GUARANTEED MANIPULATOR PRECISION VIA INTERVAL ANALYSIS OF INVERSE KINEMATICS

Muhammed R. Pac
Next Generation Systems Group
Department of Electrical Engineering
University of Texas at Arlington
Texas 76010

Micky Rakotondrabe
AS2M Department
FEMTO-ST Institute
University of Franche-Comté (UFC)
25000 Besançon, France

Sofiane Khadraoui
Texas A&M University at Qatar
Education City, PO Box 23874
Doha, Qatar

Dan O. Popa∗
Next Generation Systems Group
Department of Electrical Engineering
University of Texas at Arlington
Texas 76010
Email: popa@uta.edu

Philippe Lutz
AS2M Department
FEMTO-ST Institute
University of Franche-Comté (UFC)
25000 Besançon, France

ABSTRACT

The paper presents a new methodology for solving the inverse problem of manipulator precision design. Such design problems are often encountered when the end-effector uncertainty bounds are given, but it is not clear how to allocate precision bounds on individual robot axes. The approach presented in this paper uses interval analysis as a tool for uncertainty modelling and computational analysis. In prior work, the exponential formulation of the forward kinematics map was extended to intervals. Here, we use this result as an inclusion function in the computation of solutions to set-valued inverse kinematic problems. Simulation results are presented in two case studies to illustrate how we can go from an uncertainty interval at the end-effector to a design domain of allowable uncertainties at individual joints and links. The proposed method can be used to determine the level of precision needed in the design of a manipulator such that a predefined end-effector precision can be guaranteed. Also, the approach is general as such it can be easily extended to any degree-of-freedom and kinematic configuration.

INTRODUCTION

The success of automated assembly by robotic manipulators is highly dependent on the precision of the positioning mechanisms employed in the kinematic design of the robot. The importance of precision becomes more prominent when the desired operational accuracy is in micro/nano scale as in micro-assembly and nano-manipulation applications. Most of the parametric uncertainties that are negligible in conventional robotics become the predominant error sources in micro/nano applications. With the emergence of micro/nano-robotics in the last decade, there is a growing demand for design guidelines describing how to build these robots based on application specific precision criteria. Since the current methodology have not addressed this problem well, we propose a new approach to the kinematic analysis and design of robots using interval analysis.

In late 80's and early 90's, modelling of errors in robot kinematics was a popular topic [1–6]. However, these works are limited to analysing the effects of general error transformations and do not address how they come about or how to choose or design the parameters of a manipulator for a given end-effector preci-

---

∗Address all correspondence to this author.

1

sion. On the other hand, mechanical design of multi-axis machines based on the accuracy required at the tool tip has been investigated for precision machine design. A concept called error budget was proposed to account for the effect of each error source on the tool accuracy [7, p. 61], [8]. Using first order and small angle approximations to simplify the homogeneous transformation matrices describing geometric errors in the mechanism of a machine, an overall kinematics map can be created to analyse the effects of error terms on the tool accuracy. The results of this analysis provide the designer with insight into how to allocate mechanical tolerances to the individual system components. However, there is no systematic way of doing this allocation due to the fact that this type of an inverse problem is difficult to pose and solve.

A recent work in [9] studies modular robotic chains made up of individual axes with link/joint uncertainties for several different configurations. It was shown that some of the kinematic configurations provide more successful operation. Also, joint position uncertainties were shown to manifest themselves as end-effector inaccuracy in different magnitudes depending on the kinematic design. The analysis of error propagation in this work was done using Monte Carlo simulations which suffer from inability to provide guaranteed closed form solutions since Monte Carlo method can only solve for sample points and produce sets of points in the solution space.

Finding guaranteed solutions to a set of equations with uncertain data is possible with interval analysis. Interval analysis is a mathematical tool for computation of rigorous bounds on solutions to ideal model equations when the input arguments of the model are represented as intervals instead of point values. It extends the model equations to the interval domain and allows for analytical and computational handling of uncertain data without having to assume a distribution for it or to sample it. It also helps avoid the complex mathematical formulations involving distribution functions [10].

Intervals were used in [11] to model uncertain physical parameters of a robot and to find its forward kinematics map using Denavit-Hartenberg (D-H) notation. Optimization of the D-H parameters was done to minimize volumetric end-effector error while optimizing the cost of precision. While this work nicely approaches the problem from a mechanism design perspective, it fails to consider the fact that a mechanism design cannot be optimized practically without considering the uncertainty in joint positions. Interval analysis was also used in [12] to find the multiple solutions to the forward kinematics problem of parallel robots. Then, [13] extends the method to finding the robot parameters that guarantee a singularity-free workspace.

