

Topology Design of Surgical Reconfigurable Robots by Interval Analysis

Denny Oetomo^{1,2}, David Daney², Kanako Harada³, Jean-Pierre Merlet², Arianna Menciassi^{3,4}, Paolo Dario³

Abstract—An automated design generation algorithm for a serial kinematic chain is presented for the reconfigurable robot used in a novel endoluminal surgical procedure (European Union project ARES). The algorithm produces the possible topologies, given the design constraints, desired performance, and available modules, such that all constraints are satisfied for every point in the desired workspace. This is achieved through the use of interval analysis methods and branch-and-bound loop that searches through the end-effector pose and the design parameter spaces. The resulting algorithm is demonstrated through an example of a serial chain manipulator made of the reconfigurable modules of the surgical robot for the application. The results are presented and discussed.

I. INTRODUCTION

AN endoluminal surgical system was proposed in the European Union project ARES (Assembly of Reconfigurable Endoluminal Surgical system) [1]. The aim is to construct a reconfigurable manipulator capable of observation and interventional procedures in the Gastrointestinal (GI) tract. The modules of the robot are to be sent into the GI tract of the patient in the form of swallowed capsules. Upon reaching the stomach cavity, the modules then reconfigure themselves into a topology suitable for the desired surgical task. The modules to be utilised consist of actuated modules (1DOF revolute jointed module) and various passive modules (non-actuated, but serve other purposes, such as batteries, camera, support, etc).

The reconfiguration aspect of this robotic scheme poses the challenge of automatic generation of a suitable manipulator topology for the given task. For instance, a surgical procedure is to be carried out on a specific section of the stomach wall, requiring a given workspace reach. The actuated modules is limited in the range of displacement it can produce. Give the design constraints (joint displacement limit) and task constraints (the required workspace), the algorithm is required to produce the possible manipulator topologies such that all the constraints are satisfied (for all points in the required workspace).

The problem of automatic generation of a manipulator topology to satisfy a set of given criteria is not one that has been well studied. Most of the design strategies proposed are optimisation based [2], [3]. Drawbacks associated with

optimisation based approaches include the difficulties in constructing the suitable cost functions, especially when several measures of different physical properties are involved and the physical meanings of variables are lost in the construction of the functions. Optimisation approaches also often push the solutions to one side of the extreme without considering whether certain thresholds of the other constraints are violated. Classical optimisation is also not appropriate to tackle antagonistic constraints (e.g. accuracy versus workspace size) as only Pareto optimum exists.

Under reconfigurable robotics [4], various aspects of the problem have been studied extensively in the past, such as module design [5], docking-and-release strategies [6], reconfiguration strategy [7], as well as various gait and locomotion strategies [8]. However, there are usually a set of predefined topologies to which the reconfigurable robot can rearrange itself to. An automated generation strategy of suitable topology considering the task at hand requires the high-level intelligence that is still not well-established at the moment.

In this paper, an interval-based method is presented to provide an automated search in the design parameter space to determine the ranges of values in the robot design parameters that would yield an achievable end-effector workspace where all given constraints are satisfied for all poses inside the desired workspace. The design parameters are the parameters such as link lengths and offset angles that completely describe a serial chain topology and configuration, such as the Denavit-Hartenberg parameters. In interval-based method, these parameters, as well as the joint and task space variables, are expressed as bounded ranges of continuous values (intervals) instead of as real variables. This renders the algorithm with the ability to accommodate a continuous range of possible values for each parameter and evaluate the serial chain performance for all points within these bounds, not just on point sampling basis. This also allows the algorithm to take into account any uncertainties, such as modelling uncertainties, fabrication tolerances, and round off errors, in producing the certified solutions to the constraints. In contrast to optimisation based approaches, interval analysis techniques maintain the physical meanings of various parameters and quantitative measures. The result of the algorithm is a set of windows in the manipulator design parameter space, within which it is guaranteed that each point defines a manipulator topology which satisfies all the required constraints for all points in the desired workspace. If desired, optimisation technique can be performed afterwards on the solution boxes found by the interval techniques. This would yield an optimised solution out of the set of design

This work is supported in part by the European Commission, European NEST project No. 015653 ARES(Assembling Reconfigurable Endoluminal Surgical system).

¹Department of Mechanical Engineering, University of Melbourne, VIC 3010 Australia doetomo@unimelb.edu.au, ²Project COPRIN, INRIA Sophia-Antipolis, France FR-06902, ³CRIM Lab, Scuola Superiore Sant'Anna, Pisa 56025 Italy, ⁴The Italian Institute of Technology (IIT), Genova 16163 Italy.

parameters that are already certified to satisfy all the given constraints.

