Proceedings of the ASME 2019 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/MR 2019
August 18-21, 2019, Anaheim, CA, USA

DETC2019/MR-97945

DRAFT: COMPUTING ROBUST INVERSE KINEMATICS UNDER UNCERTAINTY

Anirban Sinha
Department of Mechanical Engineering
Stony Brook University, New York, USA
Email: anirban.sinha@stonybrook.edu

Nilanjan Chakraborty
Department of Mechanical Engineering
Stony Brook University, New York, USA
Email: nilanjan.chakraborty@stonybrook.edu

ABSTRACT

Robotic tasks, like reaching a pre-grasp configuration, are specified in the end effector space or task space, whereas, robot motion is controlled in joint space. Because of inherent actuation errors in joint space, robots cannot achieve desired configurations in task space exactly. Furthermore, for redundant manipulators, mapping of joint space error set to that of task space is a map of one to many i.e., for the same reaching task, different inverse kinematics solutions map joint space error set to task space differently. In this paper we present method to choose inverse kinematics solution for redundant manipulators with actuation uncertainties (errors) for a specified task that will have least task space error. We call this particular IK as robust-IK. To obtain such IK we propose two key steps, at first obtain all possible task space errors as bounded sets for different IKs and then choose one particular IK which has minimum task space error bound with respect to the given task. We devise set theoretic approach to bound all possible joint space errors and obtain task space error bound by mapping it through linearized position and rotation kinematics of the manipulator. Finally we present applications of robust-IK algorithm for fundamental manipulation tasks with simulated and real 7-DoF redundant manipulator.

1 INTRODUCTION

A fundamental problem in many robotics tasks is to move the end effector of a manipulator to a desired configuration (position and orientation). For example, in grasping, the end effector is usually moved to a pre-grasp configuration before closing the fingers. In assembly operations like peg-in-a-hole operations, the end effector holding a part is usually moved to a pre-insertion configuration from where the insertion of the peg in the hole is attempted. One key factor that dictates the success of operations like grasping or the peg-in-a-hole insertion is the accuracy of the placement of the end effector in the pre-grasp or pre-insertion configuration. Because of the inherent inaccuracy in actuation, it is usually not possible to place the end effector at the desired configuration exactly. Thus, success of operation depends on the amount of inaccuracy that can be tolerated for the given task and whether robot can place its end effector within that error margin. Furthermore, given a particular accuracy requirement for a given task, a question of interest is for the robot to evaluate its capability on whether it can place its end effector so that it is highly likely that the robot can perform the subsequent operations successfully. The *goal of this paper is to develop algorithms for the robot to select its joint angles such that in spite of actuation and/or sensing errors, the robot can determine if it can place its end effector within the prescribed error margin and also compute the solution that is robust to the actuation errors.*

Computing the joint angles given end effector (tool) configuration is known as inverse kinematics (IK) problem which can be formally defined as follows: *Given a desired position and orientation of the end effector of the manipulator, compute the corresponding joint angles.* It is well known that, in general, the IK problem has multiple solutions, and for redundant robots, the IK problem has infinitely many solutions. All the solutions are equivalent in a sense that end effector goes to the same configuration if each joints can be rotated exactly as desired. However, if the joints have errors, the effect of the joint error in the task space is not identical. For example consider the $3R$ manipulator

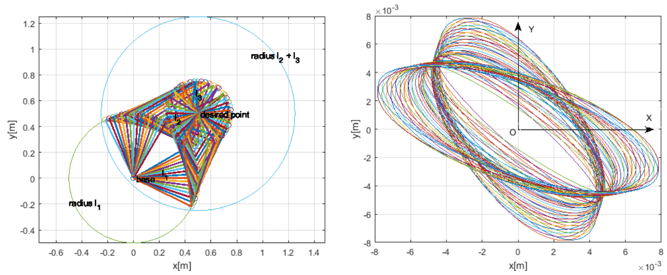

Figure 1: (LEFT) MULTIPLE IK SOLUTIONS OF 3R ROBOT FOR A REACHING TASK. (RIGHT) TASK SPACE ERROR SETS CORRESPONDING TO IKS IN LEFT FIGURE W.R.T. END-EFFECTOR FRAME, OBTAINED BY PROPAGATING JOINT SPACE ERROR SET. ALL ERROR SETS ARE CENTERED AROUND ***O*** (DESIRED POSITION) BUT WITH DIFFERENT SIZE AND ORIENTATIONS BECAUSE SAME ERROR SET IN JOINT SPACE MAPPED TO DIFFERENT ERROR SETS IN TASK SPACE FOR DIFFERENT IK(S)

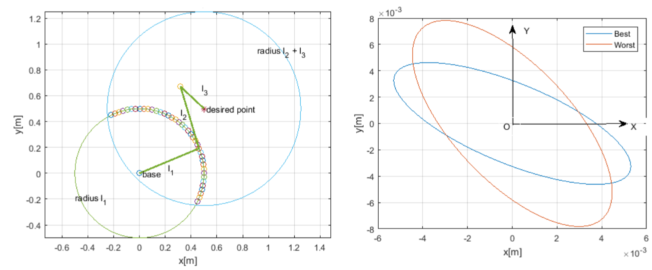

Figure 2: THE TASK SPACE ERROR SETS CORRESPONDING TO ROBUST (BLUE) AND WORST (RED) JOINT CONFIGURATIONS FOR A TSAK THAT REQUIRES MINIMIZED ERROR ALONG Y AXIS. BLUE ELLIPSE HAS SMALLER PROJECTION LENGTH AS COMPARED TO THE PROJECTION OF RED ELLIPSE ON TO Y AXIS

in Fig1, where the goal is to move the end effector to a given position in the xy-plane. Since there are only two variables in the task space (namely, (x, y) coordinates of the end effector), and the $3R$ robot has 3 degrees-of-freedom (DoF), the robot is redundant with respect to the positioning task. Hence number of IK solutions is infinity and some of them are shown in Fig1. If we bound joint space error using a n dimensional ball, then by propagation of error in task space, we obtain an error ellipsoid. A similar kind of situation with a 3R robot is shown in Fig1 (right image) where task space error ellipsoids for different IKs are plotted by propagating joint space error set modelled as a 3 dimensional ball. As is evident from the Fig1, that task space error sets for different IKs are different. Therefore if the positioning task demands minimum error in x-direction, then IK solution corresponding to the red error-ellipsoid (error set) is better than the IK solution corresponding to blue error-ellipsoid of Fig2. However if criterion for the positioning task of the end effector is to have minimized error along y-direction, then IK solution corresponding to the blue error-ellipsoid is better.

The above planar example illustrates the need for developing methods to analyze the propagation of the joint space errors and find IK solutions that are *robust* to the actuation errors. Although IK is a classical problem in robotics and there has been recent work on propagation of joint space errors to task space of robotic manipulators in [1,2], to the best of our knowledge, there is no work on finding IK solutions that are robust to actuation errors. As illustrated in the planar example, algorithms to compute robust IK solutions can enable a robot to determine whether it can successfully accomplish a task based on given error margin.

The same example as in figures 1 and 2 also answers the question of feasibility of a task in a sense that given joint error parameters and robot kinematics model if the robot can at all position its end effector within a tolerable error margin or not. On the other hand from a designer perspective a question of interest would be, what are the joint error parameters given robot kinematic model so that robot will be able to place its end-effector within a specified error margin.

We used geometry (i.e., n dimensional ball) to bound joint space uncertainty, which can be used to have both set-theoretic and probabilistic interpretation of uncertainty models. In order to map joint space error to task space we use first order approximation of the nonlinear forward kinematics function for position and orientation independently. This helped us to obtain position and orientation error set in task space separately which in turn is useful to perform optimization on these two sets independently based on a suitably defined error measure depending on task at hand.

