g_{ik}, associated with each feasible subset, A_{feas_i}, and each specification, s_k. Often, this is the most arduous phase of the reliability analysis, as it requires capturing the probabilistic behavior of a particular algorithm and expressing it in terms of the desired specifications. The performance functions should be defined in such a manner that

$$[g_{ik}(x) > 0] \Rightarrow success \tag{3.1}$$

$$[g_{ik}(x) \leq 0] \Rightarrow failure \tag{3.2}$$

where x is a vector of state variables.

Once each RPF is defined, the statistics associated with the current state of the environment can be used to find the reliability, R_{ik}, of the particular subset A_{feas_i} in meeting the desired specification s_k. If the form of the underlying distribution for the state variables is known (based on prior experience), while the distribution parameters must be found from statistical sampling, then maximum likelihood estimation may be applied.

Maximum likelihood estimation is useful because under certain regularity conditions it possesses several compelling advantages. The regularity conditions are not very restrictive. In short, if samples $\{x_1, x_2, ..., x_k\}$ are taken from an underlying cumulative distribution function $P(x;e)$, where e is a distribution parameter to be estimated, then the maximum likelihood estimate of e, $\hat{e}$, meets the regularity conditions if $P(x;e)$ is regular with respect to its first two e derivatives (directional derivative) and $\hat{e}$ is unique. $P(x;e)$ is *regular with respect to its first e derivative* if

$$E\{S(x;e)\} = \frac{\partial}{\partial e}\int_{-\infty}^{\infty} dP(x;e) = 0 \tag{3.3}$$

where

$$S(x;e) = \frac{\partial}{\partial e}\log dP(x;e) \tag{3.4}$$

$P(x;e)$ is *regular with respect to its second e derivative* if

$$E\{S'(x;e)\} + E\{S(x;e)\}^2 = \frac{\partial^2}{\partial e^2}\int_{-\infty}^{\infty} dP(x;e) = 0 \tag{3.5}$$

where

$$S'(x;e) = \frac{\partial}{\partial e}S(x;e) \tag{3.6}$$

Consult [64] or [65] for a rigorous treatment of maximum likelihood estimation. Under these conditions, the maximum likelihood estimate (MLE) is asymptotically normal, consistent, and asymptotically efficient. Furthermore, for a function of e, $f(e)$, the MLE is given by the invariance property to be $f(\hat{e})$ [65]. The distribution of the function is

$$\hat{f}(\hat{e}) \sim AN(f(\hat{e}), \nabla_e^T f(\hat{e}) Cov(\hat{e}) \nabla_e f(\hat{e})) \tag{3.7}$$

where

- $\hat{f}(\hat{e})$ is the MLE of $f(\hat{e})$,
- AN indicates asymptotic normality,
- $\nabla_e f(\hat{e})$ is the gradient of $f(\hat{e})$ with respect to $\hat{e}$, and
- $Cov(\hat{e})$ is the covariance matrix of $\hat{e}$.

Consequently, if the regularity conditions are satisfied for the underlying distribution, then from the sample ensemble $\{x_1, x_2, ..., x_k\}$ it is possible by maximizing the likelihood and using (3.7) to find an asymptotically normal maximum likelihood distribution for $g_{ik}(x)$. Since the distribution is asymptotically normal, it is a simple matter to then find the reliability corresponding to the RPF.

If maximum likelihood estimation cannot be used, a lower bound on R_{ik} may be found from the reliability index. A lower bound is extremely useful, as it provides a guaranteed minimum level of reliability. The disadvantage, of course, is that the lower bound can be a conservative estimate of the actual reliability. The *reliability index*, β, is defined as the minimum distance between the origin of a set of uncorrelated standard normal variates (derived from x), and the failure surface , $g(x) = 0$.

Consider the case in which the individual state variables, denoted now as x_i, are uncorrelated, Gaussian random variables. The variables can be replaced by a set of reduced variates with the transformation:

$$x_i' = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}}; i = 1, 2, \ldots, n \tag{3.8}$$

where μ_{x_i} is the expected value of the i^{th} random variable, and σ_{x_i} is the standard deviation of the i^{th} random variable. To estimate the reliability, Shinozuka [49] has shown that the point on the failure surface ($g(x) = 0$) with the minimum distance to the origin of the reduced variates, x', is the most probable failure point. If $g(x)$ is a

nonlinear function, the reliability index may be used as an approximate measure of reliability by placing a lower bound on the reliability. For linear $g(x)$, the reliability can be found exactly from the reliability index.

Determining the reliability index, β, may require iterative methods for nonlinear performance functions. In contrast, linear performance functions have a closed form solution. Suppose that the performance function is represented as:

$$g(x) = a_0 + \sum_i a_i x_i \tag{3.9}$$

where the a_i's are constants. Note that the reliability performance function may be negative, although this possibility corresponds to a system failure. The minimum distance to the origin of the reduced variates is then [1]:

$$\beta = \frac{a_0 + \sum_i a_i \mu_{x_i}}{\sqrt{\sum_i (a_i \sigma_{x_i})^2}} \tag{3.10}$$

For linear performance functions, the reliability can be found directly from the reliability index by the formula

$$R = \Phi(\beta) \tag{3.11}$$

where $\Phi(.)$ is the normalized, zero mean, Gaussian cumulative distribution function and R is the reliability. The minimum distance for nonlinear RPFs can be found by iteratively searching for the minimum of

$$\beta = \frac{-\nabla g^T x'}{\sqrt{(\nabla g^T \nabla g)}} \tag{3.12}$$

subject to

$$g(x') = 0 \tag{3.13}$$

where ∇g is the gradient vector of $g(x)$ with respect to x', and T denotes the transpose. Parkinson [37] offers a method for efficient solution to the minimization.

If $g(x)$ is concave toward the origin of the reduced variates, then [39]

$$\chi_n^2(\beta^2) \leq R \leq \Phi(\beta) \tag{3.14}$$

On the other hand, if $g(x)$ is convex toward the origin, then

$$\Phi(\beta) \leq R \leq 1 \tag{3.15}$$

where $\chi^2_n(.)$ is the chi-squared distribution function with n degrees of freedom. If the RPF is neither convex nor concave, (3.14) may always be used as a lower bound on the reliability. Figure 3.1 graphically depicts the bounds for a two dimensional problem.

Should the variables be correlated, the solution is still possible. However, the covariance matrix must be used [49]. For non–Gaussian variates, Ang [2] and Parkinson [39] explain techniques for transforming the variates to an equivalent Gaussian system.

3.2 Combination of Reliability Terms

Once the reliabilities in meeting each individual specification have been found, the ability of the potential algorithms (A_{feas_i}) in meeting the desired specifications, S_D, depends on the relationships between the elements of S_D. In the majority of cases, the elements s_k will have a series relationship, i.e. the success in meeting requirements will depend on all specifications being satisfied. Occasionally, the elements may have parallel relationships. For instance, it may be desired to either meet a desired overshoot, *or* a desired execution time. Since meeting either criteria is termed a success, the relationship is parallel.

To allow ease in combining reliabilities, as well as to accommodate information flows within Intelligent Machines, a new concept is now defined. The *reliability self information* (RSI), $I(R)$, is

$$I(R) = -\log R \tag{3.16}$$

, while the *failure self information* is

$$I(F) = -\log F \tag{3.17}$$

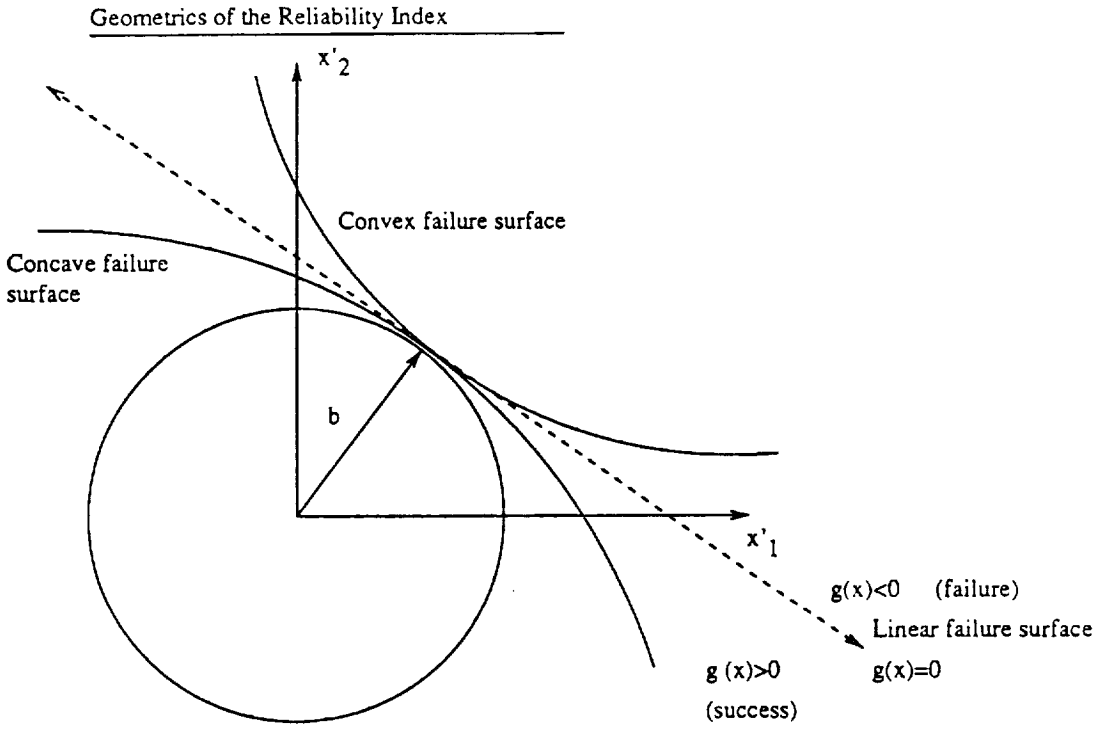


Figure 3.1: A Geometric Picture of the Reliability Index for Two Dimensions

where F is the probability of failure, i.e.

$$F = 1 - R \tag{3.18}$$

The base of the logarithm is arbitrary, as different bases will simply add a constant bias. Since these quantities are measures of self information, they enjoy many of the same properties which entropies do. For instance, multiplication can be replaced by addition, etc. The information theoretic setting makes RSI easy to interpret in terms of Intelligent Control as proposed by Saridis [48]. The flow of knowledge resulting from the reliability self information complies with the principle of Increasing Precision with Decreasing Intelligence [46].

Once the reliabilities R_{ik} have been found, it is a simple matter to find the corresponding RSI's, $I(R_{ik})$ from (3.16). It is also a simple matter to find the RSI of the entire subset if its elements are in a series relationship. The reliability of a series system (assuming independence) is [2]

$$R_s = R_1 R_2 \cdots R_n \tag{3.19}$$

where R_i denotes the reliability of the i^{th} component. The RSI for a potential subset of algorithms A_{feas_i} corresponding to n specifications (s_k) in a series relationship is found by combining (3.19) and (3.16):

$$I(R_i) = \sum_{k=1}^{n} I(R_{ik}) \tag{3.20}$$

Thus multiplication is replaced by addition when using RSI terms. On the other hand, if the set of specifications, S_D, contains parallel relationships, the parallel relationships must be simplified to one series term before (3.20) can be used. The reliability of a parallel arrangement (assuming independence) is:

$$R_s = 1 - \prod_i (1 - R_i) \tag{3.21}$$

Rewriting (3.21) in terms of failure probabilities, we find

$$F = \prod_i F_i \tag{3.22}$$

The total failure self information of a parallel connection can be found from (3.17) and (3.22)

$$I(F) = \sum_i I(F_i) \tag{3.23}$$

As before, addition replaces multiplication. The RSI can be found from the failure self information by the simple formula

$$I(R) = -\log(1 - \exp[-I(F)]) \tag{3.24}$$

The RSI found from (3.24) and (3.23) is the equivalent RSI of the parallel set of specifications, so it can be combined with specifications in series with it using (3.20).

Naturally, once the RSI, $I(R_i)$, is found for each subset of feasible algorithms, A_{feas_i}, the performance of the alternate plans can easily be compared. The smaller the RSI, the more reliable is the plan. In a more qualitative sense, the set of desired specifications operates on the feasible algorithms to produce a set of reliability performance functions. The environment then operates on the performance functions to produce a reliability self information (Figure 3.2). Figure 3.3 outlines the reliability analysis procedure in a flowchart form.

To clarify these concepts further, the case study performs this analysis in detail for a robotic problem of practical significance.

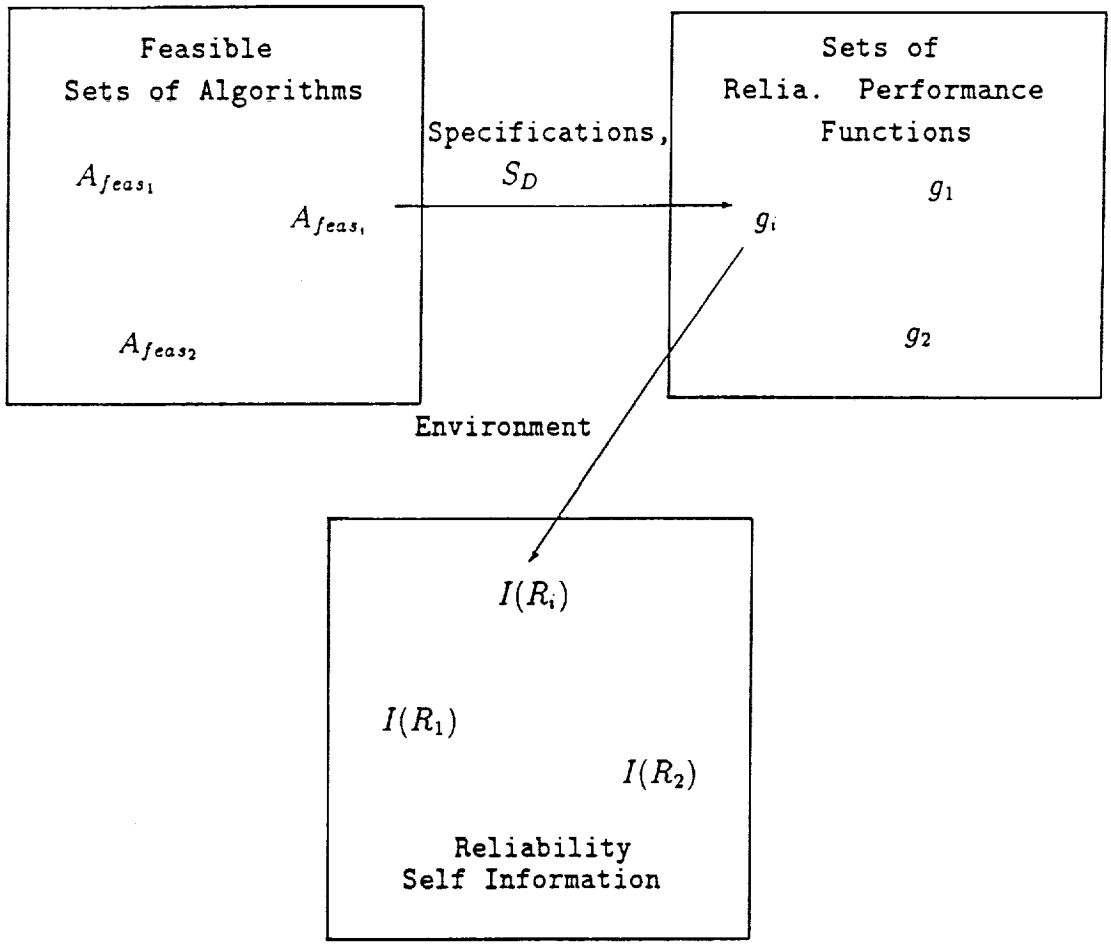


Figure 3.2: A Pictorial Description of the Reliability Analysis Procedure

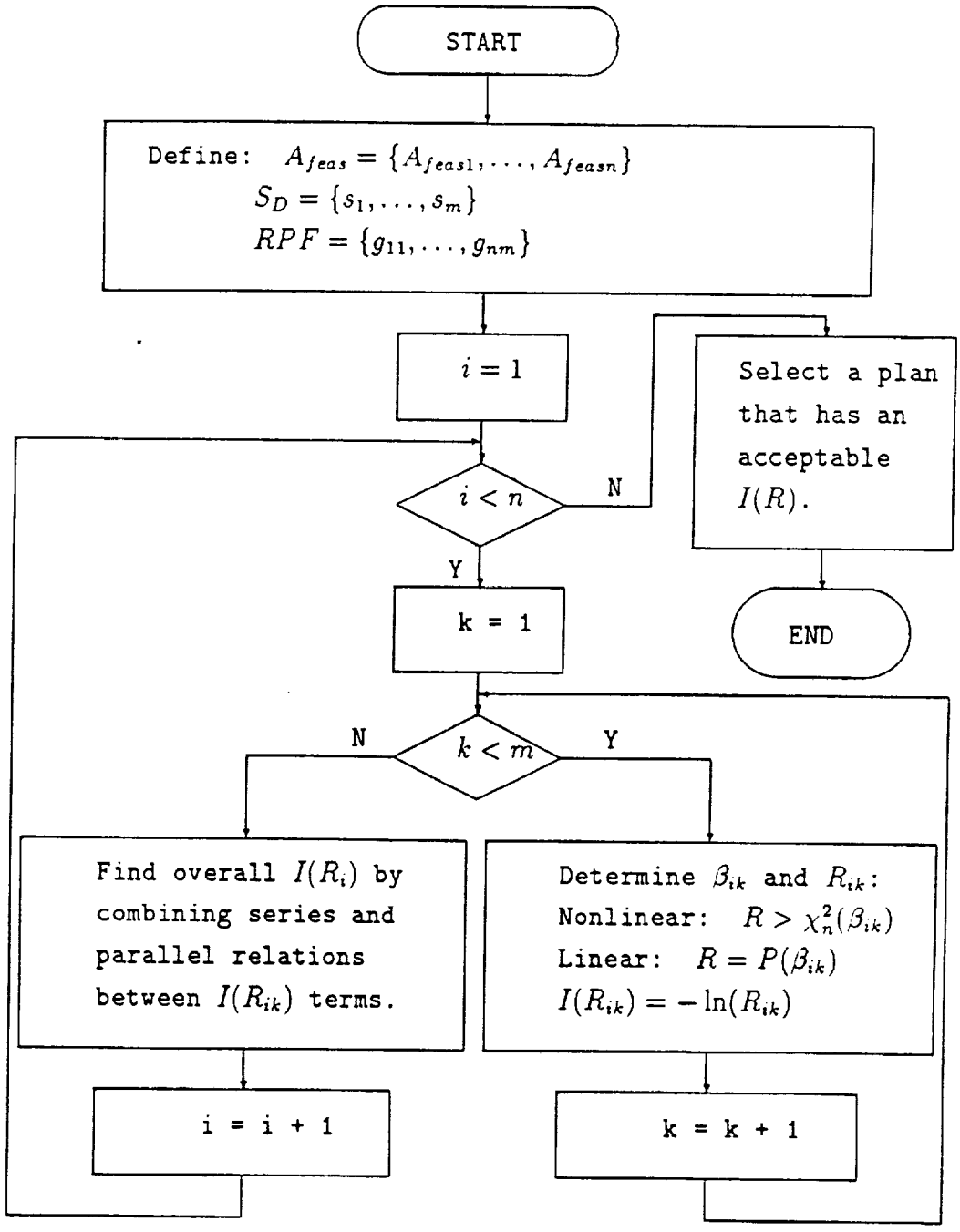


Figure 3.3: The Reliability Analysis Procedure

4. RELIABILITY ANALYSIS OF DISCRETE TIME-INVARIANT CONTROL SYSTEMS

The reliability analysis techniques developed herein have focused on the types of specifications traditionally found in manufacturing environments, i.e. tolerance ranges, mean and variance criteria, etc. These specifications constrain principally the final state and extreme states, thus they can be met successfully with classical control tools such as steady state error and overshoot. By contrast, often the behavior of a system must be ensured over a time segment. Moreover, in robotics it is very important to be able to specify the behavior of a multidimensional system.

The performance of a system over a time interval is measured in optimal control theory as the value (termed *cost*) of a functional defined over that interval (termed *the performance function*). If the performance function is defined to be the integral of a quadratic function of the state variables and the system is linear, then the control which maximizes performance (as defined by the performance function) can easily be found.

By using nonlinear feedback, Tarn [57] has proved that robotic manipulators may be viewed as linear time invariant systems. Consequently, if a quadratic performance function is defined, then the optimal control for the i^{th} sensing system can be found and the expected cost corresponding to that control and sensing strategy can be evaluated. Although the expected cost can be calculated, techniques have not been previously developed to determine the reliability of a control system in meeting a set of specifications. This chapter derives a lower bound reliability in meeting quadratic constraints on the error of discrete time invariant control systems.

4.1 Problem Statement

The notation used will be very similar to that employed by Hsieh and Skelton [18], as the concepts developed here might be used in conjunction with their ideas to form a powerful design technique.

The time invariant discrete plant is denoted

$$x_p(k+1) = A_p x_p(k) + B_p u(k) + D_p w(k) \tag{4.1}$$

$$z(k) = M_p x_p(k) + E_p v(k) \tag{4.2}$$

where $x_p \in \Re^{n_x}$, $u \in \Re^{n_u}$, $z \in \Re^{n_z}$ are state, input, and measurement vectors, while $w(k) \in \Re^{n_w}$ and $v(k) \in \Re^{n_v}$ are zero mean white Gaussian noise sources with covariance matrices $W(k)$ and $V(k)$, respectively. The plant is controlled with a dynamic compensator of a specified order n_c

$$x_c(k+1) = A_c x_c(k) + B_c z(k) \tag{4.3}$$

$$u(k) = C_c x_c(k) + D_c z(k) \tag{4.4}$$

The initial condition of x_p and x_c is assumed to either be deterministically known or normally distributed.

In order for operation of the system to be considered successful, the following set of weighted square norm specifications must be satisfied:

$$s_k : \; x^T(k) Q(k) x(k) \leq \epsilon^2, \; k = 1, \ldots, N \tag{4.5}$$

where $x \in \Re^{n_p + n_c}$ and

$$x = \begin{bmatrix} x_p \\ x_c \end{bmatrix} \tag{4.6}$$

The equations describing the evolution of x are, from (4.1-4.4)

$$x(k+1) = A_{cl} x(k) + D_{cl} v_l(k) \tag{4.7}$$

where $v_l(k) = [w^T(k)\ v^T(k)]^T$,

$$A_{tl} = \begin{bmatrix} A_p + B_p D_c M_p & B_p C_c \\ B_c M_p & A_c \end{bmatrix} \tag{4.8}$$

$$D_{tl} = \begin{bmatrix} D_p & B_p D_c E_p \\ 0 & B_c E_p \end{bmatrix} \tag{4.9}$$

4.2 Reliability Bound for Weighted Square Norms of Zero Mean Gaussian Vectors

In order to assess the reliability of meeting the quadratic specifications (4.5), a general bound on the reliability of meeting a single weighted square norm will first be derived. This result is in itself useful, as many applications require the evaluation of only a single weighted norm. The analysis is performed by finding the reliability of meeting a very special weighted norm.

Theorem 1 *If* $x \sim N(0, C_x)$, $x \in \Re^n$, *then*

$$P\{x^T C_x^{-1} x \leq \epsilon^2\} = \chi_n^2(\epsilon^2) \tag{4.10}$$

where $\chi_n^2(\cdot)$ *is the chi-squared cumulative distribution function with* n *degrees of freedom.*

Proof:

Since C_x is a covariance matrix, C_x^{-1} is symmetric and positive definite. Therefore, it can be uniquely factored into the Cholesky decomposition. $C_x^{-1} = C^T C$ [5]. Let $y = Cx$. Then

$$E\{y\} = 0 \tag{4.11}$$

$$Cov(y) = CC_xC^T = C(C^TC)^{-1}C^T \tag{4.12}$$

where $E\{\cdot\}$ denotes the expectation operator, and $Cov(\cdot)$ denotes covariance. Then

$$Cov(y)C = C(C^TC)^{-1}C^TC = C \tag{4.13}$$

Therefore $Cov(y) = I$, the nxn identity matrix. Linear transformations of Gaussian variates are also Gaussian [35], thus

$$y \sim N(0, I) \tag{4.14}$$

The weighted norm can be rewritten

$$x^T C_x^{-1} x = x^T C^T C x = y^T y = y_1^2 + y_2^2 + \cdots + y_n^2 = \chi^2 \tag{4.15}$$

The elements of y are uncorrelated from (4.14), but this implies they are independent for Gaussian variates. Hence (4.15) is a sum of n squared, independent, standard normal variates. From [35], this implies that χ^2 has a chi-squared distribution with n degrees of freedom. Hence

$$P\{x^T C_x^{-1} x = y^T y \leq \epsilon^2\} = \chi_n^2(\epsilon^2) \ \surd \tag{4.16}$$

Thus if the weighting matrix is equal to the inverse of the covariance matrix, then the reliability can immediately and easily be found from tables of the chi-squared distribution. Of course, in general, the weighting matrix used to form a specification arises from physical needs and will not be related to the inverse covariance matrix. For instance, suppose that the vector x is a 6 dimensional pose error in a robotic system, with the first three elements representing the orientation and the last three elements representing the x,y,z position. The task at hand may require high positioning accuracy in the "x" direction, while the other directions and orientations are much less important. Then the weighting matrix, Q, should reflect this need for accuracy. Fortunately, Theorem 2 uses Theorem 1 to derive conditions under which a lower bound on reliability can easily be found for a class of weighting matrices Q. As such, Theorem 2 is a major contribution of this work.

Theorem 2 *If $x \sim N(0, C_x)$, $x \in \Re^n$, and $C_x^{-1} - Q \geq 0$ (the difference between C_x^{-1} and Q is positive semi-definite) then*

$$P\{x^T Q x \leq \epsilon^2\} \geq \chi_n^2(\epsilon^2) \tag{4.17}$$

Proof:

The matrix $C_x^{-1} - Q$ is positive semi-definite, therefore $x^T(C_x^{-1} - Q)x \geq 0$ or $x^T C_x^{-1} x \geq x^T Q x$. By Theorem 1, $P\{x^T C_x^{-1} x \leq \epsilon^2\} = \chi_n^2(\epsilon^2)$. But if x is inside the hyperellipsoid defined by $x^T C_x^{-1} x \leq \epsilon^2$, then the inequality $x^T Q x \leq \epsilon^2$ is also true. Thus

$$P\{x^T Q x \leq \epsilon^2\} \geq P\{x^T C_x^{-1} x \leq \epsilon^2\} \geq \chi_n^2(\epsilon^2) \ \surd \tag{4.18}$$

Theorem 2 is useful for analyzing many robotic tasks. For instance, suppose statistics for the error due to imperfect kinematic transformations are known. Then a lower bound on the probability of meeting a desired weighted square positioning norm can be found.

4.3 Entropy Constraints for Selection of Feasible Plans

Entropy has been proposed as a computationally efficient method of selecting feasible plans in a robotic system [34]. This section extends the technique to handle specifications consisting of a weighted square l_2 norm of a zero mean Gaussian random vector, and shows that the resulting entropy constraint yields a reliability which depends only on the dimension of the weighted vector.

Feasible plans are defined in terms of a set of entropy inequalities via (2.3). This definition yields a set of entropy constraints which must be satisfied to ensure reliable operation.

In order to use (2.3), the entropy of both the plan (simply a zero mean normal response for the case considered now) and the specification (a weighted square l_2 norm) must be found. The specification entropy is found in accordance with [34] by using Jaynes Maximum Entropy Method. In order to find the maximum entropy distribution, Theorems 3 and 6 have been developed.

Theorem 3 *Given the constraint that a random variable $x \in \Re^n$ lies within the*

hypersphere $x^T x \leq \epsilon^2$, the Jaynes maximum entropy distribution is given by

$$f(x) = \begin{cases} 1/V_n(\epsilon) & , x^T x \leq \epsilon^2 \\ 0 & , otherwise \end{cases} \tag{4.19}$$

with entropy

$$H(x) = \ln V_n(\epsilon) \tag{4.20}$$

where $V_n(\epsilon)$ is the hypervolume of a n-dimensional hypersphere of radius ϵ.

Proof:

If $f(x)$ and $g(x)$ are two arbitrary probability densities, then [35]

$$-\int_{-\infty}^{\infty} g(x) \ln g(x) dx \leq -\int_{-\infty}^{\infty} g(x) \ln f(x) dx \tag{4.21}$$

Assume that

$$f(x) = \begin{cases} 1/V_n(\epsilon) & , x^T x \leq \epsilon^2 \\ 0 & , \text{otherwise} \end{cases} \tag{4.22}$$

then the entropy of x is

$$H(f(x)) = -\int_{x^T x \leq \epsilon^2} \ln[V_n(\epsilon)]^{-1} / V_n(\epsilon) dx = \frac{\ln[V_n(\epsilon)]}{V_n(\epsilon)} \int_{x^T x \leq \epsilon^2} dx = \ln[V_n(\epsilon)] \tag{4.23}$$

Now suppose $g(x)$ is any other density such that

$$\int_{x^T x \leq \epsilon^2} g(x) dx = 1 \tag{4.24}$$

Then, by (4.21),

$$\begin{aligned} H(g(x)) &= -\textstyle\int_{x^T x \leq \epsilon^2} g(x) \ln g(x) dx \quad \leq -\displaystyle\int_{x^T x \leq \epsilon^2} g(x) \ln[V_n(\epsilon)]^{-1} dx \\ &= \ln V_n(\epsilon) \textstyle\int_{x^T x \leq \epsilon^2} g(x) dx \quad = H(f(x)) \end{aligned} \tag{4.25}$$

Thus the entropy corresponding to any distribution other than $f(x)$ is less than or equal to the entropy obtained by $f(x)$, and $f(x)$ is therefore the maximum entropy distribution. $\surd$

Theorem 3 can easily be generalized to hyperellipsoids, rather than spheres.

Lemma 4 *Given the constraint that a random variable $x \in \Re^n$ lies within the hyperellipsoid $x^T Q x \leq \epsilon^2$, $Q \geq 0$, the Jaynes maximum entropy distribution is given by*

$$f(x) = \begin{cases} 1/V_n(\epsilon) & , x^T Q x \leq \epsilon^2 \\ 0 & , otherwise \end{cases} \tag{4.26}$$

with entropy

$$H(x) = \ln V_n(\epsilon) \tag{4.27}$$

where $V_n(\epsilon)$ is the hypervolume of a n-dimensional hyperellipsoid of radius ϵ, $x^T Q x \leq \epsilon^2$.

The proof is identical to Theorem 3, except all integrals are taken over the hypervolume $x^T Q x \leq \epsilon^2$

Lemma 5 *Given two random vectors, $x \in \Re^n$ and $y \in \Re^n$, subject to the constraints, $x^T Q x \leq \epsilon^2$, $y^T P y \leq \epsilon^2$, $Q \geq 0$, $P \geq 0$, $P - Q \geq 0$, then*

$$H(y) \leq H(x) \tag{4.28}$$

where $H(y)$ and $H(x)$ are the entropies found from Jaynes MEM for y and x, respectively.

Proof:

Since both Q and P are positive semi-definite, x and y are constrained to be within hyperellipsoids. $P - Q \geq 0$ implies that $x^T Q x \leq x^T P x \leq \epsilon^2$, thus the hyperellipsoid formed from the weighting matrix P is entirely within the hyperellipsoid formed by Q. This implies that the volume of Q's hyperellipsoid is greater than or equal to the volume of P's hyperellipsoid. From Lemma 4, $H(y) \leq H(x)$ $\surd$

Theorem 6 *The hypervolume inside the hypersphere $x^T x \leq \epsilon^2$, $x \in \Re^n$ is given by*

$$V_n(\epsilon) = K_n \epsilon^n \tag{4.29}$$

$$K_n = \frac{(n-1)K_{n-1}K_{n-2}}{nK_{n-3}},\ n \geq 4 \tag{4.30}$$

and $K_1 = 2,\ K_2 = \pi,\ K_3 = 4\pi/3$

Proof:

(The approach to the proof was suggested by Kostas Kyriakopoulos.) For lines, discs and spheres, the volume is given by

$$V_1(\epsilon) = 2\epsilon \text{ (line)} \tag{4.31}$$

$$V_2(\epsilon) = \int_{-\pi/2}^{\pi/2} V_1(\epsilon\cos\theta)\epsilon\cos\theta d\theta = \pi\epsilon^2 \text{ (disc)} \tag{4.32}$$

$$V_3(\epsilon) = \int_{-\pi/2}^{\pi/2} V_2(\epsilon\cos\theta)\epsilon\cos\theta d\theta = 4\pi\epsilon^3/3 \text{ (sphere)} \tag{4.33}$$

In general, the volume of a hypersphere can be expressed as

$$V_n(\epsilon) = \int_{-\pi/2}^{\pi/2} V_{n-1}(\epsilon\cos\theta)\epsilon\cos\theta d\theta \tag{4.34}$$

Assume that V_{n-1} can be written in the form

$$V_{n-1}(\epsilon) = K_{n-1}\epsilon^{n-1} \tag{4.35}$$

Then

$$V_n(\epsilon) = K_{n-1}\epsilon^n \int_{-\pi/2}^{\pi/2} \cos^n\theta d\theta \Rightarrow V_n(\epsilon) = K_n\epsilon^n \tag{4.36}$$

Since $V_1(\epsilon) = K_1\epsilon^1$, by induction $V_n(\epsilon) = K_n\epsilon^n$.

From a table of integrals,

$$\int \cos^n\theta d\theta = \frac{1}{n}\cos^{n-1}\theta\sin\theta + \frac{n-1}{n}\int \cos^{n-2}\theta d\theta \tag{4.37}$$

By combining (4.35), (4.36), and (4.37), a recursive formula for calculating the K_n coefficient can be found

$$K_n\epsilon^n = K_{n-1}\epsilon^n \int_{-\pi/2}^{\pi/2} \cos^n\theta d\theta \tag{4.38}$$

$$\Rightarrow \int_{-\pi/2}^{\pi/2} \cos^n \theta d\theta = \frac{K_n}{K_{n-1}} = \left[\frac{n-1}{n} \int_{-\pi/2}^{\pi/2} \cos^{n-2} \theta d\theta \right] \tag{4.39}$$

but

$$\int_{-\pi/2}^{\pi/2} \cos^{n-2} \theta d\theta = \frac{K_{n-2}}{K_{n-3}} \tag{4.40}$$

Therefore

$$K_n = \frac{(n-1)K_{n-1}K_{n-2}}{nK_{n-3}} \surd \tag{4.41}$$

Theorems 3 and 6 are used along with Theorem 2 to find a lower bound on the reliability imposed by the definition of feasible plans.

Theorem 7 *Given the specification* $s : x^T Q x \leq \epsilon^2$, $x \in \Re^n$, $x \sim N(0, C_x)$, $C_x^{-1} - Q \geq 0$, $Q \geq 0$ *then the entropy constraint*

$$H(x) \leq H(spec) \tag{4.42}$$

(where $H(spec)$ *is the maximum entropy of* x *consistent with the specification* $s_1 : x^T C_x^{-1} x \leq \epsilon^2$*) yields a lower bound on reliability in meeting* $x^T Q x \leq \epsilon^2$ *which depends only on the dimension of* x.

Proof:

Let $y = Cx$ where C is obtained from $C_x^{-1} = C^T C$ (Cholesky decomposition). From the proof of Theorem 1, $y \sim N(0, I)$. Since y has a standard normal distribution, its entropy is easily found [35]

$$H(y) = \frac{n}{2} \ln 2\pi e \tag{4.43}$$

Because C arises from a covariance matrix, it is nonsingular. Thus $H(y)$ can be expressed in terms of x using the formula [35]

$$H(y) = H(Cx) = H(x) + \ln |det(C)| \tag{4.44}$$

The specification $s_1 : x^T C_x^{-1} x \leq \epsilon^2$ can be rewritten as $y^T y \leq \epsilon^2$, therefore by Theorems 3 and 6, the maximum entropy of y subject to the constraint s_1 is

$$H(spec_y) = \ln V_n(\epsilon) = \ln K_n \epsilon^n \tag{4.45}$$

Since $H(spec)$ is the maximum entropy of x subject to the constraint s_1,

$$H(spec_y) = H(spec) + \ln|det(C)| \tag{4.46}$$

As a consequence, the entropy constraint $H(x) \leq H(spec)$ is equivalent to

$$H(y) - \ln|det(C)| \leq H(spec_y) - \ln|det(C)| \tag{4.47}$$

or $H(y) \leq H(spec_y)$. From (4.45) and (4.43), the inequality is

$$\frac{n}{2}\ln 2\pi e \leq \ln K_n \epsilon^n \tag{4.48}$$

Thus the squared norm threshold, ϵ, must satisfy

$$\epsilon \geq \frac{\sqrt{(2\pi e)}}{K_n^{1/n}} \tag{4.49}$$

in order for the entropy constraint to hold. Using Theorem 1, it is easy to calculate the reliability in satisfying s_1, i.e.

$$P\{x^T C_x^{-1} x \leq \epsilon^2\} = \chi_n^2(\epsilon^2) \tag{4.50}$$

Since cumulative distribution functions are monotonically increasing, (4.49) and (4.50) imply that the reliability in meeting s_1 is at least $\chi_n^2[2\pi e/(K_n^{2/n})]$. Also, by Theorem 2, the reliability in meeting $s : x^T Q x \leq \epsilon^2$ is greater than or equal to the reliability in meeting s_1. The worst case reliabilities in meeting s, given that the entropy constraint is met, are listed in Table 4.1. $\surd$

Theorem 7 extends the techniques suggested in [34] to specifications consisting of weighted norms of Gaussian vectors, and provides a lower bound on reliability of plans selected as feasible. In robotics, 6 dimensional vectors are very common, as they can be used to describe the 6 degrees of freedom available in mechanical systems. The entropy constraint (4.42) yields a lower bound on reliability of 0.87 for $n = 6$. Lower bounds for other dimensions are listed in Table 4.1.

n	K_n	ϵ	$R_{worst} = \chi^2_n(\epsilon^2)$
1	2	2.07	0.96
3	4.19	2.56	0.91
6	5.2	3.1	0.87
10	2.6	3.8	0.83
20	0.025	4.96	0.78
50	$1.7\text{x}10^{-13}$	7.44	0.72
100	$2.4\text{x}10^{-40}$	10.3	0.68

Table 4.1: Worst Case Reliabilities When the Entropy Constraint is Satisfied

4.4 Reliability of Meeting a Set of Quadratic Specifications

The problem of finding the reliability in meeting the set of specifications $S = \{s_1, s_2, \ldots, s_N\}$ given in (4.5) has not been solved exactly. Rather, a lower bound on the reliability has been found by employing Theorem 2 and the positive correlation between the specifications.

Theorem 8 *Given a discrete time system described by (4.1 - 4.4) or (4.7) with the property* $Q - A_{tl}^T Q A_{tl} \geq 0$ *where* $Q \geq 0$ *is a constant matrix, then*

$$P(s_{k+1}|s_1 s_2 \ldots s_k) \geq P(s_k) \tag{4.51}$$

i.e., the specifications are positively correlated.

Proof:

The probability density function of $x(k)$ given that the specification at sample k has been satisfied is given by the continuous form of Bayes theorem to be [35]

$$f_{x(k)}(x(k)|s_k) = \frac{f_{x(k)}(x(k))P(s_k|X(k) = x(k))}{P(s_k)} \tag{4.52}$$

where $X(k)$ denotes the random value of $x(k)$. But since $X(k)$ is known to be within the hyperellipsoid,

$$P(s_k|X(k)=x(k)) = \begin{cases} 1 & , \quad x^T(k)Qx(k) \leq \epsilon^2 \\ 0 & , \quad x^T(k)Qx(k) > \epsilon^2 \end{cases} \tag{4.53}$$

therefore

$$f_{x(k)}(x(k)|s_k) = \begin{cases} f_{x(k)}(x(k))/P(s_k) & , \quad x^T(k)Qx(k) \leq \epsilon^2 \\ 0 & , \quad x^T(k)Qx(k) > \epsilon^2 \end{cases} \tag{4.54}$$

In other words, $f_{x(k)}(x(k)|s_k)$ is concentrated entirely in the success region defined by $x^T(k)Qx(k) \leq \epsilon^2$. Now divide $x(k+1)$ into two components, the component due to the previous state and the component attributed to noise, i.e. let

$$x_s(k+1) = A_{tl}x(k) \tag{4.55}$$

$$x_N(k+1) = D_{tl}v_l(k) \tag{4.56}$$

Since $Q - A_{tl}^T Q A_{tl} \geq 0$, $x^T(k)Qx(k) \geq x^T(k)A_{tl}^T Q A_{tl}x(k)$. This implies that the weighted square norm of $x_s(k+1)$ given s_k is inside the success hyperellipsoid because

$$x_s(k+1)^T Q x_s(k+1) = x^T(k)A_{tl}^T Q A_{tl}x(k) \leq x^T(k)Qx(k) \leq \epsilon^2 \tag{4.57}$$

Thus if noise is not present, then $s_k \Rightarrow s_j$, $j > k$. The noise is white, therefore $v_l(k)$ will not be affected by the fact that s_k has occurred. Hence the conditional density $x(k+1)|s_k$ is given by

$$f_{x(k+1)}(x(k+1)|s_k) = f_{x_s(k+1)}(x_s(k+1)|s_k) \otimes f_{x_N(k+1)}(x_N(k+1)) \tag{4.58}$$

where $\otimes$ denotes n–fold convolution. The conditional density $f_{x_s(k+1)}(x_s(k+1)|s_k)$ is derived from (4.54) and (4.55), thus from (4.57) it is concentrated in the success hyperellipsoid. The density $f_{x_N(k+1)}(x_N(k+1))$ is zero mean Gaussian since $v_l(k)$ and

$x(0)$ are Gaussian. As a result, the convolution of the densities, $f_{x(k+1)}(x(k+1)|s_k)$ is more concentrated in the success region than $f_{x(k+1)}(x(k+1))$. Hence

$$P(s_{k+1}|s_k) = \int_{x^T(k+1)Qx(k+1)} f_{x(k+1)}(x(k+1)|s_k)dx(k+1) \geq P(s_{k+1}) \qquad (4.59)$$

In addition, since v_l is zero mean white Gaussian noise and the system constantly reduces the weighted square norm of the previous state, $P(s_{k+1}|s_1 s_2 \ldots s_k) \geq P(s_k)$ $\surd$

Positive correlation of the specifications can be used to find a lower bound reliability for (4.5).