II. CONSTRAINTS IN A SERIAL MANIPULATOR

In this paper, the study of the automatic generation of manipulator topology to satisfy the given task is concentrated on serial manipulators. The desired workspace is the end-effector range of poses (translations and orientations) that is required for the given task. Within this workspace, all constraints are to be satisfied, for example, it is required that all points within the desired workspace be reachable given the joint limits of the manipulator, that the workspace does not contain singular configurations, that a required amount of force at the end-effector is achievable given the available joint torques, etc. The set of constraints is defined as $\mathbf{C}(\mathbf{x}, \mathbf{h}) = \{C_1, C_2, \dots, C_{NC}\}$, where \mathbf{x} is the end-effector pose, \mathbf{h} represents the design parameters, $C_i(\mathbf{x}, \mathbf{h})$ represents a single constraint, and NC denotes the total number of constraints being imposed on the system. Each constraint $C_i(\mathbf{x}, \mathbf{h})$ could be defined as an inequality or equality constraint.

The design constraint posed by the actuated module is that it has a displacement limit of $\pm 30^\circ$ in its revolute joint motion [9]. The module is shown in Fig. 1. The modular architecture was chosen to overcome the intrinsic limitations of the current capsule endoscopy because it allows the delivery of more components inside the human body with different or similar functions. Based on this concept, the patient ingests some capsular modules consisting of modules with structural functions and modules with diagnostic and/or interventional functions (Fig. 2(a)-(c)). The modules then reconfigure into the required manipulator topology for the task (Fig. 2(d)). The modules disconnect into individual modules or into the appropriate topology to exit the GI tract (Fig. 2 (e)-(g)).

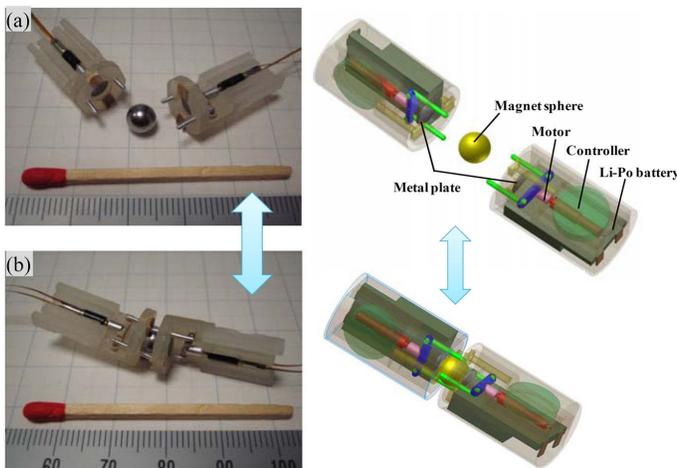


Fig. 1. Design of the revolute actuation module

In this paper, interval analysis method [10] is used as a means to express a variable as a continuous range of values. An interval extension of variable x is denoted $X = [\underline{x}, \bar{x}]$, where \underline{x} and \bar{x} are the lower and upper bounds of the interval

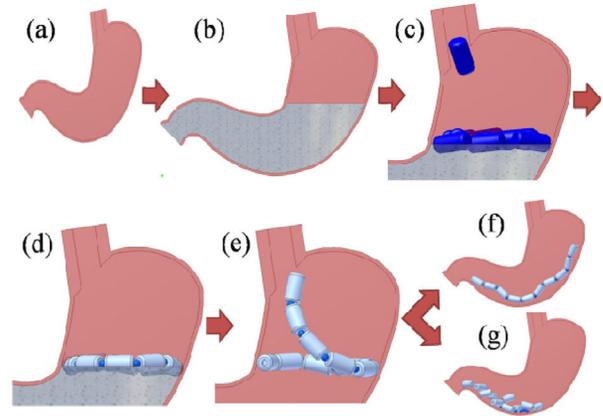


Fig. 2. Concept of the reconfigurable modular surgical robot in the stomach cavity

variable, respectively. Similarly, an interval extension of a function $f(x)$ is denoted as $F(X) = [\underline{f}, \bar{f}]$. Introduction to interval analysis can be found in the literature, such as [11].

In the case of our manipulator design problem, each design constraint, $C_i(\mathbf{x}, \mathbf{h})$, can be expressed as the required range of value, bounded within $[\underline{c}_i, \bar{c}_i]$. They represent the minimal and maximal values that a quantitative property is allowed to have. An interval box (\mathbf{X}, \mathbf{H}) is therefore an *inner box* when the interval extension of a function $F_i(\mathbf{X}, \mathbf{H})$ evaluates to an interval value such that:

$$\{\forall \mathbf{x} \in \mathbf{X}, \forall \mathbf{h} \in \mathbf{H}; \underline{f}_i \geq \underline{c}_i \text{ and } \bar{f}_i \leq \bar{c}_i\} \quad (1)$$

for *all* constraints, ($i = 1, \dots, NC$). An interval box (\mathbf{X}, \mathbf{H}) is an *outer box* when:

$$\{\forall \mathbf{x} \in \mathbf{X}, \forall \mathbf{h} \in \mathbf{H}; \bar{f}_i \leq \underline{c}_i \text{ or } \underline{f}_i \geq \bar{c}_i\} \quad (2)$$

for *any* of the constraints. When (\mathbf{X}, \mathbf{H}) does not yield an interval $\mathbf{F}(\mathbf{X}, \mathbf{H})$ that is completely contained within the limits of constraints (inner box) or completely outside these limits (outer box), it is designated as a *boundary box*.