In our prior work [14], the exponential formulation of the forward kinematics map for serial manipulators was extended to intervals. This makes it possible to use interval analysis to find guaranteed precision bounds on the end-effector pose given the uncertainty of the kinematic parameters. The contribution of the current paper is that we now propose a new method that can solve the inverse problem of bounding the allowable uncertainty in kinematic parameters of a manipulator based on given end-effector precision specifications. Besides precision machine designers, this method bears an importance for those roboticists who have to design a manipulator using elementary building blocks. For instance, custom design of multi-axis precision manipulators using individual single-axis stages is a common practice in the micro-assembly area [15–17]. The cost of such stages increase significantly with the increase in motion precision. For a given application, therefore, determining the level of precision required in each axis is an important yet insufficiently addressed consideration.

Next section provides an overview of [14], the forward kinematics problem with uncertain parameters. Then, we discuss how the precision manipulator design problem can be posed via interval analysis of inverse kinematics. We present simulation results illustrating the use of our method for a simple 2-link manipulator and a 3-DOF PPR stage, a commonly used robot in precision assembly. Finally, the paper is concluded with references at the end.

FORWARD KINEMATICS WITH UNCERTAIN JOINT PARAMETERS

Kinematic description of a robot involves certain parameters such as link lengths and joint axis vectors. In an ideal robot model, these are all assumed to be perfectly precise values. A practical implementation, however, requires calibrating the robot to minimize the uncertainty in the knowledge of such constants. Even with calibration, identification of kinematic parameters can only be done with a limited precision. Robot kinematics also involves joint position variables which are known to a limited precision as well. Therefore, there needs to be a kinematic model that can handle these uncertainties with ease and efficiency.

Using the screw-based exponential formulation [18] and interval analysis [10], we formulated the interval extension of forward kinematics map in [14]. The advantage of the exponential formulation over D-H parameters is that the joint parameters can be described with respect to the world reference frame which is the frame of reference used in the description of precision metrics. We model uncertain parameters and variables of a robot with intervals. For example, a (real) interval such as x $[1,2]$ is a closed set of real numbers represented by the two boundary values, 1 is called the left endpoint and 2 is called the right endpoint. In this paper, we follow the notation in [19] and denote intervals with square bracketed letters.

For a revolute joint, the interval extension of the homogeneous transformation matrix, $[T]$, can be represented with the function in (1) where $[\omega]$ is the joint axis vector, $[\hat{\omega}]$ is the skew-symmetric matrix of $[\omega]$, $[\theta]$ is the joint displacement, and $[q]$ is a point along the joint axis. Note that in exponential formula-

tion, joint transformations are with respect to the base reference frame.

$$[T]([\omega],[\theta],[q]) \quad \begin{array}{cc} e^{[\hat{\omega}][\theta]} & (I - e^{[\hat{\omega}][\theta]})[q] \\ 0 & 1 \end{array} \quad (1)$$

In equation (1), $[T]([\omega],[\theta],[q])$ is called an inclusion function since it covers all possible homogeneous transformation matrices $T(\omega,\theta,q)$ of the special Euclidean group $SE(3)$ that can be obtained by varying the arguments ω, θ, and q in the respective ranges of $[\omega]$, $[\theta]$, and $[q]$.

Similarly for a prismatic joint, the interval extension for the joint transformation is represented with the inclusion function in (2) where $[d]$ is the joint displacement and $[v]$ is the joint axis vector. The rotation part of the transformation denoted with $[R]$ is ideally a 3×3 identity matrix. In an actual linear axis, however, tilt error of the stage may also play an important role. Hence, this part can be modelled as a general rotation transformation $[R] = e^{[\hat{\omega}][\theta]}$.

$$[T]([d],[v]) \quad \begin{array}{cc} [R] & [d][v] \\ 0 & 1 \end{array} \quad (2)$$

Finally, interval extension of the forward kinematics map of a serial manipulator with n joints can be written as

$$[f]([x]) = [T_1] \cdots [T_n] \cdot [f]([x_0]) \quad (3)$$

where $[f]$ represents the inclusion function for the forward kinematics map and $[x]$ is the vector of joint parameters and variables. For further details on the forward kinematics map, the reader is referred to [14].

INTERVAL ANALYSIS OF INVERSE KINEMATICS

Computing the joint angles of a manipulator for a given end-effector pose is called inverse kinematics. There are two main types of solution to this problem: closed-form solutions and numerical solutions [20, p. 106]. Closed-form solutions are based on analytical expressions of the inverse relationship between joint angles and end-effector pose. Due to the complex nonlinear nature of robot kinematic equations, finding a closed-form solution is difficult in general. On the other hand, numerical solutions rely on the description of forward kinematics map and repeated evaluation of approximate joint angles until the desired end-effector configuration is sufficiently approached.

