Neural Networks 107 (2018) 34–47

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2018 Special Issue

Deeply-learnt damped least-squares (DL-DLS) method for inverse kinematics of snake-like robots

Olatunji Mumini Omisore [a,b,c,1], Shipeng Han [a,1], Lingxue Ren [a], Ahmed Elazab [d,e], Li Hui [a], Talaat Abdelhamid [f], Nureni Ayofe Azeez [g,h], Lei Wang [a,c,*]

[a] Research Centre for Medical Robotics and Minimally Invasive Surgical Devices, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen 518055, China
[b] Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China
[c] CAS Key Laboratory for Health Informatics, Shenzhen Institutes of Advanced Technology, Shenzhen 518055, China
[d] Computer Science Department, Misr Higher Institute for Commerce and Computers, Mansoura City, Egypt
[e] School of Biomedical Engineering, Health Science Center, Shenzhen University, Shenzhen 518060, China
[f] Physics and Mathematical Engineering Department, Faculty of Electronic Engineering, Menoufiya University, Menouf 32952, Egypt
[g] School of Computer Science and Information, North-West University, Vaal Triangle Campus, South Africa
[h] Department of Computer Sciences, University of Lagos, Akoka, Lagos State, Nigeria

HIGHLIGHTS

- Deeply-learnt damped least squares method is proposed for inverse kinematics of snake-like robots.
- The proposed method has a reachability measure of 91.59% with error threshold of 0.01 mm.
- The method is computationally efficient, fast, and maneuvers singular points, simultaneously.
- Validation against popular methods used for inverse kinematics shows the proposed method is better.
- Deep learning in neural networks reduces iterations required for convergence of each target point.

ARTICLE INFO

Article history:
Available online 23 August 2018

Keywords:
Radiosurgical robots
Snake-like robots
Inverse kinematics
Jacobian matrix
DLS methods
Deep neural network

ABSTRACT

Recently, snake-like robots are proposed to assist experts during medical procedures on internal organs via natural orifices. Despite their well-spelt advantages, applications in radiosurgery is still hindered by absence of suitable designs required for spatial navigations within clustered and confined parts of human body, and inexistence of precise and fast inverse kinematics (IK) models. In this study, a deeply-learnt damped least squares method is proposed for solving IK of spatial snake-like robot. The robot's model consists of several modules, and each module has a pair of serial-links connected with orthogonal twists. For precise control of the robot's end-effector, damped least-squares approach is used to minimize error magnitude in a function modeled over analytical Jacobian of the robot. This is iteratively done until an apt joint vector needed to converge the robot to desired positions is obtained. For fast control and singularity avoidance, a deep network is built for prediction of unique damping factor required for each target point in the robot's workspace. The deep network consists of 11 x 15 array of neurons at the hidden layer, and deeply-learnt with a huge dataset of 877,500 data points generated from workspace of the snake robot. Implementation results for both simulated and actual prototype of an eight-link model of the robot show the effectiveness of the proposed IK method. With error tolerance of 0.01 mm, the proposed method has a very high reachability measure of 91.59% and faster mean execution time of 9.20 (±16.92) ms for convergence. In addition, the method requires an average of 33.02 (±39.60) iterations to solve the IK problem. Hence, approximately 3.6 iterations can be executed in 1 ms. Evaluation against popularly used IK methods shows that the proposed method has very good performance in terms of accuracy and speed, simultaneously.

* Corresponding author at: Research Centre for Medical Robotics and Minimally Invasive Surgical Devices, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen 518055, China.

E-mail addresses: omisore@siat.ac.cn (O.M. Omisore), sp.han@siat.ac.cn (S. Han), lx.ren@siat.ac.cn (L. Ren), ahmedelazab@szu.edu.cn (A. Elazab), hui.li1@siat.ac.cn (L. Hui), talaat@siat.ac.cn (T. Abdelhamid), 30041708@nwu.ac.za (N.A. Azeez), wang.lei@siat.ac.cn (L. Wang).