III. INTERVAL ANALYSIS ALGORITHM FOR SERIAL KINEMATIC CHAINS

In the interval evaluation of a function, numerical values are substituted into a function, resulting in the loss of the relationships between various variables. Overestimation [11] may occur as multiple occurrences of the same variables within the function are regarded as independent variables. This makes it difficult to obtain clear decision whether a box forms an inner or outer box, therefore the Branch-and-bound loops were utilised as to consider the smaller subsections of the boxes at each iteration of the solution search algorithm. Solutions are searched throughout the end-effector workspace \mathbf{X} and design parameter \mathbf{H} .

A. Constraint Satisfaction in End-Effector Workspace

To construct the interval-based algorithm for serial kinematic chains, a function $S(\mathbf{x}, \mathbf{h})$ is defined such that it evaluates the interval extension of the kinematic functions corresponding to the design constraint of the system $\mathbf{C}(\mathbf{X}, \mathbf{H})$, compares it to the given constraints, and returns 1, 0, -1,

TABLE I

SUMMARY OF BRANCH-AND-BOUND LOOP TO OBTAIN THE DESIGN PARAMETER RANGE OF VALUES WHERE ALL CONSTRAINTS ARE SATISFIED WITHIN THE GIVEN WORKSPACE \mathbf{X}

- Input:** \mathbf{X} and \mathbf{H}_i
Output: 1, 0, -1, (representing inner, boundary, and outer solutions, respectively).
- 1 Initialise empty lists \mathcal{L}_{IN} , \mathcal{L}_{OUT} , and \mathcal{L}_B .
 - 2 Initialise list \mathcal{L} containing initial design parameter intervals \mathbf{H}_0 to be analysed.
 - 3 **While** (\mathcal{L} not empty)
 - (a) Extract design parameter box \mathbf{H}_i from \mathcal{L}
 - (b) Remove design parameter box \mathbf{H}_i from \mathcal{L}
 - (c) Evaluate function $S^*(\mathbf{X}, \mathbf{H}_i)$,
 - (d) **If** $S^*(\mathbf{X}, \mathbf{H}_i) == 1$
add \mathbf{H}_i to list \mathcal{L}_{IN}
 - (e) **Else If** $S^*(\mathbf{X}, \mathbf{H}_i) == -1$
add \mathbf{H}_i to list \mathcal{L}_{OUT}
 - (f) **Else If** $S^*(\mathbf{X}, \mathbf{H}_i) == 0$
 - (i) **If** Dimension ($\mathbf{H}_i > \epsilon_h$)
Bisect \mathbf{H}_i into $\mathbf{H}_i(1)$ and $\mathbf{H}_i(2)$
Add $\mathbf{H}_i(1)$ and $\mathbf{H}_i(2)$ to list \mathcal{L} .
 - (ii) **Else If** Dimension($\mathbf{H}_i \leq \epsilon_h$)
Add \mathbf{H}_i to \mathcal{L}_B .
 - (iii) **End If**
 - (f) **End If**
 - 4 **End While**

if interval (\mathbf{X}, \mathbf{H}) forms an inner, boundary, or outer box to the constraints, respectively. In this specific task of verifying whether a given workspace \mathbf{X} can be achieved given a constraint of joint displacement $[q_{min}, q_{max}]$, function $S(\mathbf{x}, \mathbf{h})$ calculates the inverse kinematics of the manipulator, and returns:

- -1 if $q_i > q_{max}$ or $\bar{q}_i < q_{min}$ for at least one of the joints and at least one pose in (\mathbf{X}, \mathbf{H}) ,
- 1 if $q_i \geq q_{min}$ and $\bar{q}_i \leq q_{max}$ for all joints and all poses of (\mathbf{X}, \mathbf{H}) ,
- 0 if $(q_i < q_{min}$ and $\bar{q}_i > q_{min})$ or $(q_i < q_{max}$ and $\bar{q}_i > q_{max})$ for at least one of the joints and at least one pose in (\mathbf{X}, \mathbf{H}) .

B. Constraint Satisfaction in Design Parameter Space

To search for solutions in the design parameter space \mathbf{H} such that all constraints $\mathbf{C}(\mathbf{X}, \mathbf{H})$ are satisfied for all points $\mathbf{x} \in \mathbf{X}$, two (nested) branch-and-bound loops are required to search both the \mathbf{X} and \mathbf{H} spaces. The algorithm then outputs the regions in the design parameter space ($\mathbf{h} \in \mathbf{H}$) that are certified to form serial chain mechanisms where all the given constraints \mathbf{C} are satisfied in the desired workspace $\mathbf{x} \in \mathbf{X}$.