Inverse Kinematics with Joint Parameter Uncertainty: An Example

Consider the simple example shown in Fig. 1 where a two-link planar manipulator with revolute joints is depicted. The
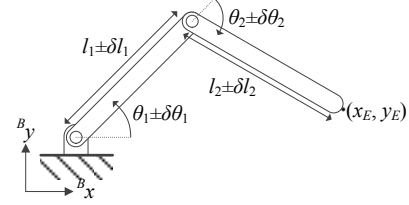

FIGURE 1: TWO-LINK MANIPULATOR WITH UNCERTAIN LINK LENGTHS AND JOINT ANGLES.

knowledge of the link lengths and joint positions are assumed to be uncertain to the extent denoted by δl_i and $\delta \theta_i$ for $i = 1, 2$, respectively. Given the nominal link lengths l_i and joint positions θ_i, the corresponding interval arguments can be expressed as

$$[x] \quad \begin{pmatrix} [l_1] \\ [\theta_1] \\ [l_2] \\ [\theta_2] \end{pmatrix} \quad \begin{pmatrix} [l_1 - \delta l_1, l_1 + \delta l_1] \\ [\theta_1 - \delta \theta_1, \theta_1 + \delta \theta_1] \\ [l_2 - \delta l_2, l_2 + \delta l_2] \\ [\theta_2 - \delta \theta_2, \theta_2 + \delta \theta_2] \end{pmatrix}. \quad (4)$$

It was shown in [14] that the rotation part of the revolute joint transformation in (1) reduces to the interval extension of the Rodrigues' formula when the joint axis is a real (point-valued) vector as in the case of this example. That is,

$$R(\omega,[\theta]) \quad e^{\hat{\omega}[\theta]} = I + \hat{\omega} \sin([\theta]) + \hat{\omega}^2(1 - \cos([\theta])) \quad (5)$$

where $\omega = (0,0,1)^T$ is a real vector for the manipulator in Fig. 1. Then, the inclusion function for the transformation of the end effector pose can be written as

$$[f]([x]) = [T](\omega_1,[\theta_1],[q_1]) \cdot [T](\omega_2,[\theta_2],[q_2]) \cdot [f]([x_0]) \quad (6)$$

where

$$[T](\omega_i,[\theta_i],[q_i]) = \begin{array}{cc} R(\omega_i,[\theta_i]) & (I - R(\omega_i,[\theta_i]))[q_i] \\ 0 & 1 \end{array}. \quad (7)$$

Then, the inverse problem for this mapping can be posed as follows:

For a given interval of end-effector position $[y] = [f]([x])$, what should be the manipulator parameters $[x] = ([l_1],[\theta_1],[l_2],[\theta_2])^T$?

Note that the answer to the above question can provide not only the range of $[\theta_i]$ for the desired interval of end-effector position but also the maximum allowable uncertainty in $[\theta_i]$ and $[l_i]$. Therefore, this methodology can be used as a tool for precision

design of manipulators. That is, determination of the required joint encoder resolution and of the manufacturing tolerance for the robot mechanism can be done based on the results of this inverse kinematic analysis.

When the kinematic mechanism is as simple as the one in Fig. 1, the point-valued inverse kinematics problem can be solved analytically as in (8) [20, p. 112]. These equations provide a couple of solutions for θ_1 and θ_2 for each end-effector position (x, y). However, as the number of joints increases, the forward kinematics map becomes non-invertible. Also, incorporation of mechanism errors and parameter uncertainties introduces additional degrees of freedom to the formulation. Then, finding the inverse solution with interval arguments requires a set-based computational method. In order to address this problem, next we will introduce a set inversion algorithm that is based on interval analysis.

$$\begin{aligned} \theta_2 &= \pm \cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1 l_2}\right) \\ \theta_1 &= \mathrm{atan2}(y, x) \quad \cos^{-1}\left(\frac{x^2+y^2+l_1^2-l_2^2}{2l_1 \sqrt{x^2+y^2}}\right) \end{aligned} \tag{8}$$

Set Inversion via Interval Analysis

Given a nonlinear function f from R^n to R^m and a set Y in R^m, finding X described as

$$X = \{x \in R^n \mid f(x) \in Y\} = f^{-1}(Y) \tag{9}$$

is the set inversion problem [19, p. 55]. Jaulin et al. addressed this problem via an algorithm called SIVIA (Set Inversion Via Interval Analysis) [21]. For a given $Y \in R^m$, X can be bounded arbitrarily closely with a lower bound $\underline{X}$ and an upper bound $\overline{X}$ such that $\underline{X} \subset X \subset \overline{X}$ provided that an inclusion function $[f]$ can be found for f. Note that $\underline{X}$ is guaranteed to be a solution set whereas $\overline{X}$ may contain non-solution points.