[1] Authors with equal contributions.

https://doi.org/10.1016/j.neunet.2018.06.018
0893-6080/

1. Background study

Medical robotics is becoming a safe and convenient assistive platform for surgery and radiotherapy of internal organs in confined areas of human body (Hadjerci et al., 2016). Conventionally, large incisions are required for proper visualization of such organs. In surgery, medical robots, such as da Vinci and Zeus surgical systems, have been proposed and used for minimally invasive procedures. Movements made by medical experts are mimicked by such systems to facilitate complex surgeries with minimal invasion (Omisore, Han, Ren, & Wang, 2016). Similarly, in radiosurgery, oncologists are required to deliver treatment dosage to affected tissues which are sometimes located in hidden areas of the human. Despite the large amounts of morphologic and functional data mostly used in pre-operative planning, experts are, sometimes, unable to deliver it appropriately to targeted organs (Li, Zou, Li, Xie, & Xiong, 2017). Therefore, medical robotic systems are employed for precise delivery of pre-planned doses to intended operative sites (Jayarao & Chin, 2007).

Radiation energy from linear particle accelerator in radiosurgical systems, like Gamma Knife (Hayashi et al., 2013), CyberKnife (Kuo, Yu, Petrovich, & Apuzzo, 2003), can be directed to destroy tumor cells without damaging the healthy organs in the body. The target is bombarded with beams of ionizing radiation made-up of gamma rays, x-rays, or sub-atomic particles like protons. External beam radiotherapy has been a conventional approach for delivering of radiation doses on tumor cells in the human body. However, the non-invasive methods require high positioning accuracy of the radiation beam in order to focus the tumor cells for a long period of time. Furthermore, directing the high-energy rays through the body exposes healthy tissues around the operative site to radiation. Recently, brachytherapy became a common approach for radiosurgery of tumor cells in cervical, prostate, and gastrointestinal areas (Schieda, Malone, Al-Dandan, Ramchandani, & Siegelman, 2014). It involves precise placement of short-range radiation isotopes injected to localize tumor cells in the body. Hence, probability of damages to healthy tissues around the targeted organs is reduced. Nonetheless, brachytherapy is an outpatient treatment modality and more convenient for cervical cancer. Since irradiation only affect tissues within few millimeters of radiation source, prolonged treatment time is required. This can lead to thrombosis of internal organs and may result in urinary and digestive problems (Pieters, De Back, Koning, & Zwinderman, 2009).

In the last three decades, serial-link robots have been developed to assist experts in surgical and rehabilitative procedures (Degani, Choset, Wolf, Ota, & Zenati, 2006; Ren, Omisore, Han, & Wang, 2017; Sardana, Sutar, & Pathak, 2013). Snake-like robots are emerging type of flexible manipulators being proposed for medical aids. They consist of serial links that are interconnected by several rotational joints for serpentine movements, and a linear joint for translation. These redundant robots usually have a non-stationary side where an end-effector is fixed, and its pose is manipulated towards given targets by kinematic models. Snake-like robots are designed as modular structures with several variable-length links and multiple twists (Degani et al., 2006; Ren et al., 2017). Thus, they can exhibit spatial serpentine poses with varying navigational patterns that can be fitted to manipulate objects in confined areas. Furthermore, such designs are useful for obstacle avoidance, kinematic modeling, and singularity avoidance (Degani et al., 2006; Parsa, Daniali, & Ghaderi, 2010; Ren et al., 2017; Sardana et al., 2013; Sheng, Yiqing, Qingwei, & Weili, 2006; Srinivasa, Bhattacharyya, Sundareswara, Lee, & Grossberg, 2012).