Contributions of this paper

Here we clearly state our contributions. Firstly, we propagate joint space error set to task space separately to rotation space ($\subset SO(3)$) and position space ($\subset \mathbb{R}^3$) utilizing linearized forward kinematics mapping for position and orientation respectively. This helped to obtain two separate error sets (position and orientation) over which we perform optimization to find robust IK depending on tasks. We do not claim that task space error sets obtain this way will bound all possible errors of position and orientation for a given task but using simulation results and with experiments we showed that the error sets can bound more than 98.5% of all possible errors. In other words we compute approximate error sets in task space which is fast and easy to implement. Unlike [3,4] our method propagates entire joint space error bound in to task space error bound without computing covariances at intermediate joints. Secondly, while computing rotation error set we have built-up on the idea of computing *error quaternion* on $SO(3)$ described in [5], to find rotation error bound of n-DoF manipulator. Thirdly, our research also proposes task based error minimization algorithm for $\mathbb{R}^3$, $SO(3)$ and both. Our research should not be confused with estimating errors in task space, rather we are interested in finding a robust IK solution instead, that results in minimum error bound in task space after propagating error from joint to task space.

2 RELATED WORK

A group-theoretic approach in dealing with robot kinematics is introduced to robotics community by [6–8]. Precise positioning of robot arm suffers from uncertainties in actuation, sensing, robot model and control algorithms. Representation of spatial uncertainty in the context of robotics can be found in the classic work by [9, 10] and recently in [2]. Authors in [5] presented a method to map rotation errors from roll-pitch-yaw to quaternion representation of rotations. However irrespective of the source of uncertainty, we can broadly categorize uncertainty in to two basic classes. Errors are in link-lengths, offset lengths and/or origin of the joints that are not known precisely, constant biases of sensor/actuator output etc which are often denoted as *static errors*. They do not change over time and hence can be estimated offline and compensated during calibration process of the robot. Works on estimating these parameters using end effector pose error between kinematics model and actual pose can be found in [11–17]. In order to accurately collecting end-effector pose data to estimate error parameters, [18] used an external laser tracker. Static pose error minimization for calibration using partial pose information can be found in [19]. The second kind of the uncertainty corresponds to random actuation and sensing errors realized in execution time of the robot. They implicitly affect accuracy in joint rotations which in turn affects accuracy at the end-effector of a manipulator. Error association in $SE(3)$ by fusing multiple sensor data can be found in [2, 9]. This second kind of error source is behind the motivation of our work. A group theoretic approach to propagate joint space random actuation error into end-effector space has been presented in [3]. Authors in [3] present a method to obtain error covariance at the end of each individual link in closed form due to errors in desired joint configurations. By repeating this procedure sequentially for each link of a manipulator they obtain final error covariance at the end effector. To capture the effect of large joint errors on error covariance authors in [4] presented a second order theory of error propagation. In essence their method relies on evaluating desired and erroneous poses at each joints for a few discrete samples of joint errors so that individual covariances at each joint can be calculated and finally combine them in sequential manner using proposed closed form covariance propagation formula to obtain final error covariance at the last distal frame. Unlike [3, 4] our method does not rely on individual error samples and corresponding frame by frame error covariance computation. Assuming small joint errors along with linearized model of forward position and rotation kinematics our method can propagate whole joint space error set into task space error set. Obviously [3, 4] are more effective if joint space errors are large and no prior knowledge about joint error bounds is available.

3 MATHEMATICAL PRELIMINARIES

In this section, we present notations and definitions that will be used throughout the paper. Let $\mathbb{R}^n$ be the real Euclidean space of dimension n, $\mathbb{R}^{m\times n}$ be the set of all $m\times n$ matrices with real entries. The set of all joint angles, $\mathcal{J}$, is called the *joint space* or the configuration space of the robot. In this paper $\mathcal{J}\subset\mathbb{R}^7$, since we are using a 7 Degree-of-Freedom (7-DoF) manipulator with joint limits.

Let $SO(3)$ be the Special Orthogonal group of dimension 3, which is the space of all rigid body rotations. Let $SE(3)$ be the Special Euclidean group of dimension 3, which is the space of rigid motions (i.e., rotations and translations). $SO(3)$ and $SE(3)$ are defined as follows [6]: $SO(3)=\{\mathbf{R}\subset\mathbb{R}^{3\times 3}|\mathbf{R}^T\mathbf{R}=\mathbf{R}\mathbf{R}^T=\mathbf{I},|\mathbf{R}|=1\}$, $SE(3)=SO(3)\times\mathbb{R}^3=\{(\mathbf{R},\mathbf{p})|\mathbf{R}\in SO(3),\mathbf{p}\in\mathbb{R}^3\}$ where $|\mathbf{R}|$ is the determinant of $\mathbf{R}$ and $\mathbf{I}$ is a 3×3 identity matrix. The set of all end effector configurations is called the *end effector space* or *task space* of the robot and is a subset of $SE(3)$.

Unit Quaternion Representation of $SO(3)$: Unit quaternions are a singularity free representation of $SO(3)$. A quaternion is a tuple $\boldsymbol{q}=(\eta,\varepsilon_x,\varepsilon_y,\varepsilon_z)$ which includes a vector $\boldsymbol{\varepsilon}\in\mathbb{R}^3$ with components ε_x, ε_y, ε_z and a scalar η. For a unit quaternion $\|\mathbf{q}\|=1$. In our paper we extensively make use of vector representation of unit quaternions, $\boldsymbol{q}=[\eta\quad\boldsymbol{\varepsilon}^T]^T$ with its conjugate $\boldsymbol{q}^{-1}=[\eta\quad-\boldsymbol{\varepsilon}^T]^T$. Rotation about an axis $\boldsymbol{\omega}$ with angle ϕ is a unit quaternion represented as, $\boldsymbol{q}(\boldsymbol{\omega},\phi)=[\cos\phi/2\quad\boldsymbol{\omega}\sin\phi/2]$. Let $\boldsymbol{\varepsilon}^\times\in\mathbb{R}^{3\times 3}$ denote *skew symmetric matrix* of vector $\boldsymbol{\varepsilon}\in\mathbb{R}^3$.

4 PROBLEM FORMULATION

In this section we formalize our problem of *obtaining robust IK corresponding to minimum task space error bound in the presence of uncertain joint space errors.* Before moving forward, we define two terms that will be using frequently as we move along. By *error bound* mean maximum possible error in position or orientation one may encounter most of the time (with very high probability) due to actuation or sensing error in joint space. By *robust IK* we mean the particular joint solution vector ($\boldsymbol{\Theta}\in\mathbb{R}^n$) that corresponds to minimum error bound in task space.

Uncertainty Modeling in Joint Space:

We use a geometric model of the uncertainty in the joint space. More specifically, we model the joint space error as a ball in $\mathbb{R}^n$ centered at the desired joint angles. The ball can be interpreted as either a set-theoretic model of uncertainty or a probabilistic model of uncertainty. Depending on the interpretation of the uncertainty model, the results can be given a worst case interpretation or a probabilistic interpretation.

In the set-theoretic model, the ball represents the 2-norm of the joint errors, and the assumption is that the joint errors always lie within this set and we have no other knowledge that can be used to define a probability measure on this set.