The process of finding X using SIVIA is illustrated in Fig. 2 for 2-dimensional x and y spaces. SIVIA starts with an initial search domain $[x_0]$ that is guaranteed to contain $\overline{X}$. Then, the following 4-step procedure is applied:

If the mapping $[f]([x])$ results in an interval (box) in the y-space that intersects with Y without being fully enclosed by Y as in Fig. 2(a), then $[x]$ is said to contain part of the solution set X but regarded as undetermined. If the width of $[x]$ is greater than a predetermined resolution parameter ϵ, then it needs to be bisected along the longest side and the procedure needs to be repeated recursively on each sub-box. If $[f]([x]) \cap Y = \emptyset$ as in Fig. 2(b), then $[x]$ is not part of X hence can be discarded.
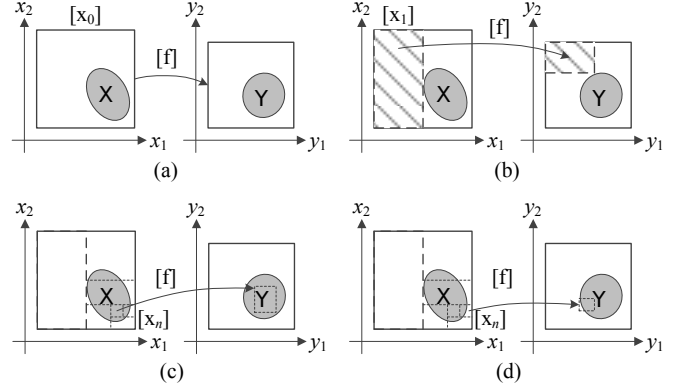If $[f]([x]) \subset Y$ as in Fig. 2(c), then $[x] \subset X$ and $[x] \in \overline{X}$ and $[x] \in \underline{X}$.

FIGURE 2: SIVIA ALGORITHM PROCEDURE (BASED ON [19, p. 57]).

Finally, if $[x]$ is undetermined and width$([x]) < \epsilon$ as in Fig. 2(d), then the procedure stops for $[x]$ and it is added to the upper bound $\overline{X}$ of X.

General Case of Inverse Kinematics via Set Inversion

Let f be the forward kinematics map of a serial manipulator from R^n to $SE(3)$ where n is the total number of joint variables and parameters that are uncertain. Also, let Y be a subset of $SE(3)$. Then, calculating X in (9) is the inverse kinematics problem in presence of joint parameter uncertainties for a given set of end-effector configurations. This is inherently a set inversion problem and can be addressed using SIVIA.

Application of SIVIA to a general inverse kinematics problem requires comparing $[f]([x])$ in (3) with Y and determining whether or not they intersect or one includes the other. This is relatively simpler if Y can be represented as an interval so that both $[f]([x])$ and Y are 4×4 interval matrices. Then, the comparison can be done element-wise. Otherwise, each member of Y has to be compared with $[f]([x])$ one by one.

SIMULATIONS

In this section, we present some example simulation results that show how the proposed method can be implemented. We carried out interval calculations using MATLAB and a toolbox called INTLAB developed by S. M. Rump [22]. INTLAB supports many operations with real and complex interval scalars, vectors, and matrices. It provides efficient functions for basic operations in algebra, trigonometry, etc.

In order to verify the results obtained using interval analysis, we also performed Monte Carlo analysis. When an analytical expression for inverse kinematics is available, it can directly be used to go from the configuration space to the parameter space. For instance, by evaluating equation (8) for various samples of

4

TABLE 1: TWO-LINK MANIPULATOR SIMULATION PARAMETERS.

l_1	δl_1	l_2	δl_2	$[\theta_1(0)]$	$[\theta_2(0)]$
1	0.001	1	0.001	$[0, \pi/2]$	$[-\pi/2, \pi/2]$

the input arguments $[l_1]$, $[l_2]$, $[x]$, and $[y]$, one can obtain a set of points in the joint space whose convex hull approximately provides the inverse solution X. We refer to this as the Monte Carlo solution in the next part.

Precision Design of the Two-Link Manipulator

In this part, we present the solution to the inverse kinematics problem of the two-link manipulator discussed in . This serves as a validation example such that we verify the interval analysis results with Monte Carlo simulation of inverse kinematic equations. The results will enable us to determine the minimum joint encoder resolution required to achieve a given end-effector precision.