Appropriate relationships between joint and Cartesian spaces of a robot are vital to solving its kinematics. This can be in terms of forward kinematics that is, solving for coordinates of a robot's end-effector based on a given joint-vector, or inverse kinematics (IK). The latter involves computation of joint configurations from Cartesian space of the robot. Snake-like robots are capable of several independent spatial articulations. Therefore, they can aid precise and timely passage of radiosurgical tools to desired areas through natural orifice or single-port incision in the human body. A major factor hindering their adoption for interventional procedures is the complex transcendental trigonometric computations required for solving IK of the spatially redundant robot (Ren et al., 2017; Sardana et al., 2013). Joint and Cartesian variables can be mapped into nonlinear models while kinematics problem can be geometrically solved. However, this is not applicable for kinematics of spatially redundant robots. Alternatively, numerical methods have been adopted for solving IKs of serial-link robots (Tchon, 2008; Yahya, Moghavvemi, & Mohamed, 2011). These are based on analyzing the Jacobian of the robot, and solving minimization problems to obtain the best configurations for given target points with an admissible tolerance error.

Deep Learning (DL) is a subfield of machine learning that involves application of neural networks with more than one hidden layer structured to function like human brain. This learning approach is achieved with deep neural architectures made-up of several nonlinear that are layers designed to process multiple levels of nonlinear operations. Recently, DL has demonstrated efficiently in different fields such as medical imaging and robotics (Liu, Liu, Sun, & Fang, 2017a). Applications of DL in robotics have been found in grasp detection (Lenz, Lee, & Saxena, 2015), manipulation in unknown environments (Levine, Wagener, & Abbeel, 2015), and robot perceptions (Liu, Qin, Sun, & Guo, 2017b). IK in snake-like robot is also a highly nonlinear problem which can be approached with the learning modality. Classical neural networks have been adopted to minimize the complexities involved in IK of redundant robots (Parsa et al., 2010; Toshani & Farrokhi, 2014). However, application of such method for precise and timely control of snake-like robots is still lacking. Furthermore, application of deep neural network (DNN) in modeling the high transcendence involved in IKs of snake-like robots is yet to be explored.

In this study, DL is employed to predict the appropriate damping factor needed for solving IK of spatially flexible snake-like robots. An important contribution of DL is its ability to automatically find compact features that can well represent high-dimensional data in a deep network. The proposed method is aimed as a new variant of damped least square (DL-DLS) method, and its novelty lies in prediction of damping factor for precise and fast IK control of snake-like robots. The DL-DLS method gains prediction knowledge by learning nonlinear functions that can be defined from specified features of data points in the robot's workspace. Significant problems in robotics can be logically solved with deep reinforcement learning, a superposition of DNN and reinforcement learning. This learning approach adopts principled frameworks for autonomous control in which the experience gained by a robot from trial-and-error interactions within an environment can be utilized for decision making. To design the DNN, an optimal damping factor is selected manually for data points in the workspace of a snake-like robot by using trial-and-error approach. The selected values are then used to train and validate the deep network. Hence, this method follows *behavioral cloning*, a deep reinforcement learning approach that takes advantage of optimal actions from human experts and supervised learning to guide future decision making process in the robot's environment.

The rest of this paper is organized as follows. Review of related works on IK studies is presented in Section 2. The design of a flexible snake-like robot is presented in Section 3. The proposed DL-DLS method for IK of the robot is given in Section 4. The experimental results and performance validation of the IK method are presented in Section 5. Finally, conclusions of this study and future works are given in Section 6.

2. Related works

Solving IK problems in serial link robots have been based on several methods (Chung, Youm, & Chung, 1994; Elgazzar, 1985; Jamali, Khan, & Rahman, 2011; Kostic, Hensen, de Jager, & Steinbuch, 2002; Kucuk & Bingul, 2014; Makondo, Claassens, Tlale, & Braae, 2012; Ren et al., 2017; Sardana et al., 2013; Sheng et al., 2006; Srinivasa et al., 2012; Tchon, 2008; Toshani & Farrokhi, 2014; Yahya et al., 2011; Yahya, Mohamed, Moghavvemi, & Yang, 2009). According to Omisore, Han, Ren, Zhang, and Wang (2017), these methods can be simply categorized as algebraic and iterative approaches. Algebraic approaches, including closed-form or geometric methods, focus on obtaining exact solutions based on a robot's model. Such methods involve solving complex transcendental equations to obtain a set of joint angles needed to converge the effector to given targets in the workspace. Closed-form methods are applicable to models with, at least, three joint axes having a common intersection. These are classes of industrial robots with simplified structure (Elgazzar, 1985; Kostic et al., 2002; Kucuk & Bingul, 2014). However, it is challenging to obtain exact solutions for snake-like robots due to spatial complexities and insufficient offsets needed to corroborate the twisted joints at consecutive joints of the robots.