A branch-and-bound loop is carried out to obtain the design parameters that produces inner and outer solutions. A function $S^*(\mathbf{X}, \mathbf{H}_i)$ is defined such that it returns 1,0,-1, if \mathbf{H}_i forms an inner, boundary, or outer boxes to the design constraints. The branch-and-bound loop that searches the design parameter space calls the function $S^*(\mathbf{X}, \mathbf{H}_i)$ to evaluate the design parameter space \mathbf{H} against the given constraints. This is summarised in Table I.

Function $S^*(\mathbf{X}, \mathbf{H}_i)$ is in turn constructed as another branch-and-bound loop which assigns \mathbf{H}_i to an inner, outer, or boundary box - given the required workspace \mathbf{X} - by calling function $S(\mathbf{X}, \mathbf{H}_i)$. Within iteration i of $S^*(\mathbf{X}, \mathbf{H}_i)$, where a box \mathbf{H}_i out of the design parameter is evaluated, a

TABLE II

SUMMARY OF ALGORITHM FOR FUNCTION $S^*(\mathbf{X}, \mathbf{H}_i)$ USED IN THE ALGORITHM IN TABLE I

- Input:** \mathbf{X}_i and \mathbf{H}_i
Output: 1, 0, -1, (classifying (\mathbf{X}, \mathbf{H}) as inner, boundary, and outer boxes, respectively).
- 1 Initialise empty lists \mathcal{L}_B .
 - 2 Initialise list \mathcal{L} containing initial pose \mathbf{X} .
 - 3 **While** (\mathcal{L} not empty)
 - (a) Extract interval pose \mathbf{X}_i from \mathcal{L}
 - (b) Evaluate function $S(\mathbf{X}_i, \mathbf{H}_i)$,
 - (c) **If** $S(\mathbf{X}_i, \mathbf{H}_i) == -1$
Return (-1)
 - (c) **Exit function** $S^*(\mathbf{X}, \mathbf{H}_i)$
 - (e) **Else If** $S(\mathbf{X}_i, \mathbf{H}_i) == 0$
 - (i) **If** Dimension ($\mathbf{X}_i > \epsilon_x$)
Bisect \mathbf{X}_i into $\mathbf{X}_i(1)$ and $\mathbf{X}_i(2)$
Add $\mathbf{X}_i(1)$ and $\mathbf{X}_i(2)$ to list \mathcal{L} .
 - (ii) **Else If** Dimension($\mathbf{X}_i \leq \epsilon_x$)
Add \mathbf{X}_i to \mathcal{L}_B .
 - (iii) **End If**
 - (f) **End If**
 - 4 **End While**
 - 5 **If** (\mathcal{L}_B is empty)
Return 1
 - 6 **Else**
Return 0
 - 7 **End If**

search in the workspace \mathbf{X} is carried out using the previously defined function $S(\mathbf{X}, \mathbf{H}_i)$. Bisection is performed to refine the search on a smaller region of the workspace (terminated at a threshold value ϵ_h). If $S(\mathbf{x}, \mathbf{H}_i) == -1$ for any point \mathbf{x} , then the set of design parameters \mathbf{H}_i is an outer box. If no outer box is found while executing $S(\mathbf{x}, \mathbf{H}_i)$, but $S(\mathbf{x}, \mathbf{H}_i) == 0$ for any point within \mathbf{X} , then \mathbf{H}_i is a boundary box. If no outer and no boundary box is found (and termination threshold is reached), then \mathbf{H}_i is an inner box. The algorithm of function $S^*(\mathbf{X}, \mathbf{H}_i)$ is summarised in Table II.

Any point $\mathbf{h} \in \mathbf{H}$ contained in the resulting inner boxes of design parameters is certified to constitute a manipulator where all points in its workspace satisfy the given constraints. Having interval design parameters means that any bounded uncertainty known to affect the design parameters is also included in the consideration when certifying the boxes. The design solution resulting from this process is therefore robust with respect to such uncertainties. In this paper, the joint limits given the required end-effector workspace is given as the design constraint. To evaluate this constraint, it is necessary to solve the inverse kinematics of the manipulator within the $S(\mathbf{X}, \mathbf{H}_i)$ function.

IV. SOLVING INVERSE KINEMATICS OF SERIAL MANIPULATORS IN INTERVAL ANALYSIS

The limit of displacements for the manipulator joints is selected as the design constraints in this paper as it is one of the most common in design problems. As joint displacement is involved given the end-effector pose requirements, it is necessary to be able to evaluate the inverse kinematics relations of the manipulator within $S(\mathbf{X}, \mathbf{H})$.

Conventionally, the inverse kinematics of non-redundant serial manipulators are solved symbolically to obtain the closed-form solutions. However, explicit solution to the

inverse kinematics is specific to the manipulator design/topology. When the topology of the mechanism is unknown, as is the case in this paper (e.g. during design process or in the case of reconfigurable robots), explicit inverse kinematics does not exist. Hence it is necessary to construct an inverse kinematics algorithm that is general enough to be able to evaluate the performance of all possible topologies within a specified window of design parameters. In our algorithm, the forward kinematic relationship is utilised as kinematic constraints in establishing the inverse kinematic solutions.