For the manipulator parameters given in Table 1 and for a sample end-effector position $([x], [y]) = (1.4 \pm 0.01, 1.2 \pm 0.01)$, the solutions of the SIVIA algorithm for a stopping criterion of $\varepsilon = \pi/1800$ is shown in Fig. 3. The picture on the left shows the 7018 subpavings (i.e. bisected arguments) of θ_1 and θ_2 each of which represent an interval sample from the joint space. SIVIA produces increasing concentration around the two solution regions as it converges by bisecting initial 'undetermined' intervals. The exploded view of one of those regions on the right shows that the remaining undetermined region $\overline{X}$ (yellow) found by SIVIA properly encloses the inverse Monte Carlo solution X (blue line) which in turn encloses the lower bound $\underline{X}$ (green region) as suggested by SIVIA.

The result in Fig. 3 shows that if the joint positions can be addressed as precisely as to fit in the lower bound (green region, $\underline{X}$), then it can be guaranteed that the end-effector can be positioned in the given interval $([x], [y]) = (1.4 \pm 0.01, 1.2 \pm 0.01)$. The question is then how to quantify the precision based on this result.

Fig. 4(a) shows $\underline{X}$ in more detail where an interval of (θ_1, θ_2) with the largest overlapping area with $\underline{X}$ is encircled. If the length of this interval along θ_1 is A and that along θ_2 is B, then the proper design choice for the joint resolution of the manipulator about these two axes can be described as in equation (10). This makes sure that the actual joint positions $[\theta_1] \quad [\theta_1 - \delta\theta_1, \theta_1 + \delta\theta_1]$ and $[\theta_2] \quad [\theta_2 - \delta\theta_2, \theta_2 + \delta\theta_2]$ can be commanded to be inside $\underline{X}$ for a certain value of θ_1 and θ_2. This is explained pictorially in Fig. 4(b) with an arbitrarily positioned grid of addressable intervals each of which represents the
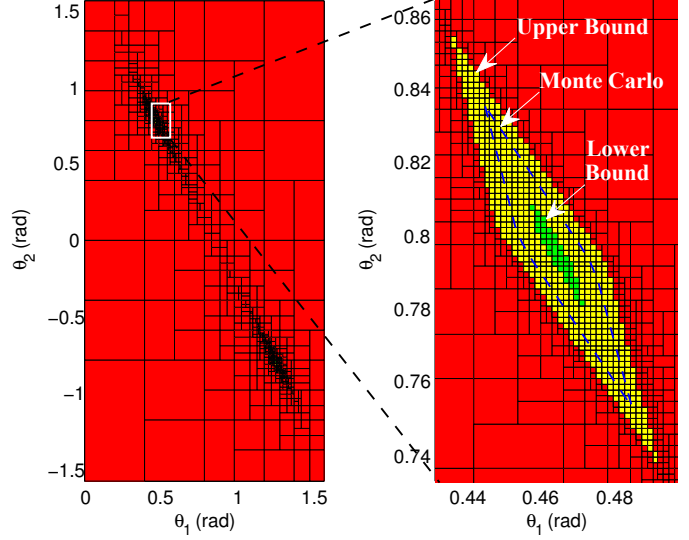

FIGURE 3: BOUNDS ON θ_1 AND θ_2 OF THE TWO-LINK MANIPULATOR FOR $([x], [y]) = (1.4 \pm 0.01, 1.2 \pm 0.01)$.

set of actual joint positions for a given position command. When the condition in (10) is satisfied, there exists at least one interval of $([\theta_1], [\theta_2])$ that completely overlaps with $\underline{X}$. In Fig. 4(b), there are five such possible joint positions shown by boxes with the slash pattern. Therefore, $\delta\theta_1^*$ and $\delta\theta_2^*$ are the coarsest resolution values for the joint encoders that guarantee the given end-effector precision. In this case, (A, B) is measured to be $(\pi/900, \pi/450)$ which correspond to $(\delta\theta_1^*, \delta\theta_2^*) = (0.05°, 0.1°)$. For this particular end-effector position, it can be seen that the resolution requirement for the first joint is higher as it can introduce more Abbe error due to its distance from the end-effector.

$$\delta\theta_1 \leq \delta\theta_1^* \quad A/4 \quad , \quad \delta\theta_2 \leq \delta\theta_2^* \quad B/4 \qquad (10)$$