In the probabilistic model, the ball represents the probability mass that the uncertain configuration is within the ball assuming a Gaussian probability measure. The radius of the ball corresponds to the confidence level with which we want to know whether the uncertain configuration will lie within the ball. Thus, the actuation uncertainty, $\boldsymbol{\delta\Theta}$, is a multivariate Gaussian with zero mean and known $n \times n$ covariance matrix ($\boldsymbol{\Sigma}$), i.e., $\boldsymbol{\delta\Theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. The ball model implicitly implies that the covariance matrix is diagonal or the error in the joints are uncorrelated and the variance of all the joints are identical, i.e., $\boldsymbol{\Sigma} = \sigma^2 \mathbb{I}$, where σ is the standard deviation and $\mathbb{I}$ is the $n \times n$ identity matrix. Thus, the uncertainty set in the joint space is

$$\boldsymbol{\delta\Theta}^T \boldsymbol{\delta\Theta} \leq c \quad \text{where} \quad c = (k\sigma)^2 \tag{1}$$

considering error up to k standard deviations about zero mean.

Remark 1. *Please note that the ball model is for ease of exposition only. We could have used a more general ellipsoidal model of the uncertainty, with the covariance matrix non-diagonal. The problem formulation and the solution techniques applies in this case also, although the expressions for the propagated uncertainty set becomes more complicated.*

We assume that the joint actuation errors are sufficiently small such that a linear approximation of the forward kinematics function about the desired position and orientation can be used to propagate joint space error. To propagate joint space error (as in equation (1)) into task space, we need to have the mapping from $\boldsymbol{\delta\Theta} \in \mathbb{R}^n$ to $\boldsymbol{\delta T} \subset SE(3)$ where $\boldsymbol{\delta T} = (\boldsymbol{\delta X}, \boldsymbol{\delta q})$, $\boldsymbol{\delta X} \in \mathbb{R}^3$, $\boldsymbol{\delta q} \in SO(3)$. To find expression of $\boldsymbol{\delta q}$ in terms of $\boldsymbol{\delta\Theta}$, we use first order Taylor's series approximation of *forward rotation kinematics function* $\boldsymbol{q}(\boldsymbol{\Theta}) : \mathbb{R}^N \to SO(3)$. Similarly we derive the expression for position error term i.e., $\boldsymbol{\delta X} \in \mathbb{R}^3$ by linearizing *forward position kinematics function* $\boldsymbol{F}(\boldsymbol{\Theta}) : \mathbb{R}^N \to \mathbb{R}^3$. Since there is no bi-invariant metric available [20] to quantify error in $SE(3)$, we optimize two metrics defined over $\mathbb{R}^3$ and $SO(3)$ independently. To compute *robust IK* we optimize an objective obtained by weighted sum of error metrics defined over $\mathbb{R}^3$ and $SO(3)$.

4.1 Error Propagation from joint space to $\mathbb{R}^3$

Suppose $\bar{\boldsymbol{\Theta}} \in \mathbb{R}^n$ denotes the nominal joint angle vector of a n DoF manipulator that takes the end-effector to desired position $\in \mathbb{R}^3$. Due to joint angle errors $\boldsymbol{\delta\Theta} \in \mathbb{R}^n$, actual position of end effector will be $\boldsymbol{F}\left(\bar{\boldsymbol{\Theta}} + \boldsymbol{\delta\Theta}\right)$ instead of desired position $\boldsymbol{F}\left(\bar{\boldsymbol{\Theta}}\right)$. The Taylor's series approximation of $\boldsymbol{F}(\bar{\boldsymbol{\Theta}} + \boldsymbol{\delta\Theta})$ gives,

$$\boldsymbol{F}\left(\bar{\boldsymbol{\Theta}} + \boldsymbol{\delta\Theta}\right) = \boldsymbol{F}\left(\bar{\boldsymbol{\Theta}}\right) + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\Theta}}|_{\bar{\boldsymbol{\Theta}}} \boldsymbol{\delta\Theta} + O\left(\boldsymbol{\delta\Theta}^2\right) \tag{2}$$

The partial derivative $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\Theta}}|_{\bar{\boldsymbol{\Theta}}} \in \mathbb{R}^{3\times n}$ corresponds to the first three rows of *manipulator Jacobian* [6, 21]. Denoting $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\Theta}}|_{\bar{\boldsymbol{\Theta}}}$ by $\boldsymbol{J}_p\left(\bar{\boldsymbol{\Theta}}\right) \in \mathbb{R}^{3\times n}$ (*position Jacobian*), *position error vector* is

$$\boldsymbol{\delta X} = \boldsymbol{F}(\bar{\boldsymbol{\Theta}} + \boldsymbol{\delta\Theta}) - \boldsymbol{F}(\bar{\Theta}) \approx \boldsymbol{J}_p \boldsymbol{\delta\Theta} \tag{3}$$

where $\boldsymbol{J}_p \in \mathbb{R}^{3\times n}$, $n > 6$. Therefore,

$$\boldsymbol{\delta\Theta} = \boldsymbol{J}_p^T \left[\boldsymbol{J}_p \boldsymbol{J}_p^T\right]^{-1} \boldsymbol{\delta X} \tag{4}$$

Substituting (4) into (1) and simplifying, we obtain

$$\boldsymbol{\delta X}^T \left[\boldsymbol{J}_p \boldsymbol{J}_p^T\right]^{-1} \boldsymbol{\delta X} \leq c \tag{5}$$

Equation (5) describes the set of position errors of the end effector due to the error in the joint space. Geometrically, the set of position errors is an ellipsoid.

4.2 Error Propagation from Joint Space to $SO(3)$

As before, let $\bar{\boldsymbol{\Theta}} \in \mathbb{R}^n$ denote nominal joint angle vector that takes end-effector to desired orientation $\boldsymbol{q}\left(\bar{\boldsymbol{\Theta}}\right)$. Because of random joint error $\boldsymbol{\delta\Theta}$, actual orientation of end effector becomes $\boldsymbol{q}\left(\bar{\boldsymbol{\Theta}} + \boldsymbol{\delta\Theta}\right)$. Using first order Taylor's series approximation we get error quaternion $\boldsymbol{\delta q}$ as,

$$\boldsymbol{\delta q} = \boldsymbol{q}(\bar{\boldsymbol{\Theta}} + \boldsymbol{\delta\Theta}) - \boldsymbol{q}(\bar{\boldsymbol{\Theta}}) \approx \frac{\partial \boldsymbol{q}(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} |_{\bar{\boldsymbol{\Theta}}} \boldsymbol{\delta\Theta} + (\boldsymbol{\delta\Theta})^2 \tag{6}$$

Following [5], for serial chain manipulators, we can show

$$\boldsymbol{\delta q} = \frac{\partial \boldsymbol{q}_r(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \boldsymbol{\delta\Theta} = \frac{1}{2} \boldsymbol{H}^T \boldsymbol{J}_r \boldsymbol{\delta\Theta} \tag{7}$$

$$\text{where } \boldsymbol{H}(\boldsymbol{q}) = \left[-\boldsymbol{\varepsilon} \; \eta \mathbf{1} - \boldsymbol{\varepsilon}^\times\right] \tag{8}$$

$$\boldsymbol{J}_r = \left[\boldsymbol{\omega}_1 \; \boldsymbol{R}_1 \boldsymbol{\omega}_2 \; \boldsymbol{R}_1 \boldsymbol{R}_2 \boldsymbol{\omega}_3 \; \cdots \; \prod_{i=1}^{n-1} \boldsymbol{R}_i \boldsymbol{\omega}_n\right]$$

where $\boldsymbol{R}_i$ is the rotation matrix of i^{th} frame and $\boldsymbol{\omega}_i$ is the axis of rotation of the same. Utilizing the fact that $\boldsymbol{H}\boldsymbol{H}^T = \mathbb{I}_{3\times 3}$ and Eq (7) we obtain the relationship between $\boldsymbol{\delta q}$ and $\boldsymbol{\delta\Theta}$ as,

$$\boldsymbol{\delta\Theta} = 2\boldsymbol{J}_r^T (\boldsymbol{J}_r \boldsymbol{J}_r^T)^{-1} \boldsymbol{H}_d \boldsymbol{\delta q} \tag{9}$$

where $\boldsymbol{H}_d = \boldsymbol{H}(\boldsymbol{q}_d)$. Substituting $\boldsymbol{\delta\Theta}$ from Eq (9) into Eq (1) we get **uncertainty set** of orientation in task space as follows,

$$\boldsymbol{\delta q}^T \boldsymbol{H}_d^T (\boldsymbol{J}_r \boldsymbol{J}_r^T)^{-1} \boldsymbol{H}_d \boldsymbol{\delta q} \leq c/4 \tag{10}$$

Thus we have propagated error set from joint to rotation task space as in equation (10). Next we formulate the optimization problem to compute robust IK for a given task defined in end-effector space.