Theorem 9 *Given a system defined by (4.1 - 4.4) or (4.7), with state covariance matrix* $C_{x(k)} = E\{x(k)x^T(k)\}$ *and a set of specifications*

$$s_k : x^T(k)Qx(k) \leq \epsilon^2,\ k = 1, \ldots, N \qquad (4.60)$$

where $C^{-1}_{x(k)} - Q \geq 0$, $k = 1, \ldots, N$, $Q \geq 0$, *and* $Q - A_{tl}^T Q A_{tl} \geq 0$ *the reliability of meeting the specifications (4.60) has a lower bound of*

$$R \geq [\chi_n^2(\epsilon^2)]^N \qquad (4.61)$$

Proof:

Since $C^{-1}_{x(k)} - Q \geq 0$ and $x(k)$ is a zero mean normal variable, by Theorem 2

$$P(s_k) \geq \chi_n^2(\epsilon^2) \qquad (4.62)$$

By the chain rule of probabilities,

$$R = P(s_1 s_2 \ldots s_N) = P(s_1)P(s_2|s_1)P(s_3|s_1 s_2) \cdots P(s_N|s_1 s_2 \cdots s_{N-1}) \qquad (4.63)$$

Theorem 8 shows that $P(s_{k+1}|s_1 s_2 \ldots s_k) \geq P(s_k)$. Therefore

$$R \geq P(s_1)P(s_2) \cdots P(s_N) \qquad (4.64)$$

From (4.62), this implies that

$$R \geq [\chi_n^2(\epsilon^2)]^N \quad \surd \tag{4.65}$$

In order to use Theorem 9, the closed loop system must satisfy the constraint

$$Q - A_{tl}^T Q A_{tl} \geq 0 \tag{4.66}$$

One method of ensuring that (4.66) is satisfied involves first picking an arbitrary matrix $Q_1 \in \Re^{nXn}$ such that $Q_1 > 0$. A weighting matrix Q can then be found by numerically solving the algebraic Lyapunov equation

$$Q - A_{tl}^T Q A_{tl} = Q_1 \tag{4.67}$$

for a positive definite real symmetric matrix Q. Such a solution exists for (4.67) as long as $x(k+1) = A_{tl}x(k)$ is asymptotically stable [27]. In the case study simulation, Q_1 is chosen as $Q_1 = kI$, where k is a positive integer scaling the identity matrix.

The Q matrix found from (4.67) does not, of course, have any relationship to the specifications. Lemma 10 allows Q to be related to the specifications.

Lemma 10 *Given a system defined by (4.1 - 4.4) or (4.7), with state covariance matrix* $C_{x(k)} = E\{x(k)x^T(k)\}$ *and a set of specifications*

$$s_k : x^T(k)Q_d(k)x(k) \leq \epsilon^2,\ k = 1, \ldots, N \tag{4.68}$$

where $C_{x(k)}^{-1} - Q_d(k) \geq 0,\ k = 1, \ldots, N,\ Q - A_{tl}^T Q A_{tl} \geq 0,\ Q \geq 0,\ Q_d(k) \geq 0,\ Q - Q_d(k) \geq 0,\ k = 1, \ldots, N$ *the reliability of meeting the specifications (4.68) has a lower bound of*

$$R \geq [\chi_n^2(\epsilon^2)]^N \tag{4.69}$$

Proof:

Since $Q - A_{tl}^T Q A_{tl} \geq 0$, the system monotonically moves into the hyperellipsoid defined by $x^T(k)Qx(k)\epsilon^2$. From $Q - Q_d(k) \geq 0,\ k = 1, \ldots, N$, if the specification

s_{q_k} : $x^T(k)Qx(k) \leq \epsilon^2$ is satisfied, then s_k : $x^T(k)Q_d(k)x(k) \leq \epsilon^2$ is also satisfied. As a consequence, the system also moves into the hyperellipsoids $x^T(k)Q_d(k)x(k) \leq \epsilon^2$. The s_k specifications are then positively correlated (Theorem 8). Since $C^{-1}_{x(k)} - Q_d(k) \geq 0$, $k = 1, \ldots, N$, Theorem 2 yields

$$P(s_k) \geq \chi^2_n(\epsilon^2) \tag{4.70}$$

The positive correlation between specifications then implies

$$R \geq [\chi^2_n(\epsilon^2)]^N \ \surd \tag{4.71}$$

Thus as long as the Q matrix found by solving the Lyapunov equation (4.67) satisfies $Q - Q_d(k)$, $k = 1, \ldots, N$, then the reliability of hyperellipsoid specifications described by $Q_d(k)$ can be found. Note that this allows treatment of time varying specifications.

A second lemma treats the case in which the specifications constrain only a portion of the state variables, obtaining an improved bound over that found directly by Lemma 10.

Lemma 11 *Given the conditions of Lemma 10 except* $C^{-1}_{x(k)} - Q_d(k) \geq 0$, $k = 1, \ldots, N$, *if*

$$Q_d(k) = \begin{bmatrix} Q_{sub}(k) & 0 \\ 0 & 0 \end{bmatrix} \tag{4.72}$$

and

$$C^{-1}_{x_{sub}(k)} - Q_{sub}(k) \geq 0, \ k = 1, \ldots, N \tag{4.73}$$

where $Q_{sub}(k) \in \Re^{m \times m}$, $m \leq n$, x_{sub} *denotes the first* m *elements of* x, *and* $C^{-1}_{x_{sub}(k)}$ *is the covariance matrix of* $x_{sub}(k)$, *then*

$$R \geq [\chi^2_m(\epsilon^2)]^N \tag{4.74}$$

Proof:

From Lemma 10, the specifications are positively correlated. The specifications can be rewritten as

$$s_k : \; x_{sub}^T(k) Q_{sub}(k) x_{sub}(k) \leq \epsilon^2 \tag{4.75}$$

Since $x(k)$ is normally distributed with zero mean, $x_{sub}(k)$ is also. From Theorem 2, $P(s_k) \geq \chi_m^2(\epsilon^2)$. The positive correlation then yields the lower bound

$$R \geq [\chi_m^2(\epsilon^2)]^N \; \surd \tag{4.76}$$

The results of this lemma are not, of course, restricted to only the first elements of the state vector, x. Rather, by rearranging the vector, any subset of the state vector may be used.

Lemma 11 is used extensively during the case study, because the case study has specifications constraining the 6 dimensional pose error of a robotic system. The state vector for the system, on the other hand, has 24 states. In order to optimize the design, the case study makes use of the following result from discrete Lyapunov control theory [27].

Theorem 12 *Given the discrete time invariant system*

$$x(k+1) = A_t x(k) + B_t u(k) \tag{4.77}$$

with A_t asymptotically stable and a real positive definite symmetric weighting matrix $Q > 0$ such that $Q - A_t^T Q A_t > 0$, the feedback

$$u^*(k) = -(B_t^T Q B_t)^{-1} B_t^T Q A_t x(k) = -K_t x(k) \tag{4.78}$$

reduces the weighted square norm $x^T(k) Q x(k)$ at the optimal rate.

Proof:

First define a discrete Lyapunov function

$$V(x(k)) = x^T(k) Q x(k) \tag{4.79}$$

Since $Q > 0$, by the Stability theorem of Lyapunov, the system is asymptotically stable in the large if

$$\Delta V(x(k)) = V(x(k+1)) - V(x(k)) < 0 \tag{4.80}$$

Let the feedback be denoted as $u(k) = -K_l x(k)$. Then

$$\Delta V(x(k)) = x^T(k)[A_t - B_t K_l]^T Q[A_t - B_t K_l]x(k) - x^T(k)Qx(k) \tag{4.81}$$

or

$$\Delta V(x(k)) = x^T(k)A_t^T Q A_t x(k) + 2x^T(k)A_t^T Q B_t u(k) + u^T(k)B_t^T Q B_t u(k) - x^T(k)Qx(k) \tag{4.82}$$

In order to find the control, $u(k)$, which will reduce the weighted square norm of $x(k)$ at its optimal rate, $\Delta V(x(k))$ must be minimized. Minimization with respect to $u(k)$ requires that

$$\frac{\partial \Delta V(x(k))}{\partial u(k)} = 0 = 2B_t^T Q A_t x(k) + 2B_t^T Q B_t u^*(k) \tag{4.83}$$

$$\Rightarrow u^*(k) = -(B_t^T Q B_t)^{-1} B_t^T Q A_t x(k) \tag{4.84}$$

This does in fact produce a minimum because

$$\frac{\partial \Delta^2 V(x(k))}{\partial u(k)^2} = P = 2B_t^T Q B_t \geq 0 \tag{4.85}$$

i.e. P is positive semi-definite, so a minimum has been reached. The positive semi-definiteness of $2B_t^T Q B_t$ follows from the fact that $Q > 0 \Rightarrow z^T Q z > 0$, $z \neq 0$. But z can be written as $z = \sqrt{2}B_t y$. Then $z^T Q z = y^T 2B_t^T Q B_t y > 0$, $z \neq 0$. Since B_t may have a nonzero null space, z may equal zero when $y \neq 0$. Thus $y^T P y \geq 0$, $y \neq 0$. $\surd$

Theorem 12 is not a new result of this work. Rather, it is a well known result from discrete Lyapunov control theory applied to the results of this work.

In order to design a control algorithm for

$$x(k+1) = A_t x(k) + B_t u(k) + D_t v(k) \tag{4.86}$$

where $E\{x(k)\} = 0$ (this implies that only the stochastic errors are penalized, not deterministic transient effects in the control law), $v(k)$ is zero mean uncorrelated white Gaussian noise, $x(k+1) = A_t x(k)$ is asymptotically stable, and the controller must be capable of reliably satisfying a set of specifications of the form $s_k : x^T(k)Q_d(k)x(k) = x_{sub}^T(k)Q_{sub}(k)x_{sub}(k) \leq \epsilon^2$, the following algorithm is suggested:

1. Select a matrix $Q_1 > 0$.

2. Find a real positive definite solution Q to the Lyapunov equation $Q - A_t^T Q A_t = Q_1$ numerically.

3. If $Q - Q_d(k) \geq 0$, $\forall k = 1, \ldots, N$, then let $u(k) = -(B_t^T Q B_t)^{-1} B_t^T Q A_t x(k)$ to produce a new (optimized) system $x(k+1) = A_{tl}x(k) + D_{tl}v_l(k)$.

4. Calculate the covariance of $x(k)$, $C_x(k)$, by solving $C_x(k+1) = A_{tl}C_x(k)A_{tl}^T + D_{tl}Cov(v_l(k))D_{tl}^T$.

5. Obtain the covariance matrix, $C_{sub}(k) = Cov[x_{sub}(k)]$, for the subset of the state variables which are constrained by the specifications. $C_{sub}(k)$ can be found from $C_x(k)$.

6. If $C_{sub}^{-1}(k) - Q_{sub}(k) \geq 0$, $\forall\, k = 1, \ldots, N$ then $R \geq [\chi_m^2(\epsilon^2)]^N$. Otherwise, try selecting a different Q_1 or changing A_t through inner loop feedback gains and go back to step 1.

This design procedure is used in the case study for reliably controlling a high precision positioning task in a robotic system.

4.5 Conclusions

The techniques developed here provide a method of easily analyzing specifications posed as a quadratic function of zero mean normal random vectors. This is

useful in itself for evaluating static errors in robotic systems such as those arising from an imperfect model of the manipulator kinematics or from a noisy set-point measured by a vision system. In addition, the method is applicable to multivariable discrete time invariant control systems. Thus noise inside the feedback loop (such as encoder noise) can also be analyzed, and the reliability of staying within a hyperellipsoid defined by the quadratic specifications over an interval of time can be found. In addition, entropy constraints are developed for the quadratic specifications, and are shown to yield a lower bound on reliability which depends only on the order of the vector.

The method has great potential for use in conjunction with Skelton's "covariance control" [18]. Skelton finds a parameterization for all compensators of a specified order which will achieve a desired steady state covariance (for linear time invariant systems). Skelton does not, on the other hand, develop any methods for specifying the desired covariance matrix. Using the concepts developed herein, it is possible to first phrase the specifications in terms of physically meaningful weighted norms. A steady state covariance matrix C_x can then be selected such that $C_x^{-1} - Q \geq 0$, and a compensator can be found which obtains C_x using Skelton's theorems. Theorem 9 can then be used to find a lower bound on the reliability of meeting the quadratic specifications over a time interval.

5. CASE STUDY–RELIABILITY OF VISUAL POSITIONING

Visual positioning is an extremely important and highly developed facet of robotics. However, surprisingly few attempts have been made at analyzing the reliability of visual positioning. Thus, this section has a two-fold purpose. First, it illustrates the ideas of the preceding chapters so that the notation and concepts are made clear. Second, it provides a general reliability solution, useful for many hand/eye coordination tasks of a robotic manipulator.

> *Problem to be Solved:* A manipulator has already moved close to a gripping post on an object using *a priori* knowledge. Due to environmental uncertainties, the final movement cannot be completed without measuring the current pose of the gripping post. Using a vision system mounted on a separate arm, it is possible to view the object from N different preprogrammed positions. Five inverse kinematics routines are available, and a library of M computed torque compensators can be used for manipulator control. It is desired to make the final movement subject to two specifications–the total execution time must be less than t_f and the gripper must not collide with the gripping post. Find reliable subsets of control and sensing which are capable of satisfying these specifications.

The first step in the analysis is the definition of the desired set of specifications, S_D. To avoid collision with the gripping post, error tolerances are generated in each of the 6 degrees of freedom from the geometry of the gripper and gripping post. Hyperellipsoids are then inscribed within these tolerance bounds as a function of the distance from the final gripping frame. The geometry of the post and gripper is illustrated in Figures 5.1 and 5.2, while the overall configuration of the case study is shown in Figure 5.3. The homogeneous transformation from the initial end effector

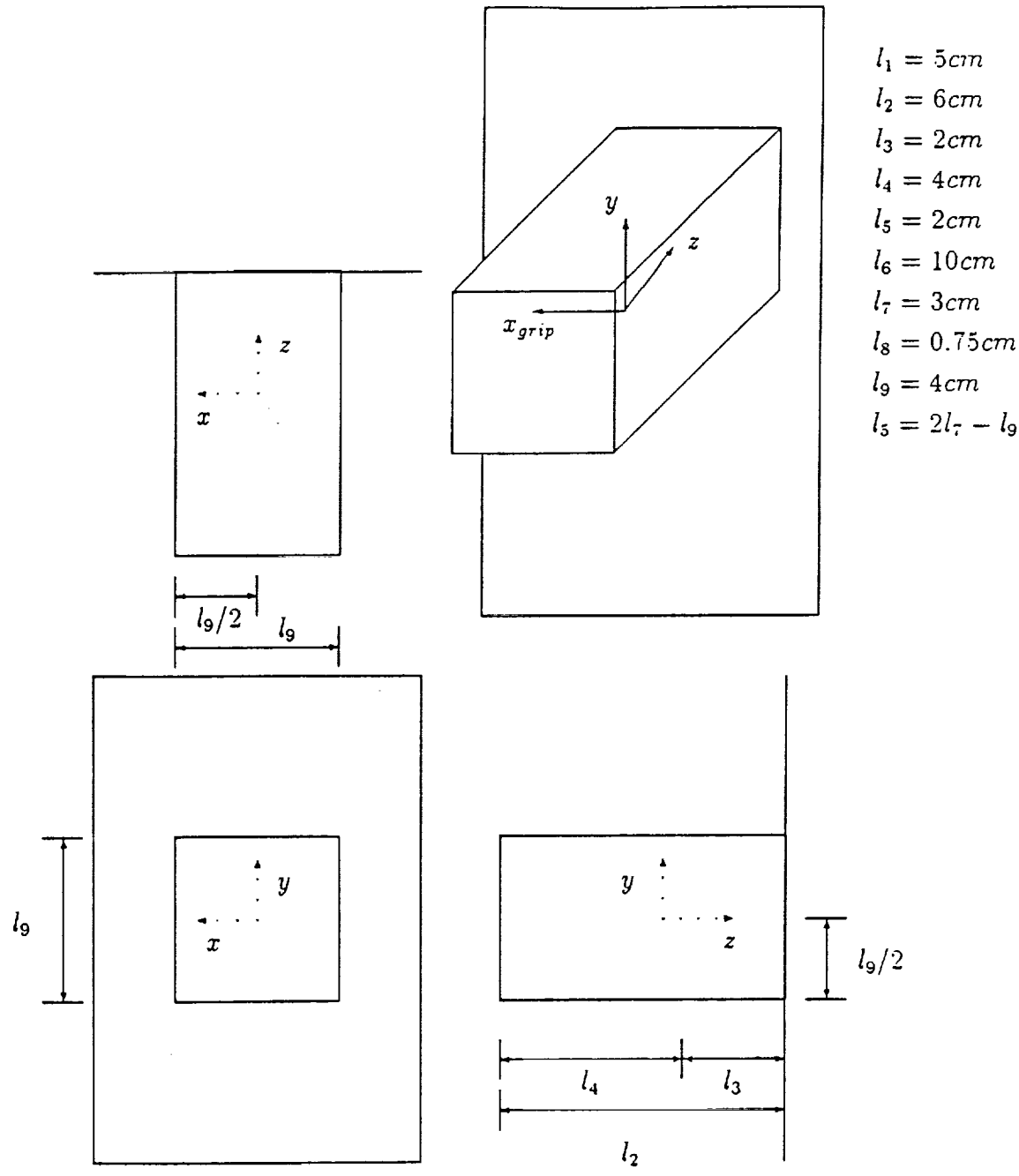


Figure 5.1: Geometry of the Gripping Post

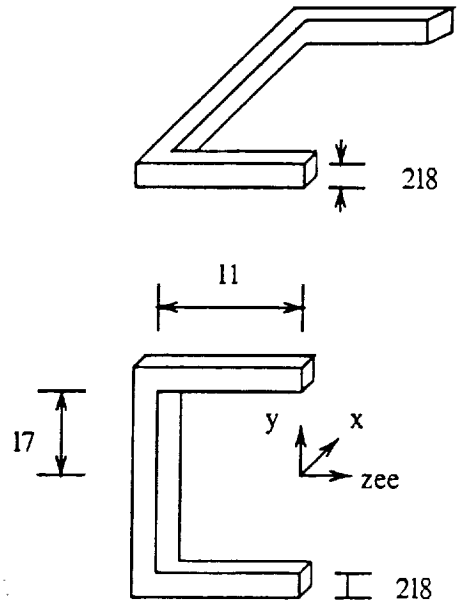


Figure 5.2: Geometry of the Gripper

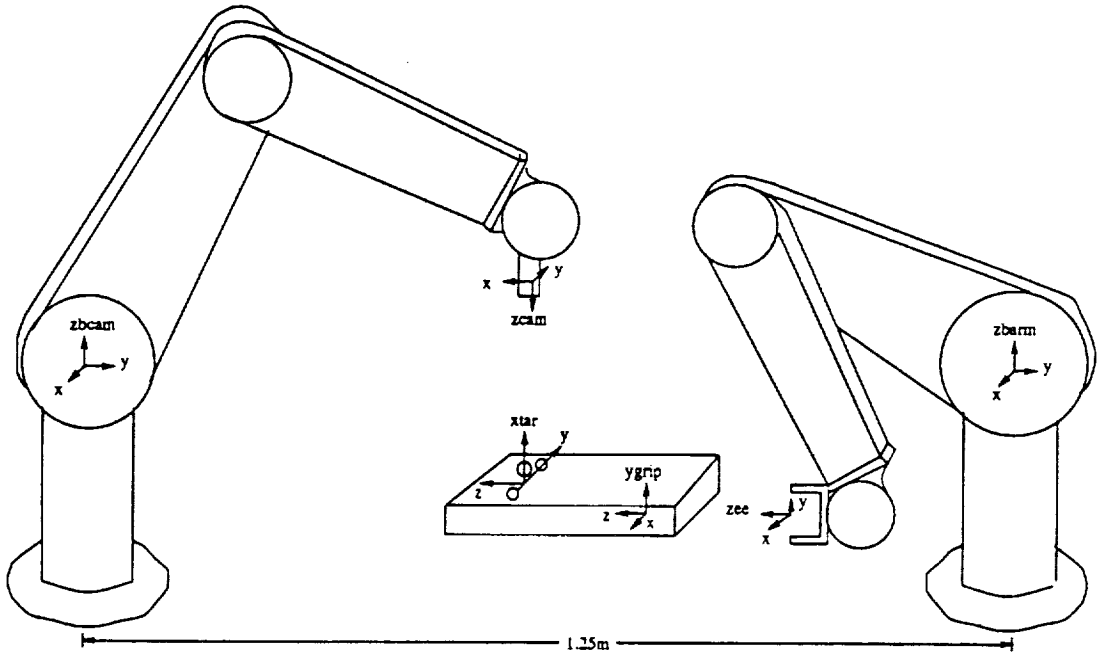


Figure 5.3: Configuration of the Case Study

frame to the base of the manipulating arm is

$$T_{ee}^{barm} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -.3 \\ 0 & 1 & 0 & -.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

The transform from the nominal gripping frame to the base of the manipulator is

$$T_{grip}^{barm} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -.5 \\ 0 & 1 & 0 & -.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

To properly grip the post, the end effector frame should be driven to coincide with the gripping frame. Thus the end effector must be moved 20cm along its "z" axis for the nominal grip to be performed. The object is not at its nominal position at all times, rather the nominal position is a noisy estimate of the object's actual pose. Consequently, a measurement of the object's pose must be made with the vision system, and then the measured transformation from the gripping frame to the end effector frame is used to generate a joint trajectory capable of docking the gripper with the gripping post.

To perform the docking reliably, the generated trajectory must prevent collisions between the gripper and post. These collisions can be modeled by defining tolerance bounds in each direction. The tolerance bound in the end effector's "x" direction is easy to define because a collision will not occur due to the errors in this direction. The only fear is that the grip will not be centered. To prevent drastically off-centered grips, a tolerance bound of $3cm$ is defined. The tolerance value in the end effector's "y" direction, on the other hand, is more critical as errors in "y" will

cause collisions. Using the data in Figure 5.1, the "y" tolerance is defined to be

$$e_{y_{max}} = \begin{cases} .9l_5 & , \ p_z < 2l_4 + l_3 \\ p_z - 2l_4 - l_3 + .9l_5 & , \ 2l_4 + l_3 \leq p_z < 2l_4 + l_3 - .9l_5 + l_6 \\ l_6 & , \ p_z \geq 2l_4 + l_3 - .9l_5 + l_6 \end{cases} \tag{5.3}$$

where p_z is the absolute value of the distance to be moved along the end effector's "z" axis to dock the gripping and end effector origins. Since the specification becomes increasingly tighter as the object is approached, the end effector is, in effect, "funneled" into the gripping pose. Once the gripping post is inside a portion of the gripper, the "y" tolerance reaches its minimum value and stays at that constant value for the remainder of the docking.

The specification for the "z" direction also becomes increasingly tighter as the object is approached. It is defined to be

$$e_{z_{max}} = \begin{cases} p_z + l_3 & , \ p_z < l_4 \\ l_2 & , \ l_4 \leq p_z < 2l_4 + l_3 \\ p_z - l_4 & , \ p_z \geq 2l_4 + l_3 \end{cases} \tag{5.4}$$

In contrast to the "y" specification, the "z" specification does not reach a minimum before the docking is complete, but instead becomes increasingly difficult to meet. This reflects the fact that, as the gripping post is moved farther into the gripper, a collision between the inside end of the gripper and the post is more difficult to avoid.

The tolerance bounds for orientation are coupled to the allowable errors in positioning. For instance, if a large error is present in the "y" direction, then even slight rotations about the end effector's "x" axis will produce a collision. The "x"

orientation tolerance is thus defined as

$$e_{\theta_{xmax}} = \begin{cases} \arctan .5l_5/l_4 & , \ p_z < 2l_4 + l_3 \\ p_z - (2l_4 + l_3) + \arctan .5l_5/l_4 & , \ 2l_4 + l_3 \leq p_z < \pi/2 + 2l_4 + l_3 \\ & \quad - \arctan .5l_5/l_4 \\ \pi/2 & , \ p_z > \pi/2 + 2l_4 + l_3 - \arctan .5l_5/l_4 \end{cases} \tag{5.5}$$

The "y" orientation error is similar to the "x" positioning error– it will not cause collisions under reasonable circumstances, but will cause off-centered grips. Thus $e_{\theta_{ymax}} = 23$ degrees. Finally, rotational errors about the end effector's "z" axis can cause collisions of the edge of the gripper with the post, thus

$$e_{\theta_{zmax}} = \begin{cases} \alpha & , \ p_z < 2l_4 + l_3 \\ p_z - (2l_4 + l_3) + \alpha & , \ 2l_4 + l_3 \leq p_z < \pi/2 + 2l_4 + l_3 - \alpha \\ \pi/2 & , \ p_z \geq \pi/2 + 2l_4 + l_3 - \alpha \end{cases} \tag{5.6}$$

where

$$\alpha = \arccos \frac{l_7 - .1l_5}{\sqrt{l_7^2 + l_8^2}} - \arctan l_8/l_7 \tag{5.7}$$

The tolerance specifications obtained from the data in Figure 5.2 is plotted in Figures B.1–B.3.

Once the tolerance specifications are found, it is a very simple matter to inscribe hyperellipsoids inside those specifications. First, the radius of the hyperellipsoid, ϵ, must be selected. If the conditions of Theorem 2 are met, the probability of staying inside the hyperellipsoid has a lower bound of $\chi_6^2(\epsilon^2)$. Thus ϵ can be selected from tables of the chi-square distribution. For the simulation, the radius is selected to be $\epsilon^2 = 22$ which yields a lower bound probability of .9988. The pose error in the case study, $e \in \Re^6$, is represented as the roll–pitch–yaw angles stacked on top of the position error. Thus the problem of inscribing hyperellipsoids can be expressed as finding a weighting matrix, Q_{px}, such that if

$$e^T Q_{px} e l e \epsilon^2 \tag{5.8}$$

is satisfied, then e is inside the tolerance hyperrectangle.

One Q_{px} which satisfies these requirements is

$$Q_{px}(p_z) = \begin{bmatrix} \epsilon^2/e^2_{\theta_{xmax}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \epsilon^2/e^2_{\theta_{ymax}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \epsilon^2/e^2_{\theta_{zmax}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \epsilon^2/e^2_{x_{max}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \epsilon^2/e^2_{y_{max}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \epsilon^2/e^2_{z_{max}} \end{bmatrix} \tag{5.9}$$

Note that, since the tolerance specifications are a function of the distance to the gripping frame, p_z, the weighting matrix (5.9) is also a function of p_z. Since the distance from the gripping frame to the end effector (p_z) is a function of the generated joint trajectory, $\Theta_d(k)$, $Q_{px}(\cdot)$ can be viewed as a function of either $\Theta_d(k)$ or k.

By using the weighting matrix found from the geometry, the specifications for the given problem are:

- s_1 = { total execution time is less than a desired time, t_f }
- s_2 = { $e^T(k)Q_{px}(k)e(k) \leq \epsilon^2$, for all of the time steps, k, present in the trajectory driving the end effector to the gripping frame }

Next, the plans must be formed. In this case, each plan will consist of the triplet $A_i = \{V_i, K_i, G_i\}$, where V_i, K_i, and G_i are the vision algorithm, inverse kinematics, and compensator used by the i^{th} plan, respectively. Each algorithm is now examined in depth to find expressions for the statistical performance of each plan.

5.1 Analysis of the Vision System

As previously stated, the vision system consists of a camera mounted on the end of a separate robot arm. Although the vision system may be positioned at N different viewpoints to reduce problems with noise, assume there is only sufficient

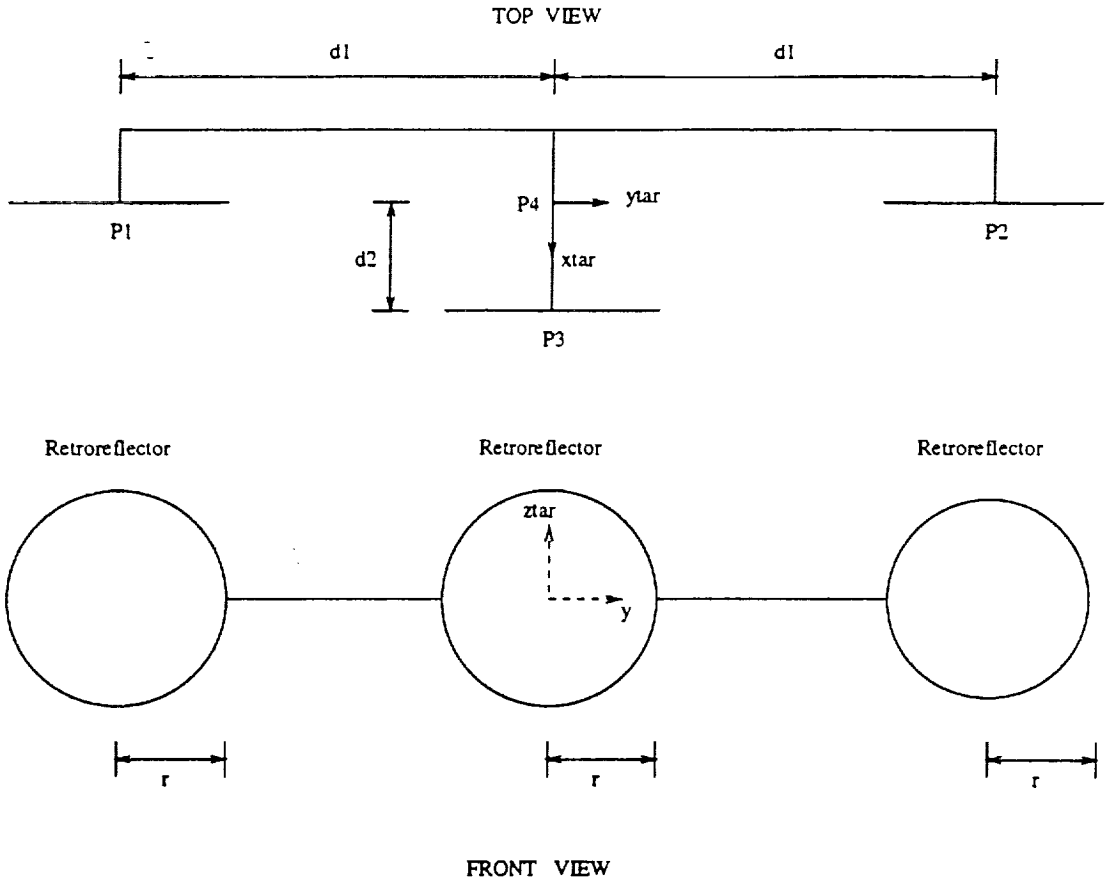


Figure 5.4: The MSFC Target Used in the Vision System

time to take measurements from one of the viewpoints. Each viewpoint will be regarded as a separate vision algorithm, thus the number of potential vision algorithms is also N. The accuracy in measuring the object's pose will depend upon the pose of the object in the camera coordinate frame. Consequently, each viewpoint will have differing measurement statistics. These statistics can be found by using an approach similar to that of Lee and Kay [29].

The vision system used is based on a system developed by NASA's Marshall Space Flight Center for autonomous docking of space ships. It consists of a single camera and a three dimensional target. The target is depicted in Figure 5.4. Since

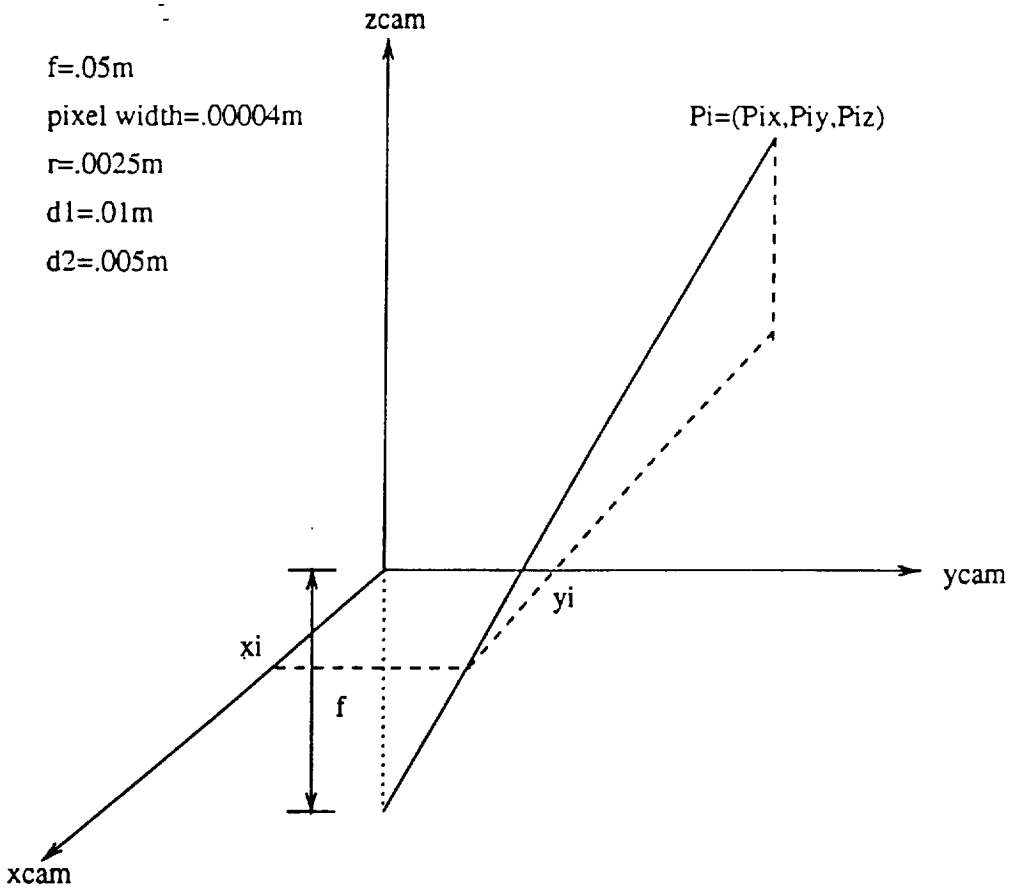


Figure 5.5: A Pinhole Model of Camera Optics

the target parameters (d_1, d_2, r) are known, it is possible to find the target's pose with only a single camera. The target pose then defines the gripping frame's pose because the transformation from the target to the gripping frame is known. The measurement is accomplished by first assigning a coordinate frame to the camera, as shown in Figure 5.5, such that the camera frame X–Y plane corresponds to the inverted image plane. Next, points are projected onto the image plane using the pin–hole model of camera optics to produce the inverted image frame points (x_i, y_i) from the camera frame coordinates, $p_i = [p_{x_i}\ p_{y_i}\ p_{z_i}\ 1]^T$

$$x_i = \frac{f p_{x_i}}{p_{z_i} + f} \tag{5.10}$$

$$y_i = \frac{f p_{y_i}}{p_{z_i} + f} \tag{5.11}$$

The positions of the target points with respect to the target coordinate system are

$$p_1^{tar} = [\ 0 \quad -d_1 \quad 0 \quad 1\ ]^T \tag{5.12}$$

$$p_2^{tar} = [\ 0 \quad d_1 \quad 0 \quad 1\ ]^T \tag{5.13}$$

$$p_3^{tar} = [\ d_2 \quad 0 \quad 0 \quad 1\ ]^T \tag{5.14}$$

$$p_4^{tar} = [\ 0 \quad 0 \quad 0 \quad 1\ ]^T \tag{5.15}$$

Points p_1^{tar}–p_3^{tar} correspond to the centers of the circular retroreflectors. In the simulation, the entire area of each circle is mapped onto the discrete grid formed by the CCD image plane. The centroid of that image is then found to determine the image position (x_i, y_i) which corresponds to the retroreflector center p_i^{tar}. Unfortunately, if the circle is close to the image plane and at an angle with respect to the image plane, then the image plane's centroid does not correspond to the center of the circle. This slight bias does not pose a problem in the system under consideration, as will be demonstrated when the simulation results are presented. Let the unknown transformation from the target coordinate system to the camera coordinate system be denoted by

$$T_{tar}^{cam} = \begin{bmatrix} R_{tar}^{cam} & & & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.16}$$

$$R_{tar}^{cam} = [nsa] = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \tag{5.17}$$

$$p = [p_x p_y p_z]^T \tag{5.18}$$

The camera coordinates corresponding to the four target points are related by

$$p_i^{cam} = T_{tar}^{cam} p_i^{tar} \tag{5.19}$$

These points are then substituted into (5.10, 5.11) to produce six nonlinear equations in nine unknowns

$$x_1(-d_1 s_z + p_z + f) = f(-d_1 s_x + p_x) \tag{5.20}$$

$$y_1(-d_1 s_z + p_z + f) = f(-d_1 s_y + p_y) \tag{5.21}$$

$$x_2(d_1 s_z + p_z + f) = f(d_1 s_x + p_x) \tag{5.22}$$

$$y_2(d_1 s_z + p_z + f) = f(d_1 s_y + p_y) \tag{5.23}$$

$$x_3(d_2 n_z + p_z + f) = f(d_2 n_x + p_x) \tag{5.24}$$

$$y_3(d_2 n_z + p_z + f) = f(d_2 n_y + p_y) \tag{5.25}$$

Also, since R_{tar}^{cam} is orthonormal,

$$n^T n = 1 \tag{5.26}$$

$$s^T s = 1 \tag{5.27}$$

$$n^T s = 0 \tag{5.28}$$

$$a = n \mathrm{x} s \tag{5.29}$$

These nonlinear equations are solved in the simulation using the Newton–Raphson iterative technique for solving nonlinear simultaneous equations. The solution yields the transformation from the target coordinate system to the camera coordinate system, T_{tar}^{cam}. Due to the iterative nature of the Newton–Raphson solution, the measurement time is stochastic.

Once the homogeneous transformation matrix is found, it can be represented as a 6 x 1 vector by expressing the rotation matrix in terms of roll–pitch–yaw angles. Following the approach in [11], the roll–pitch–yaw angles can be found as:

$$\theta_x = atan2(n_y, n_x) \tag{5.30}$$

$$\theta_z = atan2(\sin\theta_x a_x - \cos\theta_x a_y, -\sin\theta_x s_x + \cos\theta_x s_y) \tag{5.31}$$

$$\theta_y = atan2(-n_z, \cos\theta_x n_x + \sin\theta_x n_y) \tag{5.32}$$

The pose of the target can be expressed as

$$x_{tar}^{cam} = \begin{bmatrix} \theta \\ p \end{bmatrix} = \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \\ p_x \\ p_y \\ p_z \end{bmatrix} \tag{5.33}$$

Equations 5.20–5.29 combined with 5.30–5.32 are nonlinear equations relating the image measurements $((x_i, y_i),\ i = 1 \ldots 3)$ to x_{tar}^{cam}. Due to pixel truncation, the measured image positions will not be perfect. These image plane errors will cause errors in x_{tar}^{cam}. Denote the true image positions as

$$x_i' = x_i + n_{ix} \tag{5.34}$$

$$y_i' = y_i + n_{iy} \tag{5.35}$$

where

$$n_{image} = [\ n_{1x} \quad n_{1y} \quad n_{2x} \quad n_{2y} \quad n_{3x} \quad n_{3y}\ ]^T \tag{5.36}$$

is a vector of image noise. Using an approach similar to that of Lee and Kay [29], the errors in x_{tar}^{cam}, e_{tar}^{cam}, arising from n_{image} can be found through a linearization procedure. The procedure substitutes the true pose, $x_{tar}^{cam'} = x_{tar}^{cam} + e_{tar}^{cam}$ and the true image positions, (x_i', y_i'), into equations (5.20–5.29). Retaining only the first order terms yields

$$M e_{tar}^{cam} = D n_{image} \tag{5.37}$$

where

$$M = \begin{bmatrix} -x_1 d_1 s_y & x_1 d_1 s_x + f d_1 s_z & -f d_1 s_y & -f & 0 & x_1 \\ -y_1 d_1 s_y - f d_1 s_z & y_1 d_1 s_x & f d_1 s_x & 0 & -f & y_1 \\ x_2 d_1 s_y & -x_2 d_1 s_x - f d_1 s_z & f d_1 s_y & -f & 0 & x_2 \\ y_2 d_1 s_y + f d_1 s_z & -y_2 d_1 s_x & -f d_1 s_x & 0 & -f & y_2 \\ x_3 d_2 n_y & -x_3 d_2 n_x - f d_2 n_z & f d_2 n_y & -f & 0 & x_3 \\ y_3 d_2 n_y + f d_2 n_z & -y_3 d_2 n_x & -f d_2 n_x & 0 & -f & y_3 \end{bmatrix} \tag{5.38}$$

$$D = \begin{bmatrix} d_1 s_z - v & 0 & 0 & 0 & 0 & 0 \\ 0 & d_1 s_z - v & 0 & 0 & 0 & 0 \\ 0 & 0 & -d_1 s_z - v & 0 & 0 & 0 \\ 0 & 0 & 0 & -d_1 s_z - v & 0 & 0 \\ 0 & 0 & 0 & 0 & -d_2 n_z - v & 0 \\ 0 & 0 & 0 & 0 & 0 & -d_2 n_z - v \end{bmatrix} \tag{5.39}$$

$$v = p_z + f \tag{5.40}$$

The pose error is then

$$e_{tar}^{cam} = M^{-1} D n_{image} \tag{5.41}$$

If the exact focal length is not known, then the errors arising in x_{tar}^{cam} due to the focal length error, e_f, are (to a first order approximation)

$$e_{tar}^{cam} = \begin{bmatrix} -x_1 - d_1 s_x + p_x \\ -y_1 - d_1 s_y + p_y \\ -x_2 + d_1 s_x + p_x \\ -y_2 + d_1 s_y + p_y \\ -x_3 + d_2 n_x + p_x \\ -y_3 + d_2 n_y + p_x \end{bmatrix} e_f \tag{5.42}$$

Although the errors caused by focal length uncertainties can easily be accommodated using (5.42), the present simulation does not include these terms.

In order to find a set of position and orientation errors corresponding to the translations and rotations about the camera frame which will move the measured T_{tar}^{cam} into alignment with the actual T_{tar}^{cam}, the coupling between positional errors and orientation must be included. This yields the final (camera frame) errors in pose due to pixel truncation, e_{cam}, to be

$$e_{cam} = \begin{bmatrix} & 0 & & I \\ \begin{bmatrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{bmatrix} & & I \end{bmatrix} M^{-1} D n_{image} = L n_{image} \tag{5.43}$$

The covariance of the vision pose when using the i^{th} algorithm is then

$$C_{vi} = L_i Cov(n_{image} n_{image}^T) L_i^T \tag{5.44}$$

where L_i is the measurement matrix made from the i^{th} viewpoint.

In order to make use of (5.43), a statistical model must be found for the image noise, n_{image}. This noise arises strictly from pixel truncation, because the image position of a point is only known to be somewhere within the rectangle formed by the pixel. With only the knowledge that the position is within bounds, the maximum entropy distribution is given by Jaynes MEM principle to be uniform. Consequently, a uniform distribution in the image plane is assumed for each pixel. Note that this assumption is used strictly in the analysis of accuracy–uniformly distributed noise is not added to the image. Instead, the geometry of the pin–hole camera model dictates the noise level due to truncation. The variance of the image noise under the uniform distribution is

$$\sigma_x^2 = w_x^2/12 \tag{5.45}$$

$$\sigma_y^2 = w_y^2/12 \tag{5.46}$$

where w_x and w_y are the pixel widths in the "x" and "y" directions, respectively.

The image positions of the centers of the retroreflectors is found by finding the centroids of the images caused by the retroreflectors. Consequently, the image position of the center is much more accurate than a single pixel. This increase in accuracy is modeled by assuming that the pixel noise is independent and identically distributed between pixels. The covariance of the centroid is then found by

$$Cov(n_{ix}) = \sigma_x^2 \frac{m + k_x}{2m^2} \tag{5.47}$$

$$Cov(n_{iy}) = \sigma_y^2 \frac{n + k_y}{2n^2} \tag{5.48}$$

where

- m is the number of rows covered by the image,
- k_x is the number of rows consisting of a single pixel,
- n is the number of columns covered by the image, and
- k_y is the number of columns consisting of a single pixel.