The lower bound $\underline{X}$ can always be improved in expense of computational time by reducing the value of ε. Fig. 5(a) compares the previous result in Fig. (3) where ε was $\pi/1800$ with the one in Fig. 5(b) where ε is $\pi/3600$. It can be seen that a finer bisection resolution improves the lower bound $\underline{X}$ towards the Monte Carlo result. The amount of improvement in this case is from $(A, B) = (\pi/900, \pi/450)$ to $(A, B) = (7\pi/3600, \pi/400)$ which means that the $\delta\theta_1^*$ and $\delta\theta_2^*$ are now $0.0875°$ and $0.1125°$, respectively. The total number of subpavings processed in this case is 57590 which is significantly higher than the previous value 7018. Indeed, SIVIA terminates after generating less than $(\text{width}([x_0])/\varepsilon + 1)^n$ bisections while the computing time increases exponentially with the dimension of $[x]$ [23] which is propor-
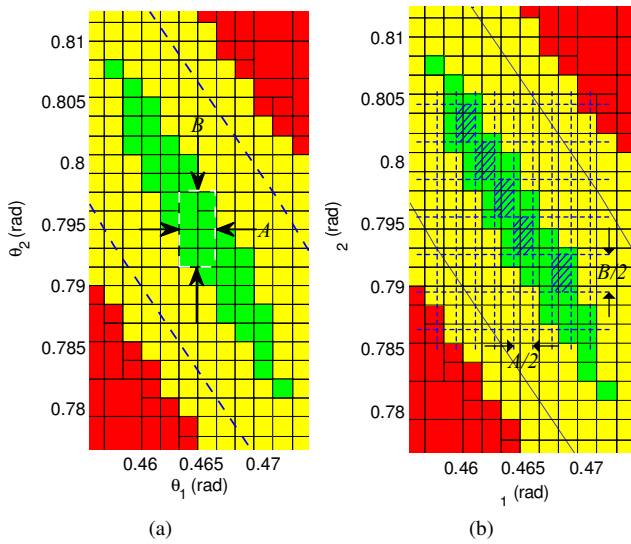
FIGURE 4: (a) THE INTERVAL OVERLAPPING WITH THE LOWER BOUND $\underline{X}$ WITH MAXIMUM AREA OF A × B (b) AN ARBITRARILY POSITIONED GRID OF ADDRESS-ABLE INTERVALS FOR $\delta\theta_1 = A/4$ AND $\delta\theta_2 = B/4$.

FIGURE 5: UPPER ($\overline{X}$) AND LOWER ($\underline{X}$) BOUNDS FOUND USING SIVIA AND (X) FOUND USING MONTE CARLO FOR (a) $\varepsilon = \pi/1800$ (b) $\varepsilon = \pi/3600$.

tional to the number of degrees of freedom of a manipulator. However, using bisections is one of the most basic techniques of interval analysis in terms of computational efficiency. There are efficient solvers that combine use of bisections, contractors, inclusion tests, and local optimization procedures to reduce the computational complexity to polynomial order [19]. Application of these solvers to the inverse kinematics problem is not covered in this paper but will be addressed in the future.

Allocation of Mechanism Tolerances in a 3-DOF Precision Stage

In this part, we will demonstrate how the presented method can also be used to allocate tolerances to the mechanical design of a 3-DOF PPR manipulator as in Fig. 6 such that a given end-effector precision can be achieved. For clarity, we will focus only on some of the parametric uncertainties of the manipulator such as prismatic joint axes and rotary joint position vectors. The presented method will enable us to bound the allowable misalignment in these vectors. The vectors $[v_x]$, $[v_y]$, and $[q]$ that parametrize the manipulator are represented as shown in (11) with error bounds denoted by $[\delta v_y]$ for the misalignment of $[v_x]$ along y-axis, $[\delta v_x]$ for the misalignment of $[v_y]$ along x-axis, and $[\delta q]$ for the misalignment of $[q]$ along x and y axes. In a practical scenario, for instance, these terms can represent the errors introduced into the geometry of the manipulator during its manufacturing or assembly.
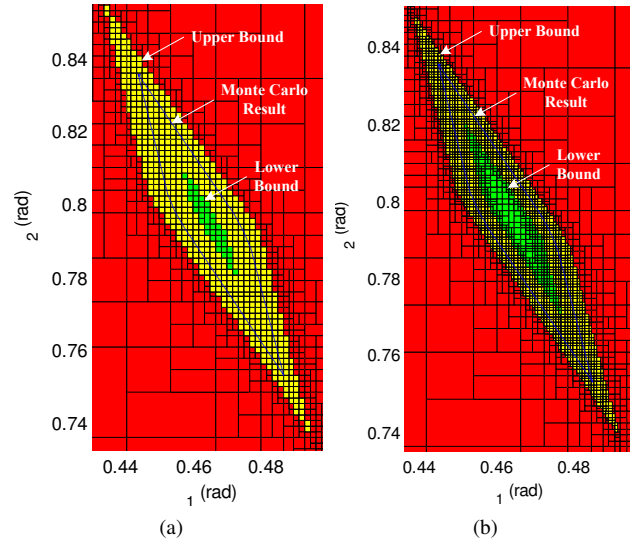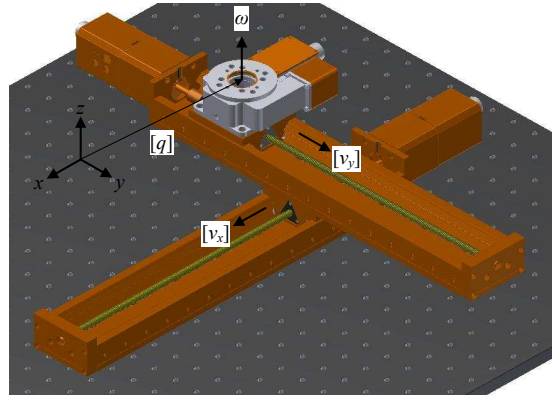