4.3 Task-dependent Robust IK Solution

Let $\boldsymbol{g}_d \in SE(3)$ be the desired tool frame configuration, where $\boldsymbol{g}_d = (\boldsymbol{x}_d, \boldsymbol{q}_d)$, with $\boldsymbol{x}_d \in \mathbb{R}^3$ denoting the desired position and $\boldsymbol{q}_d$ denoting the unit quaternion representation of the desired orientation of the tool frame. Let $\mathbb{M}$ be a task-specific metric or measure of distance between two configurations in $SE(3)$. Note that technically, there is no bi-invariant metric in $SE(3)$ [20], and the definition of a metric involves a choice of a weight relating the rotational error in $SO(3)$ to the translational error in $\mathbb{R}^3$. Thus, we assume $\mathbb{M} = \mathbb{P} + \lambda\mathbb{O}$, where $\mathbb{P}$ is a metric defined on $\mathbb{R}^3$ and $\mathbb{O}$ is a metric defined on $SO(3)$. The weight λ is a task-dependent parameter. Some examples of the metric $\mathbb{M}$ of interest are given in the later sections. Note that due to error in execution, any IK solution will result in an end effector configuration that is different from $\boldsymbol{g}_d$.

Let $\mathcal{N}(\boldsymbol{g}_d)$ be a neighborhood of the desired configuration $\boldsymbol{g}_d$, such that any end effector configuration $\boldsymbol{g} \in \mathcal{N}(\boldsymbol{g}_d)$ is an acceptable solution for the task at hand. Let ε be a task-dependent tolerance. Then $\mathcal{N}(\boldsymbol{g}_d) = \{\boldsymbol{g} : \mathbb{M}(\boldsymbol{g}, \boldsymbol{g}_d) \leq \varepsilon\}$. Formally a robust IK solution is defined as *a joint angle vector for which the end effector lies within the neighborhood $\mathcal{N}(\boldsymbol{g}_d)$ with high (user prescribed) probability irrespective of the realization of the random joint space error during task execution.* Note that for a given $\boldsymbol{g}_d$ there may be multiple IK solutions that are robust. It may also be possible that there are no robust solutions. Therefore we pose the robust IK problem as an optimization problem (instead of a feasibility problem) as follows:

$$\begin{aligned} \underset{\boldsymbol{\Theta} \in \mathbb{R}^n}{\text{argmin}} \quad \underset{\boldsymbol{x}, \boldsymbol{q}}{\max} \quad & \mathbb{M} \\ \text{subject to} \quad & \boldsymbol{g}_{st}(\boldsymbol{\Theta}) = \boldsymbol{g}_d \\ & \boldsymbol{\delta x}^T \left[\boldsymbol{J}_p \boldsymbol{J}_p^T\right]^{-1} \boldsymbol{\delta x} \leq c \\ & \boldsymbol{q}^T \boldsymbol{q} = 1 \\ & \boldsymbol{\delta q}^T \left[\boldsymbol{H}_d^T \left(\boldsymbol{J}_r \boldsymbol{J}_r^T\right)^{-1} \boldsymbol{H}_d\right] \boldsymbol{\delta q} \leq \frac{c}{4} \end{aligned} \tag{11}$$

where $\boldsymbol{x} = \boldsymbol{x}_d + \delta\boldsymbol{x}$, $\boldsymbol{q} = \boldsymbol{q}_d + \delta\boldsymbol{q}$. If $\mathbb{M} \leq \varepsilon$, then the optimal solution is a robust IK, which we will denote by $\boldsymbol{\Theta}^*$. Otherwise, there is no robust IK solution. In the above formulation, all the four constraints are dependent on joint vector $\boldsymbol{\Theta}$ but for brevity we have not showed them explicitly. The first constraint ensures that $\boldsymbol{\Theta}^*$ would indeed be a joint solution of the manipulator that would take the end effector to desired configuration $\boldsymbol{g}_d \subset SE(3)$. The second constraint ensures that position error would lie inside its respective error set as described by equation (5). The fourth constraint ensures rotation error will lie inside the error set as in equation (10). The third constraint makes sure that computed $\boldsymbol{q}$ is a unit quaternion. The parameter c is from equation (1) and depends on the confidence level considered while bounding joint space error.

5 SOLUTION APPROACH

In Eq (11), since $\mathbb{M} = \mathbb{P} + \lambda\mathbb{O}$, the objective is separable. The constraints are either in $\boldsymbol{x}$ or in $\boldsymbol{q}$. Therefore the optimization problem in Eq (11) can be written as two decoupled optimization problems as follows:

$$\begin{aligned} \underset{\bar{\boldsymbol{\Theta}}}{\text{argmin}} \quad \underset{\boldsymbol{\delta x}}{\max} \quad & \mathbb{P} \\ \text{subject to} \quad & \boldsymbol{\delta x}^T \left[\boldsymbol{J}_p(\bar{\boldsymbol{\Theta}}) \boldsymbol{J}_p^T(\bar{\boldsymbol{\Theta}})\right]^{-1} \boldsymbol{\delta x} \leq c \end{aligned} \tag{12}$$

$$\begin{aligned} \underset{\bar{\boldsymbol{\Theta}}}{\text{argmin}} \quad \underset{\boldsymbol{q}}{\max} \quad & \mathbb{O} \\ \text{subject to} \quad & \boldsymbol{q}^T \boldsymbol{q} = 1 \\ & \boldsymbol{\delta q}^T \left[\boldsymbol{H}_d^T \left(\boldsymbol{J}_r(\bar{\boldsymbol{\Theta}}) \boldsymbol{J}_r^T\right)^{-1} (\bar{\boldsymbol{\Theta}}) \boldsymbol{H}_d\right] \boldsymbol{\delta q} \leq \frac{c}{4} \end{aligned} \tag{13}$$

Now we discuss methods of solving the inner maximization problem in Eq (12) and (13) efficiently for one particular $\bar{\boldsymbol{\Theta}}$. Output of these inner max problems are error bounds in position or orientation respectively. After solving this inner max problem for multiple IKs, the outer min problem is just finding the minimum of the computed error bounds.

5.1 Computing position error bound for a given $\bar{\boldsymbol{\Theta}}$

The problem of finding error bounds for a particular $\bar{\boldsymbol{\Theta}}$ that will satisfy position error ellipsoid constraint in task space is defined in Eq (14).

$$\begin{aligned} \underset{\boldsymbol{\delta x}}{\text{maximize}} \quad & \mathbb{P} \\ \text{subject to} \quad & \boldsymbol{\delta x}^T \left[\boldsymbol{J}_p(\bar{\boldsymbol{\Theta}}) \boldsymbol{J}_p^T(\bar{\boldsymbol{\Theta}})\right]^{-1} \boldsymbol{\delta x} \leq c \end{aligned} \tag{14}$$

Please recall that origin of the error ellipsoid is desired tool position (see figures 1 or 2) whereas any other point lying inside the error set is a deviation from the desired position. The choice of metric $\mathbb{P}$ may vary depending on whether the task demands to minimize position error in $\mathbb{R}^3$ or any of its sub-spaces. Next we discuss on computing position error bounds in $\mathbb{R}^3$ and its lower dimensional sub-spaces, i.e., $\mathbb{R}$ and $\mathbb{R}^2$ respectively.