Since the centroid is found by summing independent, identically distributed random variables, the distribution of the centroid is (by the Central Limit Theorem) Gaussian. This implies that n_{image} is a Gaussian random vector, and because e_{cam} is linearly related to n_{image} through (5.43), e_{cam} is also Gaussian.

Once the image measurements have been made, then L may be computed, and the error statistics e_{cam} can be found. This information serves as an excellent performance check after the camera is positioned and measurements have been made. Unfortunately, it does not provide a method of evaluating each viewpoint without doing the time and computationally intensive re-positioning of the camera. Fortunately, an estimate of e_{cam} can be obtained without making the image measurements based on *a priori* knowledge. As the problem statement notes, an *a priori* estimate of the object's position/orientation in the base frame is known. This

frame may be mapped into the i^{th} camera coordinate system using the homogeneous coordinate transformation $T^{c_i}_{barm}$ which maps frames from the base of the manipulating arm to the i^{th} camera coordinate system. Because the camera viewpoints are pre-specified, the $T^{c_i}_{barm}$ transformations can be calculated off-line and stored for later use at little cost to the system. Hence the *a priori* estimate of the three points can be transformed from the base frame to each camera frame quite easily. Estimates of the error transformation for the i^{th} viewpoint, denoted as $\hat{L}_i$, can then be calculated. The quality of $\hat{L}_i$ depends upon the relative error of the *a priori* points in the camera frame. Since the cameras are kept at a respectable distance from the object to avoid collisions and interference with the manipulation task, the relative error is small, so initial analysis using $\hat{L}_i$ is quite accurate. If the cameras are used in an extremely close proximity to the object, then $\hat{L}_i$ could conceivably be conditioned on the probability density function of the *a priori* estimates [42] to improve the estimate.

The time which each vision algorithm consumes consists of both a stochastic and a deterministic component. First, the cameras must be re-positioned to the i^{th} viewpoint. This is a pre-programmed movement, so it takes a deterministic quantity of time. Next, the measurement of the points must be made. Suppose that the total vision time for the i^{th} algorithm (which includes the camera positioning time, image processing time, and the time required for transformation to the end effector frame) is

$$t_{vi} \sim N(\mu_{t_{vi}}, \sigma^2_{t_{vi}}) \tag{5.49}$$

Once the pose statistics are found from (5.43) and those of timing are found from (5.49), the necessary information for the vision reliability analysis is complete.

In order to test the validity of the stochastic accuracy model, the vision system has been simulated using Pro-MATLAB and the parameters listed in Fig. 5.5. The model captures the vision system errors quite nicely over a wide range of operating

conditions. Figures B.4–B.6 plot the absolute value of the measurement error as the distance from the target to the camera is varied. Also plotted is the 3σ bound on the measurement error for each direction. The actual transformation from the target to the camera frame is

$$T_{tar}^{cam} = \begin{bmatrix} 0 & 0 & 1 & 0.02 \\ 0 & 1 & 0 & 0.02 \\ -1 & 0 & 0 & s \\ 0 & 0 & 0 & 1 \end{bmatrix} Rot(y, 0.1) \tag{5.50}$$

where $Rot(y, 0.1)$ denotes the homogeneous transformation rotating about the "y" axis by 0.1 radians with zero translation. and s varies between 0.1 and 3.5meters. The accuracy in the "x" direction is plotted in Fig. 5.6 for convenience. Note that the bound increases as the distance increases because the model incorporates the decrease in accuracy as distance is increased. Note also that the "z" error is two orders of magnitude larger than the "x" and "y" errors, yet the bound still closely follows the actual error. Figure B.7 plots the pose error at each distance weighted by the inverse covariance matrix. If the distribution of e_{cam} is zero mean Gaussian as assumed and the covariance matrix is correct, then the weighted error will have a chi-square distribution with 6 degrees of freedom. Finally, the entropy of the vision measurement is plotted in Fig. 5.7 As expected, the entropy increases as the distance between the camera and the target increases.

In order to fully test the strengths and limitations of the statistical model, measurements have been taken under a variety of conditions. First, the position and orientation of the target is varied. Figures B.8–B.10 are obtained with

$$T_{tar}^{cam} = \begin{bmatrix} 0 & 0 & 1 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & s \\ 0 & 0 & 0 & 1 \end{bmatrix} Rot(z, 0.3) Rot(y, 0.1) \tag{5.51}$$

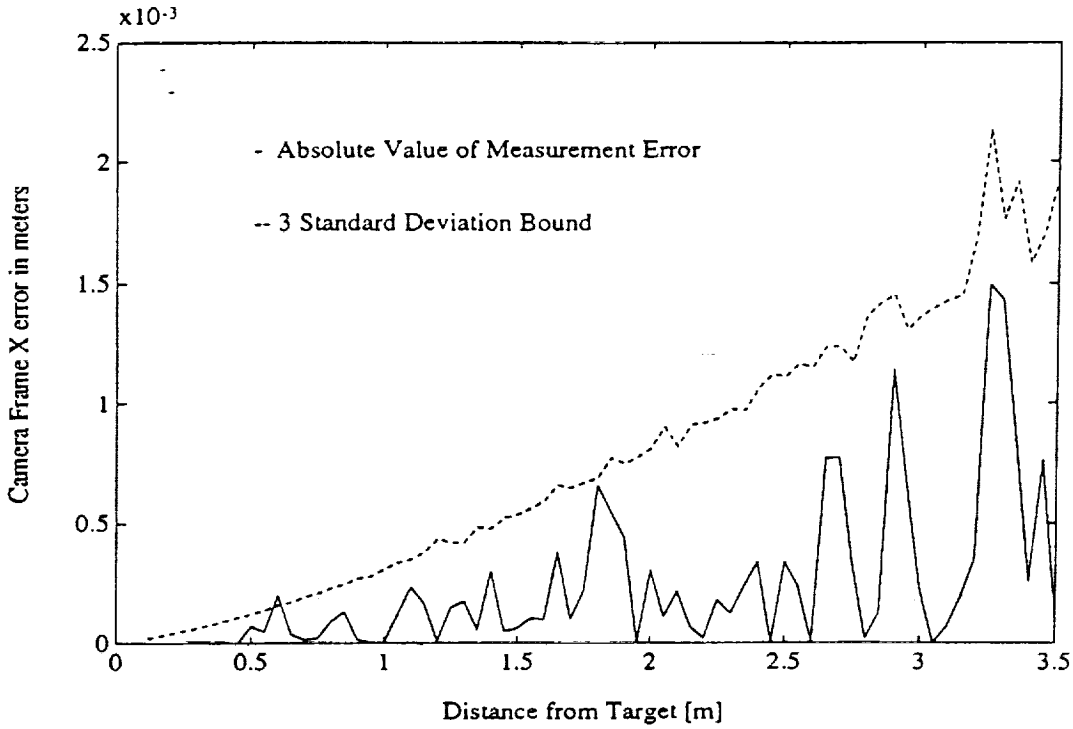


Figure 5.6: "X" Errors Due To Pixel Truncation

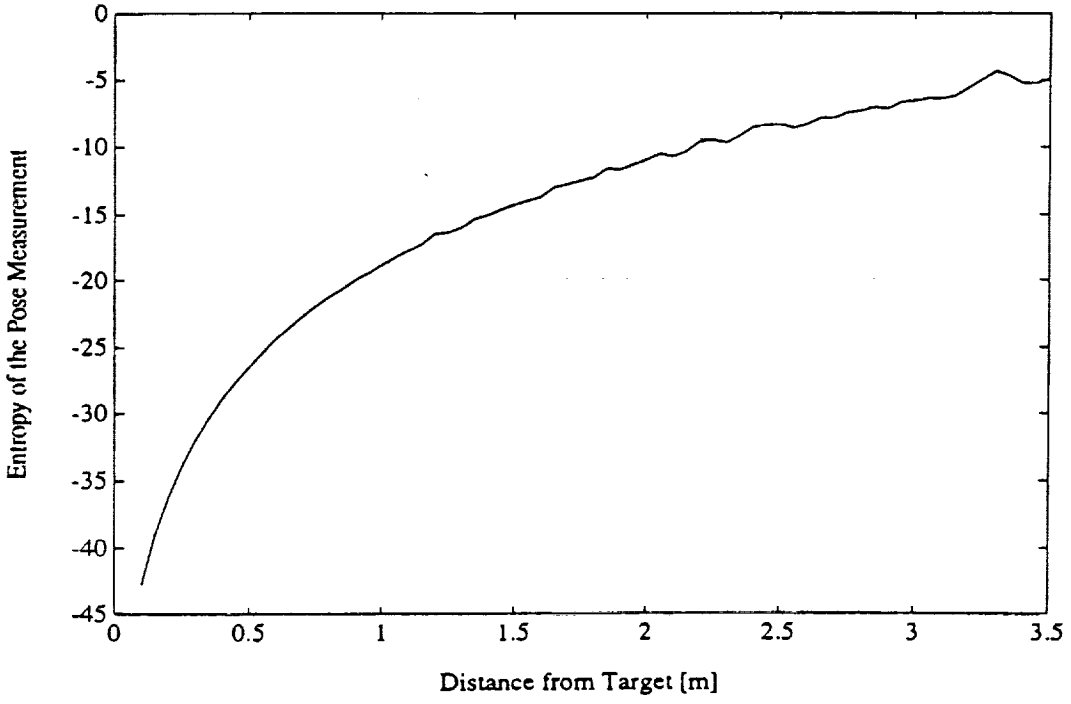


Figure 5.7: Joint Entropy of the Vision Measurement

where $Rot(z, 0.3)$ denotes the homogeneous transformation rotating about the "z" axis by 0.3 radians with zero translation, and s varies between 0.1 and 3.5m. The principle difference between the previous transformation is that the target is shifted 8cm further off the camera axis in both the "x" and "y". As expected, the shift causes a decrease in accuracy in the "x" and "y". Fortunately, the statistical model captures this phenomena, so the 3 standard deviation bound increases. Naturally, information of this sort is very useful for decision making in Intelligent Machines.

To further test the model, the pose errors are shown in Figures B.11–B.13 under conditions specifically selected to bring out the unmodeled measurement bias caused by approximating the center of the centroid to correspond to the center of the retroreflector. This is an approximation which is very often made in computer vision algorithms, and is very accurate as long as the object is much farther away than one focal length, and the angle between the image plane and the object is small. To bring out the effects, the camera is moved very close to the target, and the target is rotated sharply. The transformations are

$$T_{tar}^{cam} = \begin{bmatrix} 0 & 0 & 1 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & s \\ 0 & 0 & 0 & 1 \end{bmatrix} Rot(z, 0.6) Rot(y, 0.9) \tag{5.52}$$

where s varies between 2cm and 0.5 meters. Although the errors remain small, the 3σ bound is exceeded regularly due to the additional error arising from the centroidal bias. This should not introduce any significant problem in practice as long as the viewpoints are chosen to avoid these awkward poses.

To test the sensitivity of the statistical model to uncertainties in parameters assumed to be deterministically known, a series of chi-square goodness of fit statistics have been found as erroneous vision system parameters are used in measurement of the target's pose. First, the target length, d_1, is perturbed to ± 5 % of its nominal

value. The results of the chi-square test are shown in Figure B.14. The probability that the sample would arise from a Gaussian distribution drops off sharply in the range of ± 1 %. Thus d_1 should be known to an accuracy of at least ± 1 %. The target depth, d_2, displays a similar sensitivity. The results in Figure B.14 indicate that d_2 should be known to an accuracy of at least ± 2 %. The focal length, on the other hand, requires more precise information. As indicated in Figure B.15, the focal length should be known to within ± 0.2 %. Consequently, it is advisable to include focal length uncertainties by using (5.42).

The simulation results indicate that the stochastic error model for the vision system is very good at characterizing the error sources arising from pixel truncation. This information will soon be put to use in selecting reliable plans.

5.2 Analysis of the Kinematics Routines

In order to act upon the vision system's perception, it is necessary to use an inverse kinematics routine to calculate the joint position corresponding to the Cartesian object pose. This step is required because a joint-level computed torque algorithm is used for servoing. Several inverse kinematics algorithms have been developed, and the choice of which algorithm to use depends upon the time available and the accuracy required.

For this example, five inverse kinematics routines will be considered. The first method is probably the most commonly used inverse kinematics technique-simply use the nominal link parameters provided by the manufacturer in the standard serial calculations such as presented in [11]. This algorithm will be denoted by K_1. The nominal link parameters are not exact due to manufacturing spreads, so error is introduced by the using these parameters in the inverse kinematics computation. Since many commercially available manipulators are designed explicitly to have closed solution inverse kinematics for the nominal link parameters, it will assumed

that the nominal manipulator possesses a closed form inverse kinematics solution. In the simulation, the kinematic model of the Unimation PUMA 560 will be used. Unfortunately, if a calibration technique such as that proposed by Hayati [16] is employed, the resulting calibrated link parameters may not have closed solution inverse kinematics. Consequently, to avoid the time consuming iterative schemes for finding the inverse kinematics, nominal kinematics may be used even if calibrated link parameters are available. To measure the positioning accuracy using K_1, the errors are measured over a large number of points in the workspace, and the maximum entropy Gaussian distribution with the sample mean and variance is applied. This technique will be used to assess the accuracy of all of the kinematic routines. For a more refined analysis, it is possible to estimate the link parameters as described in [16]. Next, through maximum likelihood estimation the positioning error (and its distribution) as a function of joint position is estimated [32]. In either case, since the nominal kinematics have a closed form solution, the time required to perform the computation is fixed.

The second algorithm, denoted as K_2, is very similar to K_1. The closed form solution is retained by calibrating the arm using Hayati's method, while constraining the calibrated link parameters. Since the solution is constrained to be of closed form, it results in larger errors than are obtained when the link parameters are not artificially inhibited. On the other hand, it shows some increase in accuracy over the nominal kinematics.

The third method, proposed by Hayati [16] and denoted as K_3, approximately updates the link parameters without iterative schemes by using the inverse manipulator Jacobian. Since the Jacobian is employed, the method is ill-conditioned in the neighborhoods of Jacobian singular points. Using singular value properties and maximum likelihood estimation, an asymptotically normal approximate upper bound on the normed positioning error can be found [32]. The bound can be used as

a conservative estimate of the positioning accuracy. Although this statistical model does capture singularity effects quite nicely, only statistics for the norm (and not individual elements) are derived. Since several inverse kinematics algorithms are available, the refined estimate is not needed because the other algorithms can be employed when in the vicinity of singular points. As a result, accuracy statistics obtained by the sample mean and variance (as above) will be used. As in the previous algorithms, solution time is stochastic only in that a few conditional loops may or may not be executed depending on the particular joint angle.

The fourth algorithm, denoted as K_4, makes use of Hayati's calibration technique to find the updated link parameters for the particular manipulator under consideration. As alluded to previously, the updated kinematics may fail to possess a closed form solution, so the iterative Newton–Raphson algorithm will be utilized (see [56] for details). Again, this algorithm offers a trade–off: increased accuracy for increased time. As before, the errors are measured, and a Gaussian distribution is used to model the error. Since the time for executing K_4 depends upon the number of iterations required, the execution time is stochastic. To model the statistics, the sample mean and variance of the execution time are found, and the most pessimistic Gaussian distribution is assumed as a conservative measure.

The fifth algorithm, K_5, also uses the calibrated link parameters. However, instead of using the Newton–Raphson technique, the Jacobi iterative method outlined in [56] is employed.

To summarize, five inverse kinematics algorithms are available which play time against accuracy. The algorithms are:

$$K_1 = \{\text{Inverse kinematics using nominal link parameters}\} \tag{5.53}$$

$$K_2 = \{\text{Calibrated kinematics constrained to a closed form}\} \tag{5.54}$$

$$K_3 = \{\text{Update kinematics using the manipulator Jacobian}\} \tag{5.55}$$

$$K_4 = \{\text{Newton inverse kinematics using updated link parameters}\} \quad (5.56)$$

$$K_5 = \{\text{Jacobi inverse kinematics using updated link parameters}\} \quad (5.57)$$

The choice of inverse kinematics routine depends upon the set of specifications, S_D. The specifications, in turn, interact with the statistical parameters of the algorithm. These statistical parameters will be denoted as

$$e_{ki} \sim N(0, C_{ki}) \quad (5.58)$$

$$t_{ki} \sim N(\mu_{t_{ki}}, \sigma^2_{t_{ki}}) \quad (5.59)$$

where e_{ki} is the end effector frame Cartesian pose error attributed to the i^{th} inverse kinematics algorithm, and t_{ki} is the time required for the inverse kinematics.

In addition to the inverse kinematic error present in the manipulating arm, forward kinematic errors are also present because the pose of the camera arm is not known exactly. These forward kinematic errors can be modeled in a very similar fashion to the inverse kinematic errors. Through sampled measurements, the covariance matrix of the zero mean forward kinematic error is estimated to be C_{FK}. Lacking any other information, the most conservative distribution is Gaussian. One major difference between the forward and inverse kinematic solutions is that the forward kinematic solution has a closed form for all joint positions. Consequently, rather than having a library of forward kinematic algorithms, it is only necessary to have a single forward kinematic algorithm and use the most accurate estimates of the link parameters.

To test the stochastic kinematic models, several tests have been performed. First, the normed pose error for three of the inverse kinematics algorithms is plotted in Figure 5.8. Note that the Jacobi iterative technique (K_5) produces the smallest error, while the Jacobian update method (K_3) yields a small error at most points, but actually exceeds the nominal (K_1) errors for three of the fifty random points.

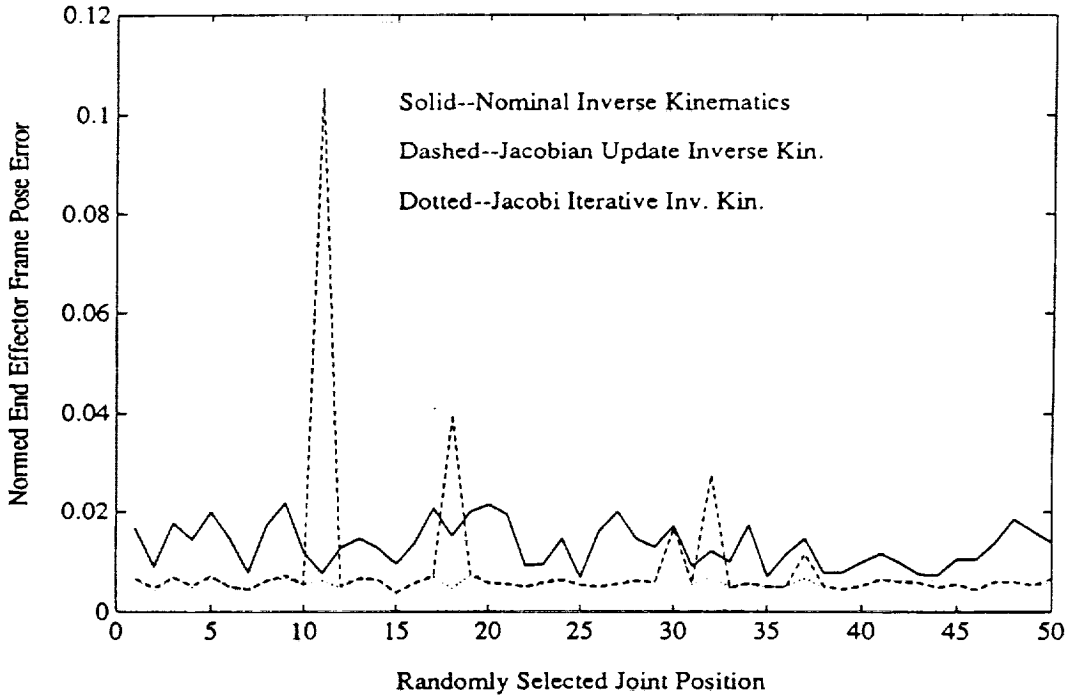


Figure 5.8: Normed Pose Error for 3 Inverse Kinematics Routines

The normed error and floating point operations required for all inverse kinematic algorithms is illustrated in Figures C.1–C.5. In addition, the 3σ bound on the floating point operations is also plotted. As expected, the most accurate algorithms are the iterative techniques, but they are also the most computationally intensive.

To test the assumption that the kinematic errors are Gaussian, the pose error for all of the kinematics routines is weighted by the inverse of the covariance matrix obtained through the sample statistics (Figures C.6–C.8). Weighting a Gaussian random vector by its inverse covariance matrix yields a chi–square distribution with degrees of freedom equal to the dimension of the vector. Thus the plots should display a chi–square distribution with six degrees of freedom. In addition to the weighted norm, Figures C.6–C.8 also plot the median value for the chi–square distribution with 6 degrees of freedom (5.6). Thus half of the norms should be above this mark, and half below. Chi–square goodness of fit tests indicate that the kinematic algorithms are nicely modeled by a Gaussian distribution with one notable exception. The vast majority of the weighted norms for the Jacobian update algorithm (K_3) are below the chi–square median. Thus a Gaussian distribution is a rather coarse and conservative model for K_3.

To test the effect of joint backlash upon the stochastic models, a backlash deadband of 0.25 degrees has been added to all joints of the PUMA, and the kinematic models have been found. The weighted norms are plotted in Figures C.9–C.11. As before, a zero mean Gaussian distribution fits well for all algorithms except K_3.

Finally, to examine the effect that joint noise and joint backlash has upon the entropy of the kinematic algorithms, the joint entropy of each kinematic algorithm is plotted as the noise and backlash is increased (Figures C.12–C.17). As expected, the entropy increases as either the joint noise or the backlash is increased.

5.3 Analysis of the Control System

To control the movements of the manipulator, a computed torque control law is used. This control law is based on the Euler-Lagrange dynamic equation:

$$\tau = D(\Theta)\ddot{\Theta} + NL(\Theta, \dot{\Theta}) \tag{5.60}$$

where

- τ is the torque applied by the motors,
- Θ is the joint position vector,
- $\dot{\Theta}$ is the joint velocity vector,
- $\ddot{\Theta}$ is the joint acceleration vector,
- $D(\Theta)$ is the inertia matrix,
- $NL(\Theta, \dot{\Theta})$ is a vector of torques due to Coriolis, gravity, centripetal, and friction nonlinearities.

An effective method of controlling an arm with dynamics of the form (5.60) is the computed torque technique described in [47], where the control function is defined as:

$$\tau = D(\Theta)E \tag{5.61}$$

where E is a compensation term. Note that the nonlinear torques due to Coriolis, gravity, centripetal and friction effects are not calculated in an open loop fashion as is often proposed for computed torque control. Instead, since the concern of this study is high precision movements, the movements are made over small distances at low velocities. Under these conditions, a PID controller can remove the nonlinear disturbances quite effectively. The nonlinear terms can, of course, be calculated if a particular application demands the extra level of performance.

Since $D(\Theta)$ is invertible due to the physical laws of inertia, this implies that

$$s^2\Theta = E(s) = G(s)(\Theta_d(s) - \Theta(s)) \tag{5.62}$$

where s denotes Laplace transformation, $G(s)$ is a compensator, and Θ_d is the desired position. The multivariable transfer function for the system is:

$$\Theta = (1/s^2)[I + (1/s^2)G(s)]^{-1}G(s)\Theta_d \tag{5.63}$$

The joint position, Θ, is subject to three sources of error:

1. Errors in the commanded position Θ_d, due to uncertainties in the vision and kinematic algorithms. These errors are not affected by the control algorithm because a PID control law is used, thus the expected value of the final Θ equals Θ_d.

2. Errors due to the nonlinear (Coriolis, etc.) disturbances. Because these disturbances are almost constant when moving over small distances at a low velocity, the PID compensator can remove these errors when it is operating at steady state. The settling time of the closed loop system, therefore, determines the time required to remove these effects.

3. Errors arising from joint position and velocity measurement noise. Although the expected value of these errors is zero, the variance depends upon the compensator used.

As the bandwidth of the closed loop system is increased, the settling time decreases. Unfortunately, the sensitivity of the system to noise is increased. The compensator thus has two conflicting criteria to meet, and the choice of compensator depends upon the specifications at hand. For the simulation, two compensators have been designed using state space techniques. A continuous time model of the plant

(the robot) can be written as:

$$\dot{x}_p = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x_p + \begin{bmatrix} 0 & -I \end{bmatrix} u \tag{5.64}$$

$$z = [I\ 0]x_p + v \tag{5.65}$$

where u is the compensation term (the inverse Laplace transform of E), v is the joint position sensor noise, and

$$x_p = \begin{bmatrix} \Theta_d - \Theta \\ \dot{\Theta}_d - \dot{\Theta} \end{bmatrix} \tag{5.66}$$

This model assumes that, since $D(\Theta)$ pre-multiplies E, the inertia term is canceled out, and $NL(\Theta, \dot{\Theta})$ is removed by the PID compensation at steady state.

An exact discretization of the plant then yields

$$x_p(k+1) = \begin{bmatrix} I & t_s I \\ 0 & t_s I \end{bmatrix} x_p(k) + \begin{bmatrix} -\frac{t_s^2}{2} I \\ -t_s I \end{bmatrix} u(k) \tag{5.67}$$

$$z(k) = [I\ 0]x_p(k) + v(k) \tag{5.68}$$

or

$$x_p(k+1) = A_p x_p(k) + B_p u(k) \tag{5.69}$$

$$z(k) = M_p x_p(k) + E_p v(k) \tag{5.70}$$

where t_s is the sampling time. The structure of the discrete PID compensator is

$$x_c(k+1) = \begin{bmatrix} 0 & I \\ 0 & I \end{bmatrix} x_c(k) + \begin{bmatrix} 0 \\ I \end{bmatrix} z(k) \tag{5.71}$$

$$u(k) = [K_v/t_s \quad -K_v/t_s + K_i t_s] x_c(k) + (K_v/t_s + K_p + t_s K_i) z(k) \tag{5.72}$$

or

$$x_c(k+1) = A_c x_c(k) + B_c z(k) \tag{5.73}$$

$$u(k) = C_c x_c(k) + D_c z(k) \tag{5.74}$$

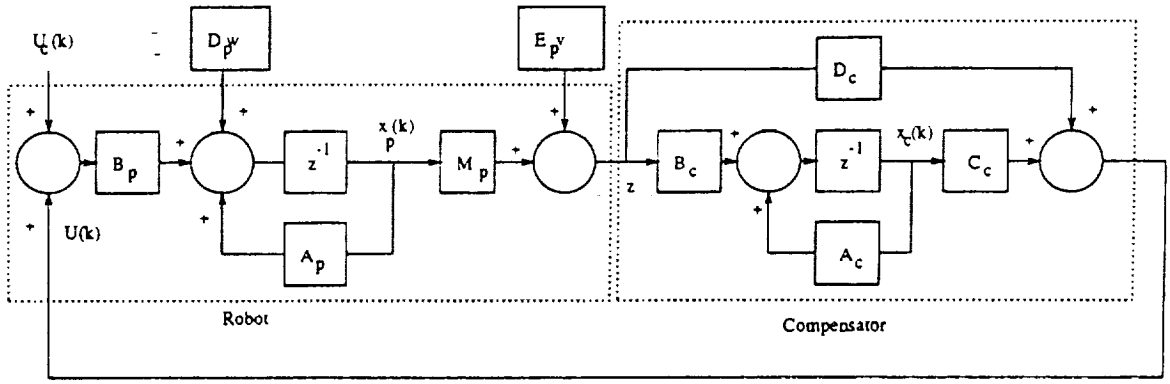


Figure 5.9: Block Diagram of the Discretized PID Controller

where the set (K_v, K_p, K_i) are the velocity, position, and integral of position gain matrices. The closed loop discrete system is illustrated in Figure 5.9 (neglecting dynamics due to sampling and a zero order hold). The figure includes process noise, w, which is not used in this study.

Without any compensation term, the linearized robot plant in Figure 5.9 is marginally stable. In order to use the Lyapunov design technique presented in Theorem 12, the system must be asymptotically stable [27]. The PID compensator makes the closed loop system asymptotically stable so that the Lyapunov design technique may be employed to yield an optimal control capable of reliably staying within the specified hyperellipsoid. In addition, the PID compensator removes positional errors due to constant disturbances.

To ensure reliable control, an outer loop gain is designed using the Lyapunov design technique (4.78). The outer loop is found by first formulating the closed loop system of Figure 5.9 as

$$x(k+1) = A_t x(k) + D_t v_t(k) + B_t u_c(k) \tag{5.75}$$

where

$$x(k) = [x_p^T(k)\ x_c^T(k)]^T \tag{5.76}$$

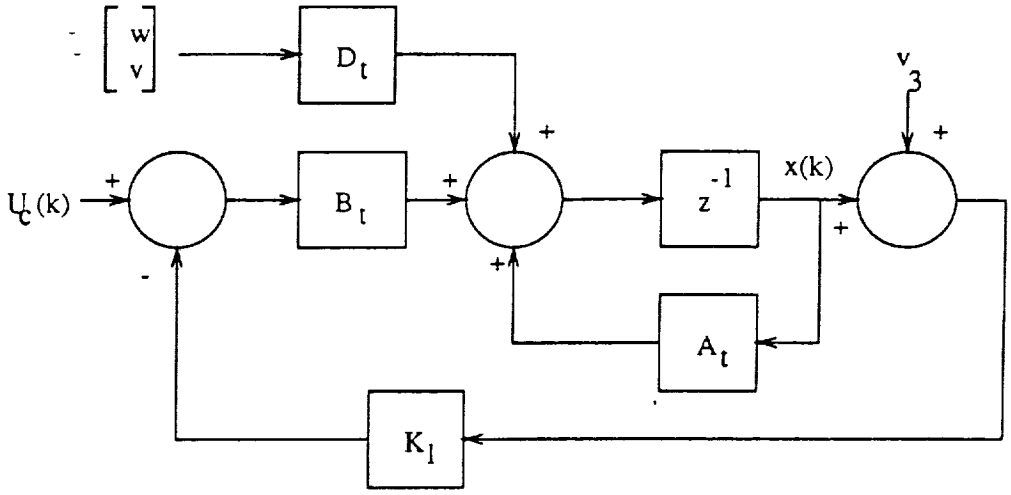


Figure 5.10: Block Diagram with Lyapunov Outer Loop

$$A_t = \begin{bmatrix} A_p + B_p D_c M_p & B_p C_c \\ B_c M_p & A_c \end{bmatrix} \tag{5.77}$$

$$D_t = \begin{bmatrix} D_p & B_p D_c E_p \\ 0 & B_c E_p \end{bmatrix} \tag{5.78}$$

$$v_t(k) = [w^T(k)\ v^T(k)]^T \tag{5.79}$$

$$B_t = \begin{bmatrix} B_p \\ 0 \end{bmatrix} \tag{5.80}$$

The outer loop is illustrated in Figure 5.10, where v_3 is the 24x1 vector of state measurement noise. Since the compensator states are known,

$$v_3 = [v^T\ v_1^T\ 0]^T \tag{5.81}$$

where v_1 is the joint velocity measurement noise. One distinct disadvantage of this control scheme is that full state feedback is required. In this case, it means that both the joint position and velocity must be measured. The Lyapunov gain, K_l, is found by first choosing a positive semi-definite matrix, Q_1, and then numerically

solving the algebraic Lyapunov equation

$$Q - A_t^T Q A_t = Q_1 \tag{5.82}$$

for Q. Note that, from Lemma 10, Q must satisfy

$$Q - Q_d \geq 0 \tag{5.83}$$

However, Q is in joint coordinates, while the desired weighting matrix (Q_{px}) is in Cartesian coordinates. Furthermore, Q_{px} penalizes only Cartesian pose, while Q is a weight on all of the state variables (this includes joint position and velocity as well as the compensator state variables). Fortunately, it is still a simple matter to find an appropriate Q_d. First, note that the manipulator Jacobian forms the first order term in the Taylor series expansion

$$x_{cart}(k) \approx x_{cart_d}(k) + J(\Theta_d(k))[\Theta(k) - \Theta_d(k)] \tag{5.84}$$

where $x_{cart}(k)$ is the actual Cartesian pose at sample k, $x_{cart_d}(k)$ is the desired Cartesian pose at sample k, $J(\cdot)$ is the manipulator Jacobian, $\Theta_d(k)$ is the joint position at sample k, and $\Theta(k)$ is the actual joint position at sample k. The accuracy of this approximation depends, of course, on the magnitude of the difference between the desired and actual positions. Since high precision motion control is of concern in this study, the difference is very small, so the approximation is very good. By combining (5.8) and (5.84), the weighting matrix can be projected from the Cartesian space into the joint space. Adding zero terms to account for those quantities that are not weighted produces the following matrix inequality which must be satisfied for all $\Theta_d(k)$ and k in the trajectory

$$Q - \begin{bmatrix} J^T(\Theta_d(k))Q_{px}(k)J(\Theta_d(k)) & 0 \\ 0 & 0 \end{bmatrix} \geq 0 \tag{5.85}$$

Choosing a Q_1 which will yield a Q satisfying (5.85) is not straightforward, and is a topic of further research. Fortunately, finding a Q_1 has not posed a problem

in this study, although the chosen Q_1 matrices are undoubtedly conservative. Once a satisfactory Q_1 has been found, the Lyapunov gain is from (4.78)

$$K_l = (B_t^T Q B_t)^{-1} B_t^T Q A_t \tag{5.86}$$

The final closed loop system is then

$$x(k+1) = A_{tl}x(k) + D_{tl}v_l(k) + B_t u_c(k) \tag{5.87}$$

where

$$A_{tl} = A_t - B_t K_l \tag{5.88}$$

$$D_{tl} = \begin{bmatrix} D_p & B_p D_c E_p & 0 \\ 0 & B_c E_p & 0 \end{bmatrix} - \begin{bmatrix} 0 & B_p K_{ll} \\ 0 & 0 \end{bmatrix} \tag{5.89}$$

$$v_l = [w^T \; v^T \; v_1^T]^T \tag{5.90}$$

with K_{ll} a matrix of the first twelve columns of K_l. The steady state covariance of x, C_{ctl}, is found from (5.87) by solving the discrete Lyapunov equation

$$C_{ctl} = A_{tl} C_{ctl} A_{tl}^T + D_{tl} C_{vl} D_{tl}^T \tag{5.91}$$

where C_{vl} is the covariance matrix of v_l. This equation can be easily solved numerically.

Two compensators have been developed for use in this study which form a basic trade-off between time and accuracy. Both algorithms have a sampling time of $t_s = 10ms$, have Gaussian joint position noise with standard deviation of 0.167 degrees, and Gaussian joint velocity noise with standard deviation 2.4 deg/sec. These large standard deviations are used to highlight effects due to noise. In the simulation of the combined vision/kinematic/control system, the noise levels are at more realistic levels for the PUMA. The joint position noise then has a standard deviation of 0.0033 degrees, while the joint velocity noise has standard deviation of 0.047 deg/sec. The first compensator, denoted by G_1, has the following PID gain matrices

$$K_v = 15I \tag{5.92}$$

C-2

$$K_p = 75I \tag{5.93}$$

$$K_i = 125I \tag{5.94}$$

By choosing Q_1 matrices and evaluating (5.85) over the desired trajectory, it has been determined that $Q_1 = 50I$ will produce a satisfactory K_l via (5.82) and (5.86). The final closed loop system has a settling time (determined by examining plots of step responses) of 3 seconds.

A second compensator (G_2) has been designed by using the PID gains

$$K_v = 30I \tag{5.95}$$

$$K_p = 300I \tag{5.96}$$

$$K_i = 1000I \tag{5.97}$$

It has been found that $Q_1 = 100I$ will produce a Q and K_l satisfying the Lyapunov matrix inequality (5.85). This system yields a settling time of 1 second.

The position error for joint 1 of the robot as it moves through the desired trajectory is plotted in Figure 5.11. Note that G_1 has a longer settling time than G_2, but it also has a smaller steady state covariance. Thus the choice of which algorithm to use depends upon the task at hand. The settling time of the algorithm affects the execution time in two ways:

1. In order to use the reliable control techniques developed in chapter 4, the response must be at steady state. Thus the trajectory is scaled so that the manipulator is able to achieve a steady state response to the constant Cartesian velocity command before a critical region (as defined by the tolerance specifications, Figures B.1–B.3) is entered. The constant Cartesian velocity command does not, of course, yield a constant joint velocity command. However, since the movement is over a small distance and is kept away from manipulator

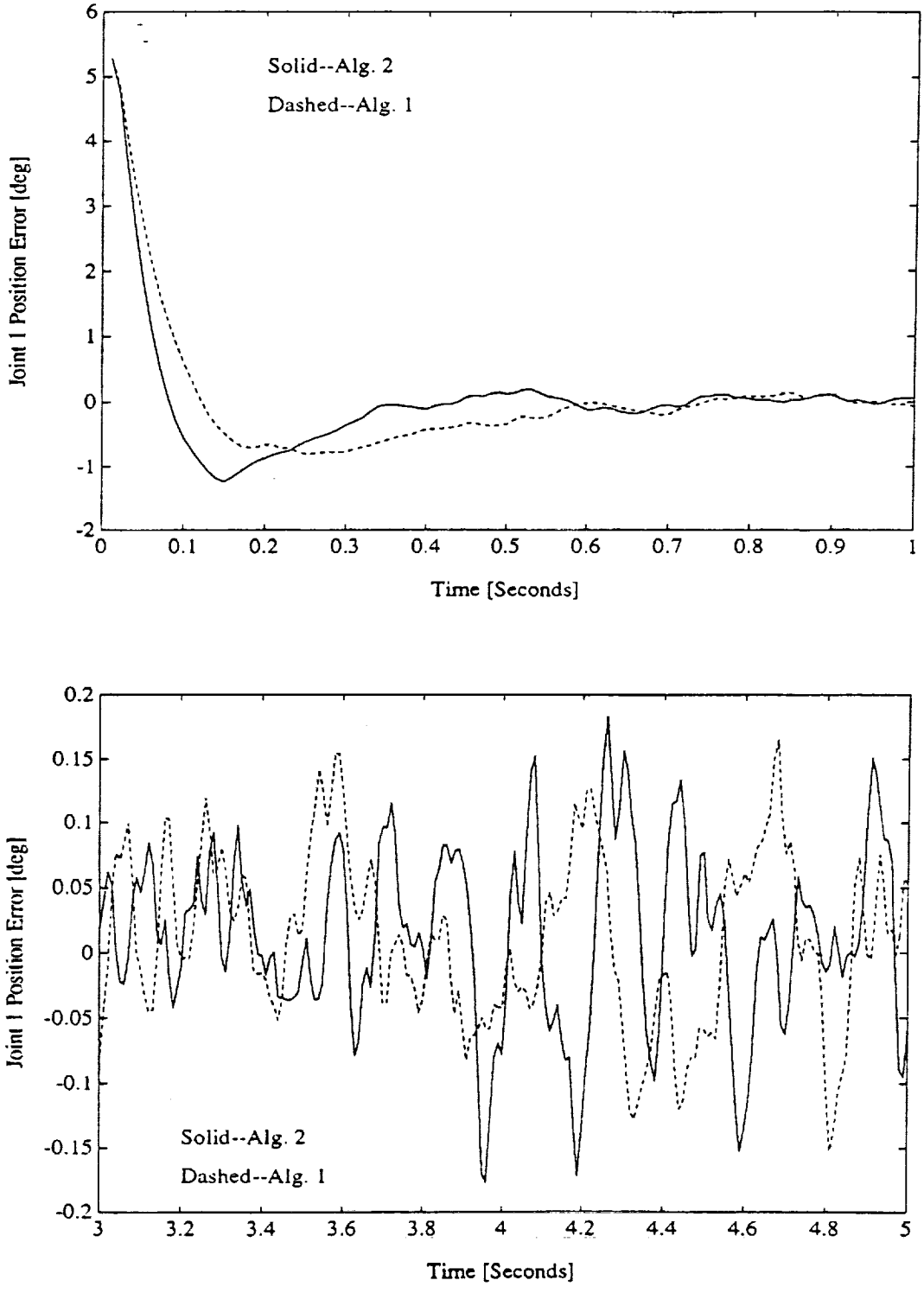


Figure 5.11: Joint Position Responses for Compensators G_1 and G_2

Jacobian singularities, the Jacobian relating joint velocities to Cartesian velocities changes very little during the movement. The joint velocity command is, therefore, almost constant over the fine motion trajectory.

2. A faster control law requires fewer sampling times in the trajectory. Because a joint level control law is used, the inverse kinematics must be solved at each sample time. Thus the slower trajectory of a slow compensator requires more calculations before movement can begin.

The steady state covariance, on the other hand, adds to the uncertainty in maintaining a desired pose of the gripper with respect to the gripping post. In contrast to the pose errors introduced by vision errors and faulty kinematics, which add a constant bias error in the commanded pose, the errors arising from joint position and velocity noise change as the manipulator moves. In order to apply Theorem 9, the steady state value of the pose error inverse covariance matrix must be greater than or equal to $Q_{px}(\Theta_d(k))$. To find the pose error covariance, the joint position covariance, C_{pq}, is first determined from C_{ctl} by extracting the position covariance terms. Next, C_{pq} is projected into Cartesian space using (5.84)

$$C_{px} = J(\Theta_d(k))C_{pq}J(\Theta_d(k))^T \tag{5.98}$$

The resulting matrix inequality is

$$[J(\Theta_d(k))C_{pq}J(\Theta_d(k))^T]^{-1} - Q_{px}(\Theta_d(k)) \geq 0 \tag{5.99}$$

The matrix inequality (5.99) holds for both compensators over the desired trajectory. Since both (5.85) and (5.99) hold over the trajectory of interest, Theorem 9 may be employed. In the case study, $\epsilon^2 = 22$, thus $\chi_6(\epsilon^2) = 0.9988$. For a two second trajectory, 200 samples are generated ($t_s = 10ms$), therefore the lower bound reliability of staying within the hyperellipsoid defined by the most restrictive $Q_{px}(\Theta_d(k))$ is

$$R \geq (.9988)^{200} = 0.79 \tag{5.100}$$

Figure 5.12 plots the weighted square norm of both compensators when using the most restrictive weighting matrix (Q_{px}) generated by the tolerance specifications. The weighted square norm stays well within the hyperellipsoid, because its maximum value is on the order of 2, while the hyperellipsoid includes all weighted square norms less than or equal to 22.

5.4 Selection of Feasible Plans

Because the total number of possible plans for accomplishing the task is $5N$xM (where $N = 10$ and $M = 2$), detailed analysis of every possibility can be very time consuming. To avoid a complete analysis for all possibilities, a subset of the plans (the feasible plans) will be found which satisfy necessary, but not sufficient, conditions for reliable operation as described in chapter 2.