FIGURE 6: 3D MODEL OF A 3-DOF PPR PRECISION MOTION STAGE.

$$[v_x] = \begin{pmatrix} 1 \\ [\delta v_y] \\ 0 \end{pmatrix} \quad [v_y] = \begin{pmatrix} [\delta v_x] \\ 1 \\ 0 \end{pmatrix} \quad [q] = \begin{pmatrix} [\delta q] \\ [\delta q] \\ 0 \end{pmatrix} \quad (11)$$

The objective in this simulation is to find how large $[\delta v_y]$, $[\delta v_x]$, $[\delta q_x]$, and $[\delta q_y]$ can be for a given stage positioning precision. For example, assume a configuration attached to the rotary stage needs to be driven from an initial pose p_i to a final pose p_f
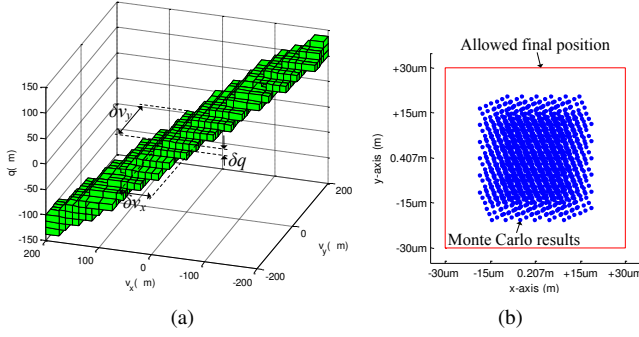
6

FIGURE 7: (a) BOUNDING OF ERROR TERMS $[\delta v_y]$, $[\delta v_x]$, and $[\delta q]$ (b) END-EFFECTOR POSITION BY MONTE CARLO SAMPLING OF (13).

through the manipulator transformation that can be described as

$$[p_f] = \begin{bmatrix} I & d_x[v_x] & I & d_y[v_y] & e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})[q] \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} p_i \quad (12)$$

where d_x, d_y, and θ are the desired displacements of the axes. We chose the initial end-effector position to be $(10mm, 0mm, 0°)$ and final joint position to be $(200mm, 400mm, 45°)$. Then, we solved for $[\delta v_y]$, $[\delta v_x]$, and $[\delta q]$ that guarantee a maximum deviation of $30\mu m$ from the desired final position. Fig. 7(a) shows the lower bound solution of this simulation. Similar to the previous case, the largest interval box that can be fit into the lower bound has approximate dimensions of $(60,140,13)$ μm. Then from (10), the allowable maximum uncertainty in the axes vectors can be expressed as

$$\begin{aligned} [\delta v_x] &= [-15, 15]\mu m \\ [\delta v_y] &= [-35, 35]\mu m \quad . \\ [\delta q] &= [-3.25, 2.25]\mu m \end{aligned} \quad (13)$$

Fig. 7(b) shows the distribution of end-effector position by sampling the intervals in (13) using Monte Carlo method. It can be seen that the results all lie inside the allowed boundary for final end-effector position hence (13) is a guaranteed tolerance allocation. It can also be seen in (13) that the amount of allowable uncertainty is different for different parameters. This depends on the given initial and final positions, desired stage positioning precision, and the kinematic design. Hence, the result can also be used to do sensitivity analysis to find out which parameter is likely to cause more deviation at the end-effector.