Computing position error bound in $\mathbb{R}^3$: Maximum possible position error or position error bound in this case would be the distance from the center of the task space position error ellipsoid (which is desired tool position) to the furthest point in the error ellipsoid. Following the fact the maximum eigenvalue of an ellipsoid describes the distance of the furthest point from its center, the optimization problem in equation (12) is reduced to as in equation (15).

$$\mathbb{P}^* = \max_{\lambda} \quad \text{eig}\left(c\left[\boldsymbol{J}_p(\bar{\boldsymbol{\Theta}})\boldsymbol{J}_p(\bar{\boldsymbol{\Theta}})^T\right]\right) \tag{15}$$

where $\mathbb{P}^*$ denotes $max(\mathbb{P})$ and *eig* a function that computes all eigenvalues of an input matrix. $\boldsymbol{J}_p$ and $\bar{\boldsymbol{\Theta}}$ are defined as before.

Computing position error bound in along a direction: Computing position error bound along a direction becomes necessary for tasks such as, drawing a horizontal line by a robot arm. In that case the arm needs minimum deviation of end-effector along vertical direction. Therefore position error bound for this case would be projection length of error ellipsoid onto the vertical axis. In general if we want to minimize error along a direction $\boldsymbol{v} \in \mathbb{R}^3$ then position error bound would be $\mathbb{P}^* = \left|\frac{\boldsymbol{L}_c^{-1}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}\right|$, where $\boldsymbol{L}_c$ is lower-triangular matrix obtained by Cholesky decomposition of $\frac{1}{c}\left[\boldsymbol{J}_p\boldsymbol{J}_p^T\right]^{-1}$.

Computing position error bound in $\mathbb{R}^2$ or along a plane: In this case position error bound is the projected area of position error ellipsoid onto the plane along which error minimization is required. Suppose the given plane equation is defined as $PL \equiv \{\boldsymbol{x}|\boldsymbol{x} = \boldsymbol{T}\boldsymbol{t}\}$ where $\boldsymbol{T}$ is the basis matrix defining the plane, then position error bound $\mathbb{P}^* = det\left|\left(\boldsymbol{T}\boldsymbol{T}^T\right)^{-T}\boldsymbol{L}_c\boldsymbol{L}_c^T\left(\boldsymbol{T}\boldsymbol{T}^T\right)^{-1}\right|$ where $\boldsymbol{L}_c$ is defined as before.

5.2 Computing rotation error bound for given $\bar{\boldsymbol{\Theta}}$

To compute rotation error bound for an IK, say $\bar{\boldsymbol{\Theta}}$, we need to solve the inner max-problem as in equation (13). If we consider orientation error metric to be $\mathbb{O} = \boldsymbol{q}^T\boldsymbol{q}_d$ following [22], then the inner max problem turns into a min problem since $\mathbb{O}$ in this case would define *cosine* of the included angle. Then inner optimal problem now turns out as in equation (16).

$$\begin{aligned} \min_{\boldsymbol{q}} \quad & \boldsymbol{q}_d^T\boldsymbol{q} \\ \text{s.t.} \quad & \boldsymbol{q}^T\boldsymbol{q} = 1 \\ & \boldsymbol{\delta q}^T\boldsymbol{H}_d^T\left(\boldsymbol{J}_r\boldsymbol{J}_r^T\right)^{-1}\boldsymbol{H}_d\boldsymbol{\delta q} \le \frac{c}{4} \end{aligned} \tag{16}$$

The optimization problem in equation (16) is non-convex because of the strict quadratic equality unit quaternion constraint.

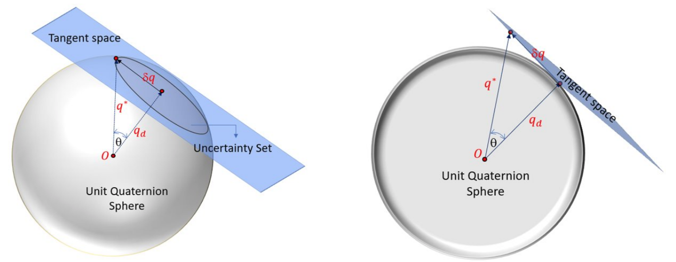

Figure 3: FROM LEFT: UNIT QUATERNION SPHERE WITH UNCERTAINTY SET (ELLIPSOID) IN ITS TANGENT SPACE. A SLICE OF UNIT QUATERNION SPHERE IN THE PLANE DEFINED BY $\boldsymbol{q}_d$ AND $\boldsymbol{\delta q}$

Nonconvex problems are hard in general and optimization solvers ends up to local minimum instead of global minimum while solving them. We can relax the problem by ignoring unit quaternion constraint. However to ensure that we still minimize the included angle between $\boldsymbol{q}_d$ and $\boldsymbol{q}$, we need to scale the objective in Eq (16) with the length of $\boldsymbol{q}$. The relaxed optimization problem is presented in Eq (17).

$$\begin{aligned} \boldsymbol{q}^* = \min_{\boldsymbol{q}} \; & \boldsymbol{q}^T\boldsymbol{q}_d/||\boldsymbol{q}|| \\ \text{s.t.} \; & \boldsymbol{\delta q}^T\boldsymbol{H}_d^T(\boldsymbol{J}_r\boldsymbol{J}_r^T)^{-1}\boldsymbol{H}_d\boldsymbol{\delta q} \le c/4 \end{aligned} \tag{17}$$

We have presented a schematic in figure 3 to visualize optimization problem in equation (17). The $3D$ ball represents a unit quaternion sphere (in gray) whereas blue plane represents the tangent space of unit quaternion sphere touching it at $\boldsymbol{q}_d$. The ellipse, lying on the tangent space is the uncertainty set centered around $\boldsymbol{q}_d$. Any vector lying inside this ellipsoid with the starting point fixed at $\boldsymbol{q}_d$ will represent random rotation error $\boldsymbol{\delta q}$ with respect to $\boldsymbol{q}_d$. Assuming $\boldsymbol{H}_d\boldsymbol{\delta q} = \boldsymbol{v}$ we get the expression of uncertainty set as $\boldsymbol{v}^T(\boldsymbol{J}_r\boldsymbol{J}_r^T)^{-1}\boldsymbol{v} \le c/4$. Further we can compute $\boldsymbol{q}$ from definition of $\boldsymbol{v}$ as following,

$$\boldsymbol{H}_d\boldsymbol{\delta q} = \boldsymbol{v} \implies \boldsymbol{\delta q} = \boldsymbol{H}_d^T\boldsymbol{v} \implies \boldsymbol{q} = \boldsymbol{q}_d + \boldsymbol{H}_d^T\boldsymbol{v} \tag{18}$$

Utilizing Eq (18) and the fact that $\boldsymbol{H}_d\boldsymbol{q}_d = 0$ we can write optimization objective in Eq (17) as, $\boldsymbol{q}^T\boldsymbol{q}_d/||\boldsymbol{q}|| = 1/\sqrt{1+\boldsymbol{v}^T\boldsymbol{v}}$ Then optimization problem in Eq (17) can be written as,

$$\begin{aligned} \boldsymbol{v}^* = \max_{\boldsymbol{v}} \; & \boldsymbol{v}^T\boldsymbol{v} \\ \text{s.t.} \; & \boldsymbol{v}^T(\boldsymbol{J}_r\boldsymbol{J}_r^T)^{-1}\boldsymbol{v} \le c/4 \end{aligned} \tag{19}$$