The kinematic relationship between the joint and task space displacement of a serial manipulator can always be expressed in the forward kinematic form of:

$$\mathbf{x} = f_{FK}(\mathbf{q}) = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \dots {}^{N-1}\mathbf{T}_N, \quad (3)$$

where \mathbf{x} is the task space pose of the end-effector, \mathbf{q} is the vector containing joint space displacement (q_1, q_2, \dots, q_N) , ${}^{i-1}\mathbf{T}_i$ is the transformation matrix from link $i-1$ to link i , and N denotes the total number of joints / articulated links in the serial manipulator. Solving for \mathbf{q} through interval analysis yields the N element joint displacement (interval) vector \mathbf{Q} for the M dimensional interval box containing the range of the end-effector pose variable \mathbf{X} . M is the number of degrees of freedom in task space that the manipulator possesses.

As previously mentioned, the evaluation of interval functions results in the overestimation due to multiple occurrences of a single variable. Therefore, evaluating the forward kinematics would result in an overestimation of the resulting workspace. Hence, to certify that a specific workspace is achievable by a given joint displacement range, it is necessary to obtain the inverse kinematics - which yields an overestimated joint space displacement required to achieve the given workspace. If the overestimated joint displacement required is within the joint limits, then it is guaranteed that the required end-effector workspace is achievable - although it is an underestimation of the true achievable workspace. A major challenge is to obtain the minimum overestimation possible for an accurate result.

A. Interval Newton Method

The inverse kinematics in this paper is solved using the modified Interval Newton method, although this is not the only method. Interval Newton method is a solving method based on evaluation of a function about the center point of the intervals [11]. It is based on a first order Taylor series expansion, where:

$$F(Q) = f(q_m) + J_q(Q) \cdot (Q - q_m) \quad (4)$$

where Q is the interval extension of variable q , (i.e. $q \in Q$), $F(Q)$ is the interval extension of function $f(q)$, $J_q(Q)$ is the derivative of function $f(q)$ with respect to variable q , evaluated at interval Q , and q_m is the centre of interval Q .

In solving for the inverse kinematics problem, let the given end-effector interval pose be described as \mathbf{X} , the interval joint displacement vector as \mathbf{Q} , and the forward kinematics of the serial manipulator as $f_{FK}(\mathbf{q})$. Solving the inverse

kinematics solution becomes a problem of solving for joint displacement interval Q , given the task space displacement X , with the following constraints [12]:

$$F_{FK}(\mathbf{Q}) - X = 0; \quad (5)$$

$$f_{FK}(\mathbf{q}_m) - X + \mathbf{J}(\mathbf{Q}, \mathbf{q}_m) \cdot (\mathbf{Q} - \mathbf{q}_m) = 0; \quad (6)$$

The joint displacement intervals were initialised at the allowable joint displacement limits, and the middle values \mathbf{q}_m are calculated accordingly. Solving for $(\mathbf{Q} - \mathbf{q}_m)$ through the linear equations:

$$X - f_{FK}(\mathbf{q}_m) = \mathbf{J}(\mathbf{Q}, \mathbf{q}_m) \cdot (\mathbf{Q} - \mathbf{q}_m); \quad (7)$$

in iterative manner, the interval inverse kinematic solutions of the serial manipulator is obtained.

B. Bisection on the Joint (Solution) Space

It is well known that inverse kinematics problem can possess multiple solutions, even for well constrained problems (non-redundant serial manipulators). For example, a planar two-link RR manipulator can have an *elbow up* and *elbow down* joint displacement solutions for one end-effector position. When such case exists, the consistency filtering techniques, such as 2B and 3B, will converge only to the outer bounds of the multiple solutions, instead of to the individual sets of solution. A quick *bisection* procedure on the interval joint variables \mathbf{Q} is therefore necessary. The summary of the procedure is described in Table III. In the procedure, each set of possible solution goes through a bisection process, which splits the interval along joint displacement interval variable Q_i , for $i = 1, \dots, N$, where N is the number of joints (steps 6(b) in Table III). Each time a set of possible solution is bisected, the algorithm tries to solve for the inverse kinematics (steps 1(b)-(d)). If the bisected branch produces a solution, then an inverse kinematic solution within the joint limit is found. If it is confirmed to have no solution, then the branch of possible solution is eliminated. If the result is not sufficient to conclude either way, then the branch is kept for bisection on subsequent joint variable (steps 1(e)-(j)).