Alternatively, Denavit–Hartenberg (DH) parameters of such robots can be geometrically analyzed to obtain suitable IK solutions. Geometric methods are best used for solving forward kinematics problem since only direct substitution of the robot's parameters into the DH matrix is required (Elgazzar, 1985). Nonetheless, geometric methods can be of advantage in solving the IK problem of robotic configurations where exact and iterative methods are found inappropriate. Some geometry-based solutions were proposed in Chung et al. (1994), Jamali et al. (2011), Kostic et al. (2002), Sheng et al. (2006) and Yahya et al. (2009) to solve IK of planar manipulators. Success attained in those studies can be attributed to the fact that geometric models of planar robots are not mathematically complex. However, spatial navigation is very important for surgical robots. Some studies have designed snake-like robots to reach target points with tri-axial coordinates (Degani et al., 2006; Ren et al., 2017). Initially, rotation of first link was made orthogonal to the direction of rotations of other links for spatial reachability, and the IK problem can then be solved in just two steps (Makondo et al., 2012; Omisore, Han, Ren, Zhang, and Wang, 2017; Sardana et al., 2013; Yahya et al., 2011). For practical applications, high articulation and flexibility are important for snake-like robots while trying to reach organs in confined areas via natural orifices or minimal incision. Prototypes of highly articulated robots were presented in Degani et al. (2006), Ren et al. (2017) and Shang et al. (2011). To the best of our knowledge, existing IK models cannot achieve precise and fast control of such prototypes, simultaneously. Omisore, Han, Ren, Zhang and Ivanov (2017) proposed a non-iterative geometric method for solving IK of the robotic configuration in Ren et al. (2017). However, the geometric method losses accuracy with every increase in the robot's links.

Iterative methods have been successfully applied as alternative approach to solving IK of serial-link robots. This category of IK methods includes heuristic and numeric iterative solutions, which have been used for intuitive manipulation of flexible snake-like robots. A popular heuristic IK method is the cyclic coordinate descent (CCD) proposed to solve IK problems by considering one joint at a time (Wang & Chen, 1991). Also, Aristidou and Lasenby (2011) proposed using both forward and backward reaching to solve IK problem in a similar manner to CCD. The latter computes a set of joint angular values required to place an end-effector at given target points, using forward and backward reaching with the objective to minimize error. Both methods require low computational cost per iteration since matrix manipulations are not involved. In addition, both forward and backward reaching in each iteration are executed simultaneous in the latter; hence, it requires reduced number of iterations for convergence. Both methods suffer low convergence for structures with different orientations at each joint, especially, when targeted points are far away from mid-planes of the robot's workspace (Omisore, Han, Ren, Zhang and Ivanov, 2017; Ren et al., 2017).

Numerical methods, which are essentially based on numeric analysis of Jacobian matrix, are another iterative approach to solving IK problems in robotics. These methods involve linear approximations of actual velocities at each joint of a robot with respect to that of its end-effector. Sometimes, the Jacobian is non-invertible at singular points. Wolovich and Elliott (1984) proposed transpose of Jacobian as a numeric technique for solving IK problems of the Stanford Arm. This method can be efficient and fast since matrix inversions are not required. However, many iterations are required for convergence in cases of robots with different orientations at each joint. Moreover, transpose-based methods mostly fail to deliver solutions for rank deficient Jacobians (Pechev, 2008). In other variants, inverse of the Jacobian is approximated to provide more stable, efficient, and singularity robust solutions to IK problems. Moore–Penrose or pseudoinverse of the Jacobian is a common and baseline approach used for approximation of Jacobian inverse (Whitney, 1969). Analogous to the Newton-based approaches, the method involves finding a minimum change in joint-vector (angular values: $\Delta\theta$) required to move a robot's end-effector within the coordinates of a target point at a predefined tolerance value. This method is a better estimate for IK problems, and it is often discussed in literature. However, its performance can be very poor due to unstable configurations near singularities in robot's workspace.