The optimization problem in equation (19) represents eigenvalue problem. Mathematically $\boldsymbol{v}^* = \frac{1}{2}\sqrt{c\lambda_{max}}\boldsymbol{V}_{max}$ where λ_{max} is maximum eigenvalue of $[\boldsymbol{J}_r\boldsymbol{J}_r^T]$ and $\boldsymbol{V}_{max}$ is the eigenvector associated to λ_{max}. Once we have computed $\boldsymbol{v}^*$ we can compute bounding quaternion $\boldsymbol{q}^*$ by following equation (18) as,

$$\boldsymbol{q}^* = \boldsymbol{q}_d + \boldsymbol{H}_d^T \boldsymbol{v}^* / ||\boldsymbol{q}_d + \boldsymbol{H}_d^T \boldsymbol{v}^*|| \tag{20}$$

Once we obtain $\boldsymbol{q}^*$, we find rotation error bound as,

$$\mathbb{O} = \arccos \boldsymbol{q}_d^T \boldsymbol{q}^* \tag{21}$$

In Algorithm 1 we have presented the steps required to compute robustIK in concise manner for the ease of implementation.

6 ALGORITHM

Algorithm 1 takes desired end-effector configuration $\boldsymbol{g}_d$, $c = (k\sigma)^2$, metric weighing factor λ and allowable error tolerance ε as input. Output of the algorithm is robust IK, $\boldsymbol{\Theta}^*$. Line 1 separates desired position $\boldsymbol{p}_d$ and rotation $\boldsymbol{q}_d$ from $\boldsymbol{g}_d$. Line 2 computes M IK solutions [23] and stores them in $\boldsymbol{\Gamma}$. Then we iterate over IK solution set between lines 3 and 8. For each IK solution we compute error bounds for position (line 5), rotation (line 6) and finally weighted error bound in line 7. If computed error bound is lesser or equal to ε, we store the error and corresponding IK solutions in Line 8 and 9 respectively. Line 12 finds the index j corresponding to minimum error bound IK. In line 13 we retrieve the robust-IK from the IK solution set.

Algorithm 1 Steps to compute RobustIK
Input: $\boldsymbol{g}_d, c, \lambda, \varepsilon$ **Output**: $\boldsymbol{\Theta}^*$

1: $\boldsymbol{p}_d \leftarrow \boldsymbol{g}_d[1:3,4]$ and $\boldsymbol{q}_d \leftarrow$ **rotm_2_q**$(\boldsymbol{g}_d[1:3,1:3])$
2: $\boldsymbol{\Gamma} \leftarrow$ **IKsolver**$(\boldsymbol{g}_d)$ where $\boldsymbol{\Gamma} \in \mathbb{R}^{M\times N}$
3: **for** $(i \leftarrow 1 \ \ to \ \ M)$ **do**
4: $\quad \bar{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Gamma}[i]$
5: $\quad \mathbb{P} \leftarrow$ solve Eq (14) using $\boldsymbol{p}_d, \bar{\boldsymbol{\Theta}}, c$
6: $\quad \mathbb{O} \leftarrow$ solve Eq (21) using $\boldsymbol{q}_d, \bar{\boldsymbol{\Theta}}, c$
7: $\quad$ **if** $\mathbb{P} + \lambda\mathbb{O} \leq \varepsilon$ **then**
8: $\quad\quad \mathbb{D}$.append$(\mathbb{P} + \lambda\mathbb{O})$
9: $\quad\quad sol$.append$(\bar{\boldsymbol{\Theta}})$
10: $\quad$ **end if**
11: **end for**
12: $j \leftarrow$ **MinIndx**$(\mathbb{D})$
13: $\boldsymbol{\Theta}^* \leftarrow sol[j,:]$ **return** $\boldsymbol{\Theta}^*$

7 NUMERICAL EXAMPLES

Here we present applications of robust-IK algorithm in two different situations. The first application is pertaining to deciding feasibility of an assigned task when robot kinematics and joint errors are known. The second application is from the perspective of robot designer to design joint actuators. The application tasks are chosen in such a manner that in one case only position error at the end-effector dictates success of the that task whereas pose error at end-effector space dictates success of the task in second application. We present simulation and experimental results using 7-DoF redundant Baxter research robot [24, 25].

7.1 ROBUST IK FOR GRASPING TASK

Objective of this example is to check feasibility of placing end-effector of Baxter arm within a small neighborhood of a desired pre-grasp configuration. While the desired end-effector configuration is decided by the pose of the block to grasp, size of the neighborhood around that desired configuration is dictated by the available clearance between the block-width and the gripper opening. More specifically, if the clearance is substantially large, the allowable error neighborhood will also be large, then execution of any one of the IK solution should be good in accomplishing the task. However this is not the case if the relative clearance is small. In this case we need to use Algorithm 1 which would return robust-IK ensuring success of the task with high-probability as opposed to any randomly chosen IK solution.

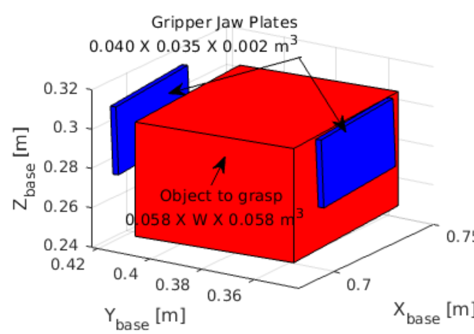

Figure 4: SIMULATION SETUP FOR PLACING END-EFFECTOR IN PRE-GRASP CONFIGURATION TASK

Simulation Result: The simulation setup is shown in figure 4. The opening of the grippers is such that small position errors along X_{base} and Z_{base} directions does not affect success of the task. However because of small clearance between the object and gripper opening, success of the task is sensitive to small position error along Y_{base}. Hence we are interested in computing robust-IK corresponding to minimized position error bound along Y_{base}. The desired end-effector pose is $\boldsymbol{p}_d = [0.71305, 0.3786, 0.300]$ and $\boldsymbol{q}_d = [0.0086, 0.9992, 0.0370, 0.0155]$. Each joint error is considered to be zero-mean Gaussian $\mathcal{N}(\mu = 0, \sigma = 0.0045\text{rad})$ with a spread upto 2 standard-deviations i.e., $k = 2$.

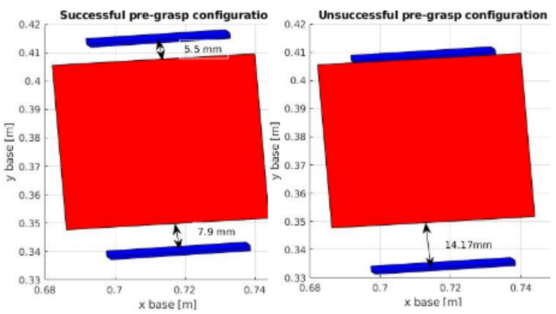

Figure 5: INSTANCES OF SUCCESSFUL AND FAILED POSITIONING TASKS BY $\mathbf{\Theta}^*$ and $\mathbf{\Theta}^-$ RESPECTIVELY

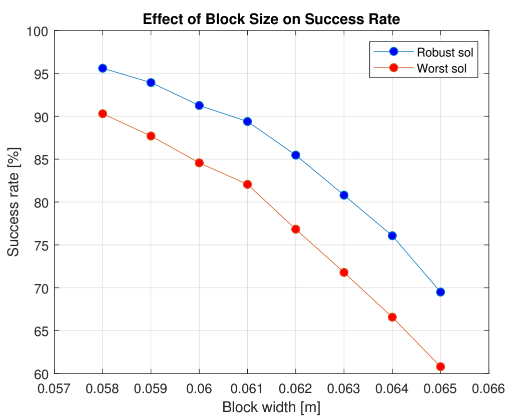

Figure 6: SUCCESS RATE OF BEST AND WORST SOLUTIONS WITH VARYING BLOCK SIZE WHILE THE GRIPPER OPENING IS FIXED AT 72mm.