Consider the individual tasks which must be performed in order to solve the given problem. First, the cameras must be moved to one of the N viewpoints. For the simulation, ten viewpoints are specified. The initial pose of the camera manipulator with respect to the nominal target frame is known from *a priori* information to be (see Figure 5.3)

$$T_{tar}^{cam} = \begin{bmatrix} 0 & 0 & 1 & x \\ 0 & 1 & 0 & y \\ -1 & 0 & 0 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.101}$$

where $x = 0$, $y = 0$, and $z = 1m$. Each viewpoint is considered to be another vision algorithm, so denote the initial viewpoint by V_1. Four more viewpoints ($V_2 - V_5$) are found by decreasing z (this means the camera must be moved closer to the target) in $20cm$ decrements. Finally, five more viewpoints ($V_6 - V_{10}$) are found by letting $x = 20cm$, $y = 10cm$, and again varying z in $20cm$ decrements from $1m$ to $20cm$. For the sake of brevity, it is assumed that the camera manipulator can

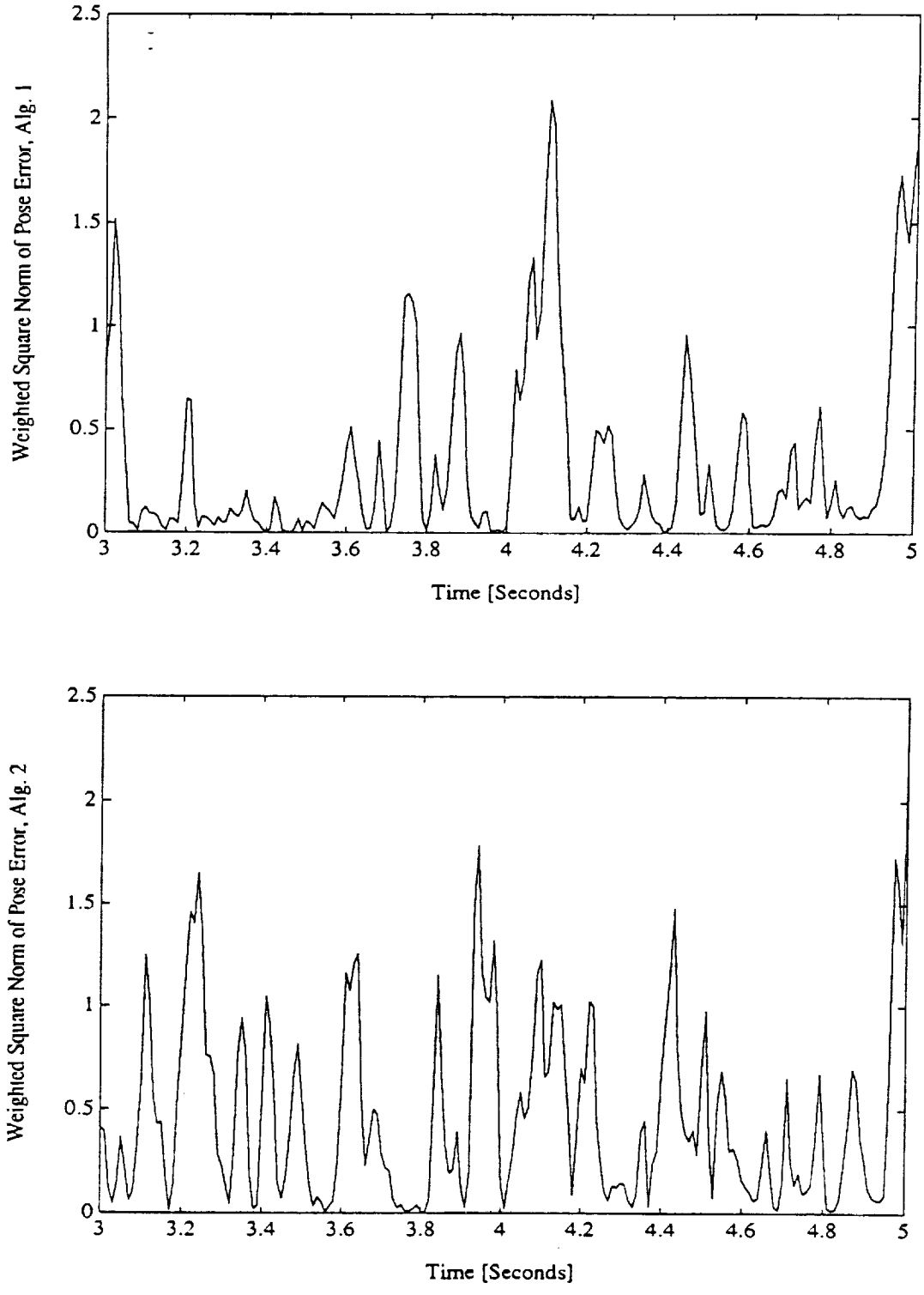


Figure 5.12: Weighted Square Norms of G_1 and G_2

T_{tar}^{cam} for V_1	T_{tar}^{cam} for V_2	T_{tar}^{cam} for V_3	T_{tar}^{cam} for V_4
$\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1. \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
T_{tar}^{cam} for V_5	T_{tar}^{cam} for V_6	T_{tar}^{cam} for V_7	T_{tar}^{cam} for V_8
$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & .2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & 1. \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
T_{tar}^{cam} for V_9	T_{tar}^{cam} for V_{10}		
$\begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 0 & 1 & 0 & 0.1 \\ -1 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$		

Table 5.1: The Viewpoints Used in the Simulation

move at $0.2m/sec$ from the initial camera position towards other viewpoints. In other words, the PUMA 560 dynamics are not included in the camera manipulator as a simplification. The dynamics are included in the gripping manipulator, as high precision motion is required. The viewpoints, T_{tar}^{cam}, are listed in Table 5.1.

From the viewpoint, image processing is used to measure the object's position in the camera coordinate system. The measurement will, of course, be imperfect due to image noise, and it will take a variable length of time due to the iterative nature of the solution. The measured pose, $T_{tar_m}^{cam}$, is transformed to the end effector frame using

$$T_{grip_m}^{barm} = T_{bcam}^{barm} T_{cam_m}^{bcam} T_{tar_m}^{cam} T_{grip}^{tar} \tag{5.102}$$

where the subscript m denotes "measured". Both T_{bcam}^{barm} (the transformation between the two manipulator base frames) and T_{grip}^{tar} (the object grip to target transformation) are assumed to be perfectly known as they are fixed. Hence the accuracy of $T_{grip_m}^{barm}$ depends upon the forward kinematic accuracy of the camera manipulator as well as

the accuracy of the vision system.

A Cartesian space trajectory is then generated which will drive the end effector from the initial pose to the grip pose. The trajectory is found by first calculating from the known initial end effector pose the grip to end effector frame

$$T^{ee}_{grip_m} = T^{ee}_{barm} T^{barm}_{grip_m} \tag{5.103}$$

Because the nominal scenario requires a 20cm movement along the end effector's "z" frame, a set of desired grip to end effector frames is specified to be

$$T^{ee}_{grip_d}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & p_z - \frac{kp_z}{N_{ctl_i}-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.104}$$

where $p_z = T^{ee}_{grip_m}(3,4)$ is the measured distance along the end effector's "z" frame, k is the sample number, and N_{ctl_i} is the total number of samples used by the i^{th} control algorithm. N_{ctl_i} is determined from the geometry of the gripper to gripping post interface and the settling time of the controller. The control should achieve a steady state response to the step velocity command before any fine motion volume is entered. For the geometry given in Figures 5.1 and 5.2, the fine motion begins when the distance along the end effector's "z" axis is less than $2l_4 + l_3 = 10cm$. After finding the measured initial distance (p_z), a constant velocity command for the i^{th} compensator is found to be

$$v_{zi} = \frac{p_z - (2l_4 + l_3)}{t_{set_i}} \tag{5.105}$$

where t_{set_i} is the settling time of the i^{th} compensator, G_i. This choice of v_{zi} allows the controller to reach steady state before the fine motion volume is entered. The number of samples required for control when using G_i is then

$$N_{ctl_i} = p_z / v_{zi} \tag{5.106}$$

The number of samples required to achieve steady state is

$$N_{ctlss_i} = t_{set_i}/t_s \tag{5.107}$$

The set of desired $T^{ee}_{grip_d}(k)$ frames is used to calculate a desired set of end effector frames

$$T^{barm}_{ee_d}(k) = T^{barm}_{grip_m} T^{ee^{-1}}_{grip_d}(k) \tag{5.108}$$

The time required to perform the forward kinematic calculations for a single $T^{barm}_{ee_d}(k)$ is modeled as

$$t_{FK} \sim N(\mu_{t_{FK}}, \sigma^2_{t_{FK}}) \tag{5.109}$$

Based on the desired frames, an inverse kinematics algorithm is selected to find a joint trajectory for servoing the manipulator toward, i.e.

$$\Theta_d(k) = K_i[T^{barm}_{ee_d}(k)] \tag{5.110}$$

where K_i specifies the i^{th} inverse kinematics algorithm. The inverse kinematics introduces stochastic position errors and requires a stochastic period of time. Finally, the joint servoing of the motors using the computed torque technique is performed.

The first specification, s_1, constrains execution time. The total execution time for plan $A_i = \{V_i, K_i, G_i\}$ is

$$t_i = t_{vi} + N_{ctl_i}(t_{FK} + t_{ki} + t_s) \tag{5.111}$$

Assuming independence (5.49), (5.59), and (5.109) yield

$$E\{t_i\} = \mu_{t_i} = \mu_{t_{vi}} + N_{ctl_i}(\mu_{t_{ki}} + \mu_{t_{FK}} + t_s) \tag{5.112}$$

$$Cov(t_i) = \sigma^2_{t_{vi}} + N^2_{ctl_i}(\sigma^2_{t_{FK}} + \sigma^2_{t_{ki}}) \tag{5.113}$$

To produce an entropy specification with zero mean, the timing specification will be considered as bounds $(-[t_f - \mu_{t_i}], [t_f - \mu_{t_i}])$. The quantity $[t_f - \mu_{t_i}]$ is positive for reliable systems because otherwise the mean time is less than the maximum

execution time, so the resulting entropy from Table 2.1 is $H(s_{1_i}) = \ln(2[t_f - \mu_{t_i}])$. The entropy of (5.113) is $H(A_{i_1}) = \ln\sqrt{2\pi e(\sigma^2_{t_{v_i}} + N^2_{ctl_i}(\sigma^2_{t_{FK}} + \sigma^2_{t_{k_i}}))}$ As a result, the timing constraints will be met if

$$\ln\sqrt{2\pi e(\sigma^2_{t_{v_i}} + N^2_{ctl_i}(\sigma^2_{t_{FK}} + \sigma^2_{t_{k_i}}))} \leq \ln(2[t_f - \mu_{t_i}]) \tag{5.114}$$

is satisfied.

Using a final time specification of $25sec$, and determining the execution time by counting the number of floating point operations, the time feasibility analysis has been performed. From the original set of 100 plans, 58 of the plans have been selected as feasible in meeting the timing specification. The plans selected as feasible are intuitively pleasing and are listed in Table 5.2. Each plan in Table 5.2 is expressed as a triplet, $A = \{V_i, K_i, G_i\}$. The integer V_i denotes the vision algorithm (viewpoint) from Table 5.1. The integer K_i denotes the inverse kinematics algorithm from (5.53–5.57). Finally, the integer G_i denotes the compensator used, G_1 or G_2. Note that those plans which require large amounts of time are not included. For instance, $A = \{10, 4, 1\}$, uses the viewpoint farthest from the initial camera position (V_{10}), the most computationally intensive inverse kinematics routine (K_4), and the slowest compensator (G_1), so it is not included. In fact, no plan which uses the Newton iterative inverse kinematics, K_4, is selected as feasible because the algorithm requires too many calculations.

In order to reduce the number of plans considered to be feasible, the ability of the plan in meeting the restriction imposed by Lemma 11 is checked at the most constraining weighting matrix. Since, from (5.85), $Q_{sub}(k) = Q_{px}(\Theta_d(k))$. Lemma 11 implies that the inverse of the pose covariance matrix minus $Q_{px}(\Theta_d(k))$ must be positive semi–definite. As the gripper moves closer to the gripping frame, the tolerance to positional errors becomes smaller (see Figures B.1–B.3). This in turn causes an increase in $Q_{px}(\Theta_d(k))$ in accordance with (5.9). The largest weighting

V_i	K_i	G_i	V_i	K_i	G_i	V_i	K_i	G_i
1	1	1	4	2	1	7	5	2
1	1	2	4	2	2	8	1	1
1	2	1	4	3	2	8	1	2
1	2	2	4	5	2	8	2	1
1	3	2	5	1	1	8	2	2
1	5	2	5	1	2	8	3	2
2	1	1	5	2	1	8	5	2
2	1	2	5	2	2	9	1	1
2	2	1	5	3	2	9	1	2
2	2	2	6	1	1	9	2	1
2	3	2	6	1	2	9	2	2
2	5	2	6	2	1	9	3	2
3	1	1	6	2	2	9	5	2
3	1	2	6	3	2	10	1	1
3	2	1	6	5	2	10	1	2
3	2	2	7	1	1	10	2	1
3	3	2	7	1	2	10	2	2
3	5	2	7	2	1	10	3	2
4	1	1	7	2	2			
4	1	2	7	3	2			

Table 5.2: Plans Feasible in Meeting the Timing Specification

occurs when the trajectory reaches its final point because the end effector and gripping frames should coincide in the ideal case. Thus $Q_{px}(\Theta_d(N))$, where N is the final sample, is the most constraining weighting matrix.

The pose covariance matrix has terms arising from four sources: the vision system errors, forward kinematic errors, inverse kinematic errors, and joint control errors. In order to propagate the vision system and forward kinematic errors to the end effector frame, (5.102) and (5.103) are combined to produce the pose of the gripping post with respect to the end effector

$$T^{ee}_{grip_m} = T^{ee}_{barm} T^{barm}_{bcam} T^{bcam}_{cam_m} T^{cam}_{tar_m} T^{tar}_{grip} \tag{5.115}$$

This pose can be represented (excepting kinematic singularity points) as a 6 x 1 vector by using (5.30)–(5.32). Let

$$x^{ee}_{grip} = \Gamma^{ee}_{cam}(x^{cam}_{grip}) \tag{5.116}$$

be the nonlinear function mapping the pose of the gripping post in the camera frame to the pose of the gripping post in the end effector frame. Then by retaining the first order terms in a Taylor series expansion of (5.116), the covariance matrix in the end effector frame due to the vision and forward kinematic errors is from (2.14) [8]

$$J^{ee}_{cam}(\Theta_d(k))(C_{vi} + C_{FK})J^{ee^T}_{cam}(\Theta_d(k)) \tag{5.117}$$

where $J^{ee}_{cam}(\cdot)$ is the interlink Jacobian matrix from the camera frame to the end effector frame (see Equation (2.12)). The covariance due to inverse kinematic errors, C_{ki}, is measured in the end effector frame, so no transformation is necessary. Finally, the control errors in joint space, C_{pq}, are projected to the end effector frame (to first order) using (5.98). The total covariance of the gripping post's pose with respect to the end effector for the i^{th} plan is then

$$C^{ee}_{grip}(i, \Theta_d(k)) = J^{ee}_{cam}(\Theta_d(k))(C_{vi}+C_{FK})J^{ee^T}_{cam}(\Theta_d(k))+C_{ki}+J(\Theta_d(k))C_{pq}J(\Theta_d(k))^T \tag{5.118}$$

	V_i	K_i	G_i	V_i	K_i	G_i	V_i	K_i	G_i	V_i	K_i	G_i
Plan	4	5	2	9	5	2	10	2	1	10	2	2
R_{i1}	0.9919			0.9901			1.0			1.0		
$I(R_{i1})$	0.0081			0.0099			2.8e-8			2.2e-16		

Table 5.3: Feasible Plans

From Theorem 2, if

$$C_{grip}^{ee^{-1}}(i, \Theta_d(k)) - Q_{px}(\Theta_d(k)) \geq 0 \tag{5.119}$$

then a lower bound reliability of the i^{th} plan staying inside the specified hyper-ellipsoid at sample k can be found. Since the covariance matrices are constants multiplied by Jacobian matrices, and the movement is over a small distance away from Jacobian singularities, $C_{grip}^{ee^{-1}}(i, \Theta_d(k))$ is almost constant. Consequently, to assess positional feasibility, those plans which are feasible in meeting the timing specification are tested to see if

$$C_{grip}^{ee^{-1}}(i, \Theta_d(0)) - Q_{px}(\Theta_d(N)) \geq 0 \tag{5.120}$$

is satisfied. If (5.120) is true, then the plan is considered to be a feasible plan, and further analysis is performed to determine if the plan will execute reliably. The four plans deemed to be feasible are listed in Table 5.3. Note that the feasible plans require moving the camera much closer to the target (from 1m to within 0.4m). In addition, note that it is possible to trade the increased inverse kinematic accuracy present in the Jacobi iterative technique (K_5) for extra accuracy in the vision system by moving from viewpoint 9 to viewpoint 10. It is not, on the other hand, possible to use K_5 from viewpoint 10 because the available time is too short.

5.5 Calculation of Reliability

The entropy constraints generated in the last section implicitly assure a level of reliability. In order to explicitly find the reliability for a particular plan, further analysis is warranted.

First, the specifications (S_D) are used in conjunction with the mathematical laws governing the system to produce reliability performance functions corresponding to each algorithm. The timing RPF for the i^{th} plan follows in a straightforward manner:

$$g_{i1} = t_f - t_i \tag{5.121}$$

Since t_i consists of a linear combination of times, using (5.111) the reliability index corresponding to g_{i1} can be found from (3.10). The timing reliability index is thus:

$$\beta_{i1} = \frac{t_f - [\mu_{t_{v_i}} + N_{ctl_i}(\mu_{t_{k_i}} + \mu_{t_{FK}} + t_s)]}{\sqrt{\sigma^2_{t_{v_i}} + N^2_{ctl_i}(\sigma^2_{t_{FK}} + \sigma^2_{t_{k_i}})}} \tag{5.122}$$

By using a look up table for the zero mean normal cumulative distribution function, the reliability can easily be found from (3.11). Moreover, the RSI is found from (3.16):

$$I(R_{i1}) = -\ln \Phi [\beta_{i1}] \tag{5.123}$$

Note that all of the reliability self information terms for execution time can be determined off-line. This is quite a time saving feature because it allows many of the unreliable plans (i.e. plans with large $I(R_{i1})$ terms) to be identified before execution begins. The timing reliabilities and RSI terms for the feasible plans are listed in Table 5.3.

In order to determine which plans are not only feasible, but are in fact reliable at meeting the desired set of specifications, S_D, it must first be ensured that the conditions of Lemma 11 are satisfied. For this case study, the conditions become,

	V_i	K_i	G_i	V_i	K_i	G_i	V_i	K_i	G_i
Plan	4	5	2	9	5	2	10	2	1
Time Relia.	0.9919			0.9901			1.0		
Time RSI	0.0081			0.0099			2.8e-8		
Accuracy Relia.	0.8555			0.8555			0.6261		
Accuracy RSI	0.1561			0.1561			0.4682		
Overall Relia.	0.8486			0.8470			0.6261		
Overall RSI	0.1642			0.1660			0.4682		

Table 5.4: Plans Satisfying the Matrix Inequalities

from (5.85) and (5.119)

$$Q - \begin{bmatrix} J^T(\Theta_d(k))Q_{px}(\Theta_d(k))J(\Theta_d(k)) & 0 \\ 0 & 0 \end{bmatrix} \geq 0 \tag{5.124}$$

$$C_{grip}^{ee^{-1}}(i, \Theta_d(k)) - Q_{px}(\Theta_d(k)) \geq 0 \tag{5.125}$$

These matrix inequalities must be satisfied for all of the desired joint positions in the fine motion portion of the trajectory, $\Theta_d(k)$, $k = N_{ctlss_i}, \ldots, N_{ctl_i}$. Since the fine motion requires between 100 and 400 samples, this reliability check is quite computationally intensive. However, the fine motion joint trajectory is over a very small range. As a consequence, a smaller number of samples can be checked as a very good approximation. In the simulation, 20 sample points evenly distributed in the fine motion trajectory are substituted into (5.124) and (5.125). Of the feasible plans, the plans listed in Table 5.4 satisfy the matrix inequalities at all 20 sample points. Since the reliable plans meet the conditions of Theorem 9, it is very easy to find a lower bound reliability of meeting the accuracy specification, s_2. Thus, for a single sample, $\epsilon^2 = 22$, $R_k \geq \chi_6(\epsilon^2) = 0.9988$. The probability of A_{feas_i} staying within s_2 during fine motion control has a lower bound of

$$R_{i2} \geq [0.9988]^{N_{ctl_i} - N_{ctlss_i}} \tag{5.126}$$

The RSI is

$$I(R_{i2}) = -(N_{ctl_i} - N_{ctlss_i}) \ln(\chi^2_6(22)) \tag{5.127}$$

Equations (5.123) and (5.127) are explicit functions relating the i^{th} feasible subset of algorithms, A_{feas_i}, and the environment and design stochastic variables to the RSI. The equations must now be combined to form one total RSI for each subset of algorithms.

Since all of the specifications must be satisfied to ensure that the desired standards are met and the specifications are independent, the system forms a series connection. The total RSI for a particular subset of algorithms, A_{feas_i}, is found from (3.20) to be the sum of the RSI's for each specification, i.e.

$$I(R_i) = I(R_{i1}) + I(R_{i2}) \tag{5.128}$$

A threshold can be set on $I(R_i)$ to discriminate reliable from unreliable plans. Those plans with a total reliability self information less than the threshold will then be considered reliable. In this manner, a specified level of reliability in meeting the desired requirements can be achieved. For instance, suppose a reliability of 0.75 is desired. The RSI threshold is then 0.2877. From Table 5.4, the RSI for $A = (10, 2, 1)$ is 0.4682. thus $A = (10, 2, 1)$ is not considered to be a reliable plan. On the other hand, both plans $(4, 5, 2)$ and $(9, 5, 2)$ have acceptable overall RSI values and will therefore execute reliably.

5.6 Comments

To test the results of the analysis, several of the plans have been simulated using the vision system depicted in Figures 5.4 and 5.5, PUMA 560 kinematics for both the camera and end effector manipulator, and PUMA 560 dynamics for the end effector manipulator. The pose errors for plan $(4, 5, 2)$ are plotted in Figures 5.13–5.15. In addition, the upper and lower tolerance bounds from Figures B.1–B.3

are also plotted, as dashed and dotted lines, respectively. The errors in all directions stay well within the tolerance bounds, and the plan is executed within the timing constraint, so the plan is successful in achieving the task at hand. Similar results are obtained for the second reliable plan, $(9, 5, 2)$. The results are illustrated in Figures D.1–D.3. To check repeatability, the plans are repeatedly tested. Both plans have reliably executed the entire task twenty consecutive times.

In contrast, $A = (1, 1, 1)$ is not selected by the analysis as a reliable plan. The results of this plan are depicted in Figures D.4–D.6. In this case, both the "Y" and "Z" errors exceed the tolerances.

As these examples illustrate, the reliability analysis techniques developed herein are very effective at selecting reliable plans from a set of plans. In addition, a means is illustrated whereby the positioning tolerances arising from the geometry of a gripping interface can be recast as a position–varying hyperellipsoid specification. Since the case study example is a very high fidelity simulation of realistic multi–arm robotic systems, the algorithms analyzed are fully capable of being implemented on standard robotic hardware. Thus the results of the case study may be directly applied to actual systems.

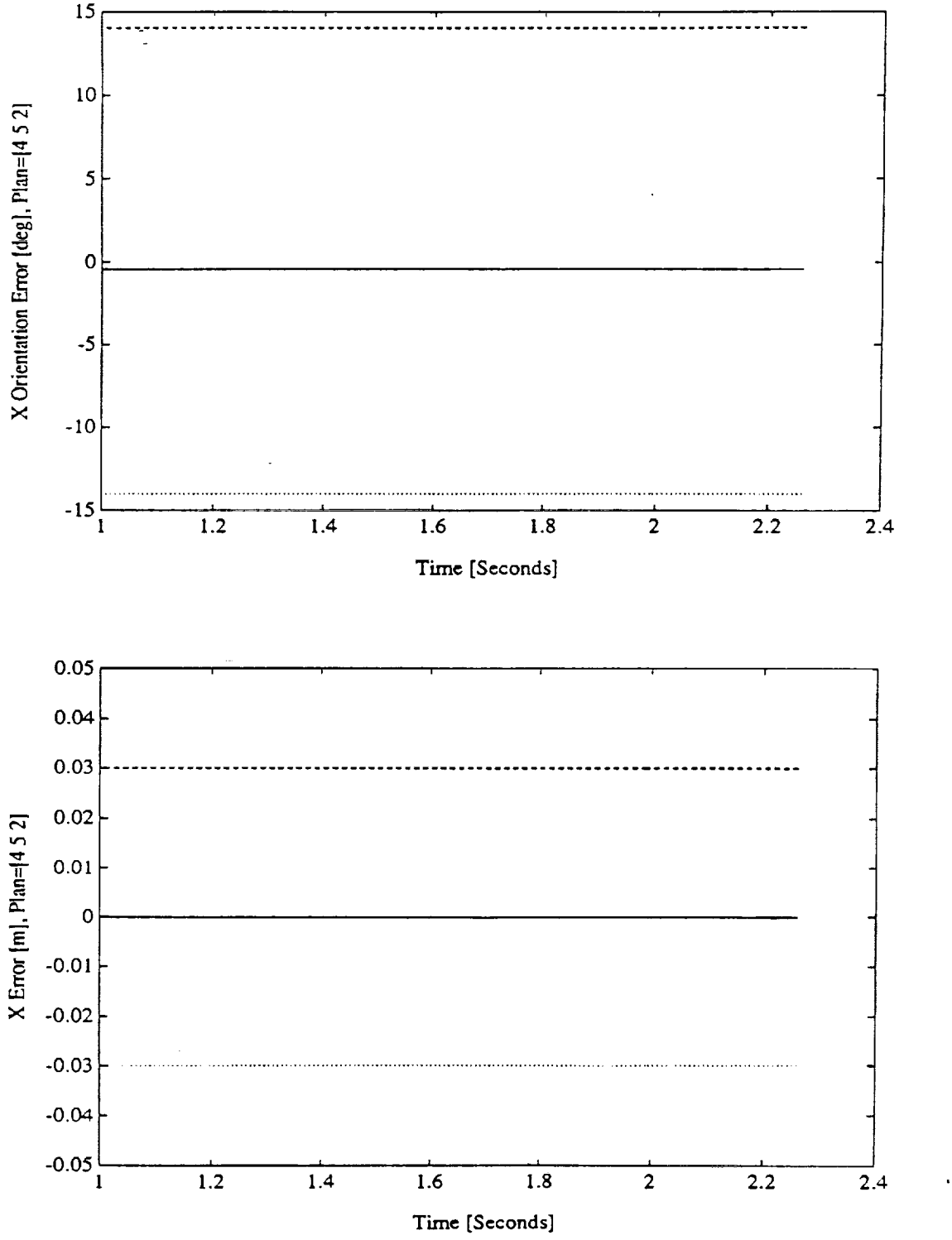


Figure 5.13: Error and Tolerance Bounds for the "X" Axis, Plan (4,5,2)

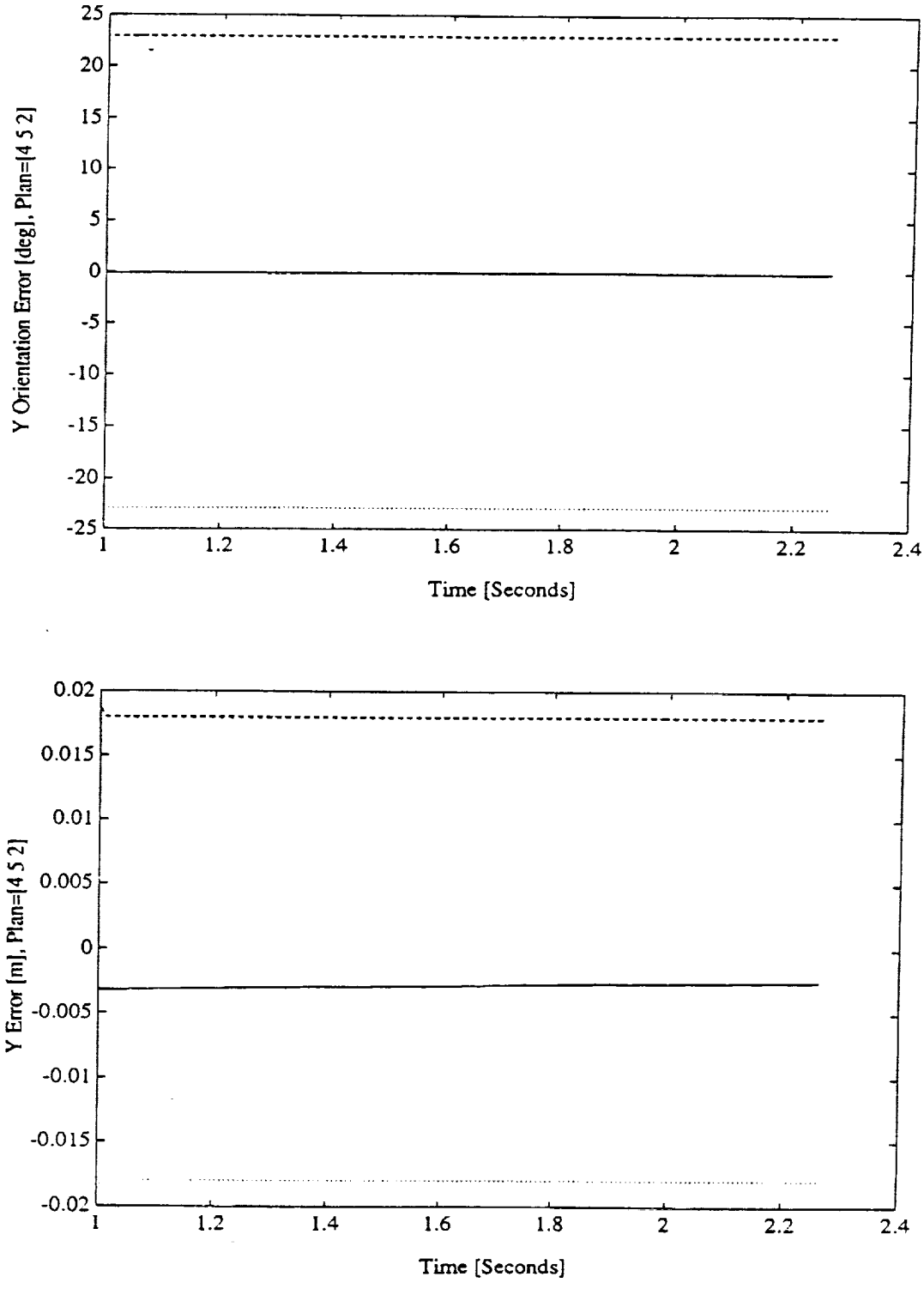


Figure 5.14: Error and Tolerance Bounds for the "Y" Axis, Plan (4,5,2)

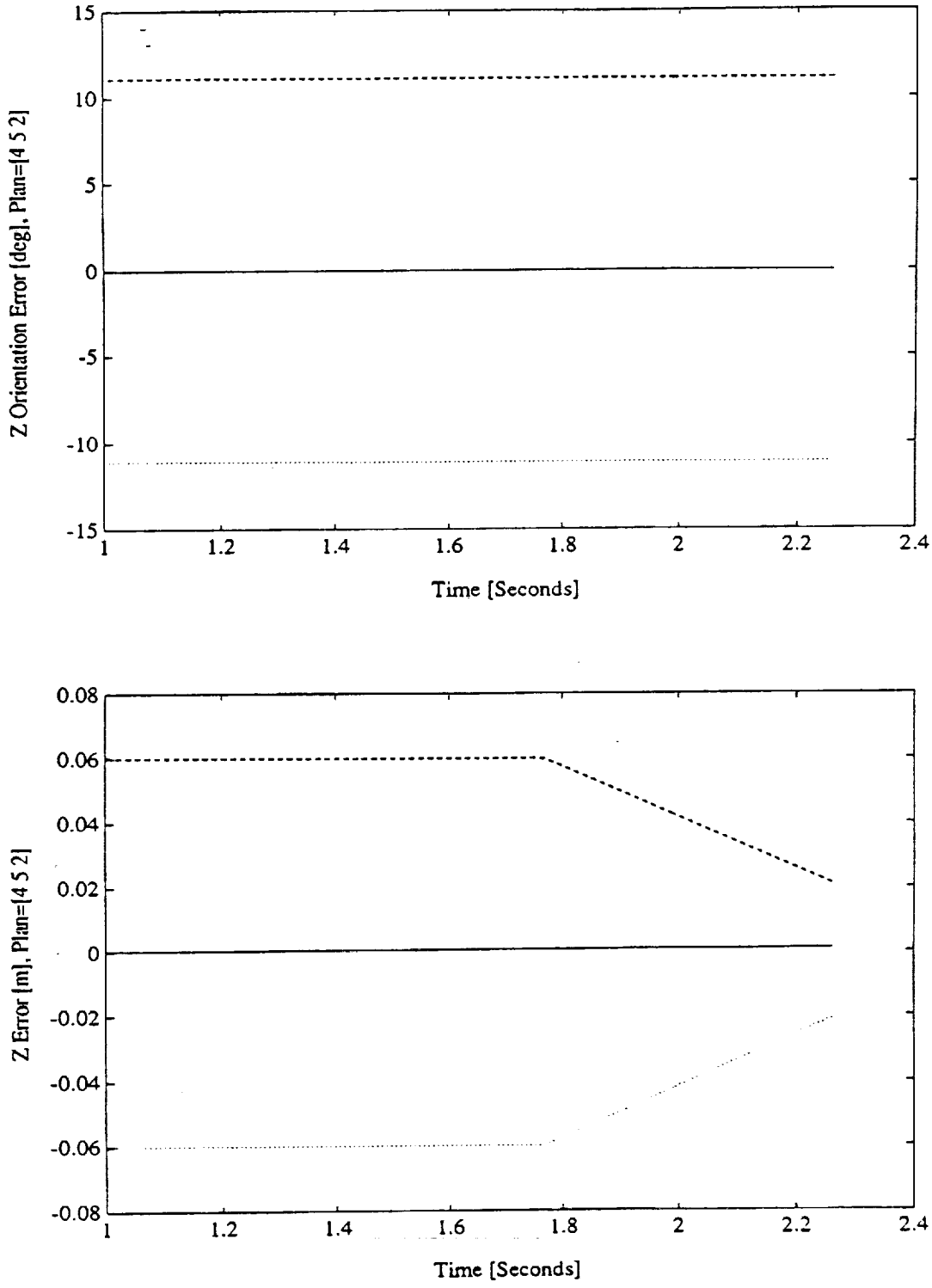


Figure 5.15: Error and Tolerance Bounds for the "Z" Axis, Plan (4,5,2)

6. CONCLUSIONS AND FUTURE WORK

By bringing together concepts from reliability and information theory, a new technique for systematically selecting reliable designs capable of executing a given task has been developed. The two stage procedure has several advantages. In particular, because the first stage proposes a coarse but computationally efficient entropy based elimination algorithm, a large number of unsuitable plans can quickly be excluded from further analysis. For scalar random variables subject to tolerance specifications, a lower bound on the reliability imposed by satisfaction of the entropy constraints is derived as long as the distribution of the random variable is either Gaussian, uniform, double exponential, Poisson, or Rayleigh. A lower bound reliability is also derived for correlated zero mean Gaussian vectors subject to quadratic specifications. The approach is especially adept at analyzing physically distributed systems due to the invariance of joint entropy to homogeneous coordinate frame transformations.

The second stage provides a practical framework for explicitly calculating the probability of success of a feasible plan. In this manner, plans of sufficient reliability can be found. Two methods are proposed for calculating the probability. Maximum likelihood estimation is suggested because it possesses the property of asymptotic normality. A large number of samples is required, of course, for asymptotic normality to be safely assumed. The first order, second moment methods from structural reliability offer a second alternative for calculating the probability of success. Lower bounds on reliability can be found even for nonlinear reliability performance functions. The reliabilities calculated by either technique can be combined to find the probability of meeting all of the specifications simultaneously as long as the specifications are independent. Plans of sufficient reliability may then be executed.

To allow analysis of performance over a time interval, a lower bound on the

reliability of a discrete time-invariant control in meeting a set of quadratic specifications is derived. This result can easily be applied to robotic systems after the system is linearized using nonlinear feedback. The quadratic specifications delineate the maximum acceptable value of a weighted square l_2 norm of the system's state error. The state error must be a zero mean Gaussian vector, so the measurement and process noise must be zero mean Gaussian variates. Although the analysis is conservative because it provides only a lower bound on the reliability, it does produce useful results in the case study.

The case study of visual positioning successfully applies these new techniques to a problem of practical significance. Because the errors in visual sensing, kinematic transformations, and control can all be accommodated in the analysis framework, an unprecedented level of statistical modeling for visual positioning has been achieved. Given realistic dimensions for a gripper and a gripping post, actual camera parameters, and the kinematic and dynamic models for the PUMA 560, the analysis deemed 3 out of 100 plans to be reliable. Each of the reliable plans did, in fact, execute reliably. Plans which were not deemed reliable also performed as expected, as they exhibited failures. This statistical model should prove very useful in high precision robotic tasks such as VLSI etching, medical manipulators, micro-robotics, etc.

6.1 Contributions

The contributions of this work are the following:

- A new method of representing specifications which delineates the maximum level of uncertainty allowed by the specification has been suggested.
- A new approach to reliability analysis based upon satisfaction of a set of entropy constraints has been developed.

- An original application of entropy invariance to homogeneous coordinate frame transformations is proposed for simplifying analysis of distributed systems.
- A systematic framework has been developed for calculating reliability of robotic systems.
- A lower bound on the reliability of a zero mean Gaussian vector meeting a quadratic specification is derived.
- The reliability of a control/sensing system over a time interval has been evaluated by deriving lower bounds on reliability for discrete time-invariant control systems.
- The first complete statistical analysis of visual positioning in the full six degree of freedom case has been developed. Errors arising from image noise, forward kinematic transformations, inverse kinematic transformations, and joint measurement noise are stochastically modeled.

6.2 Future Work

To extend the research presented in this thesis, the following issues may be addressed:

- The reliability analysis finds the probability of success in achieving a set of specifications, but does not assign a cost to alternative strategies. Decision making may be enhanced by adding cost criteria.
- Rather than visual positioning, the case study can be extended to visual servoing. This will desensitize the closed loop system to forward and inverse kinematic errors if a Cartesian control law is used.
- Methods for analyzing digital controls with assorted update times (i.e. a fast sampling time for closed loop control, but a slower sampling rate available

from a visual measurement system) need to be developed.

- Some applications may mandate a small number of samples when estimating statistics for individual algorithms. The small sample properties and the levels of confidence arising from the estimates can be investigated.

- An analysis of the reliability when discrete distributions are used may offer some advantages. First, since the actual distributions are discrete due to the presence of analog to digital converters, the modeling accuracy may increase. Second, many entropy properties have been developed in information theory for discrete distributions, so these properties may be employed.

- It might be possible to blend the results of the control analysis with Skelton's covariance control to yield a combined analysis/design technique.

- The reliability of other sensing algorithms can be analyzed and incorporated into the framework.

In conclusion, the framework for reliability analysis presented seems to be effective at selecting reliable plans for robotic systems. The case study example is a very high fidelity simulation of realistic multi-arm robotic systems. Moreover, the algorithms developed are fully capable of being implemented on standard robotic hardware.

LITERATURE CITED

[1] Ang, A. H. and W.H. Tang. *Probability Concepts in Engineering Planning and Design. Volume II–Decision, Risk and Reliability.* John Wiley & Sons, New York, 1984.

[2] Ang, A.H. and W.H. Tang. *Probability Concepts in Engineering Planning and Design Volume II–Decision, Risk, Reliability.* John Wiley and Sons, New York, 1984.

[3] Azadivar, Farhad. The Effect of Joint Position Errors of Industrial Robots on Their Performance in Manufacturing Operations. *IEEE Journal of Robotics and Automation*, Vol. RA–3, No. 2, pp 109–114, April 1987.

[4] Blostein, Steven D. and Thomas S. Huang. Error Analysis in Stereo Determination of 3–D Point Positions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI–9, No. 6, pp 752–765, November 1987.

[5] Brogan, William L. *Modern Control Theory.* Prentice–Hall, Englewood Cliffs, NJ, 1982.

[6] Brooks, R.A. Symbolic Error Analysis and Robot Planning. *The International Journal of Robotics Research*, Vol. 1, No. 4, pp 29–68, 1982.

[7] Conant, R.C. Laws of Information Which Govern Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC–6, No. 4, pp 240–255, April 1976.

[8] Durrant–Whyte, Hugh F. *Integration, Coordination and Control of Multi–Sensor Robot Systems.* Kluwer Academic Publishers, Boston, MA, 1988.

[9] Durrant-Whyte, Hugh F. Consistent Integration and Propagation of Disparate Sensor Observations. In *IEEE International Conference on Robotics and Automation*, pages 1464–1469, 1986.

[10] Durrant-Whyte, Hugh F. Uncertain Geometry in Robotics. *IEEE Journal of Robotics and Automation*, Vol. 4, No. 1, pp 23–31, February 1988.

[11] Fu, K.S., R.C. Gonzalez, and C.S.G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence.* McGraw–Hill, New York, New York, 1987.

[12] Grandjean, Pierrick and Arnaud Robert de Saint Vincent. 3–D Modeling of Indoor Scenes by Fusion of Noisy Range and Stereo Data. In *IEEE International Conference on Robotics and Automation*, pages 681–687, 1989.

[13] Han, Jia-Yuan. Fault-Tolerant Computing for Robot Kinematics Using Linear Arithmetic Codes. In *IEEE International Conference on Robotics and Automation*, volume I, pages 285–290, 1990.

[14] Harr, M.E. *Reliability-Based Design in Civil Engineering.* McGraw-Hill, New York, 1987.

[15] Havel, I.M. and I. Kramosil. A Stochastic Approach to Robot Plan Formation. *Kybernetika*, Vol. 14, No. 3, pp 143–173, 1978.

[16] Hayati, S., K. Tso, and G. Roston. Robot Geometry Calibration. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 947–951, 1988.

[17] Henderson, Thomas C., Eliot Weitz, Chuck Hansen, Rod Grupen, C.C. Ho, and Bir Bhanu. CAD-Based Robotics. In *IEEE International Conference on Robotics and Automation*, pages 631–635, 1987.

[18] Hsieh, C. and R.E. Skelton. All Covariance Controllers for Linear Discrete-Time Systems. *IEEE Transactions on Automatic Control*, Vol. 35, No. 8, pp 908–915, August 1990.

[19] Jaynes, Edwin T. Information Theory and Statistical Mechanics. *Physical Review*, Vol. 106, No. 4, pp 620–630, 1957.

[20] Jaynes, Edwin T. On the Rationale of Maximum-Entropy Methods. *Proceedings of the IEEE*, Vol. 70, No. 9, pp 939–952, September 1982.

[21] Kalata, Paul and Roland Premier. On System Identification With and Without Uncertainty. *Journal of Cybernetics*, Vol. 8, pp 31–50, 1978.

[22] Kalata, Paul and Roland Priemer. On Minimal Error Entropy Stochastic Approximation. *International Journal of Systems Science*, Vol. 5, No. 9, pp 895–906, 1974.

[23] Kalata, Paul and Roland Priemer. Linear Prediction, Filtering, and Smoothing: An Information-Theoretic Approach. *Information Sciences*, Vol. 17, pp 1–14, 1979.

[24] Kalata, Paul R. and Roland Priemer. When Should Smoothing Cease? *Proceedings of the IEEE*, pages 1289–1290, September 1974.

[25] Koomen, C.J. The Entropy of Design: A Study on the Meaning of Creativity. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 1, pp 16–30, January/February 1985.

[26] Koomen, C.J. The Entropy of Design: A Study on the Meaning of Creativity. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 1, pp 16–30, January/February 1985.