Damped least-square (DLS) methods have been well-known as stabilizer of pseudoinverse for near-singular points. This numerical solution was first used in Wampler (1986) and Wampler and Leifer (1988) for solving IK problems in robotic manipulators. It involves selection of constant damping factor to approximate solution of the Jacobian near-singular points. Nakamura and Hanafusa (1986) extended the basic DLS by adding singular value decomposition (SVD-DLS) to form a singularity robust solution for IK problems. SVD-DLS shares many features of the basic DLS; hence, both methods operate just like traditional pseudoinverse method when the damping factor is very close to zero. Deo and Walker studied the use of DLS at velocity level in Deo and Walker (1992), and later proposed an adaptive non-linear least-squares method for solving IK problems in robotic manipulators in Deo and Walker (1993). Rather than having the DLS ill-behaved, the adaptive method switches to an alternate second-order model when the target position is near singularity. Solutions of IK problem depend on motion of the end-effector caused by entries of its Jacobian. Buss and Kim (2005) proposed a selectively damped least-square (SDLS) method for solving IK problems. The study considered how each joint contributes to relative positions of the end-effector and distance of the end-effector from target. Although SDLS is better than conventional methods in terms of position tracking; however, it is slower due to computation of different singular vectors required to prevent large changes in the joint angles.

Adoption of damping factor in Jacobian based methods has been considered to minimize the effects of singularity problems. Amongst several recent studies, Flacco, de Luca, and Khatib (2012), Flacco and de Luca (2013), Kenwright (2011), and Vargas, Leite, and Costa (2014) have investigated the selection of optimal damping factor such that the end-effectors in a flexible snake-like robot can converge to given target points in a more stable and robust way. However, none of these existing studies is found applicable for medical procedures where accuracy and response time are simultaneously important. In this study, a deeply-learnt damped least square (DL-DLS) method is proposed for very precise and fast IK control of spatially flexible snake-like robots that can be used as assistive platforms for medical procedures.

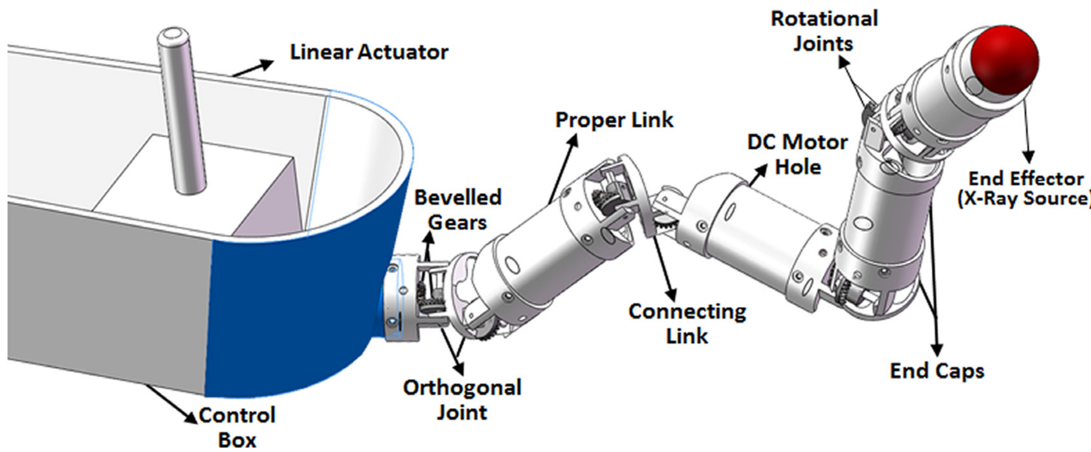

Fig. 1. CAD model of the snake-like robotic model for radiosurgery.