This error parameters are in accordance with Baxter robot [24]. Using Algorithm 1 we found robust-IK as $\mathbf{\Theta}^* = [0.0052, -0.1660, -2.0927, 1.1777, 1.6105, 2.0793, 2.6467]$rad. In figure 6 we have plotted success-rates in accomplishing the pre-grasp positioning task by robust-IK ($\mathbf{\Theta}^*$) along with the worst-IK ($\mathbf{\Theta}^-$) (IK associated to maximum error bound) while the clearance is varied by varying block width. Notice that when block width is $w = 58$mm and desired success-rate is $> 80\%$ it does not matter if $\mathbf{\Theta}^*$ or $\mathbf{\Theta}^-$ is executed because both have success-rate more than 90%. However if block width is $w = 63$mm only $\mathbf{\Theta}^*$ has success-rate $> 80\%$, hence this solution should be executed instead of $\mathbf{\Theta}^-$. Further if block width is $w = 65$mm none of the IK solution has desired success-rate indicating infeasibility of the task with that block width.

Experimental Result: We performed experiments with Baxter robot for the same task as in simulation with $\mathbf{\Theta}^*$ and $\mathbf{\Theta}^-$ computed as before. For 10 different trials, $\mathbf{\Theta}^*$ and $\mathbf{\Theta}^-$ are executed from different initial end-effector configurations and outcomes are tabulated in Tab 1. In Fig 7 we have shown instances of

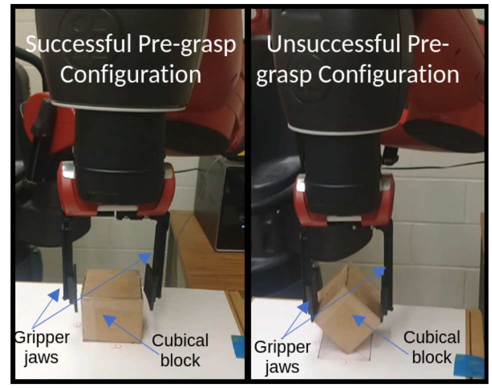

Figure 7: INSTANCES OF SUCCESSFUL AND FAILED PRE-GRASP WITH ROBUST AND WORST IK SOLUTIONS.

Trials	1	2	3	4	5	6	7	8	9	10
Θ_{rob}	1	1	1	1	1	0	1	1	1	1
Θ_{wst}	0	1	0	1	1	0	1	1	0	1

Table 1: SUCCESSFUL AND UNSUCCESSFUL EVENTS FOR ROBUST AND WORST SOLUTIONS WITH RESPECT TO EACH TRIAL. SUCCESSFUL AND UNSUCCESSFUL EVENTS ARE DENOTED AS 1 AND 0 RESPECTIVELY

successful and failed pre-grasp positioning of end-effector after executing $\mathbf{\Theta}^*$ and $\mathbf{\Theta}^-$ respectively.

7.2 ROBUST IK FOR PEG-IN-HOLE TYPE PROBLEMS

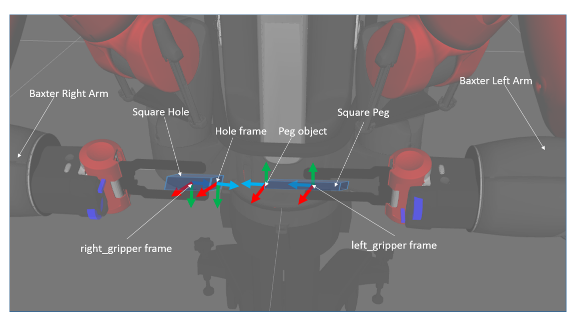

Figure 8: SIMULATION SETUP FOR PEG-IN-HOLE TASK. HOLE IS KEPT FIXED BY RIGHT ARM WHILE THE LEFT ARM HAS TO POSITION PEG_TIP FRAME WITHIN AN ERROR MARGIN (BASED ON AVAILABLE CLEARANCE) AROUND THE DESIRED PRE-INSERTION POSITION. FRAMES OF INTERESTS ARE ALSO SHOWN

Objective of this example is to show another application of computing robust-IK solution to choose robot actuator to ensure high success-rate for an assigned tasks. In this example we are given to accomplish a peg-in-hole type of task (see Fig 10) along with lengths of peg, hole, relative clearance between peg and hole and kinematic model of the Baxter robot. Assuming joint errors are zero-mean Gaussian, we need to choose error parameter σ so that error at the peg-tip will be within an error margin. The allowable error margin again depends on the relative clearance between the mating parts. Also note that any pose error at the end-effector frame i.e., left_gripper frame manifest as position error at the peg_tip frame. Let $(\boldsymbol{p}_d, \boldsymbol{q}_d)$ is the desired and $(\boldsymbol{p}, \boldsymbol{q})$ is the achieved configurations at peg_tip frame. Then from position error (e_{peg_tip}) at peg_tip frame as in Eq (22) we can derive error metric to minimize in Eq (23).

$$e_{peg\_tip} = (\boldsymbol{p} - \boldsymbol{p}_d) + l_p * (\boldsymbol{R}_z - \boldsymbol{R}_{d_z}) \tag{22}$$

$$\begin{aligned} ||e_{peg\_tip}|| &= ||(\boldsymbol{p} - \boldsymbol{p}_d) + l_p * (\boldsymbol{R}_z - \boldsymbol{R}_{d_z})|| \\ &\leq ||\boldsymbol{p} - \boldsymbol{p}_d|| + l_p * ||\boldsymbol{R}_z - \boldsymbol{R}_{d_z}|| \\ &= ||\boldsymbol{p} - \boldsymbol{p}_d|| + l_p * |\theta_z| \end{aligned} \tag{23}$$

where $\boldsymbol{R}_z$ and $\boldsymbol{R}_{d_z}$ represents $z-$axes of realized and desired rotation matrices and θ_z is the included angle between them. The term $||\boldsymbol{p} - \boldsymbol{p}_d||$ is equivalent to $\mathbb{P}$ of Eq (12), $|\theta_z|$ is equivalent to $\mathbb{O}$ of Eq (13) and l_p the length of the peg extended from left_gripper frame is equivalent to λ.