The essential point about the bisection process in the solution space (\mathbf{Q}) is to be able to isolate the individual sets of solutions, such that the interval inverse kinematic algorithm is able to converge. When there are two sets of solutions, for instance, splitting the solution space in between the solutions would tremendously increase the efficiency of the solving or filtering techniques. However, not knowing where the solutions are, it is possible that the bisection is carried out at the point of solution / in the range that contains the solution. Hence, the proposed procedure also checks whether or not the bisection process produces two separate solutions (step 1(f)-(i) in Table IV). If no separate solutions are found after bisection, the set of joint displacement \mathbf{Q}_i is returned as the *Hull* (i.e. the union box) of the two bisected halves.

Variable *SolFound* in summary table (Table IV) denotes the state of whether or not inverse kinematics solution exists

TABLE III

SUMMARY OF ALGORITHM $IK(\mathbf{X}_i, \mathbf{H}, \mathcal{F})$: INTERVAL INVERSE KINEMATICS FOR A SERIAL MANIPULATOR WITH JOINT LIMIT CONSTRAINTS

- Input:** \mathbf{X}_i, \mathbf{H} , and \mathcal{F}
Output: 1, 0, -1, (representing inner, boundary, and outer solutions, respectively).
- 1 Initialise empty list for solutions \mathcal{L}_S .
 - 2 Initialise initial estimates for the joint displacements $\mathbf{Q}^0 = (Q_1^0, Q_2^0, \dots, Q_N^0)$ at the joint limits.
 - 3 $\mathbf{Q}' = \text{Solve}Q(\mathbf{X}_i, \mathbf{H}, \mathbf{Q}, \mathcal{F})$
 - 4 **If** (solution \mathbf{Q}' does not exist)
 - (a) Return (-1)
 - (b) Exit function $IK(\mathbf{X}_i, \mathbf{H}, \mathcal{F})$
 - 5 **Else If** (solution \mathbf{Q}' exists)
 - (a) Load \mathbf{Q}' onto \mathcal{L}_S
 - (b) Return (1)
 - (c) Exit function $IK(\mathbf{X}_i, \mathbf{H}, \mathcal{F})$
 - 6 **Else**
 - (a) SolFound = 0; $i = 1$;
 - (b) **While** $i \leq N$ AND SolFound $\neq 1$
 - SolFound = Bisect-and-Solve($\mathcal{L}_{IN}, \mathcal{L}_{OUT}, i$);
 - $\mathcal{L}_{IN} = \mathcal{L}_{OUT}$;
 - (c) **End While**
 - (d) Return (SolFound)
 - 7 **End If**
 - 8 Bisect-and-Solve is detailed in Table IV.

for a box \mathbf{X}_i, \mathbf{H} . SolFound = 1 when an inverse kinematic solution is found (step 1(e), Table IV), SolFound = -1 when all possible bisections of the joint space \mathbf{Q}_i are not solutions to the inverse kinematics (step 3, Table IV), or SolFound = 0 otherwise (step 4, Table IV). Please note also that the input \mathcal{F} in algorithm $IK(\mathbf{X}_i, \mathbf{H}, \mathcal{F})$ in Table III is the forward kinematics relationship of the manipulator, used as a constraint in obtaining the inverse kinematics solutions.

V. IMPLEMENTATION ON AN EXAMPLE

As an example, the algorithm described above is implemented to search for a set of design parameters such that a desired end-effector workspace is reachable by the resulting manipulator, given joint limit of $\pm 30^\circ$ for all the revolute joints. The inverse kinematic function $IK(\mathbf{X}_i, \mathbf{H})$ is defined, taking place of function $S(\mathbf{x}, \mathbf{H})$. It returns 1,0,-1, if \mathbf{X}_i yields an inner, boundary, and outer box, respectively, to the inverse kinematics, given the allowable joint displacement limits. Solving the inverse kinematics through the Interval Newton method can be done by enforcing the kinematic relationship of the joint and task spaces. In the summary of inverse kinematic algorithm in Table III, the solving technique is labelled as $\text{Solve}Q$, which is the Interval Newton method, as described earlier in Section IV.

Specific to the application addressed in this paper, namely the modular robotic system for endoluminal surgery, there exist simplifications to the possible topologies that can be achieved through the combinations of the proposed robotic modules. Due to the limitation in the paper length, an example is given in this paper specific to the proposed application. The example of implementation of a general case serial manipulator design algorithm through interval constraint satisfaction can be found in [12].

As an example, the suitable topology of a serial chain mechanism made up of three modules of the surgical re-

TABLE IV

SUB-PROCEDURE BISECT-AND-SOLVE FOR ALGORITHM IN TABLE III.