3. Snake-like robotic model for radiosurgery

This section presents the design model of a highly articulated snake-like robot for radiosurgical treatment. The robotic model is designed for effective delivery of pre-planned radiation doses on cancerous tissues around the gastrointestinal area. Mostly, operation of the robot requires steering the connected serial links towards targeted tissues via minimal invasion. For this proposition, the snake-like robotic model in Fig. 1 is designed for minimally invasive procedures. The robot's effector (or last link) is to be placed at an entry point such as the mouth or a single small incision point, and steered towards the targeted part.

The robotic model has series of consecutive orthogonally-paired joint mechanisms for steering the end-effector towards given points in its workspace. A linear actuator is connected with a non-rotary joint to the base link (the first serial link) for advancing or receding the whole robot as specified in a trajectory. Each module of the robot has a pair of links connected by a rotational actuator and also connected with subsequent modules by a second rotational actuator. In other words, the paired-links within each module are a set of *proper link* and *connecting link*. The *proper links* are distinguished by their longer length, which is 53.12 mm, and two end-caps fixed at both ends for connection with the *connecting links*. The latter have a shorter length of 11 mm and are used for interconnection between consecutive proper links in the robot. Both *proper and connecting links* in each module have cylindrical shapes with a diameter of 20 mm. The rotational joint between the modules have orthogonally twisted relationship with the joints within the modules. Hence, each module of the snake-like robot is capable of rotations in 3D Cartesian space. Furthermore, the whole mechanism can be controlled to achieve complex serpentine motions fitted for confined and cluttered regions in human abdominal.

Regarding the other important design details for operation of the snake-like robot, each rotational joint is driven by an actuator coupled with brushless DC Micromotor. The motors are made-up of rotor with permanent magnets and stator with windings. Beveled gears with transmission ratio of 2:1 and reduction ratio of 625:1 are connected with the motors for speed and torque control, respectively. Also, the end-effector can be surgical scissors, needle, or a carbon nanotube for delivery of high doses of pre-planned radiation on cancerous tissues (Fig. 1). By controlling the respective geared motors, orthogonal movement of the joints allows roll, pitch, and yaw rotations at each module of the snake-like robot and thereby the end-effector can reach targeted coordinates with high accuracy.

4. Proposed deeply-learnt DLS IK method

In this section, a deeply-learnt damped least square (DL-DLS) IK method is proposed for precise and fast control of the snake-like model in Fig. 1. The proposed method uses differential Jacobian of the robot's end-effector to solve its IK for given targets. Optimal damping factor, that can converge the end-effector precisely around the target, is rapidly predicted by a DNN. Details of the proposed method are presented in the following subsections:

4.1. Jacobian DLS method

Despite the inexistence of closed-form solutions, analytical Jacobian can be formulated to solve IK problems for spatially dexterous robots. By direct transformation, forward kinematics of the snake-like robot in Fig. 1 can be expressed as in Eq. (1).

$$T_e(\theta) = \begin{bmatrix} R_e(\theta) & P_e(\theta)^T \\ 0^T & 1 \end{bmatrix} \tag{1}$$

The matrix, $T_e(\theta)$, is a cumulative product of the transformation matrices from the base joint of the robot to its end-effector for a given vector of joint variables: $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$. In other words, T_e can be achieved using:

$$^0T_n = \prod_{i=1}^{n} {}^{i-1}T_i \tag{2}$$

Each transformation operation is defined based on the standard DH convention using:

$$^{i-1}T_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $^{i-1}T_i$ is the transformation between two consecutive frames over an angle, θ_i.