[27] Kuo, Benjamin C. *Digital Control Systems*. Holt, Rinehart, and Winston, New York, NY, 1980.

[28] Kyriakopoulos, Konstantinos. Private Communication, December 1988.

[29] Lee, Sukhan and Youngchul Kay. An Accurate Estimation of 3-D Position and Orientation of a Moving Object for Robot Stereo Vision: Kalman Filter Approach. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 414-419, Cincinnati, OH, May 1990.

[30] Lee, W.J. and T.C. Woo. Optimum Selection of Discrete Tolerances. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 111, pp 243-251, June 1989.

[31] Mazon, Isabelle and Rachid Alami. Representation and Propagation of Positioning Uncertainities through Manipulation Robot Programs. Integration into a Task-Level Programming System. In *IEEE International Conference on Robotics and Automation*, pages 646-652, 1989.

[32] McInroy, John E. and George N. Saridis. Reliable High Precision Positioning for Intelligent Machines. In 5^{th} *IEEE Symposium on Intelligent Control*, volume I, Philadelphia, PA, September 1990.

[33] McInroy, John E. and George N. Saridis. Reliability Based Control of Intelligent Machines. In Graham, James H., editor, *Safety, Reliability, and Human Factors in Robotic Systems*. Van Nostrand Reinhold, New York, 1991.

[34] McInroy, John E. and George N. Saridis. Reliable Control and Sensor Fusion. In *IEEE International Conference on Robotics and Automation*, volume I, pages 487-491, Sacramento, CA, April 1991.

[35] Papoulis, Athanasios. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, second edition, 1984.

[36] Parkinson, David B. Solution for Second Moment Reliability Index. *ASCE Journal of the Engineering Mechanics Division*, Vol. 104, pp 1267-1275, October 1978.

[37] Parkinson, David B. Computer Solution for the Reliability Index. *Engineering Structures*, Vol. 2, pp 57-62, January 1980.

[38] Parkinson, D.B. First-Order Reliability Analysis Employing Translation Systems. *Engineering Structures*, Vol. 1, pp 31-40, October 1978.

[39] Parkinson, D.B. The Application of Reliability Methods to Tolerancing. *ASME Journal of Mechanical Design*, Vol. 104, pp 612-618, July 1982.

[40] Ray, Laura Ryan and Robert F. Stengel. Computer-Aided-Analysis of Linear Control System Robustness. In *1990 IEEE Conference on Decision and Control*, Honolulu, HI, December 1990.

[41] Ray, Laura Ryan and Robert F. Stengel. Stochastic Performance Robustness of Aircraft Control Systems. In *1990 AIAA Guidance, Navigation, and Control Conference*, Portland, OR, 1990.

[42] Rodriguez, Jeffrey J. and J. K. Aggarwal. Stochastic Analysis of Stereo Quantization Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 5, pp 467–470, May 1990.

[43] Ryan, Laura E. and Robert F. Stengel. Stochastic Stability and Performance Robustness of Linear Multivariable Systems. In *1990 American Control Conference*, San Diego, CA, 1990.

[44] Sanderson, A.C. Parts Entropy Methods for Robotic Assembly System Design. In *IEEE International Conference on Robotics and Automation*, pages 600–608, Atlanta, GA, 1984.

[45] Saridis, George N. Entropy Formulation of Optimal and Adaptive Control. *IEEE Transactions on Automatic Control*, Vol. 33, No. 8, pp 713–721, August 1988.

[46] Saridis, George N. Analytic Formulation of the Principle of Increasing Precision with Decreasing Intelligence for Intelligent Machines. *Automatica*, Vol. 25, No. 3, pp 461–467, 1989.

[47] Saridis, G.N., editor. *Advances in Automation and Robotics: Volume 1.* JAI Press, Greenwich, Connecticut, 1985.

[48] Saridis, G.N. and K.P. Valavanis. Analytical Design of Intelligent Machines. *Automatica*, Vol. 24, No. 2, pp 123–133, 1988.

[49] Shinozuka, M. Basic Analysis of Structural Safety. *ASCE Journal of Structural Division*, Vol. 3, No. 109,, March 1983.

[50] Skelton, R.E. and Masao Ikeda. Covariance Controllers for Linear Continuous-Time Systems. *International Journal of Control*, Vol. 49, No. 5, pp 1773–1785, 1989.

[51] Skelton, Robert. *Dynamic Systems Control.* John Wiley and Sons, New York, New York, 1988.

[52] Smith, Randall C. and Peter Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, Vol. 5, No. 4, pp 56–68, 1986.

[53] Smith, R.E. and M. Gini. Reliable Real-Time Robot Operation Employing Intelligent Forward Recovery. *Journal of Robotic Systems*, Vol. 3, No. 3, pp 281–300, 1986.

[54] Stengel, R.F. and L.R. Ray. Stochastic Robustness of Linear Time-Invariant Control Systems. *IEEE Transactions on Automatic Control*, Vol. 36, No. 1, pp 82–86, January 1991.

[55] Stengel, Robert F. and Laura E. Ryan. Application of Stochastic Robustness to Aircraft Control Systems. In *AIAA Guidance and Control Conference*, August 1989.

[56] Stone, Henry Wallentin. *Kinematic Modeling, Identification, and Control of Robotic Manipulators*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA 15213, December 1986.

[57] Tarn, T.J., A.K. Bejczy, A. Isidori, and Y. Chen. Nonlinear Feedback in Robot Arm Control. In *Proceedings of the 23rd Conference on Decision and Control*, pages 736–751, December 1984.

[58] Taylor, G.E. and P.M. Taylor. Dynamic Error Probability Vectors: A Framework for Sensory Decision Making. In *IEEE International Conference on Robotics and Automation*, volume II, pages 1096–1100, 1988.

[59] Taylor, P.M., I. Halleron, and X.K. Song. The Application of a Dynamic Error Framework to Robotic Assembly. In *IEEE International Conference on Robotics and Automation*, volume I, pages 170–175, 1990.

[60] Valavanis, K.P. and G.N. Saridis. Information-Theoretic Modeling of Intelligent Robotic Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 6, pp 852–872, November 1988.

[61] Viswanadham, N. and T.L. Johnson. Fault Detection and Diagnosis of Automated Manufacturing Systems. In *Proceedings of the 27th Conference on Decision and Control*, pages 2301–2306, Austin, TX, December 1988.

[62] Viswanadham, N., V.V.S. Sarma, and M.G. Singh. *Reliability of Computer and Control Systems*. North-Holland, 1987. Systems and Control Series, 8.

[63] Wang, Fei-Yue and George N. Saridis. The Coordination of Intelligent Robots: A Case Study. In *IEEE International Symposium on Intelligent Control*, pages 506–511, Albany, NY, 1989.

[64] Wilks, Samuel S. *Mathematical Statistics*. John Wiley and Sons, Inc., New York, 1962.

[65] Zacks, Shelemyahu. *The Theory of Statistical Inference*. John Wiley and Sons, Inc., New York, 1971.

APPENDIX A
IMPLICIT RELIABILITY FOR TOLERANCE SPECIFICATIONS

For a tolerance range represented as $\mu_x \pm d$,

$$R = P\{|x - \mu_x| < d\} \tag{A.1}$$

The entropy constraint requires that

$$H(x) \leq \ln 2d \tag{A.2}$$

For a class of distributions including the Gaussian, uniform, double exponential, exponential, Poisson, and Rayleigh

$$H(x) = \ln \sigma_x + \ln B \tag{A.3}$$

where B is a constant. This implies that

$$\ln \sigma_x + \ln B \leq \ln 2d \tag{A.4}$$

$$\frac{\sigma_x}{d} \leq \frac{2}{B} \tag{A.5}$$

From the Chebyshev inequality,

$$R = P\{|x - \mu_x| < d\} \geq 1 - \sigma_x^2/d^2 \tag{A.6}$$

or

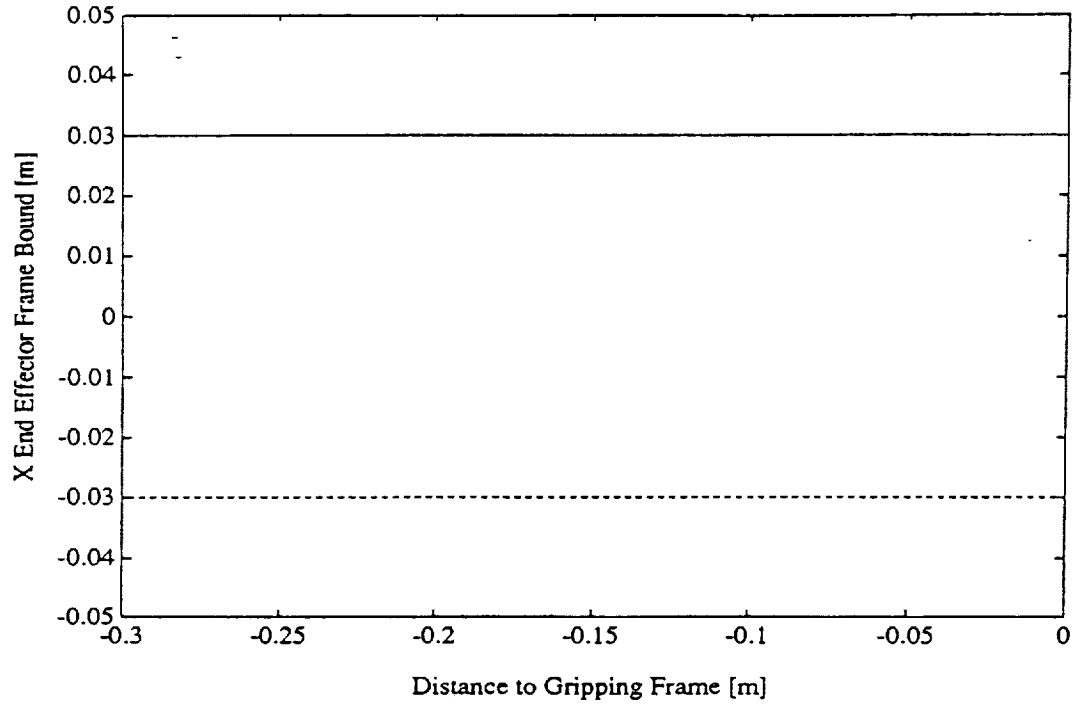
$$R \geq 1 - 4/B^2 \tag{A.7}$$

Thus a conservative lower bound on reliability implicitly guaranteed by the entropy constraints (for tolerance specifications) can be found for a large class of distributions by determining the constant, B, for the particular distribution.

APPENDIX B
VISION SYSTEM SIMULATION RESULTS

Simulation results for the Marshall Space Flight Center vision system are presented in this appendix. Figures B.1–B.3 display the tolerance bounds of the gripping interface as the distance between the two frames is varied. Figures B.4–B.6 show the actual measurement errors of the vision system as well as the 3σ bound predicted by the stochastic model while the pose of the target is kept at a typical value. Figure B.7 plots the normed pose error as it is weighted by the predicted inverse covariance. If the distribution is zero mean Gaussian as assumed and the predicted covariance is correct, then the plot should have a chi-square distribution with 6 degrees of freedom.

To test the robustness of the stochastic model for the vision system, Figures B.8–B.13 plot the measurement errors and 3σ bound for two poses which violate the assumptions of the statistical model. The first test places the pose of the target at an extreme angle so that the cross sectional area of the retroreflector in the image plane is very small. The second test purposely moves into positions which are not likely in a practical environment, but which induce a large bias due to the centroid not corresponding to the center of the retroreflector. Finally, Figures B.14 and B.15 provide the chi-square goodness of fit probability as the target parameters d_1 and d_2 as well as the focal length are varied.



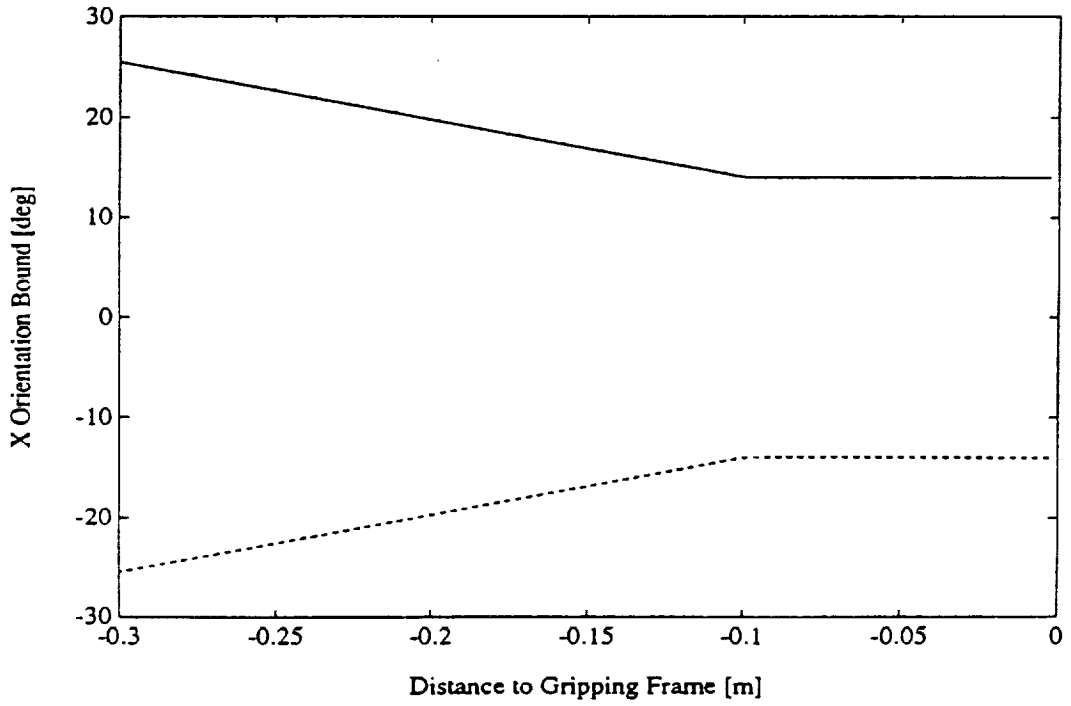
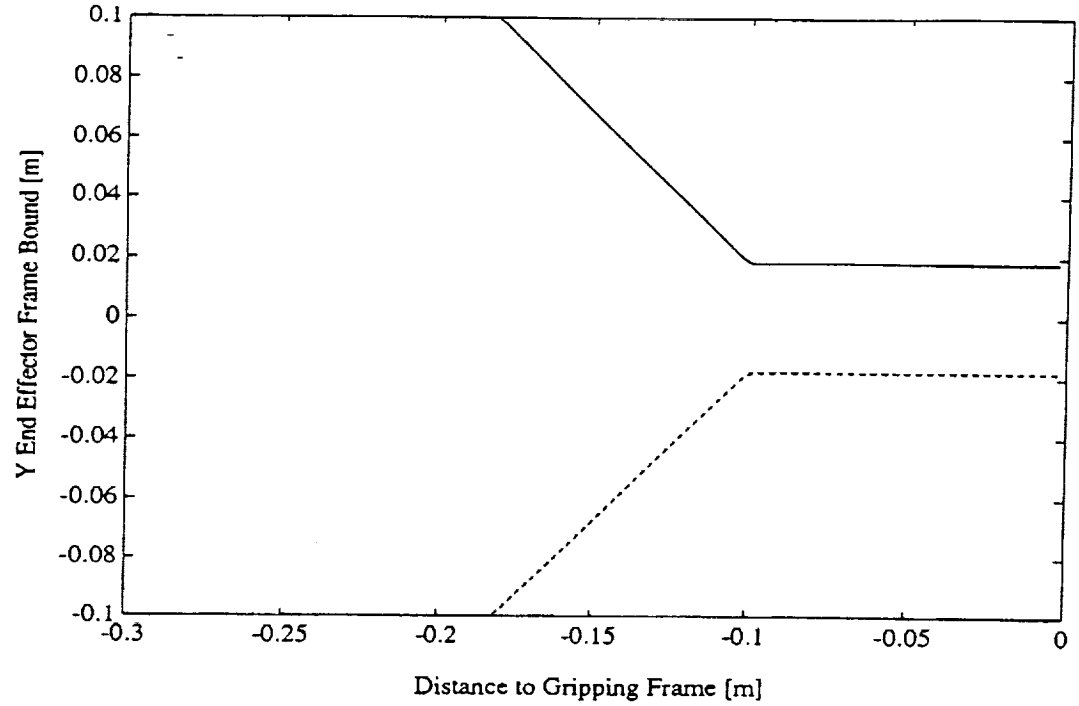


Figure B.1: Tolerance Bounds for the "X" Axis



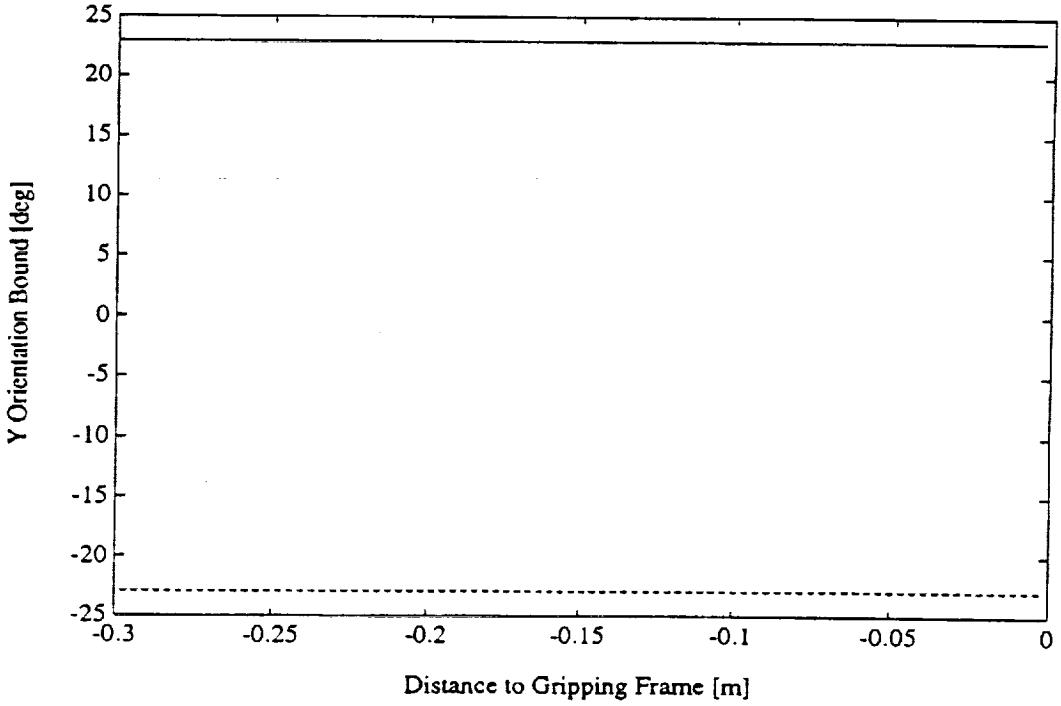
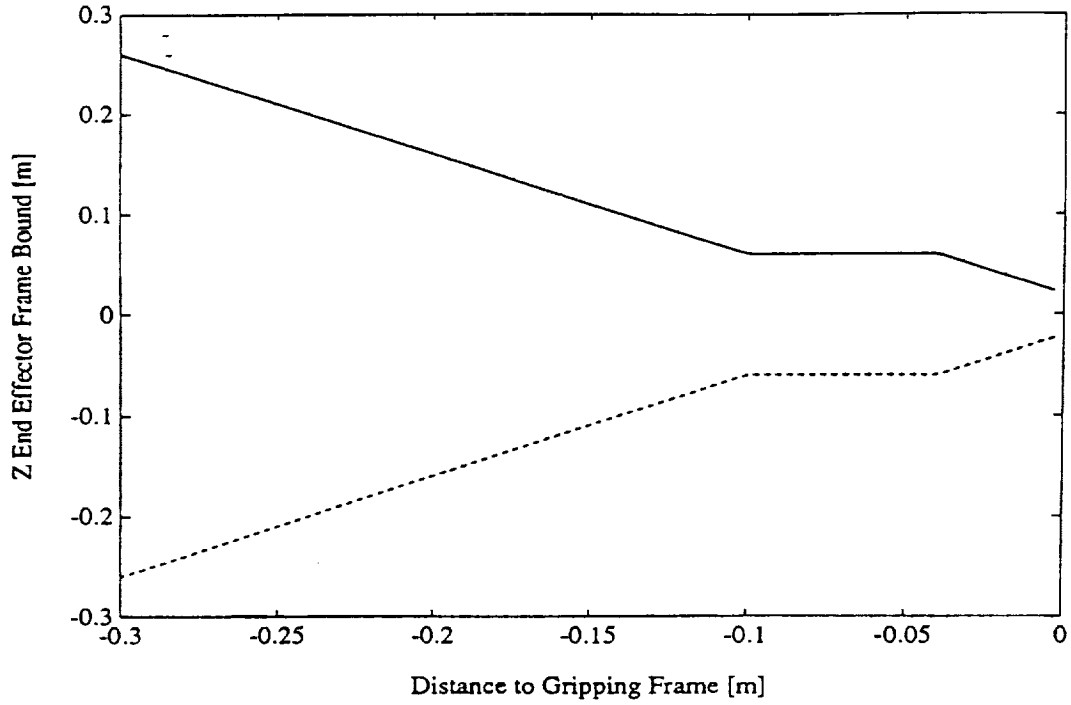


Figure B.2: Tolerance Bounds for the "Y" Axis



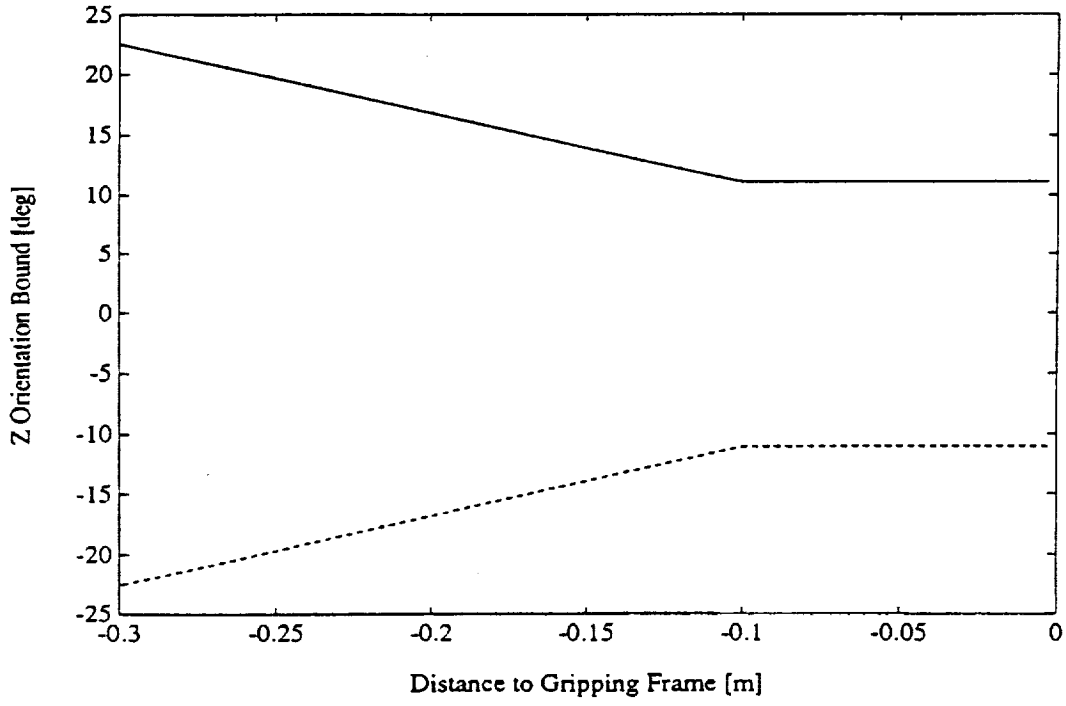


Figure B.3: Tolerance Bounds for the "Z" Axis

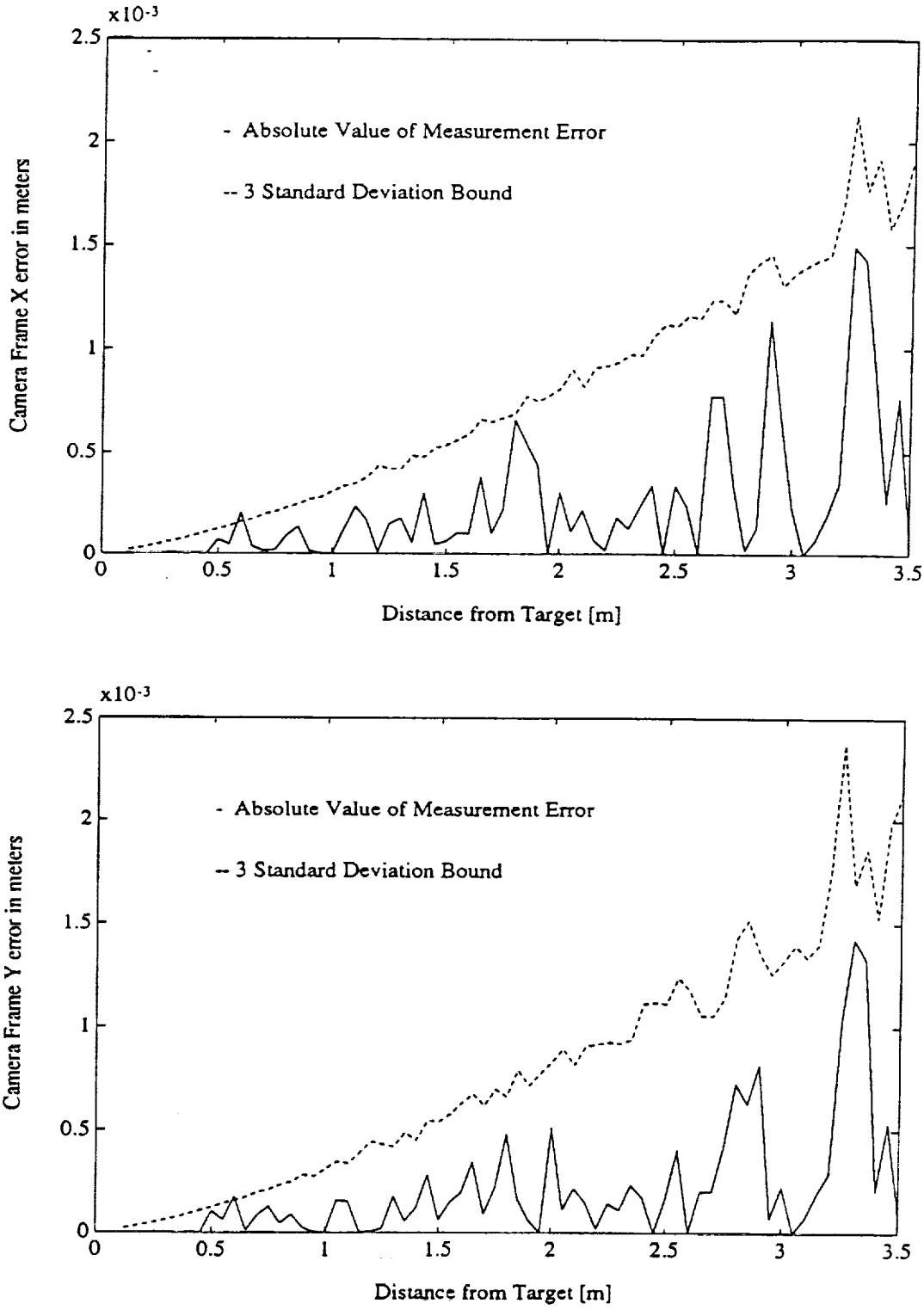


Figure B.4: Nominal Case–"X" and "Y" Errors

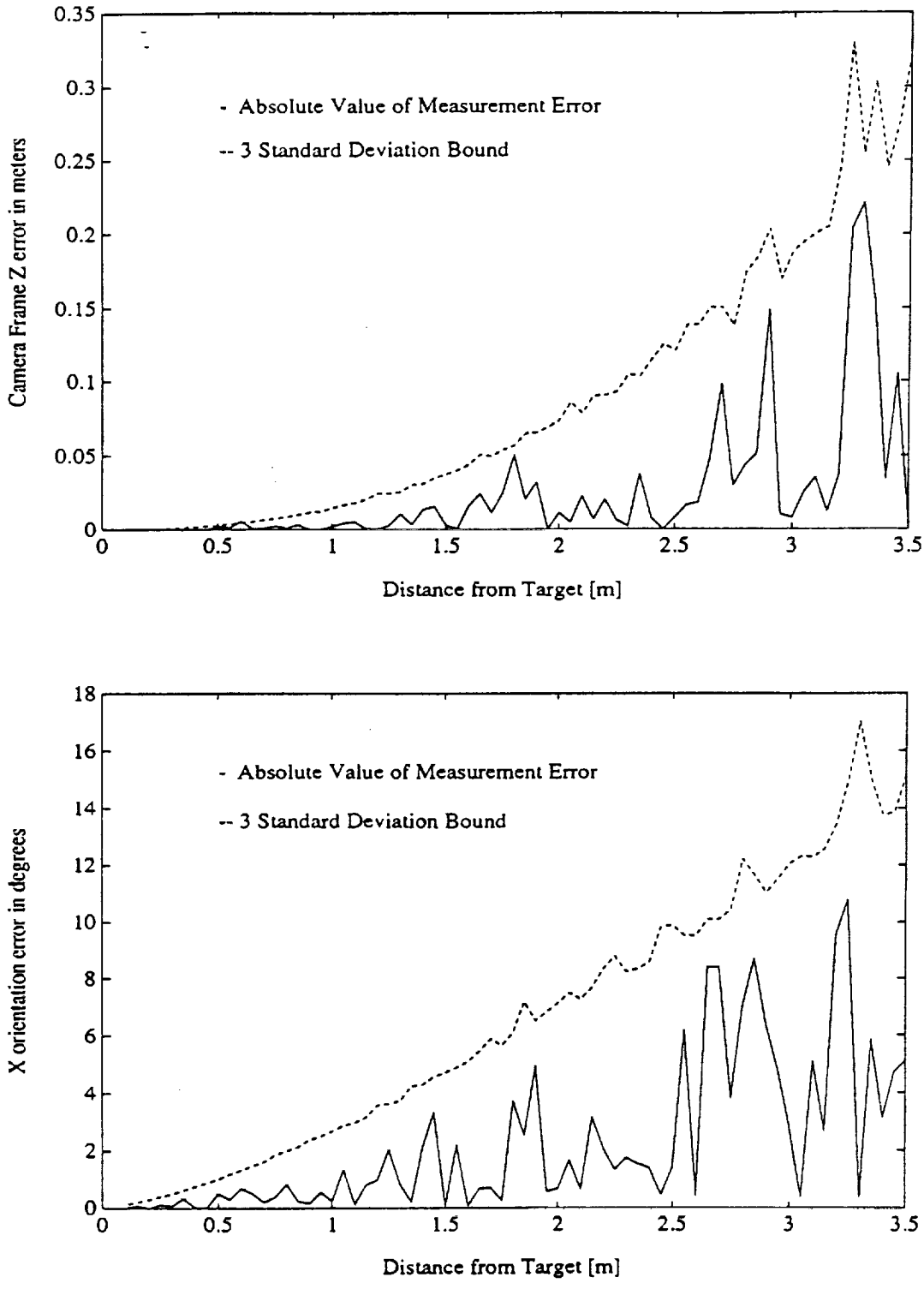
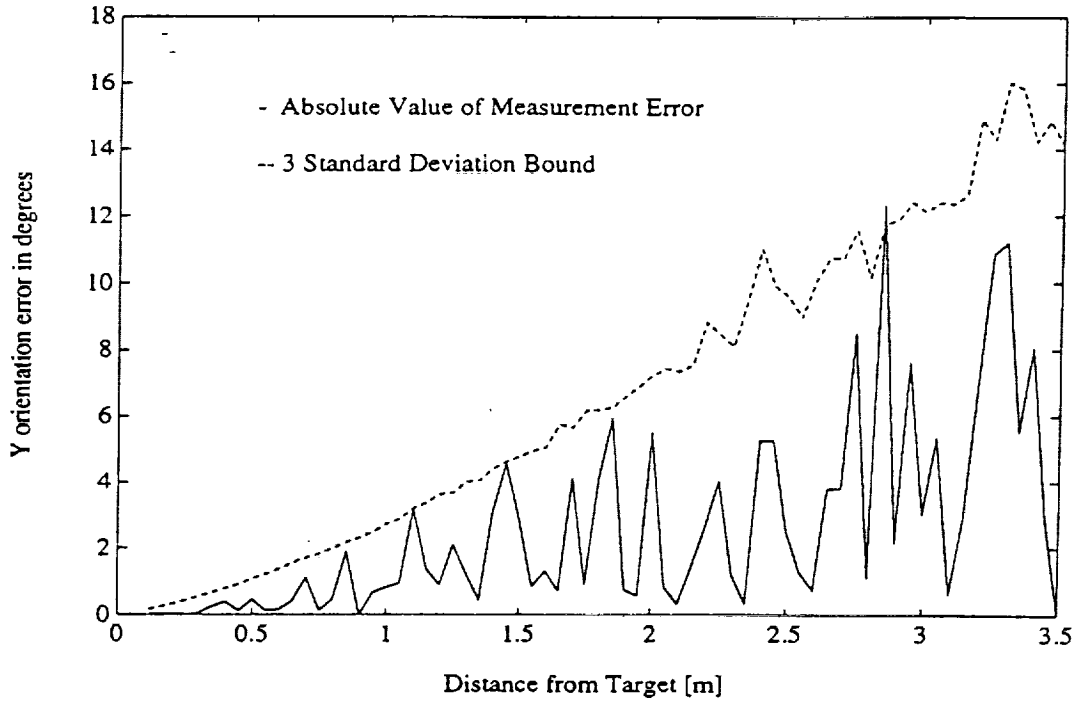


Figure B.5: Nominal Case–"Z" Position and "X" Orientation Errors



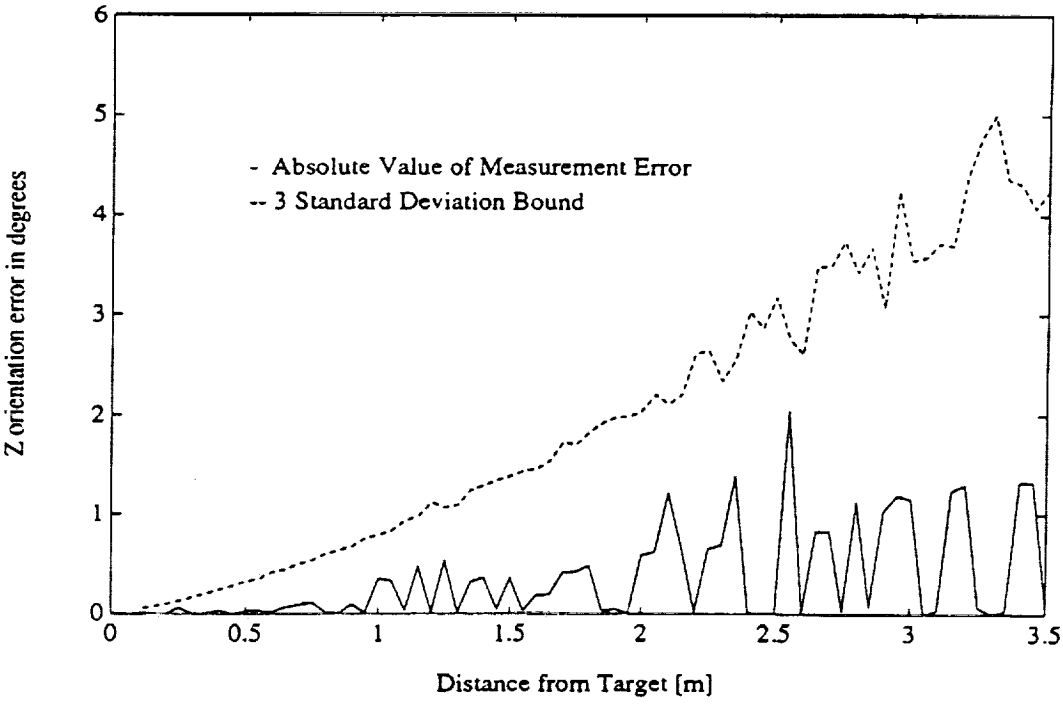


Figure B.6: Nominal Case–"Y" and "Z" Orientation Errors

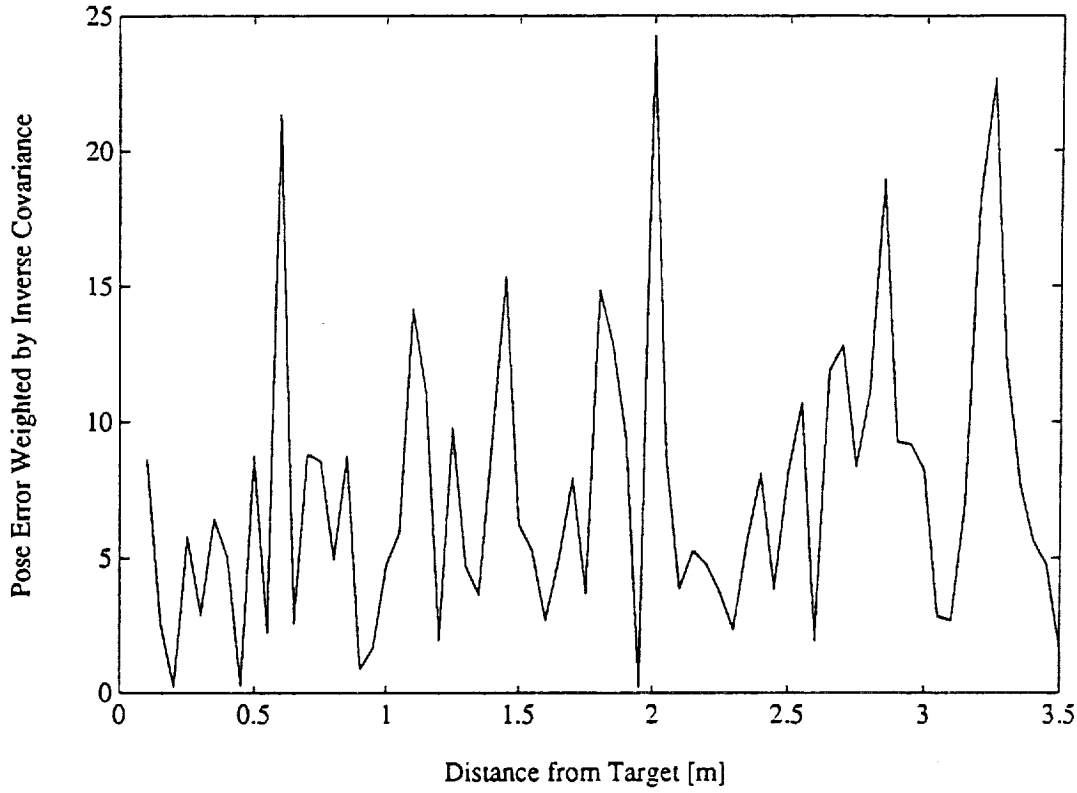
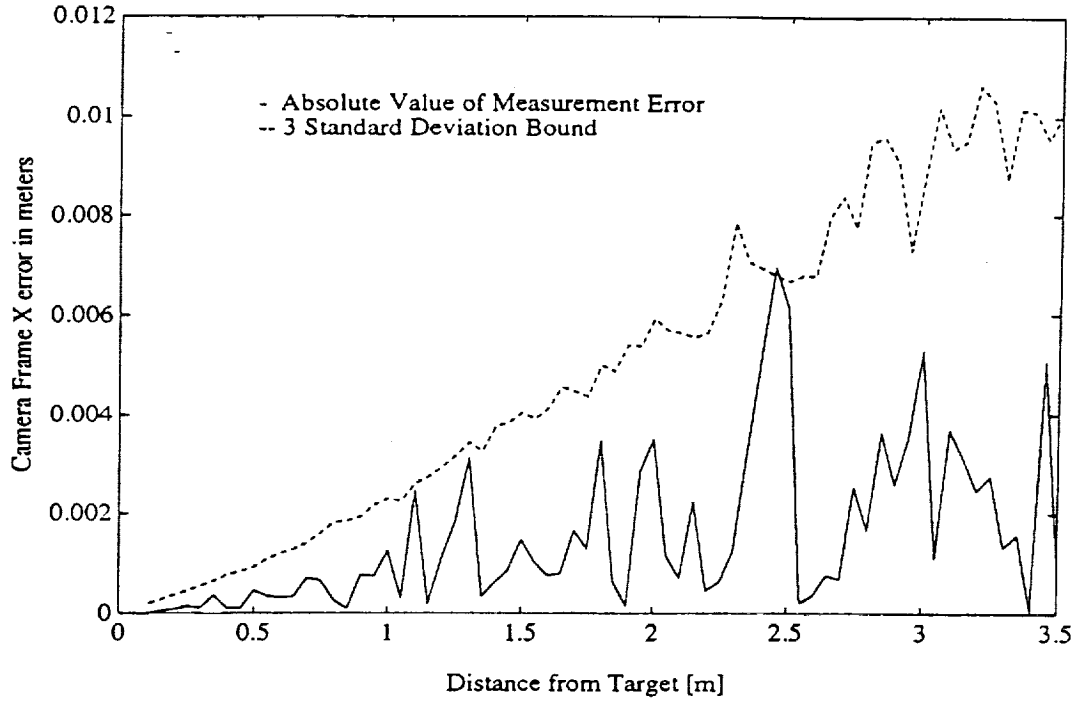