- Sub-Procedure:
 SolFound = **Bisect-and-Solve**($\mathcal{L}_{IN}, \mathcal{L}_{OUT}, i$)
Input: \mathcal{L}_{IN} : list of joint displacement \mathbf{Q} which possibly contain solutions.
 i : index of joint displacement Q_i to be bisection this round.
Output: SolFound (1 solution is found, -1 no solution, and 0 when it is not clear).
 \mathcal{L}_{OUT} list of joint displacement \mathbf{Q} which possibly contain solutions for next round of bisections.
- 1 **While** (\mathcal{L}_{IN} not empty AND SolFound==0)
 - (a) Extract \mathbf{Q} from \mathcal{L}_{IN} .
 - (b) Bisect Q_i to $Q'_i(1)$ and $Q'_i(2)$ to form $\mathbf{Q}'(1)$ and $\mathbf{Q}'(2)$
 - (c) $\mathbf{Q}''(1) = \text{Solve}Q(\mathbf{X}_i, \mathbf{H}, \mathbf{Q}'(1), \mathcal{F})$
 - (d) $\mathbf{Q}''(2) = \text{Solve}Q(\mathbf{X}_i, \mathbf{H}, \mathbf{Q}'(2), \mathcal{F})$
 - (e) **If** $\mathbf{Q}''(1)$ or $\mathbf{Q}''(2)$ are inverse kinematic solutions
 - Return SolFound = 1;
 - exit *Bisect – and – Solve*.
 - (f) **Else If** $\mathbf{Q}''(1)$ and $\mathbf{Q}''(2)$ do not form separate solutions
 - Push Hull($\mathbf{Q}''(1), \mathbf{Q}''(2)$) onto \mathcal{L}_{OUT} .
 - (g) **Else If** both $\mathbf{Q}''(1)$ and $\mathbf{Q}''(2)$ do not exist
 - This entry does not contain solution.
 - (h) **Else If** $\mathbf{Q}''(1)$ exists
 - Load $\mathbf{Q}''(1)$ onto \mathcal{L}_{OUT} .
 - (i) **Else If** $\mathbf{Q}''(2)$ exists
 - Load $\mathbf{Q}''(2)$ onto \mathcal{L}_{OUT} .
 - (j) **End If**
 - 2 **End While**
 - 3 **If** no entries contain inverse kinematic solution:
 - Return SolFound = -1;
 - 4 **Else**
 - Return SolFound = 0;
 - 5 **End If**

configurable robot is to be determined. A homogeneous reconfigurable robot scheme is used in this example, meaning that all modules used in forming the robot are identical. Each module is designed with an actuated end capable of producing 1DOF rotation within $\pm 30^\circ$ and a no-actuated end capable of fixed docking to another modules. The actuated end is designed to dock with the actuated end of another module, with docking orientation of 90° , resulting in 2DOF rotation with common centres of rotation and perpendicular axes of rotation. This results in a very compact 2DOF joint design constructed out of two identical modules (See Fig. 1).

Due to the symmetry of the modules and the repetitive nature of the kinematic chain arising from combining identical modules used to form the chain, there are only finite number of possible topologies to be obtained. Furthermore, kinematic parameters, such as link length and offset angles, can not be arbitrary values. For example, link lengths can only be multiples of the module length and offset angle between modules can only be 0° and 90° . Out of three homogeneous modules, it turned out that there can only be 2 possible topologies, as shown in Fig. 3, where the only variation is the docking orientation at point A (a docking of non-actuated ends). A surgical tool is assumed to be attached to the end of the chain.

Out of the two possible topologies (H_1 and H_2), topology H_1 is evaluated to demonstrate the workspace reachable by the robot of such topology given a joint limit of $\pm 30^\circ$. Another constraint is added to evaluate the reachable workspace only through an “elbow down” configuration, i.e. only for positive values of joint q_3 . This is done because for the

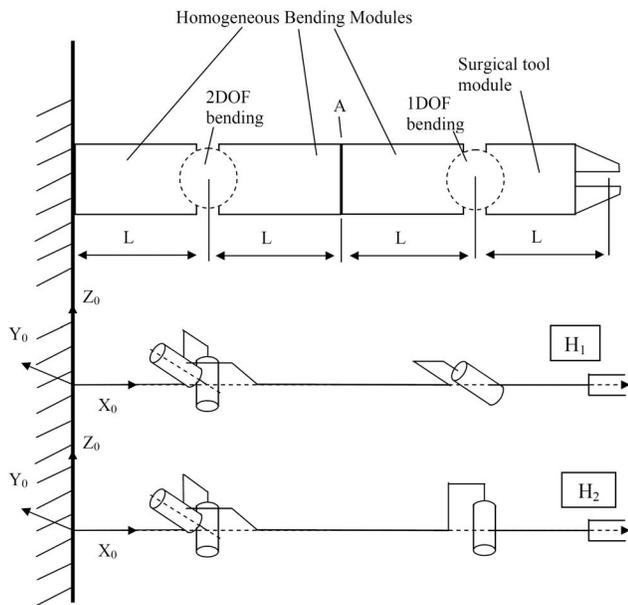


Fig. 3. Possible kinematic chains when using three homogeneous bending modules.