Results Suppose desired *left_gripper* frame configuration to ensure successful insertion of the peg into the hole is $\boldsymbol{p}_d = [0.6165, 0.077, 0.4025]$m, $\boldsymbol{q}_d = [0.6839, 0.7174, 0.0799, -0.1064]$. The extended peg length l_p is assumed to be 0.10 m. For a given value of σ, using Algorithm 1 we can compute robust-IK solution corresponding to minimum error bound. In Fig 9 we have shown change in maximum possible error at peg_tip frame with change in joint error parameter *i.e,* σ for robust and worst IK solutions. Notice that if relative clearance between peg and hole is 15mm, designer should choose actuators which has standard deviation $\sigma \leq 0.005$rad. Similarly for 10mm clearance $\sigma \leq 0.003$rad needs to be chosen. Figure 9 can also be used as a lookup table to choose allowable clearance between peg and hole when joint errors of the robot is known. In Fig 10 we have shown instances of peg positioning task for robust and worst IK solutions when $\sigma = 0.0045$rad and clearance 20mm by simulated Baxter robot. We find robust and worst IK solutions are $\boldsymbol{\Theta}^* = [0.365997, -0.205692, -1.45802, 1.66477, 2.93037, -1.12361, -0.142083]$ and $\boldsymbol{\Theta}^- = [-0.15771, 0.880958, -2.75321, 1.71041, 1.1743, 1.69088, 2.11322]$ respectively. Notice in Fig 9 that maximum position error at the peg-tip for robust-IK is well below 20mm but for worst-IK it is close to maximum allowable error. Hence it is safe to execute robust-IK instead of worst-IK to obtain reliable performance while performing the task. After running the simulation multiple times we found the average execution time to compute robust-IK solution was less than 0.40 second using computer with intel core *i*5 processor and 8GB memory. Therefore we can compute robust-IK solution in real-time.

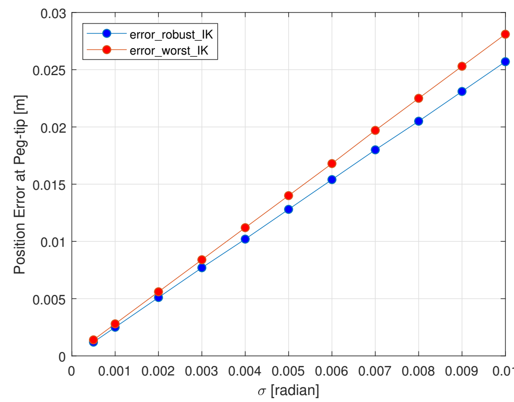

Figure 9: CHANGE IN POSITION ERROR AT THE PEG-TIP DUE TO POSE ERROR AT *left_gripper* FRAME WITH CHANGE IN STANDARD DEVIATION (σ) OF JOINT ERRORS FOR ROBUST AND WORST IK(S)

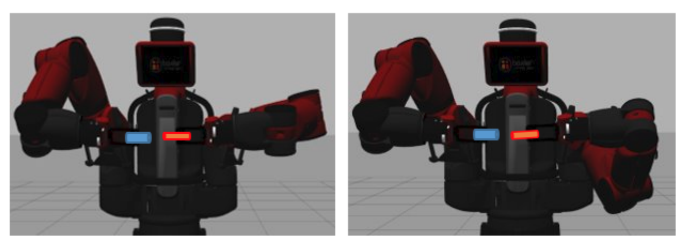

Figure 10: INSTANCES AFTER EXECUTING BEST(LEFT) AND WORST(RIGHT) IK SOLUTIONS FOR PEG-IN-HOLE TASK WHILE HOLE IS KEPT FIXED IN RIGHT ARM AND PEG IN THE LEFT HAS TO BE PLACED AT A DESIRED POSE FOR SUCCESSFUL ASSEMBLY

8 CONCLUSION

We present method of computing robust-IK that corresponds to minimum error bound in task space. We have formalized robust-IK problem as min-max type constraint optimization problem first. Then exploiting the dependencies of the constraint variables, we were able to reduce the main optimization problem into two smaller optimization problems (sub-problems). Then we showed that each of these two sub-problems can be posed as finding maximum eigenvalue problems. Finally using simulation and experimental results we show that computing robust-IK is indeed helpful in deciding feasibility of tasks knowing robot kinematic model and joint error information. We also showed an application where robust-IK can be used as tool in selecting actuators of robots ensuring high success-rate in assigned tasks.

Future Work: In future we plan to see how self evaluation application of robust-IK algorithm can be utilized when there are uncertainties both in actuation of the robot and pose of the block to grasp. We also plan to extend this research in propagating errors to task space of a mobile manipulator with actuation errors in its base and arm joints.

ACKNOWLEDGEMENT

This work was supported in part by AFOSR award FA9550-15-1-0442.

REFERENCES

[1] Chikjian, G. S., and Kyatkin, A. B., 2001. *Engineering Applications of Noncommutative Harmonic Analysis*. CRC Press.

[2] Barfoot, T. D., and Furgale, P. T., 2014. "Associating uncertainty with three-dimensional poses for use in estimation problems". *IEEE Transaction on Robotics,* ***30***(3).

[3] Wang, Y., and Chirikjian, G. S., 2006. "Propagation of errors in hybrid manipulators". *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*.

[4] Wang, Y., and Chirikjian, G., 2008. "Nonparametric second-order theory of error propagation on motion groups". *The International journal of robotics research,* ***27(11-12)***, pp. 1258–1273.

[5] Barfoot, T., Forbes, J. R., and Furgale, P. T., 2011. "Pose estimation using linearized rotations and quaternion algebra". *Acta Astronautica,* ***68***(1-2), pp. 101–112.

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

[7] McCarthy, J. M., 1990. *An Introduction to Theoretical Kinematics*. MIT Press, Cambridge, Mass.

[8] Selig, J., 1996. *Geometrical Methods in Robotics*. Springer, New York.

[9] Smith, R. C., and Cheeseman, P., 1986. "On the representation and estimation of spatial uncertainty". *The International Journal of Robotics Research,* ***5***(4), pp. 56–68.

[10] Su, S.-F., and Lee, C. S. G., 1992. "Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks". *IEEE T. Syst. Man. Cyb.,* ***22***(6), pp. 1376 – 1389.

[11] Meggiolaro, M., Mavroidis, C., and Dubowsky, S. "Identification and compensation of geometric and elastic errors in large manipulators: Application to a high accuracy medical robot".

[12] Mavroidis, C., Dubowsky, S. ., Drouet, P., Hintersteiner, J., and Flanz, J., 1997. "A systematic error analysis of robotic manipulators: Application to a high performance medical robot". *Intemational Conference on Robotics and Automation*.

[13] Wu, C.-H., 1983. "The kinematic error model for the design of robot manipulator". *IEEE American Control Conference*.

[14] Wu, C.-H., 1984. "A kinematic cad tool for the design and control of a robot manipulator". *The International Journal of Robotics Research,* ***3***(1), pp. 58–67.

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

[16] Chen, J., Wang, B., and Yang, J. S., 1984. "Robot positioning in accuracy improvement through kinematic parameter identification". *in Proc. 3rd Canadian CAD/CAM and Robotics Conf., Toronto, ON. Canada*, June.

[17] Chen, J., and Chao, L. M., 1987. "Positioning error analysis for robot manipulators with all rotary joints". *IEEE Journal of Robotics and Automation,* ***RA-3***(6), pp. 539–545.

[18] Yin, J., and Gao, Y., 2012. "Pose accuracy calibration of a serial five dof robot". *Energy Procedia,* ***14***, pp. 977–982.

[19] Goswami, A., Quaid, A., and Peshkin, M., 1993. "Identifying robot parameters using partial pose information". *IEEE Control Systems*.

[20] Belta, C. A., and Kumar, V. R., 2002. "Euclidean metrics for motion generation on se(3)". *Departmental Papers (MEAM).150*.

[21] Craig, J. J., 2005. *Introduction to Robotics: Mechanics and Control*.

[22] Huynh, D. Q., 2009. "Metrics for 3d rotations: Comparison and analysis". *J Math Imaging Vis*.

[23] Sinha, A., and Chakraborty, N., 2019. "Geometric search based inverse kinematics of 7-dof redundant manipulator with multiple joint offsets". *IEEE International Conference on Robotics and Automation*.

[24] Baxter robot.

[25] II, R. L. W., 2017. "Baxter humanoid robot kinematics". *Internet Publication*, April.