Figure B.7: Nominal Case–Weighted Pose Error



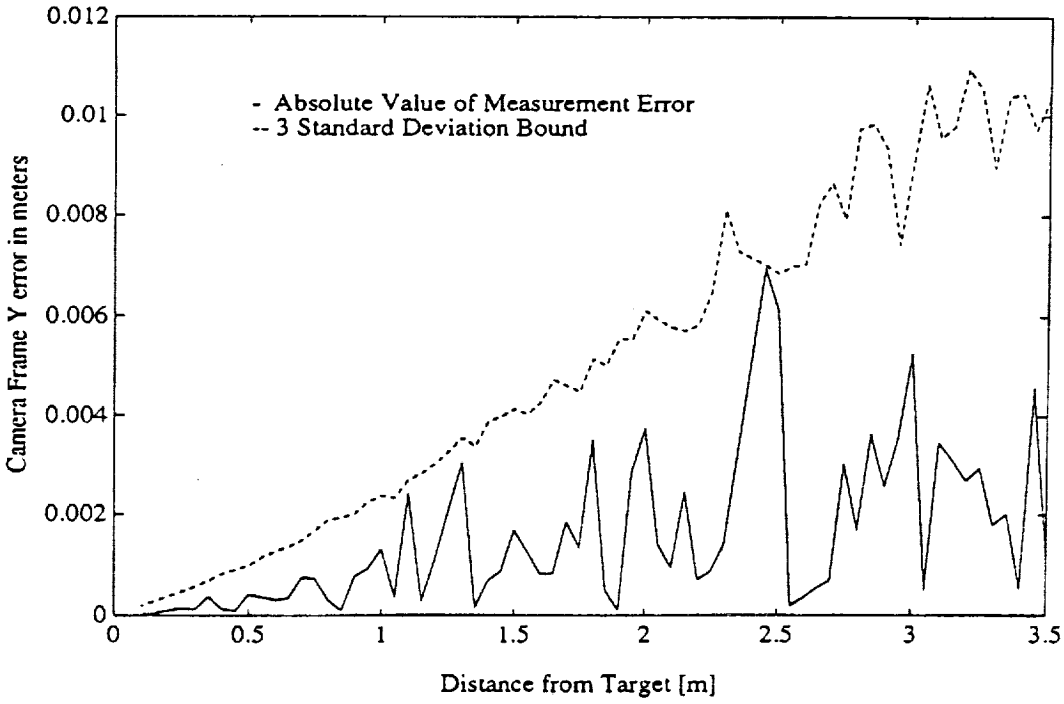


Figure B.8: Offset Case–"X" and "Y" Errors

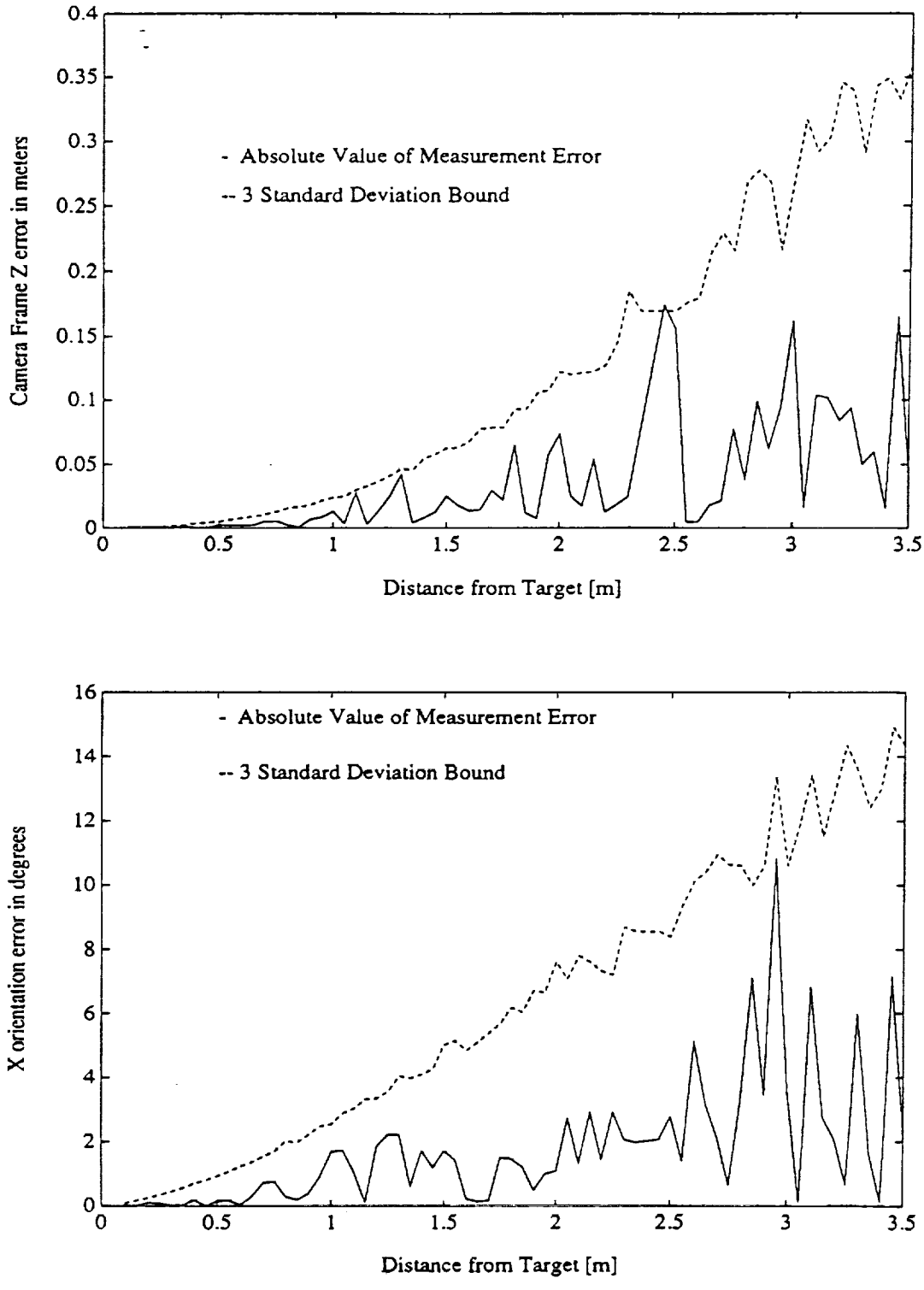


Figure B.9: Offset Case–"Z" Position and "X" Orientation Errors

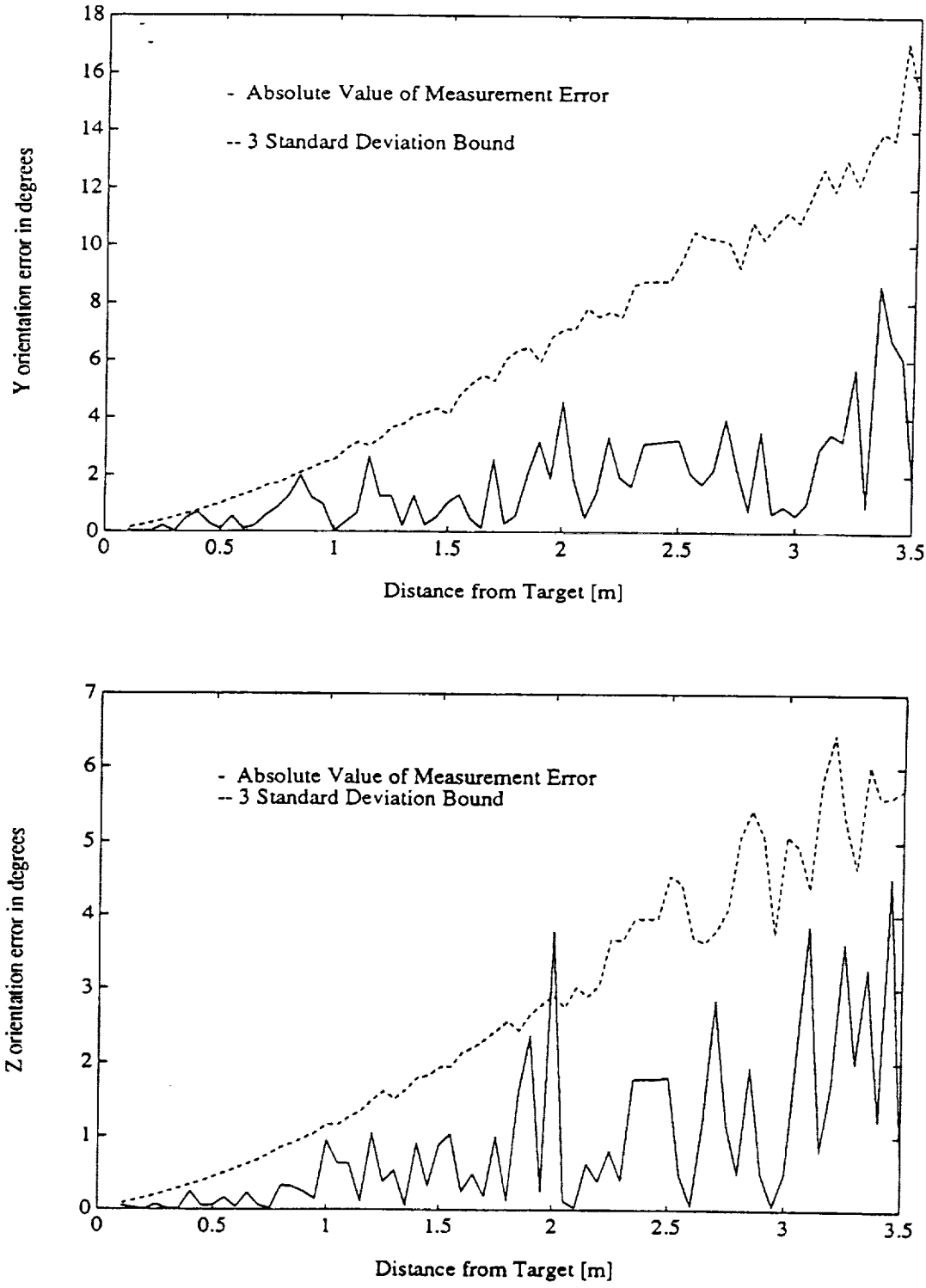
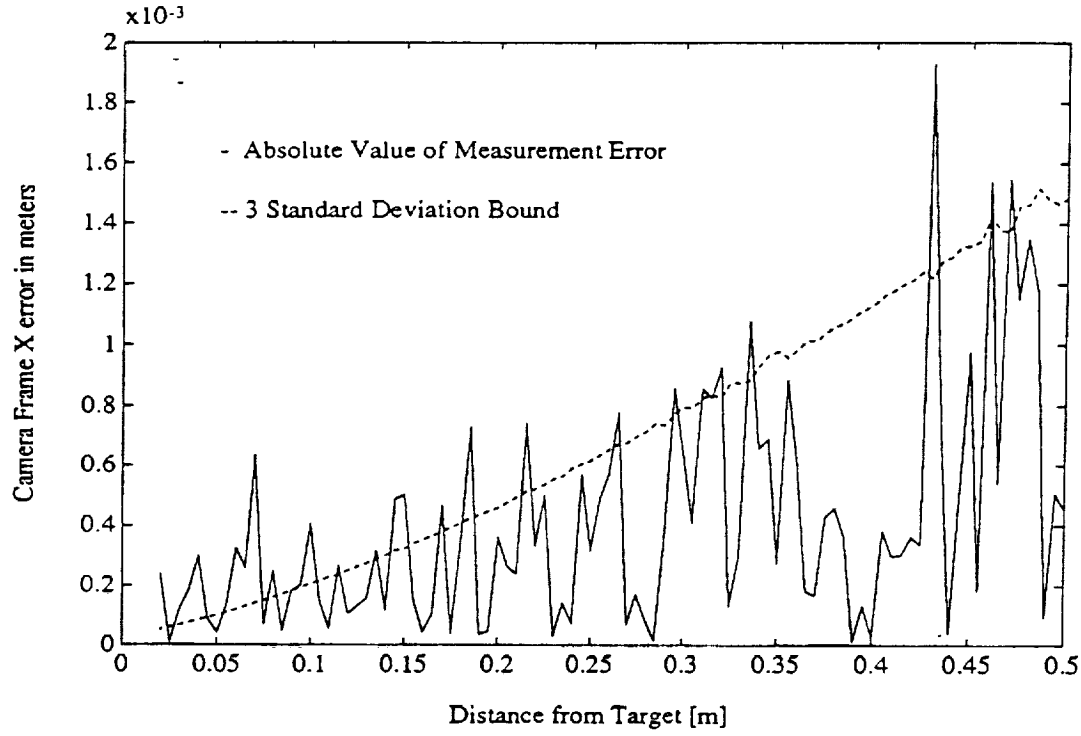


Figure B.10: Offset Case–"Y" and "Z" Orientation Errors



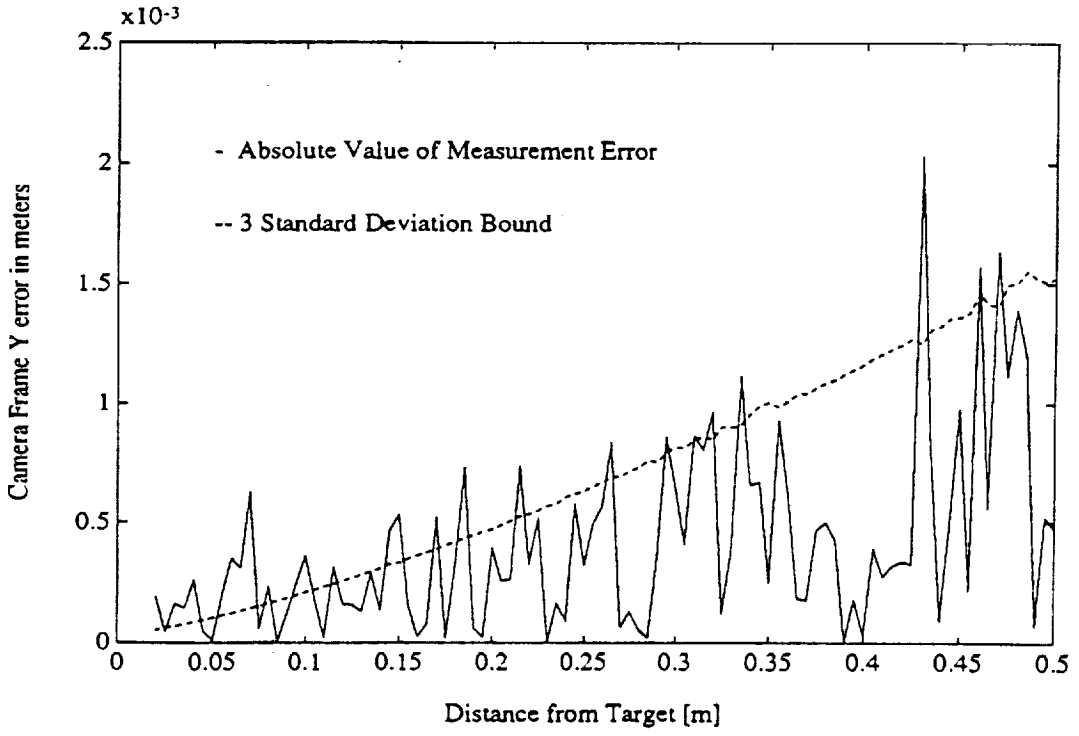
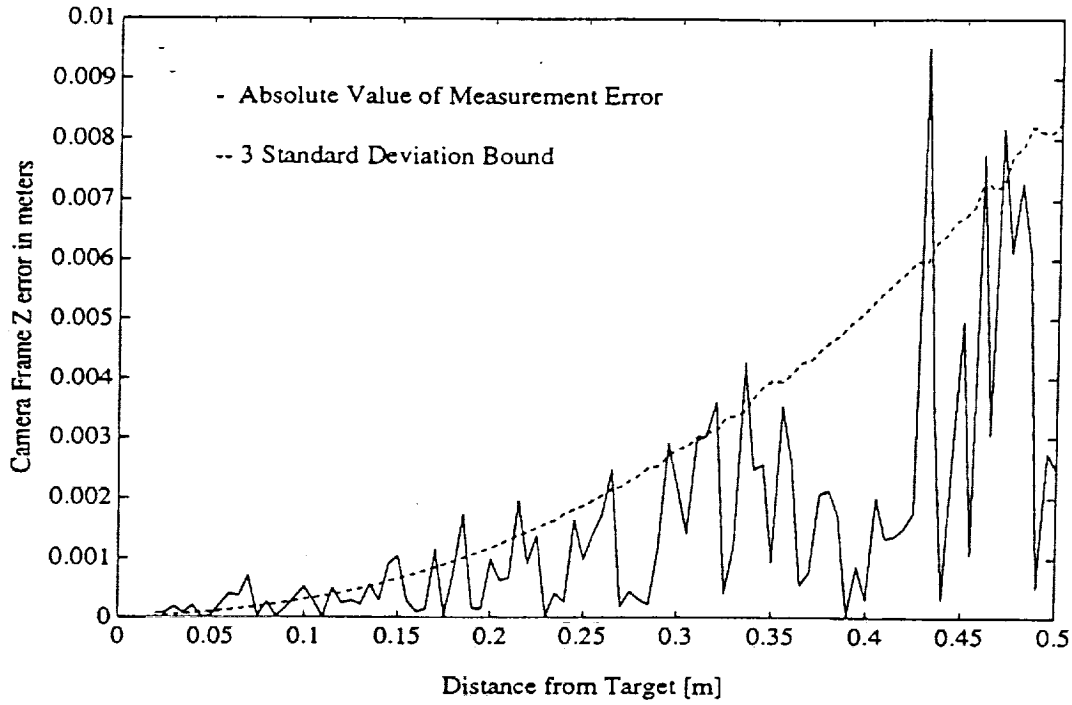


Figure B.11: Centroidal Bias Case–"X" and "Y" Errors



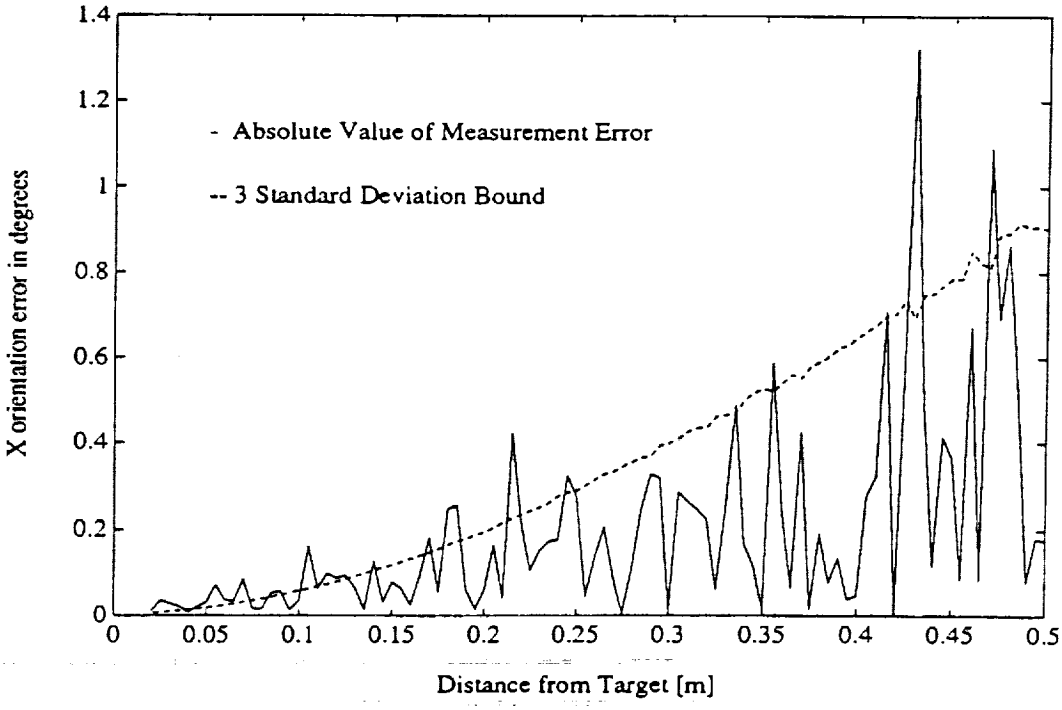


Figure B.12: Centroidal Bias Case–"Z" Position and "X" Orientation Errors

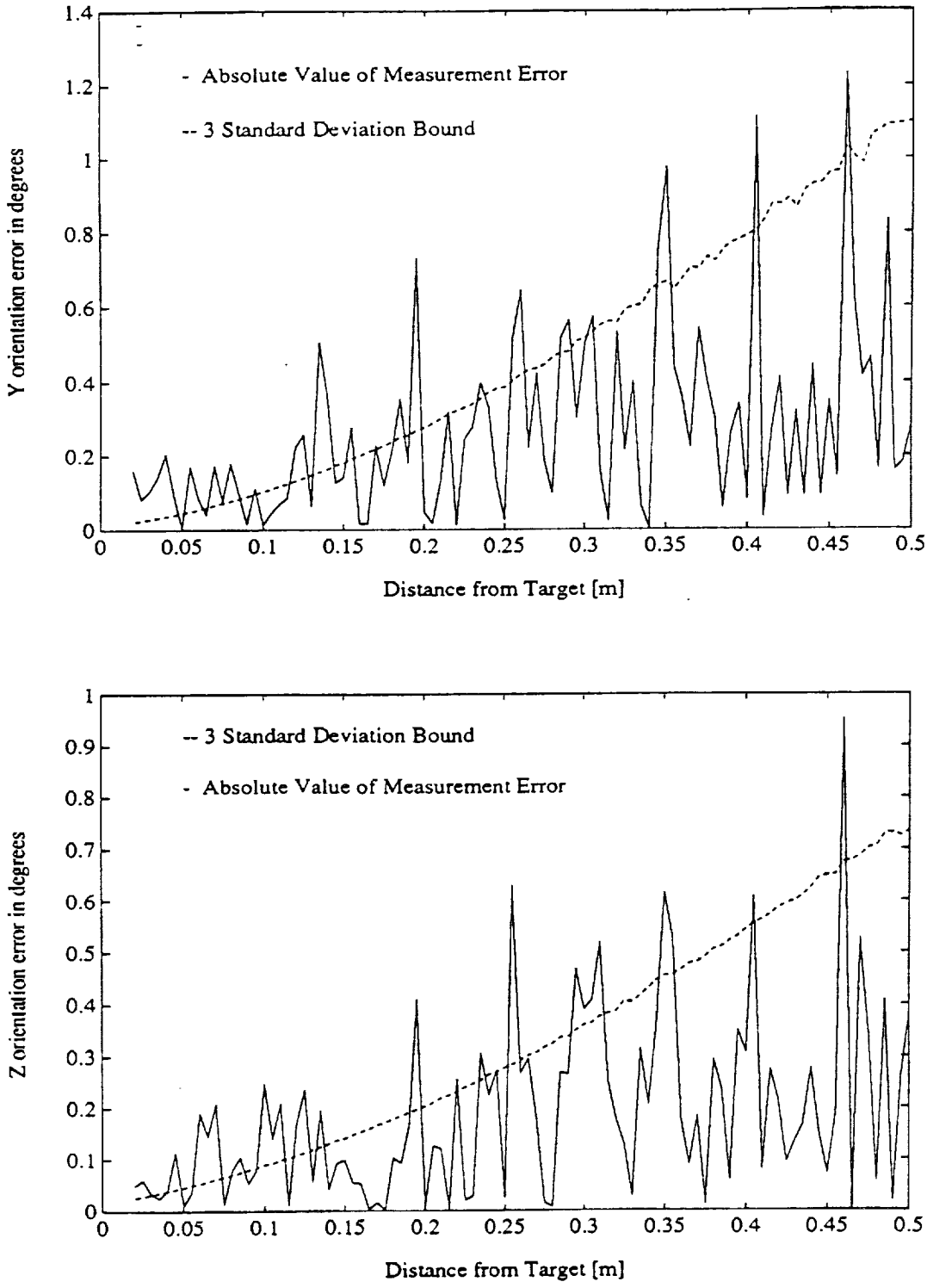
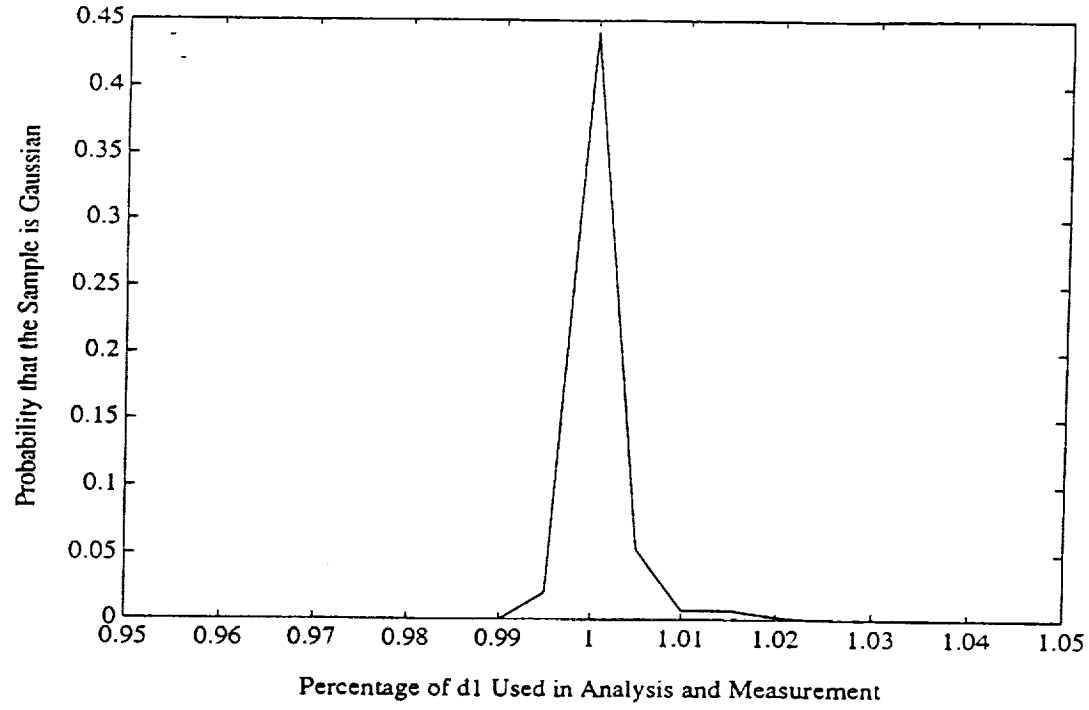


Figure B.13: Centroidal Bias Case–"Y" and "Z" Orientation Errors



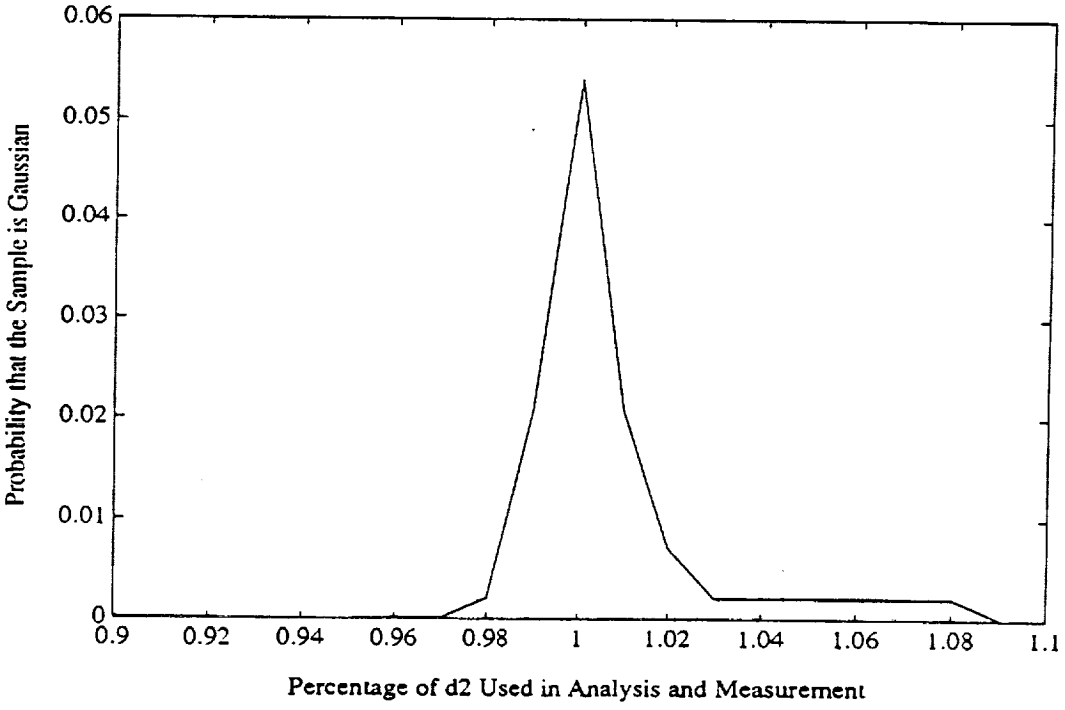


Figure B.14: Chi-Square Probabilities for d1 and d2

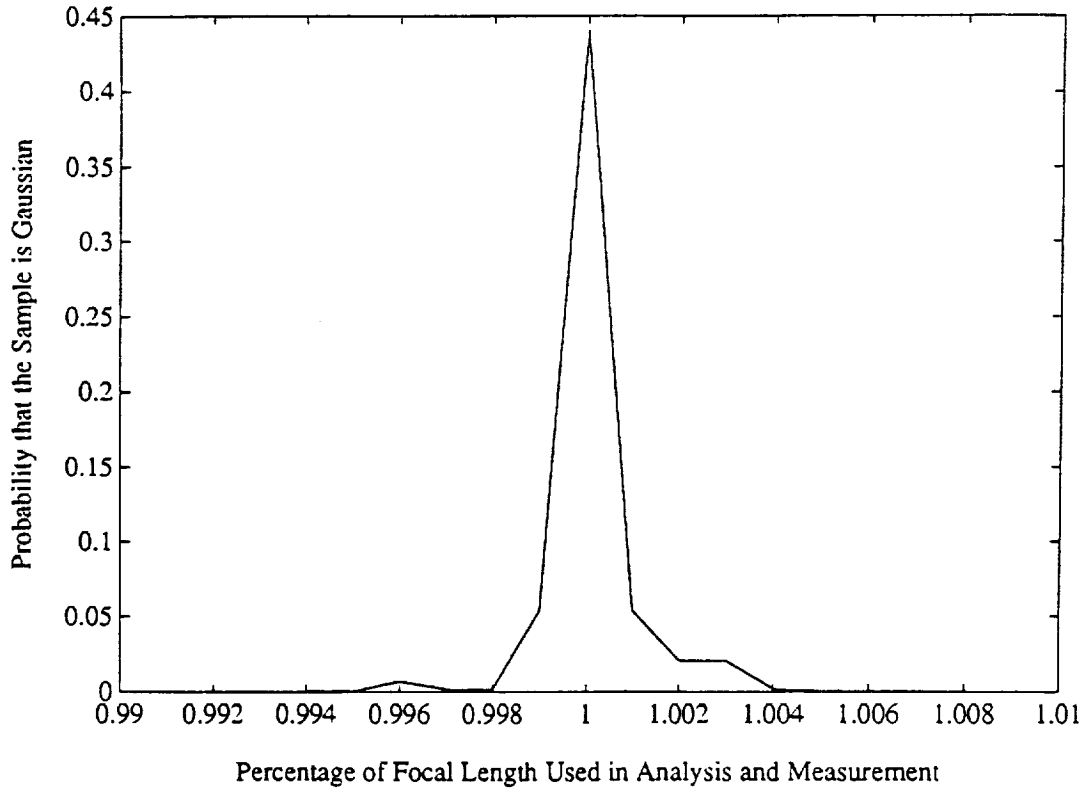


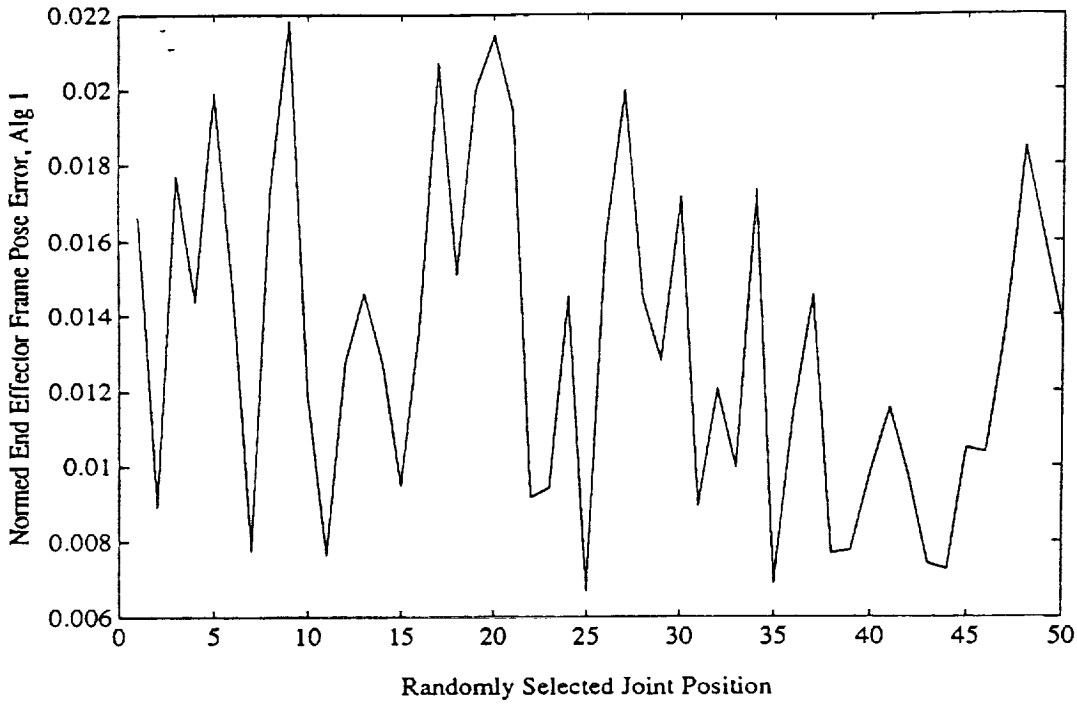
Figure B.15: Chi-Square Probabilities for Focal Length

APPENDIX C
KINEMATIC SIMULATION RESULTS

Appendix C provides the kinematic simulation results. Figures C.1–C.5 plot the l_2 norm of the end effector frame pose error for each of the five inverse kinematic routines. The errors are found at a number of randomly selected joint positions, with the same set of joint positions used for each inverse kinematic algorithm. In addition, the floating point operations required to execute each of the algorithms is also displayed, along with the sample mean and 3σ bound.

Figures C.6–C.8 plot the norm squared end effector pose error as it is weighted by the predicted inverse covariance matrix. The joint noise and backlash are kept at values normally expected for the PUMA 560. The resulting distribution should be $\chi^2_6(\cdot)$, so the median of the chi–square distribution is also plotted (it is 5.6). Figures C.9–C.11 are very similar to the previous plots, but in this case a large backlash deadband is added. The statistical properties do not greatly change.

Figures C.12–C.17 plot the joint entropy of the end effector frame pose error for each kinematic algorithm as two statistical parameters are varied. First, the standard deviation of the joint noise is increased from 0 to 0.35 degrees. As expected, the entropy also increases. Finally, the width of the backlash deadband is varied from 0 to 2 degrees as the entropy is plotted.



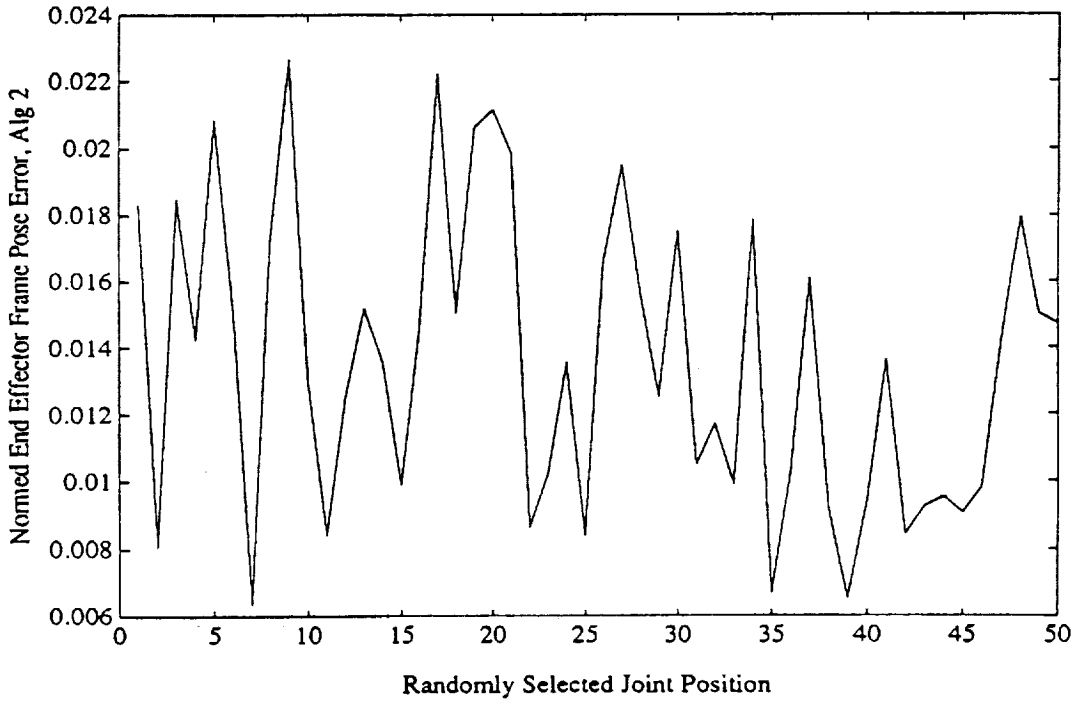
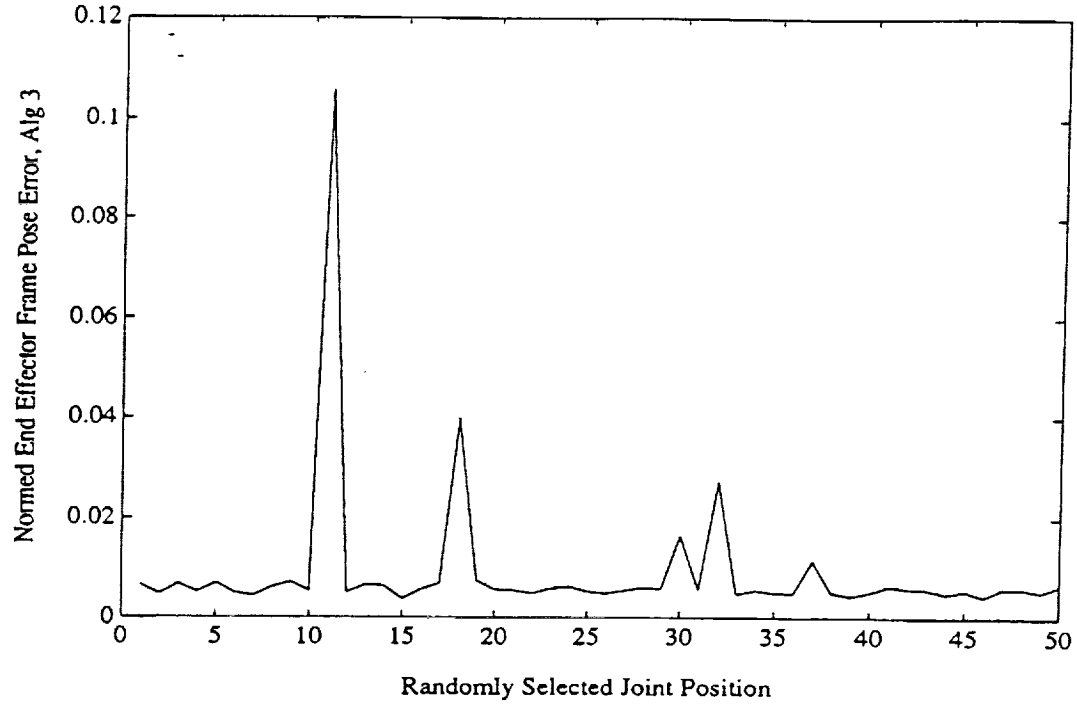