CONCLUSIONS

In this paper we propose a methodology for solving manipulator design problems with precision bounds. The simulation results show that the proposed method is effective in determining the level of precision needed in the design of a manipulator for a given end-effector precision. Solution of the inverse kinematics problem using interval analysis not only provides the range of joint space variables for a desired end-effector position but also allows calculation of the maximum allowable uncertainty in the joint sensor (encoder) feedback and the mechanism geometry tolerances. Therefore, the method can be used as a design aid in robotic applications that involve designing, building, or choosing individual axes of a multi-degree-of-freedom manipulator. Since we use standard interval analysis tools such as SIVIA algorithm, our approach can easily be applied to any kinematic configuration. Future work will include using heuristics to speed up the computational efficiency of the algorithm and exercising this method on the design of a modular micro-assembly robot.

REFERENCES

[1] Benhabib, B., Fenton, R., and Goldenberg, A., 1987. "Computer-aided joint error analysis of robots". IEEE Journal of Robotics and Automation, 3(4), Aug., pp. 317–322.

[2] Lin, P., and Ehmann, K., 1993. "Direct volumetric error evaluation for multi-axis machines". International Journal of Machine Tools and Manufacture, 33(5), Oct., pp. 675–693.

[3] Shin, Y., and Wei, Y., 1992. "A statistical analysis of positional errors of a multiaxis machine tool". Precision Engineering, 14(3), July, pp. 139–146.

[4] Soons, J., Theuws, F., and Schellekens, P., 1992. "Modeling the errors of multi-axis machines: a general methodology". Precision Engineering, 14(1), Jan., pp. 5–19.

[5] Vaishnav, R. N., and Magrab, E. B., 1987. "A general procedure to evaluate robot positioning errors". International Journal of Robotics Research, 6(59), Mar., pp. 59–74.

[6] Veitschegger, W., and Wu, C.-H., 1986. "Robot accuracy analysis based on kinematics". IEEE Journal of Robotics and Automation, 2(3), Sept., pp. 171–179.

[7] Slocum, A. H., 1992. Precision Machine Design. Society of Manufacturing Engineers, Dearborn, MI, USA.

[8] Frey, D. D., 2004. Robotics and Automation Handbook. CRC Press, ch. 10.

[9] Das, A. N., and Popa, D. O., 2011. "Precision evaluation of modular multiscale robots for peg-in-hole microassembly tasks". In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1699–1704.

[10] Moore, R. E., Kearfott, R. B., and Cloud, M. J., 2009. Introduction to Interval Analysis. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[11] Wu, W., and Rao, S. S., 2007. "Uncertainty analysis and allocation of joint tolerances in robot manipulators based on interval analysis". Reliability Eng. and System Safety, 92, pp. 54–64.

[12] Merlet, J. P., 2004. "Solving the forward kinematics of a goughtype parallel manipulator with interval analysis". Int. J. of Robotics Research, 23(3), p. 221236.

[13] Merlet, J. P., 2009. "Interval analysis for certified numerical solution of problems in robotics". International Journal of Applied Mathematics and Computer Science, 19(3), pp. 399–412.

[14] Pac, M., and Popa, D., 2012. "Interval analysis for robot precision evaluation". In Proc. 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 1087–1092.

[15] Dechev, N., Cleghorn, W. L., and Mills, J. K., 2004. "Microassembly of 3-d microstructures using a compliant, passive microgripper". Journal of Microelectromechanical Systems, 13(2), Apr., pp. 176–189.

[16] Probst, M., C. Hrzeler, R. B., and Nelson, B. J., 2009. "A microassembly system for the flexible assembly of hybrid robotic mems devices". International Journal of Optomechatronics, 3(2), pp. 69–90.

[17] Das, A. N., Murthy, R., Popa, D. O., and Stephanou, H. E., 2012. "A multiscale assembly and packaging system for manufacturing of complex micro-nano devices". IEEE Transactions on Automation Science and Engineering, 9(1), pp. 160–170.

[18] Murray, R. M., Li, Z., and Sastry, S. S., 1994. A Mathematical Introduction to Robotic Manipulation. CRC Press.

[19] Jaulin, L., Kieffer, M., Ditrit, O., and Walter, E., 2001. Applied Interval Analysis. Springer-Verlag, London, Great Britain.

[20] Craig, J. J., 2005. Introduction to Robotics: Mechanics and Control, third ed. Pearson Prentice Hall, NJ, USA.

[21] Jaulin, L., and Walter, E., 1993. "Set inversion via interval analysis for nonlinear bounded-error estimation". Automatica, 29(4), Nov., pp. 1053–1064.

[22] Rump, S. M., 1999. "Intlab interval laboratory". In Developments in Realiable Computing, T. Csendes, ed. Kluwer Academic Publishers, Dordrecht, pp. 77–104.

[23] Jaulin, L., and Walter, E., 1993. "Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis". Mathematics and Computers in Simulation, 35(2), Apr., pp. 123–137.