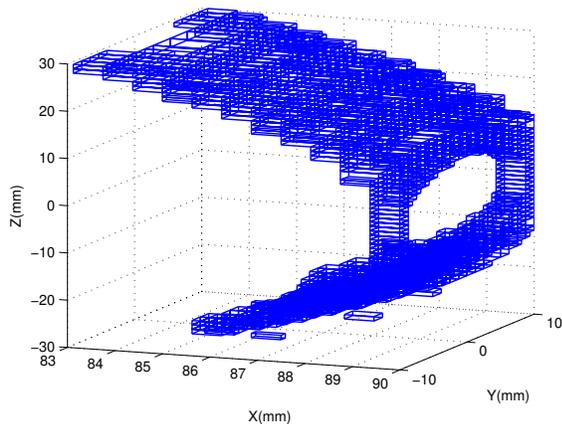


Fig. 4. Reachable workspace by topology H_1 , given joint limit of $[-30^\circ, 30^\circ]$

manipulator to cross from the “elbow up to “elbow down” configuration, it would encounter singular configuration. The subset of the workspace that is achievable by topology H_1 given the joint limits of $[-30^\circ, 30^\circ]$ calculated through the proposed algorithm can be displayed on a 3D plot as shown in Fig. 4. Note that because inverse kinematics is utilised, then workspace associated with singular configurations has not been certified as inner boxes.

If the desired workspace is defined as $\mathcal{W} = ([88, 90], [0, 4], [12, 15])^T mm$, then executing the complete algorithm (as detailed in Table I) to evaluate both topologies (H_1 and H_2) will yield an inner box for topologies H_1 and H_2 . This means both topologies are capable of achieving the given desired workspace.

VI. CONCLUSIONS

Motivated by the need to automatically generate a suitable manipulator for a surgical procedure, a serial chain design algorithm with certified satisfaction of design and task constraints is proposed and presented. In the process, an effective interval inverse kinematics algorithm for serial chain, without explicit inverse kinematic expressions was also proposed and presented. The proposed algorithm is also demonstrated to be effective in obtaining all variations of the kinematic topology of a serial manipulator such that the given constraints are satisfied in all points within the desired workspace. The design constraints were derived to describe the boundary of operation of the surgical modules. Improvement can be made on the computational efficiency of the algorithm in admitting or rejecting the various interval boxes, such as by exploring an efficient interval representation of serial kinematic transformations, hence reducing the necessity for further bisections in the process. As this study was carried out for endoluminal surgery through reconfigurable modular robots, future work also includes improving and adapting the proposed strategy to the challenging biomedical environment and requirements to compute the most suitable topology for the task.

ACKNOWLEDGEMENTS

The authors thank Professor Alfred Cuschieri (medical consultancy) and EU NEST project No.015653.

REFERENCES

- [1] “The ARES (Assembling Reconfigurable Endoluminal Surgical system),” Project Website <http://www.ares-nest.org>, 2006.
- [2] O. Ma and J. Angeles, “Optimum design of manipulators under dynamic isotropy conditions,” *Procs. IEEE Int’l. Conf. Rob. and Autom.*, vol. 1, pp. 470–475, 1993.
- [3] A. Bowling and O. Khatib, “Design of non-redundant manipulators for optimal dynamic performance,” in *Procs. Int’l. Conf. Adv. Rob.*, Monterey, CA USA, 7–9 July 1997, pp. 865–872.
- [4] M. Yim *et al.*, “Modular self-reconfigurable robot systems [Grand Challenges of Robotics],” *IEEE Rob. and Autom. Mag.*, vol. 14, no. 1, pp. 865–872, March 2007.
- [5] F. Masahiro, H. Kitano, and K. Kageyama, “Reconfigurable robot platform,” *Rob. and Auton. Syst.*, vol. 29, no. 2, pp. 119–132, Nov 1999.
- [6] Y. Fei and X. Zhao, “Design and dock analysis for the interactive module of a lattice-based self-reconfigurable robot,” *Rob. and Auton. Syst.*, vol. 55, no. 2, pp. 87–95, Feb 2007.
- [7] M. Yim, D. Duff, and K. Roufas, “Polybot: a modular reconfigurable robot,” *Procs. IEEE Int’l. Conf. Rob. and Autom.*, vol. 1, pp. 514–520, 2000.
- [8] K. Stoy, W. Shen, and P. Will, “A simple approach to the control of locomotion in self-reconfigurable robots,” *Rob. and Auton. Syst.*, vol. 44, no. 3–4, pp. 191–199, Mar 2003.
- [9] K. Harada, E. Susilo, N. Ng Pak, A. Menciassi, and P. Dario, “Design of a bending module for assembling reconfigurable endoluminal surgical system,” in *Procs. the 6th Conf. Intl. Soc. Gerontechnology (ISG’08)*, Pisa, Italy, 4–7 June 2008, pp. ID–186.
- [10] R. Moore, *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ., 1966.
- [11] E. Hansen and G. Walster, *Global Optimization using Interval Analysis*, 2nd ed. Marcel Dekker, NY., 2004.
- [12] D. Oetomo, D. Daney, and J.-P. Merlet, “Design strategy of serial manipulators with certified constraint satisfaction,” *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 1–11, Feb 2009.