Figure C.1: Normed Error for the Nominal Inverse Kinematics Routines



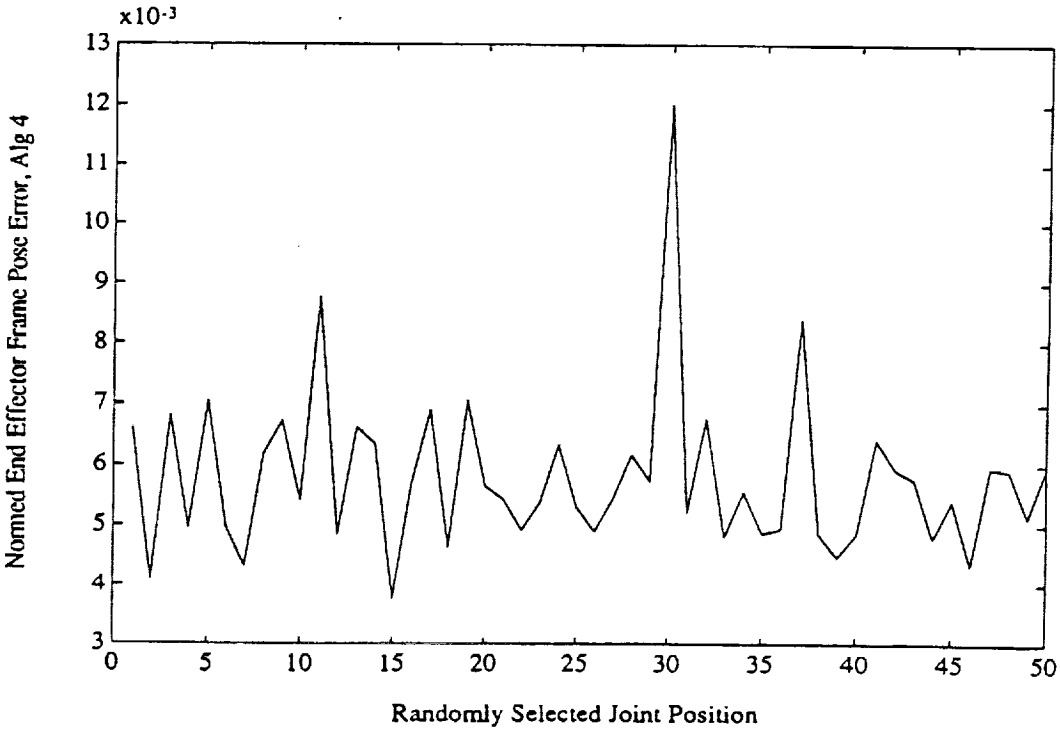


Figure C.2: Normed Error for the Jacobian Inverse Kinematics Routines

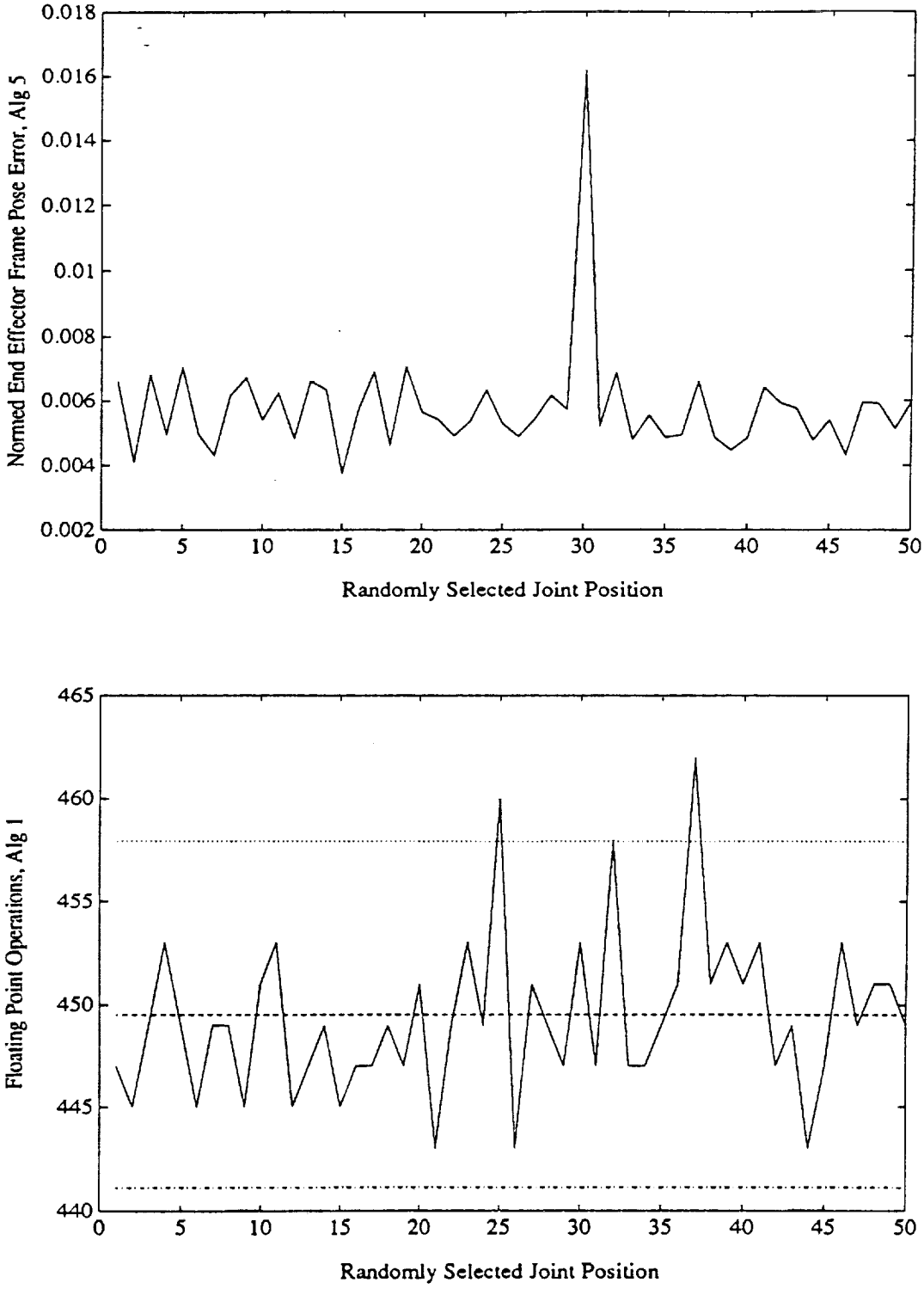


Figure C.3: Jacobi Normed Error and Nominal FLOPS

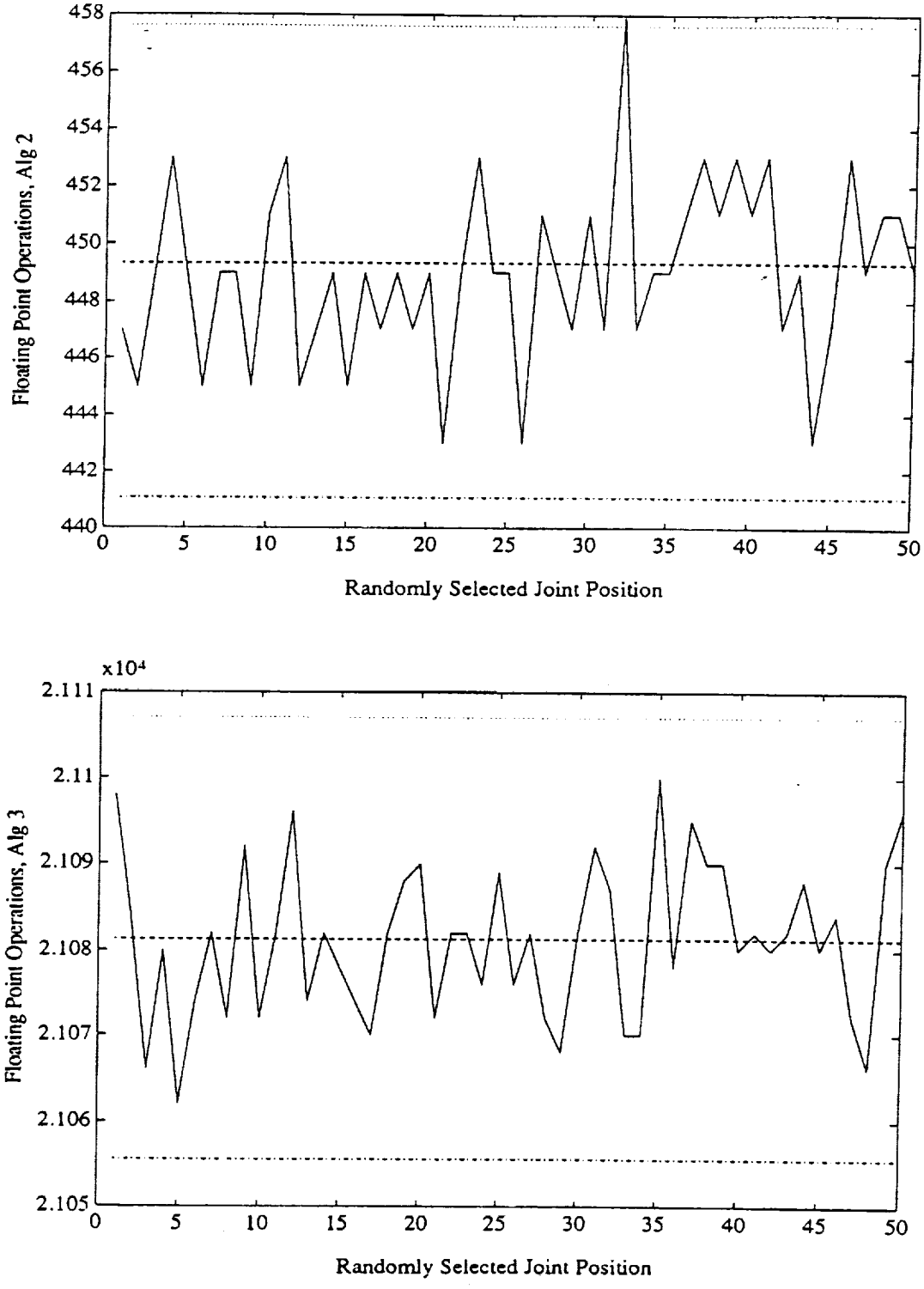
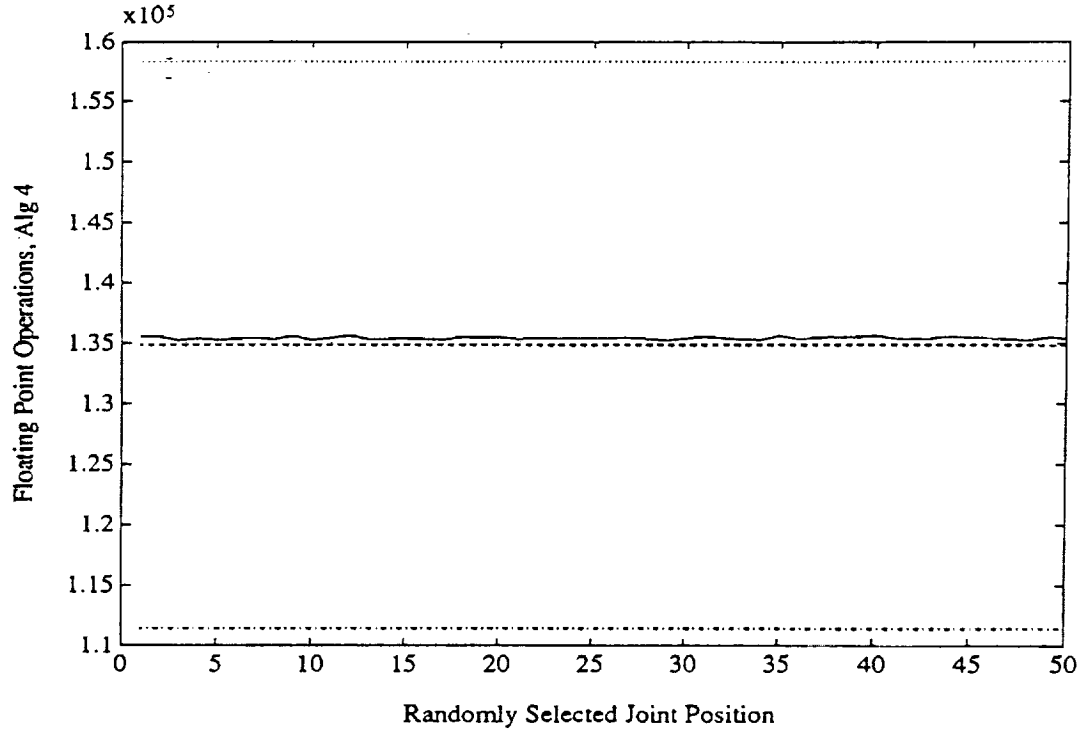


Figure C.4: K_2 and K_3 Floating Point Operations



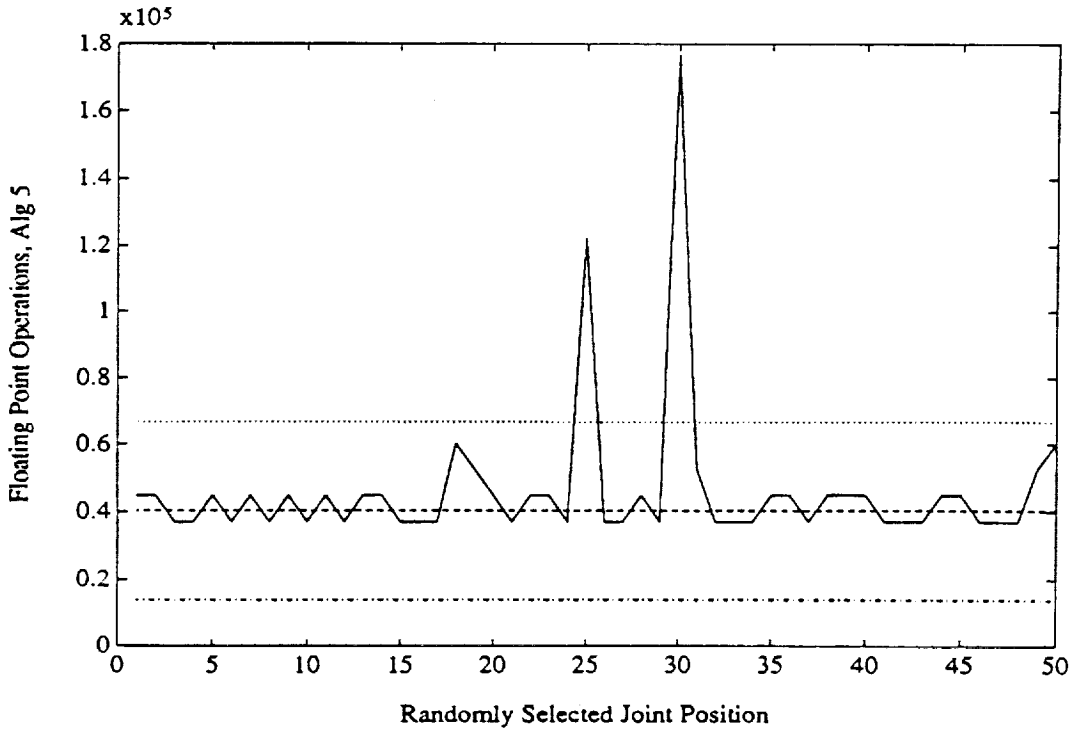


Figure C.5: K_4 and K_5 Floating Point Operations

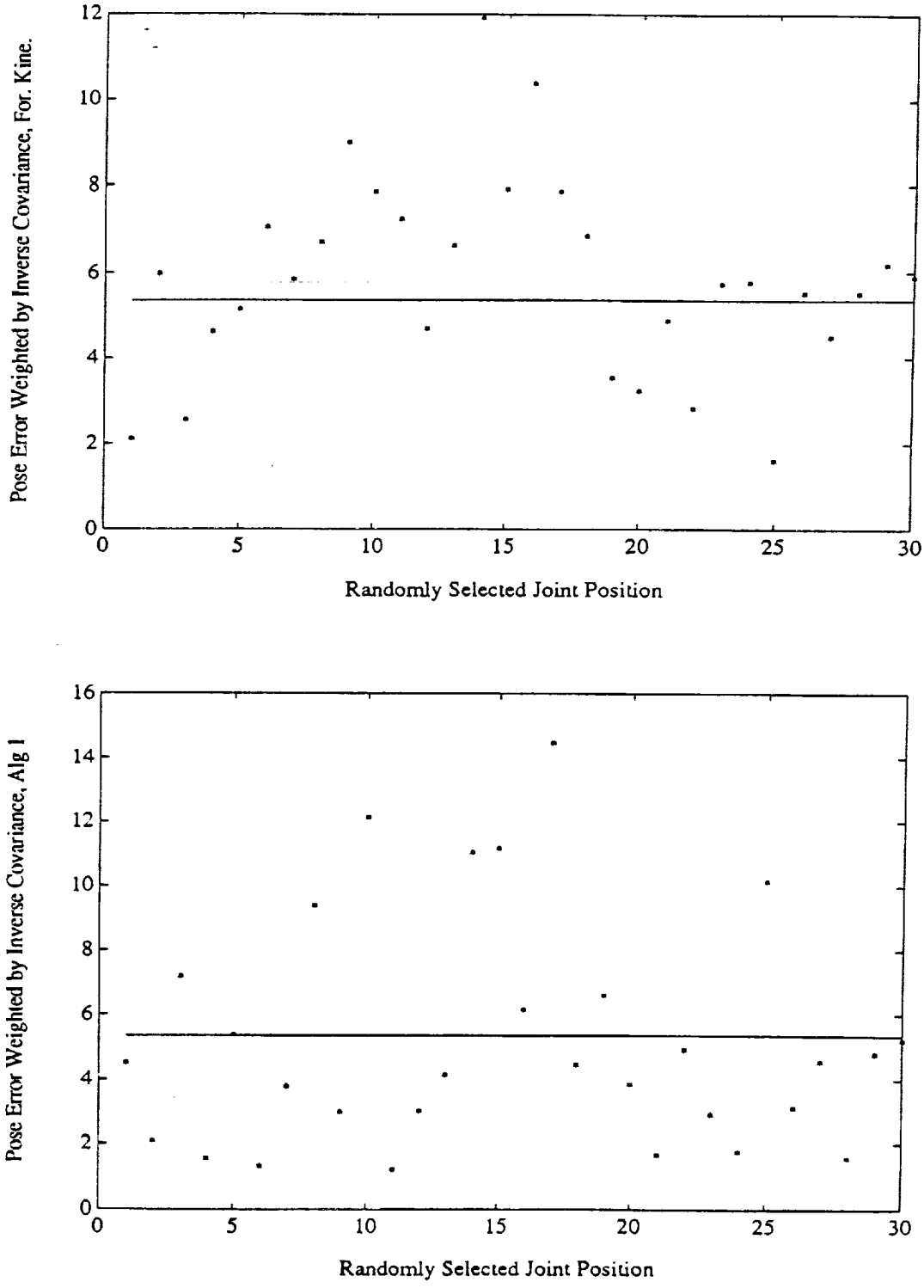
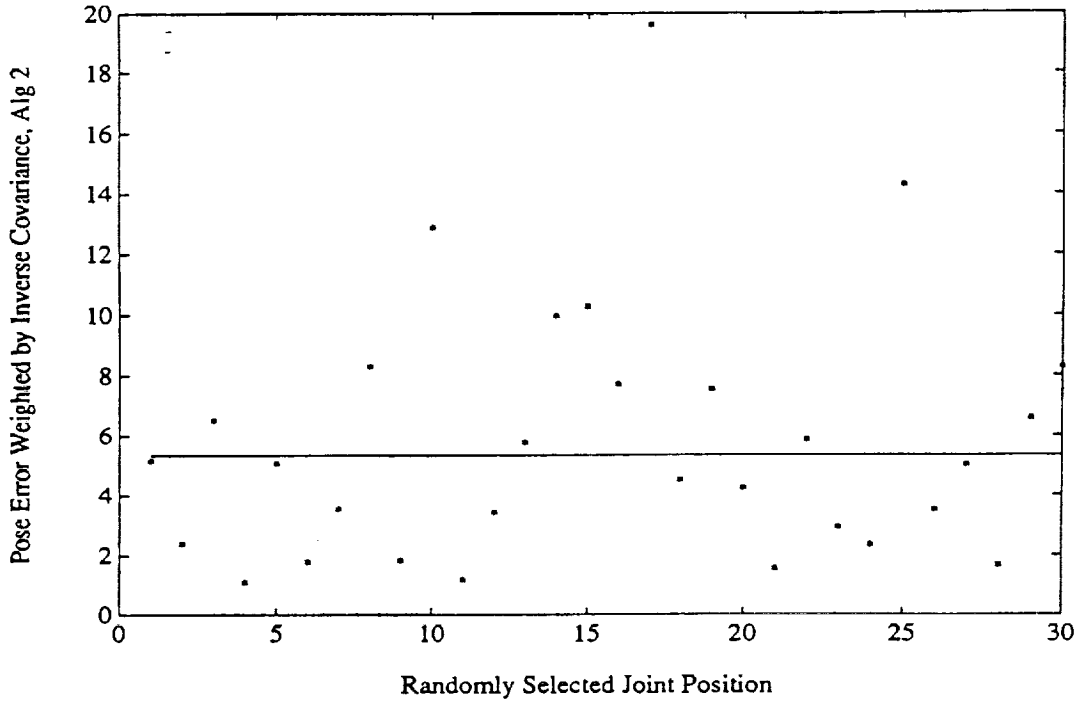


Figure C.6: Forward Kinematic and K_1 Weighted Norms



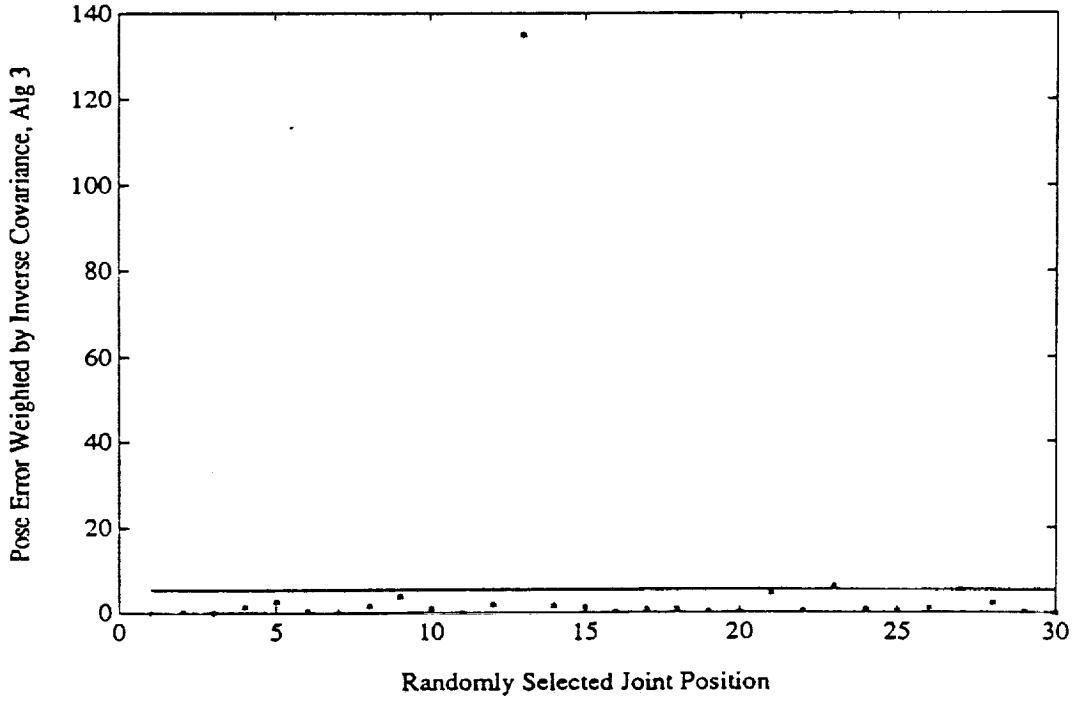
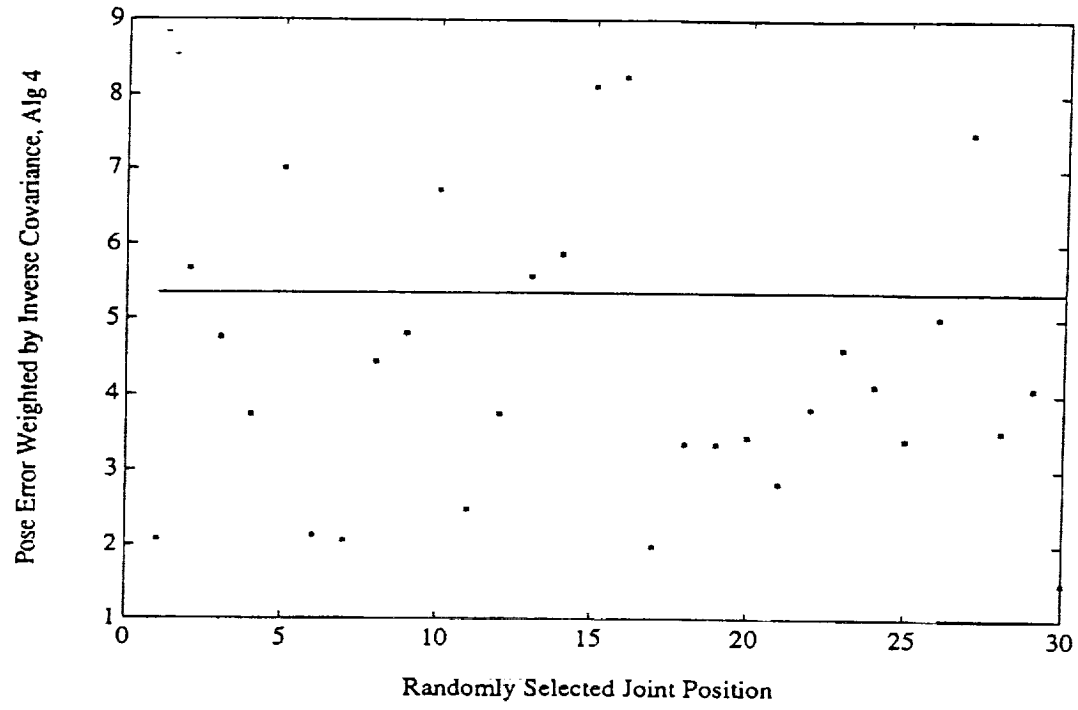


Figure C.7: K_2 and K_3 Weighted Norms



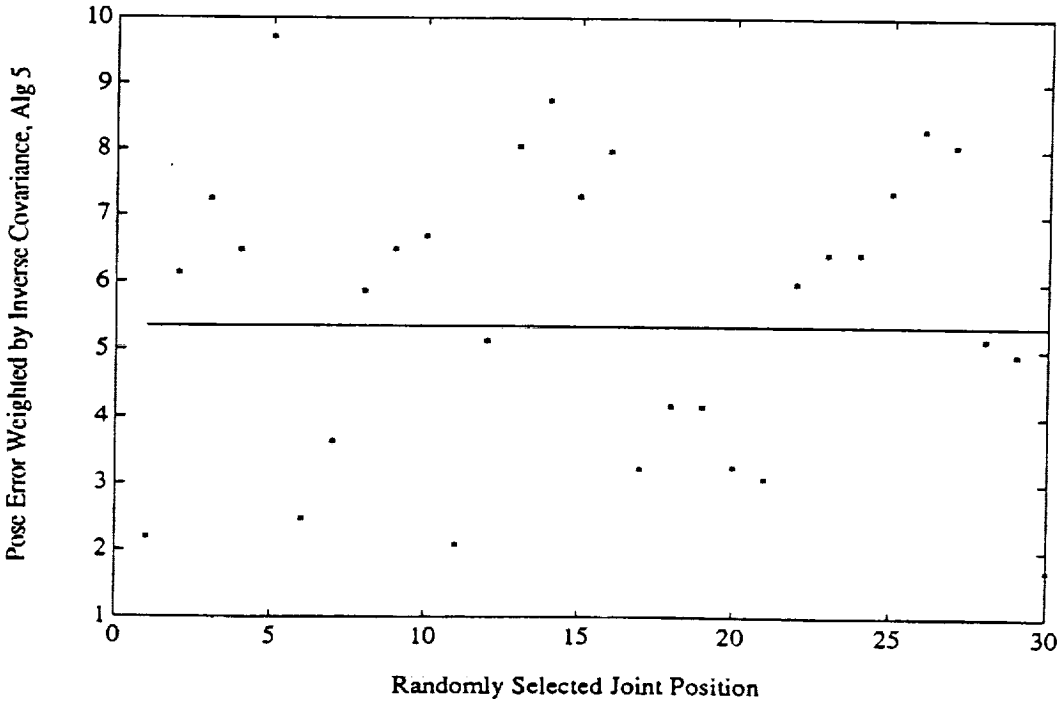
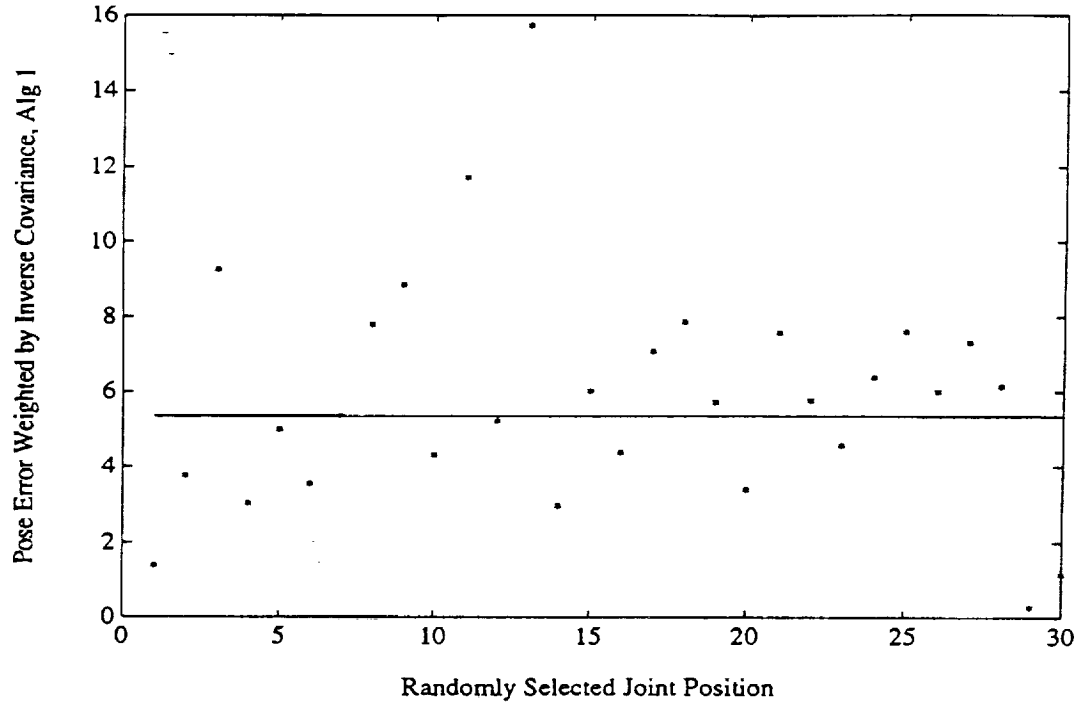


Figure C.8: K_4 and K_5 Weighted Norms



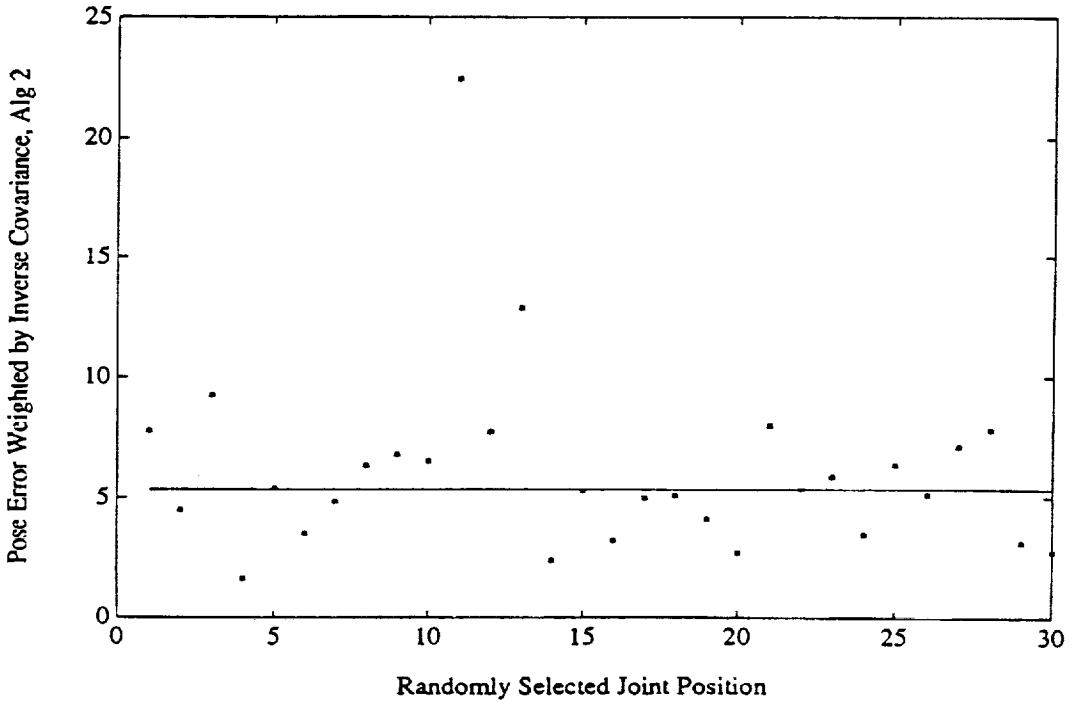


Figure C.9: K_1 and K_2 Weighted Norms with Backlash

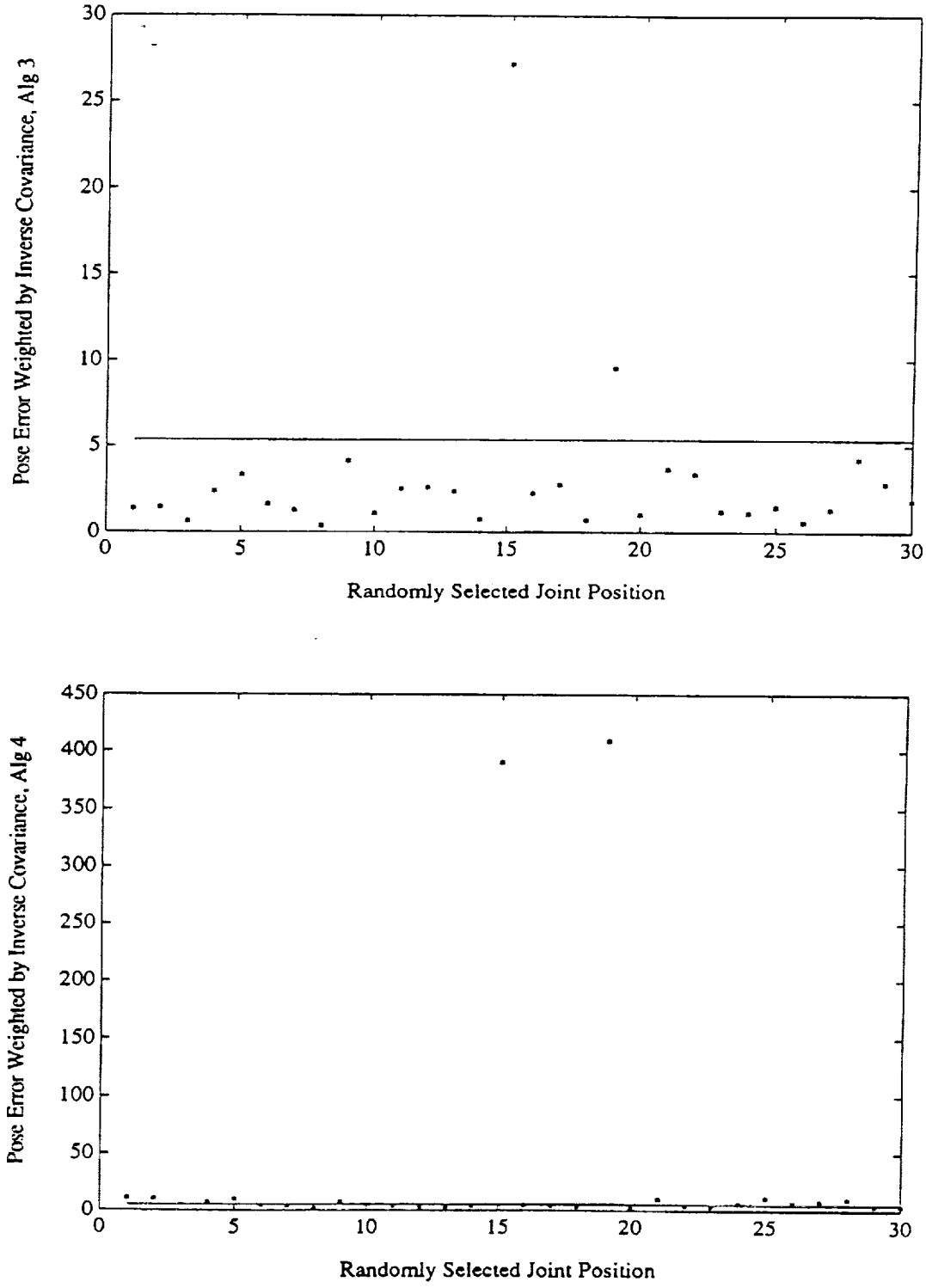


Figure C.10: K_3 and K_4 Weighted Norms with Backlash

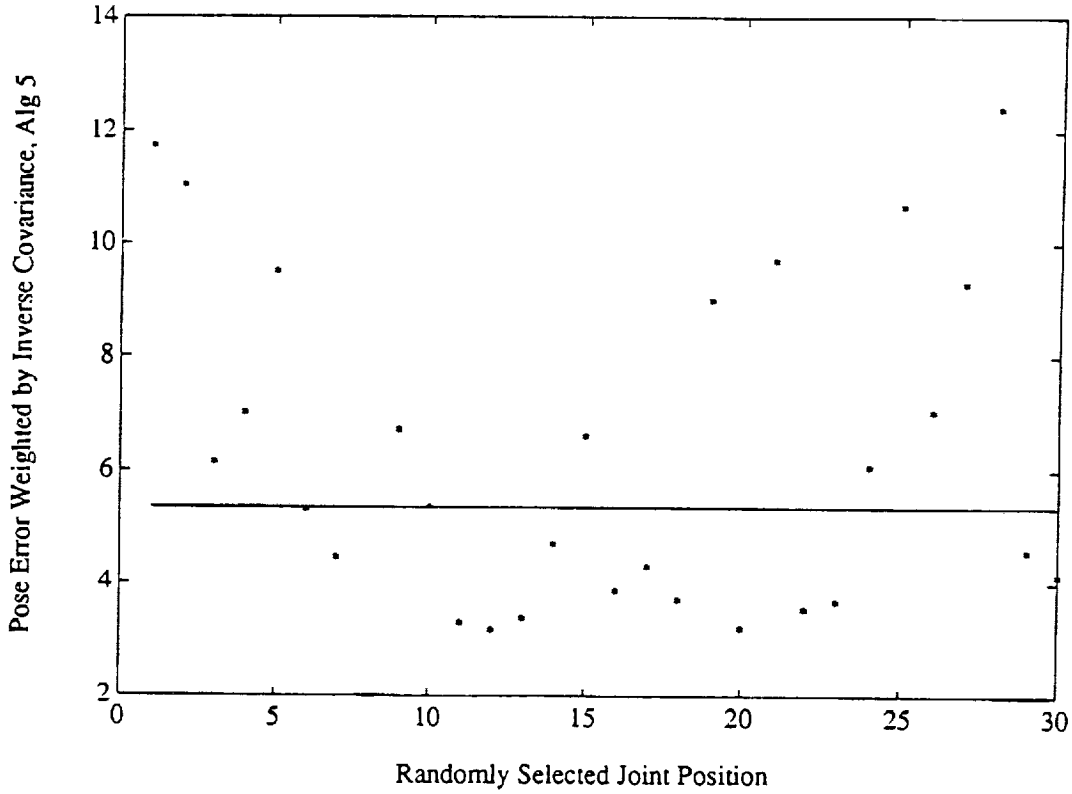
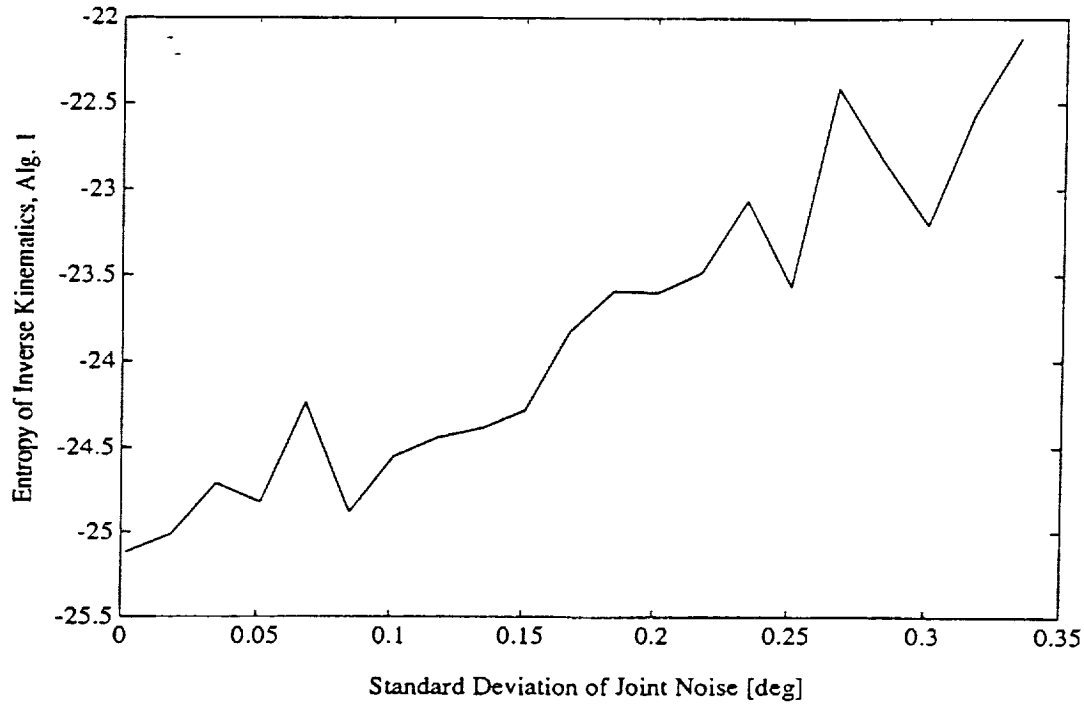


Figure C.11: K_5 Weighted Norm with Backlash



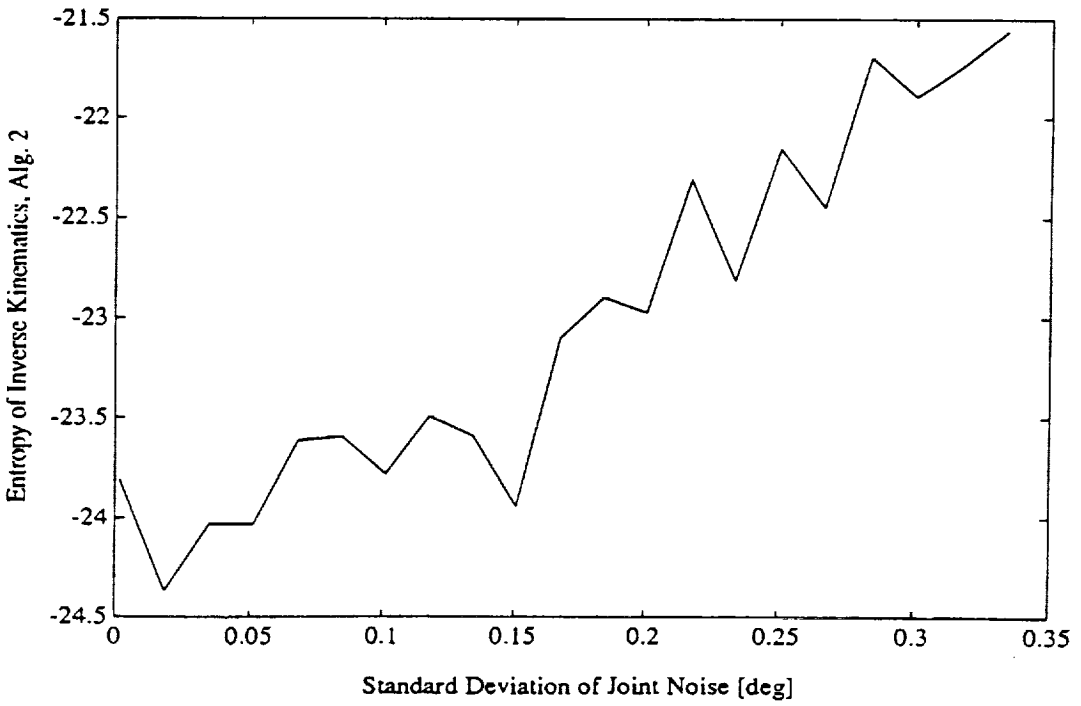


Figure C.12: Entropy of K_1 and K_2 with Joint Noise

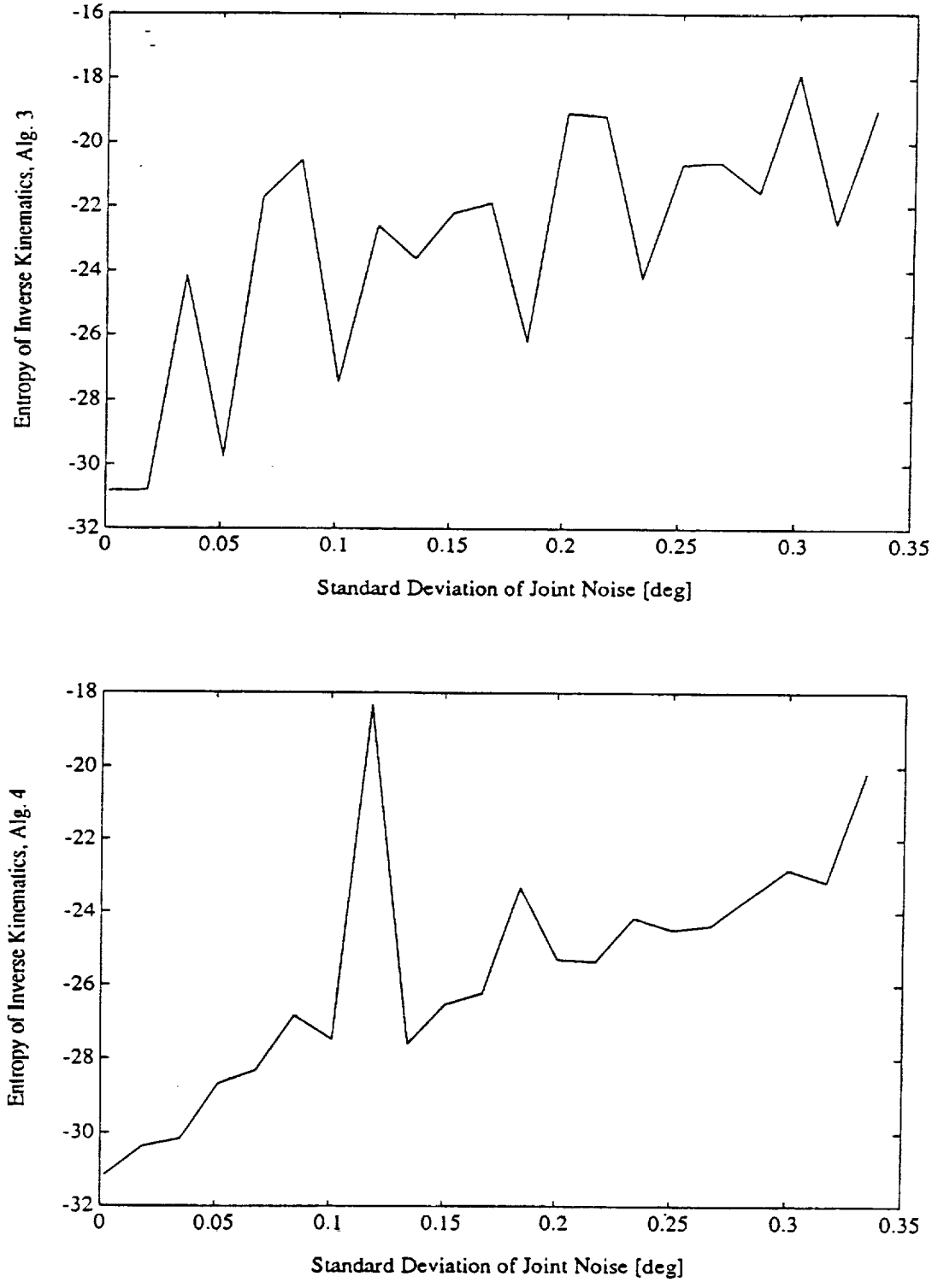


Figure C.13: Entropy of K_3 and K_4 with Joint Noise

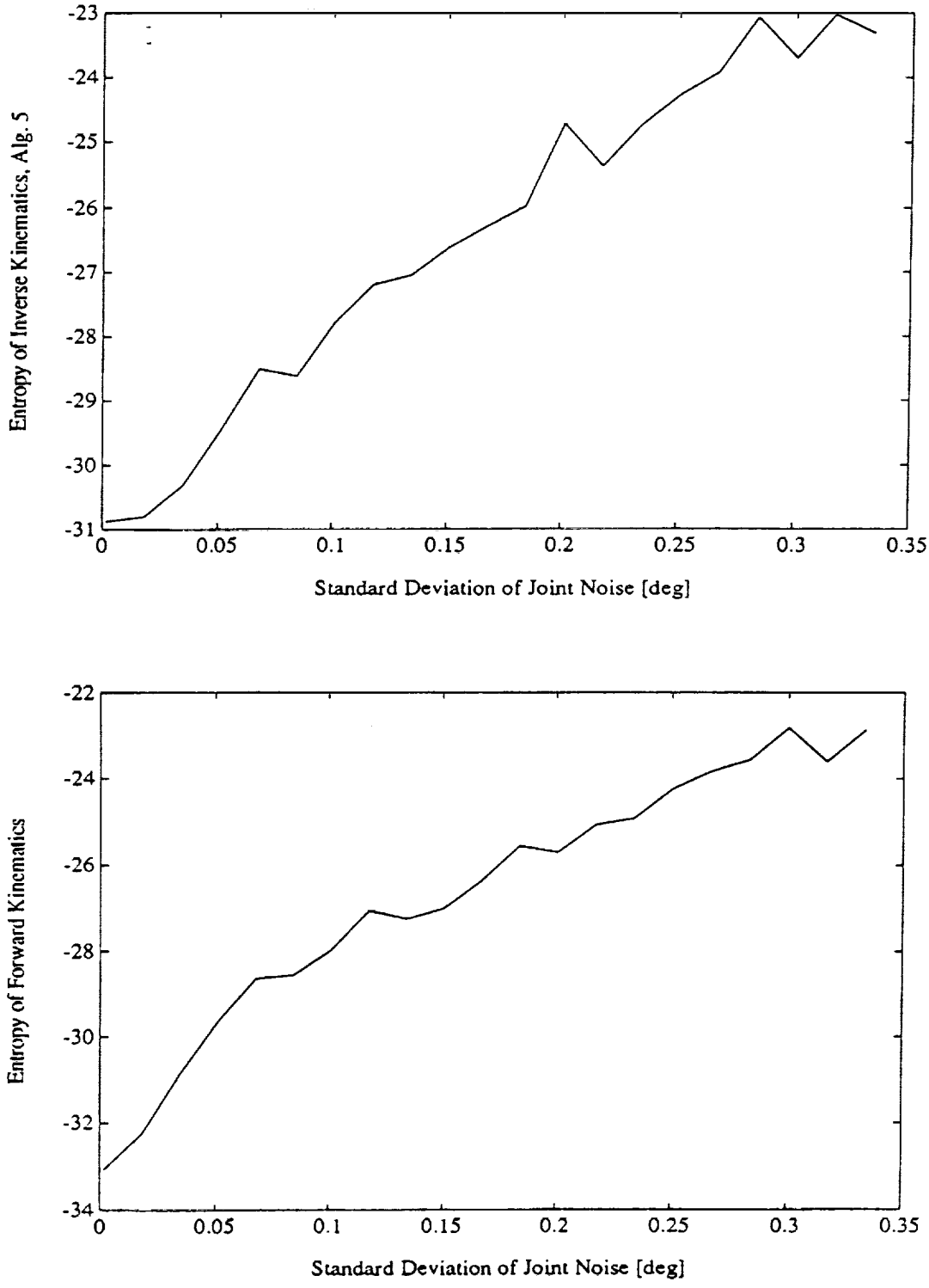
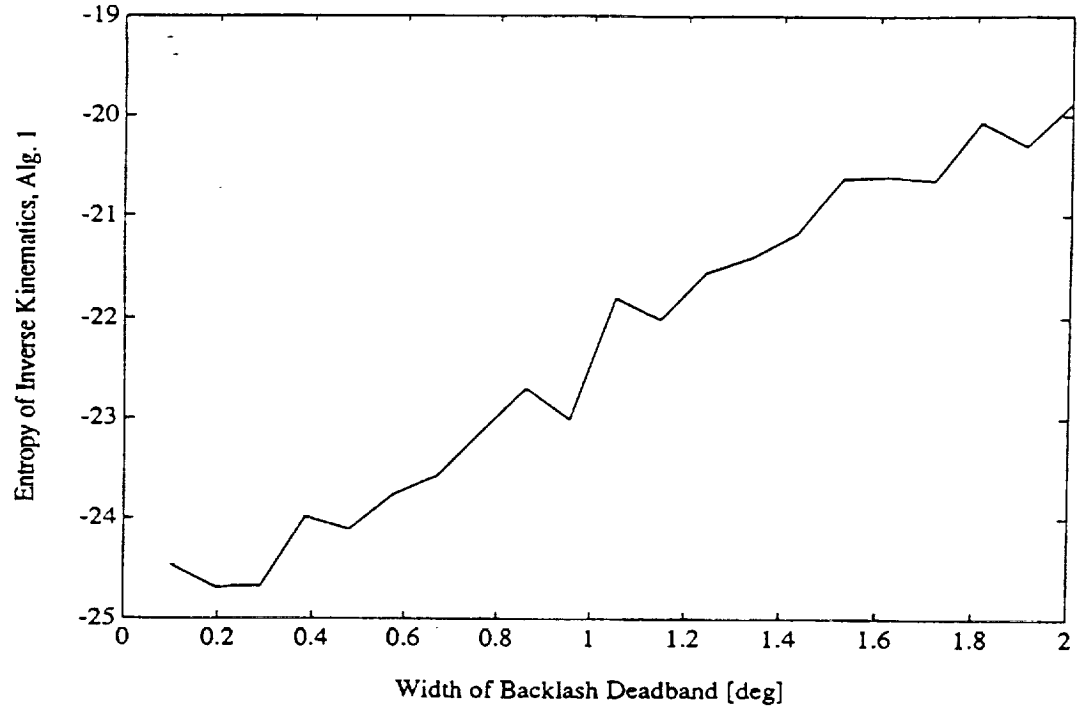


Figure C.14: Entropy of K_5 and Forward Kinematics with Joint Noise



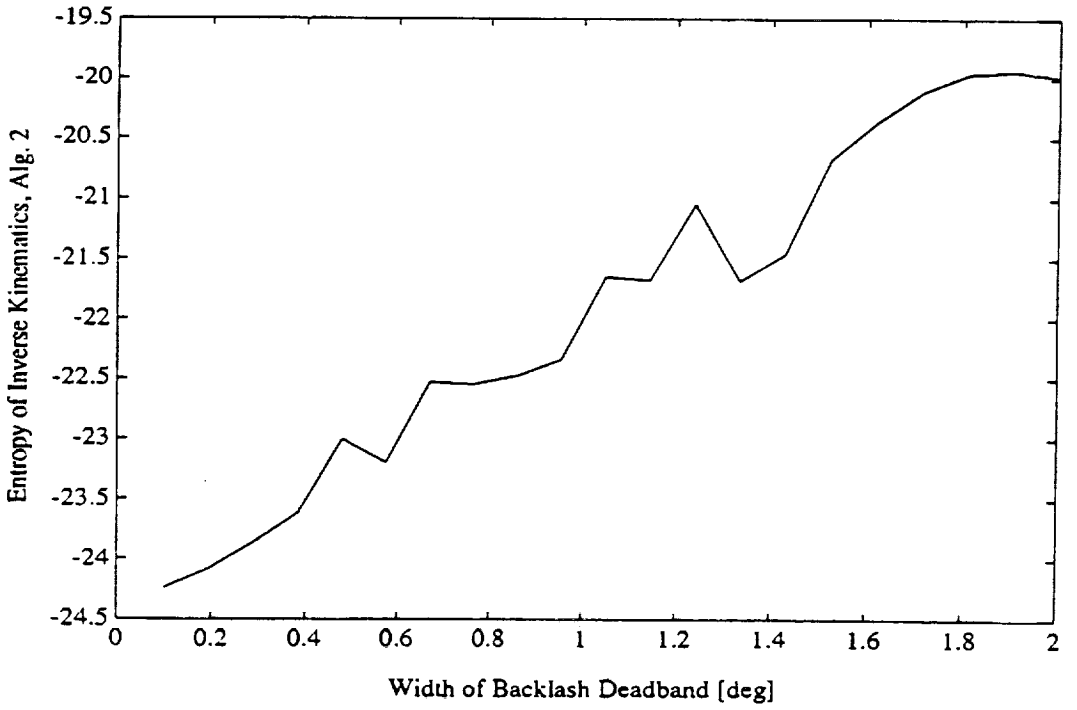
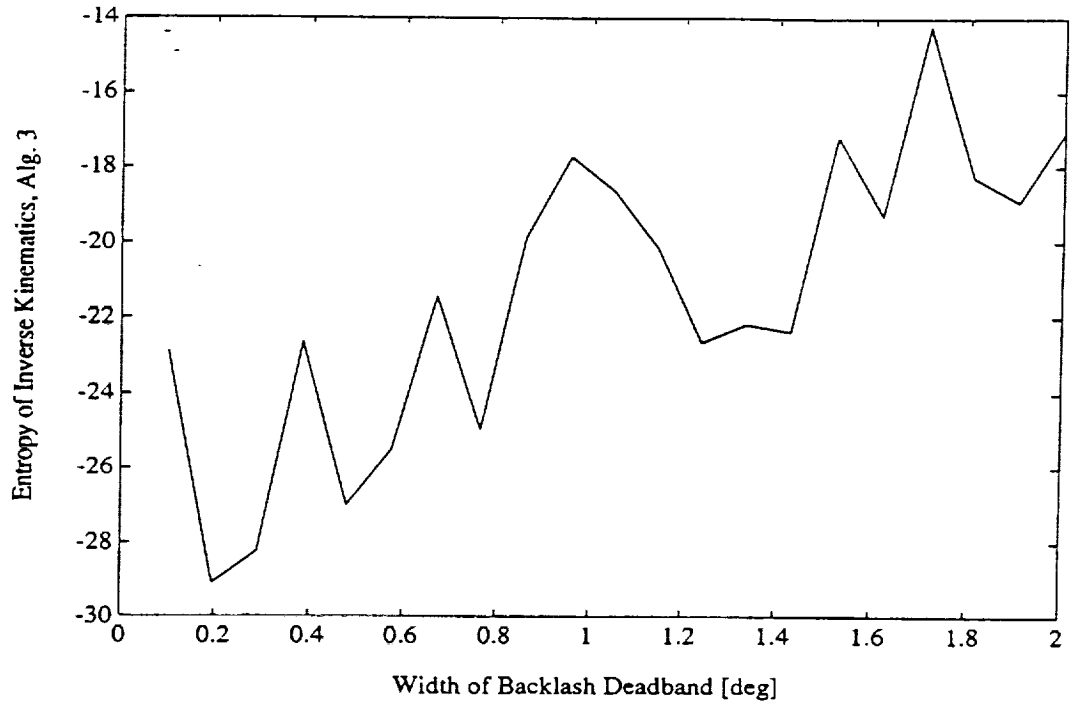


Figure C.15: Entropy of K_1 and K_2 with Backlash



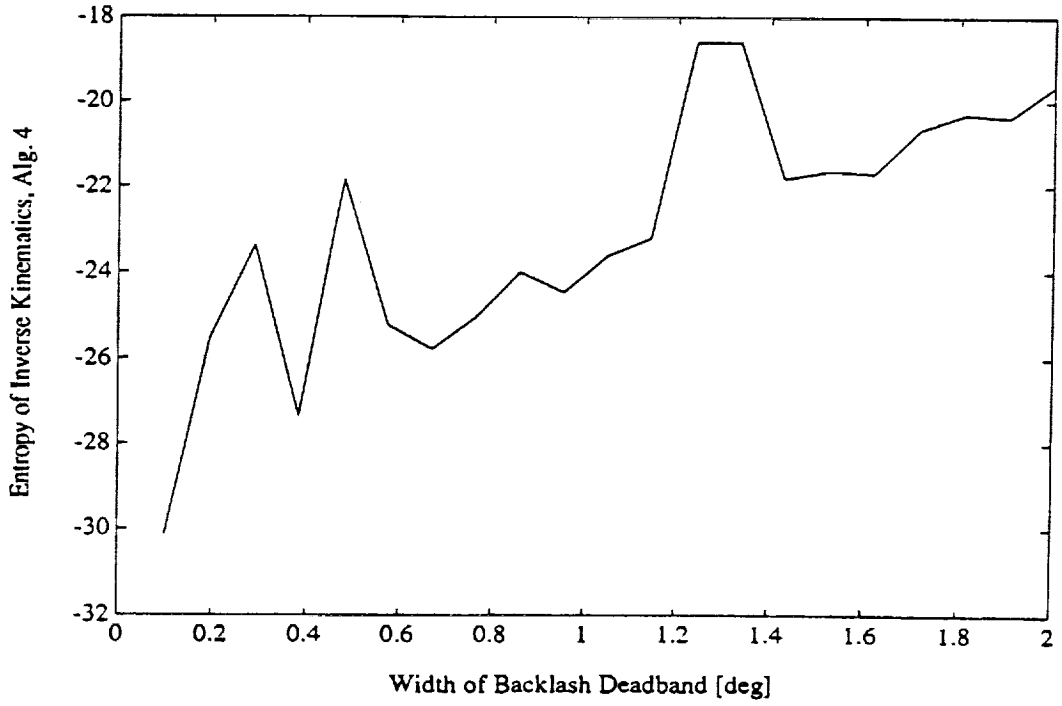


Figure C.16: Entropy of K_3 and K_4 with Backlash

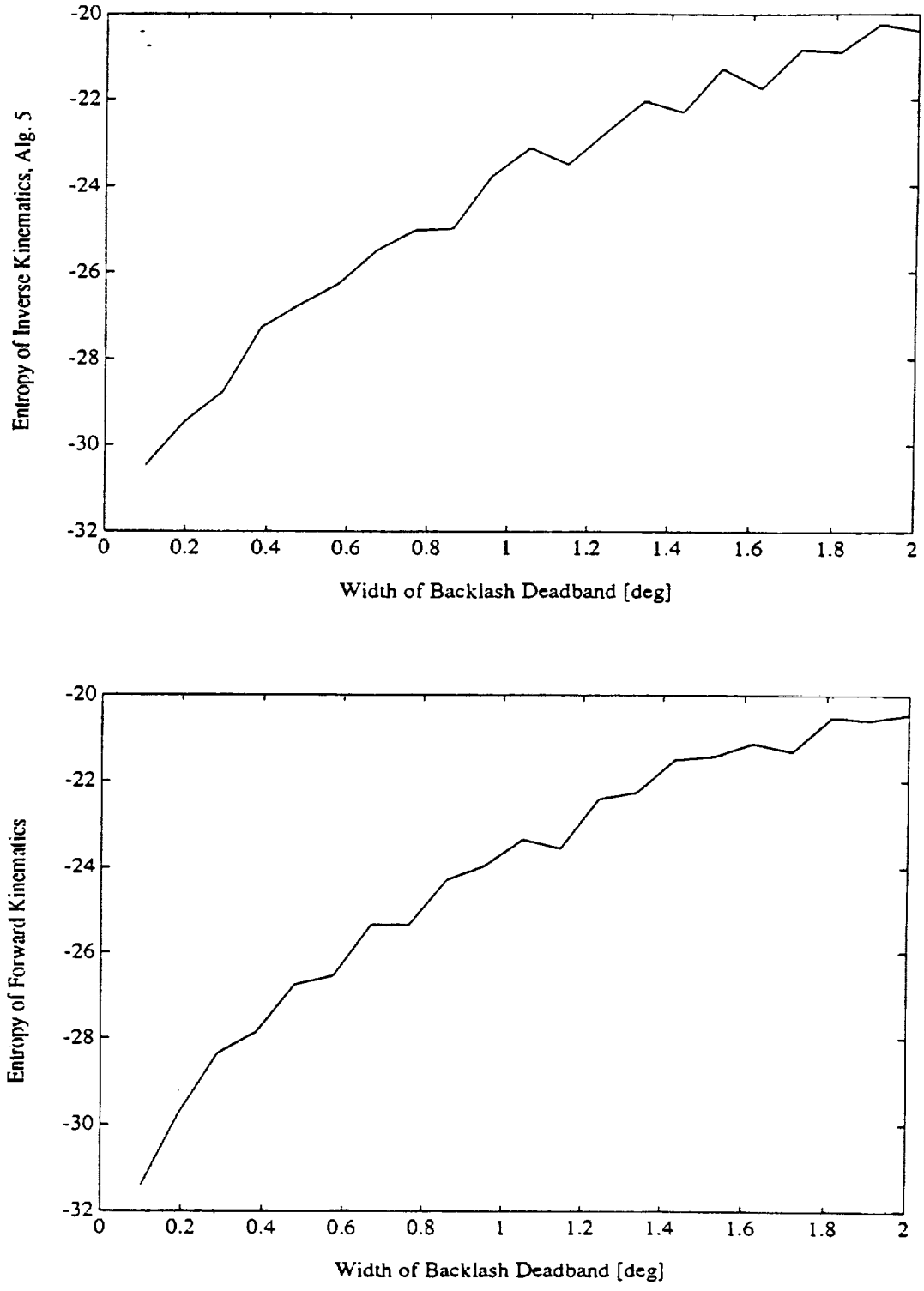


Figure C.17: Entropy of K_5 and Forward Kinematics with Backlash

APPENDIX D
PLAN EXECUTION SIMULATION RESULTS

This appendix plots the pose errors in each direction as the gripping frame is approached. The results for two plans are presented. First, plan $A = \{9, 5, 2\}$ is simulated. Since it uses vision algorithm 9, the camera is 0.4m from the target when the picture is taken. Additionally, K_5 (the Jacobi interative inverse kinematics method) is used, so errors due to the inverse kinematics are minimized. Finally, G_2 is used as a compensator. Since it has a short settling time, it allows the trajectory to be traversed quickly. The errors are plotted in Figures D.1–D.3, along with the tolerance bounds obtained from the geometry of the gripping interface. The errors are well within the tolerance bounds throughout the entire trajectory. This is the expected result because $A = \{9, 5, 2\}$ has been validated by the analysis procedure to be a reliable plan.

Plan $A = \{1, 1, 1\}$, on the other hand, is not deemed to be a reliable plan. The errors and tolerance bounds for $A = \{1, 1, 1\}$ are plotted in Figures D.4–D.6. The plan does not meet the specifications. In fact, the tolerance bounds are exceeded in both the "Y" and "Z" directions.

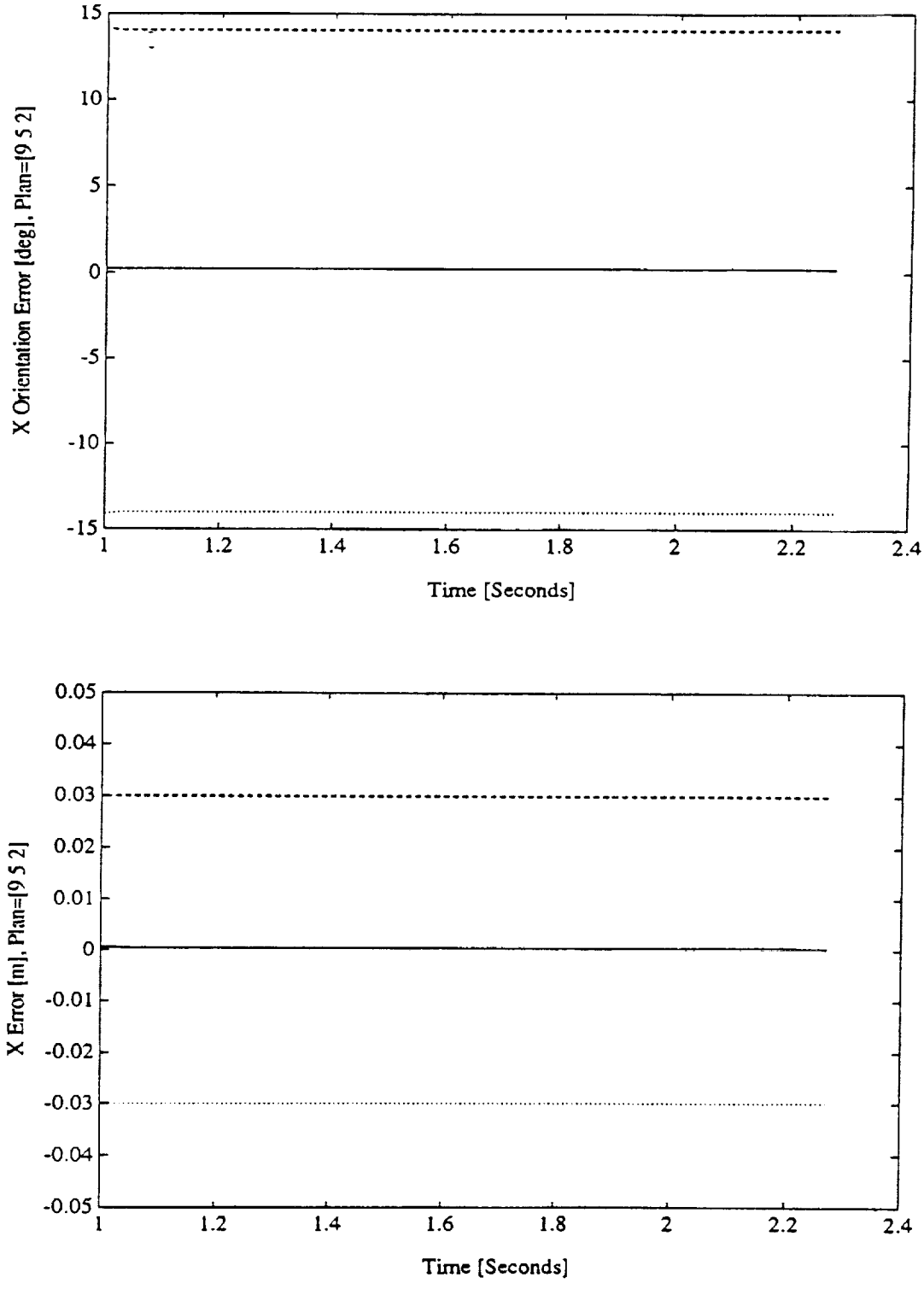


Figure D.1: Error and Tolerance Bounds for the "X" Axis, Plan (9,5,2)

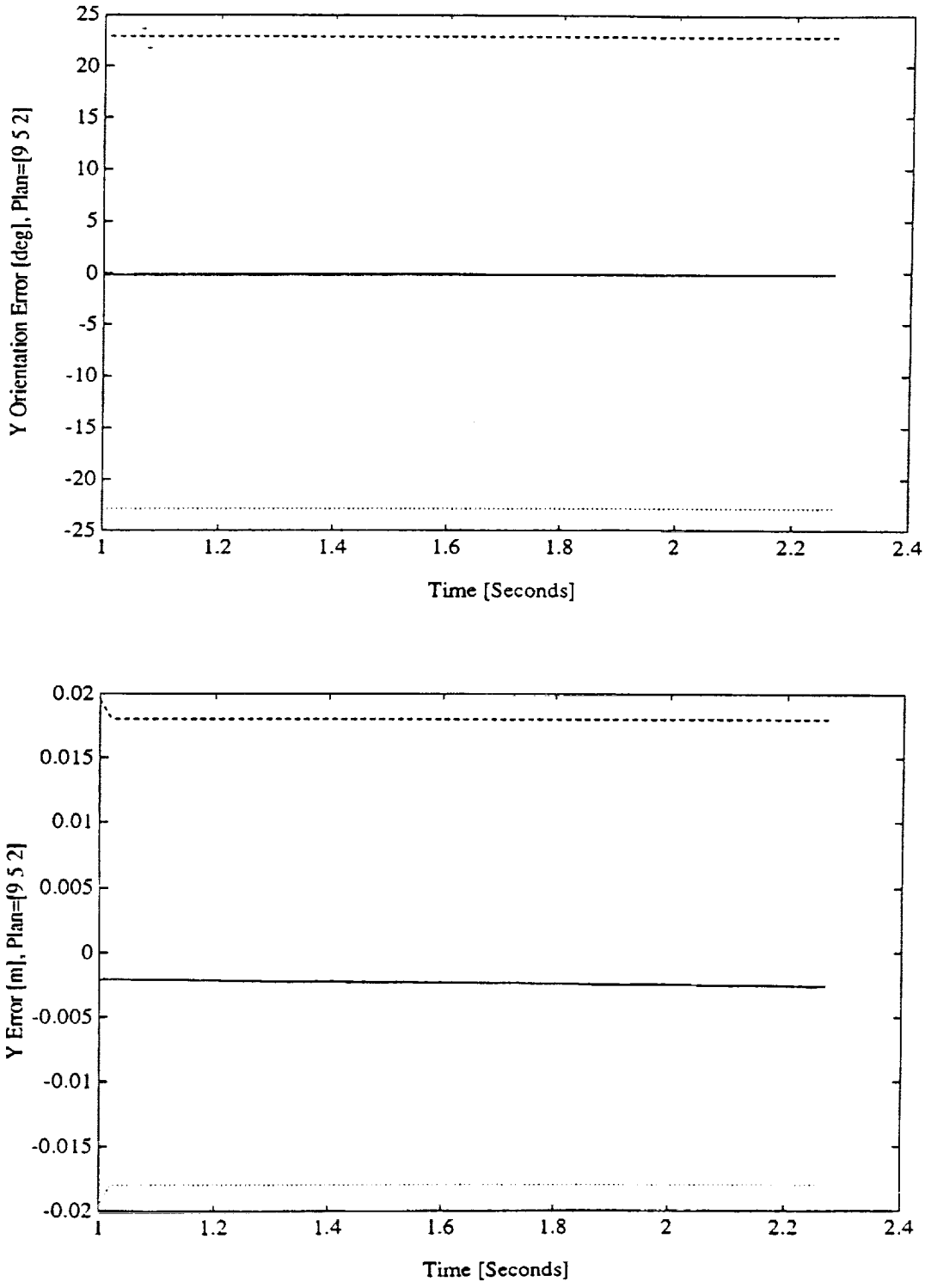


Figure D.2: Error and Tolerance Bounds for the "Y" Axis, Plan (9,5,2)

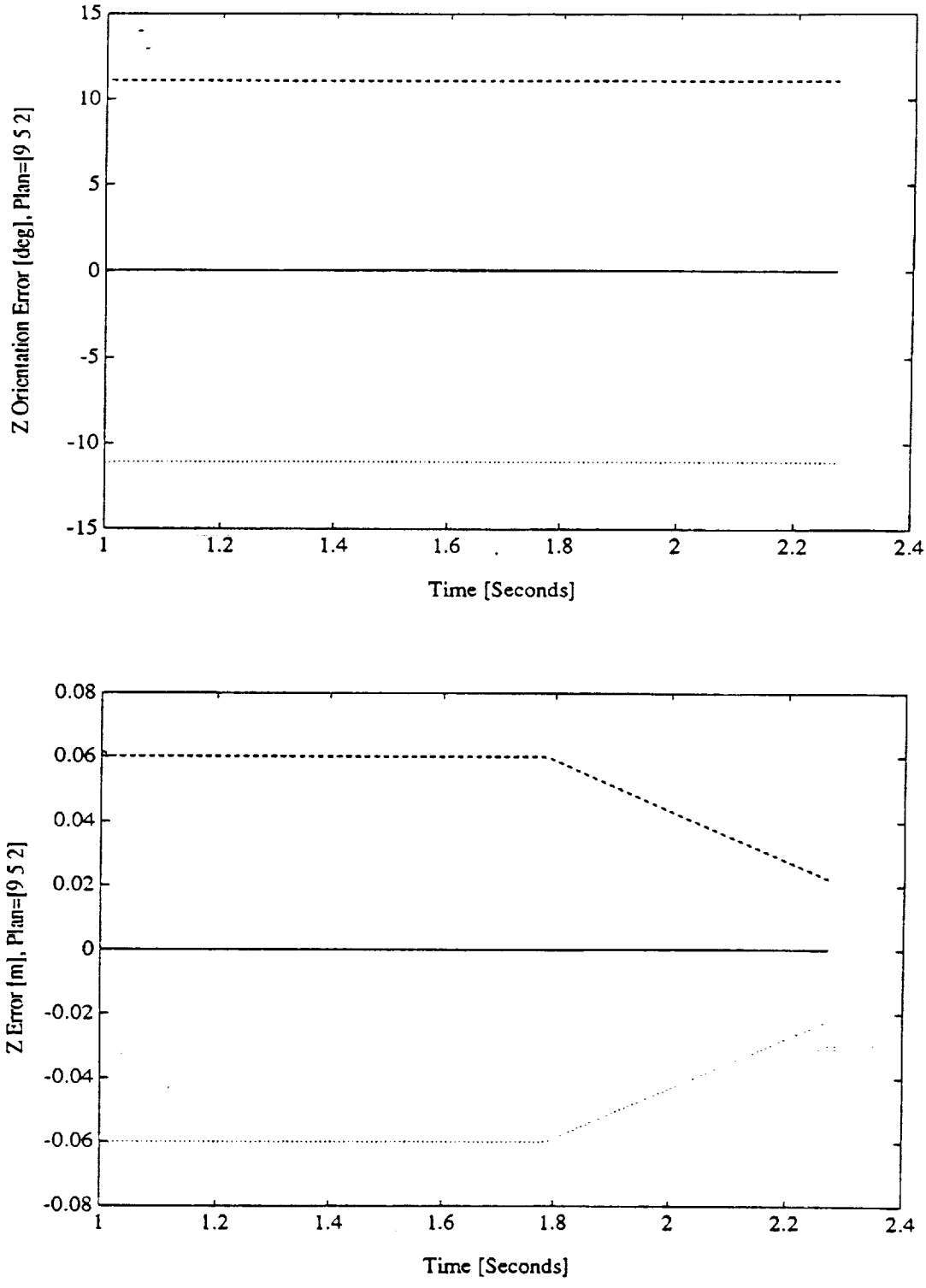


Figure D.3: Error and Tolerance Bounds for the "Z" Axis, Plan (9,5,2)

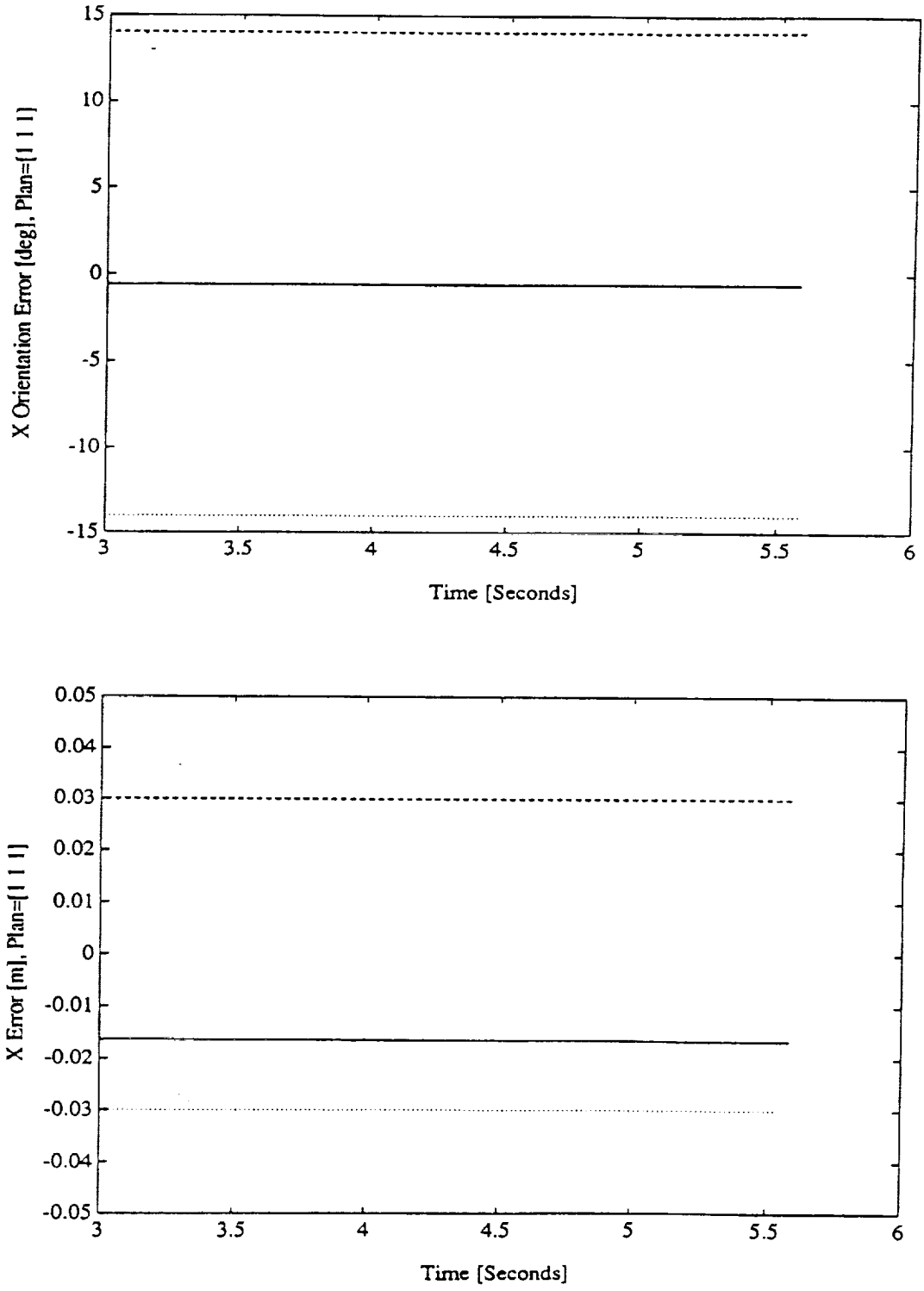


Figure D.4: Error and Tolerance Bounds for the "X" Axis, Plan (1,1,1)

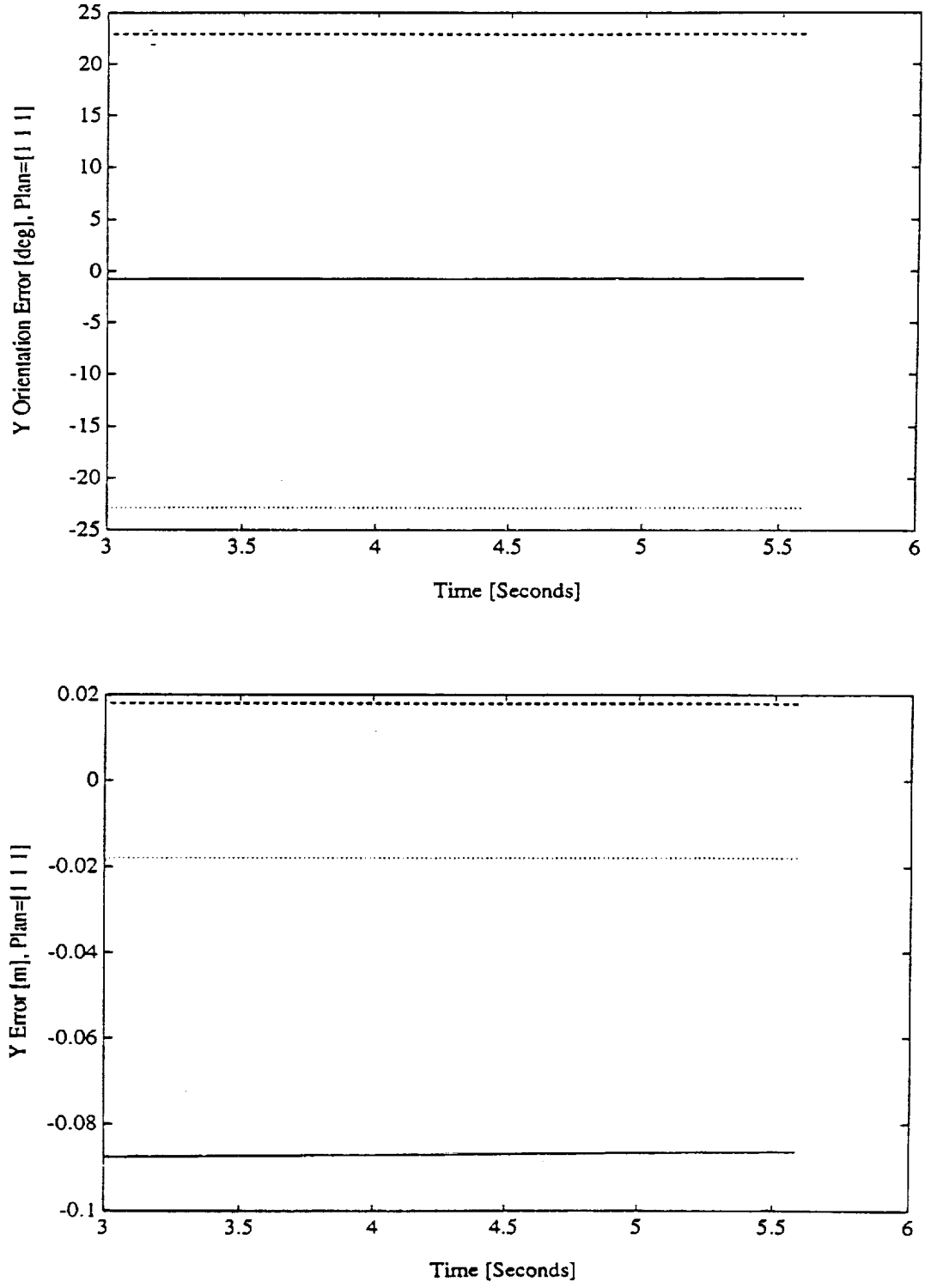


Figure D.5: Error and Tolerance Bounds for the "Y" Axis, Plan (1,1,1)

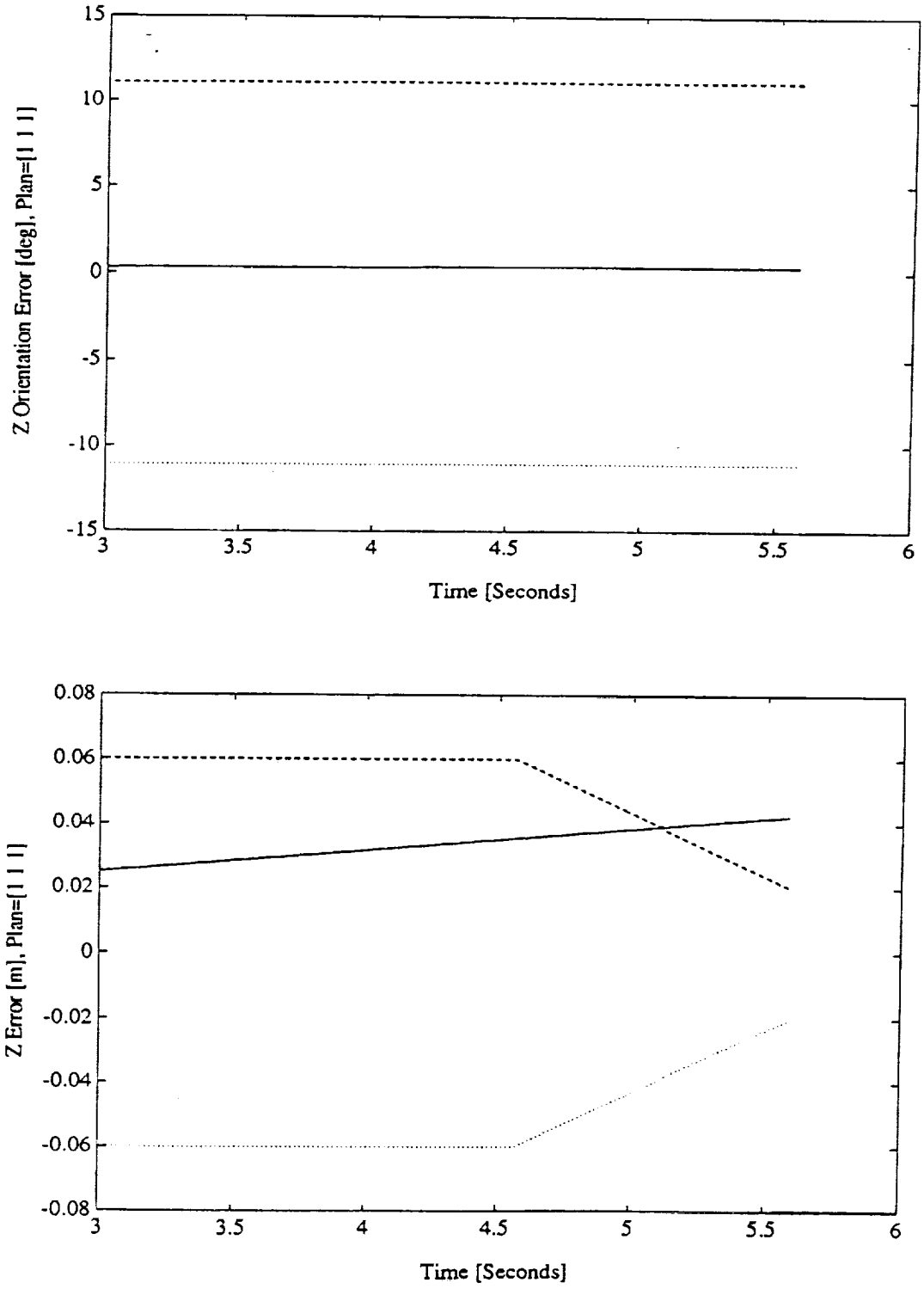


Figure D.6: Error and Tolerance Bounds for the "Z" Axis, Plan (1,1,1)