Hence, T_e is the final transformation matrix made-up of position vector $P_e^{\Re^{3\times1}}$ and orientation matrix $R_e^{\Re^{3\times3}}$ of the robot based on a given joint-vector, and the matrix vary with respect to the joint-vector θ. Wolovich and Elliott (1984) showed that rate of change of the end-effector's pose depends on rate of change of the joint-vector ($\theta^{n\times1}$).

4.1.1. Jacobian computation

The nonlinear relationship is used to model the rates of change of the end-effector's poses and joint space in form of Jacobian matrix. This approach involves expressing the differential kinematics of linear and angular velocities of the robot's end-effector as a function of its joint-space velocities. With first-order partial derivatives, IK of the snake-like robot can be approximated as:

$$J = \begin{bmatrix} \frac{\partial P_e^i}{\partial \theta_j} & \frac{\partial R_e^i}{\partial \theta_j} \end{bmatrix}^T \tag{4}$$

where $\frac{\partial P_e^i}{\partial \theta_j}$ is the linear velocity of the end-effector's position in ith axis with respect to change in jth joint.

Therefore, the IK problem becomes solving for the best joint-vector $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$ in Eq. (4) such that P_e is approximately equal to a desired target point, P_d. The Jacobian matrix in Eq. (4) is iteratively updated by modifying the angular values at the robot's joints until the end-effector converges to the given target. For an updated joint-vector ($\hat{\theta} := \theta + \Delta\theta$), rate of change of end-effector's velocity to that of jth joint's velocity is:

$$J\left(\hat{\theta}\right) = \left(\frac{\partial K_i^{Se}}{\partial \theta_j}\right) \tag{5}$$

where: ∂K_i^{Se} are axial values of the end-effector's pose in the Cartesian space. $\forall i = \{x, y, z\}\,;\ \forall j = \{1, 2, \ldots, n\}$

Finally, the Jacobian is iteratively solved until the instantaneous error ($\vec{e}$) in Eq. (6) is approximately equal to zero or less than a given threshold.

$$\vec{e} = J\left(\hat{\theta}\right)\Delta\theta \tag{6}$$

4.1.2. Damped least-square method

For a given P_d, it is important to solve Eq. (6) for optimal change ($\Delta\theta$) in joint-vector ($\hat{\theta}$) such that $\vec{e} \approx 0$. Conventionally, an optimal value of $\Delta\theta$ is computed by solving the pseudoinverse of J in Eq. (6) as follows:

$$\Delta\theta = J^{\dagger}\vec{e} \tag{7}$$

where $J^{\dagger}$ is the pseudoinverse of J with an $n \times m$ matrix.

From the basic properties of pseudoinverse, there exist a unique $\Delta\theta$ that minimizes $\left\|J\Delta\theta - \vec{e}\right\|^2$ such that the matrix $J(I - J^{\dagger}J)$ is a projector on the null space of J. Hence, the solution vector $\Delta\theta$ in Eq. (7) can be obtained from any vector (φ) that sets $J\left(I - J^{\dagger}J\right)\varphi = 0$. Thereby, a new relation to the norm of the joints' velocities can be written as a local minimizer as given in Eq. (8). Thus, the optimal change for the joint-vector can be optimized using the right pseudoinverse of J in Eq. (9).

$$\Delta\theta = J^{\dagger}\vec{e} + J\left(I - J^{\dagger}J\right)\varphi \tag{8}$$

$$\Delta\theta = J^T(JJ^T)^{-1}\vec{e} \tag{9}$$

where $\Delta\theta$ is the optimal change in the joint-vector that gives the best solution.

The pseudoinverse, $J^{\dagger} = J^T(JJ^T)^{-1}$ in Eq. (9) is computationally inexpensive but JJ^T is only invertible when J has a full row rank. Furthermore, least squares solution of the pseudoinverse is completely dominated by errors. To solve this problem, DLS (Wampler & Leifer, 1988) can be adopted to invert the differential kinematics in cases of singular positions in the workspace. Thus, Eq. (9) is modified to find minimum-norm of joint speed that minimizes $\left\|J\Delta\theta - \vec{e}\right\|^2$ such that the robot is numerically stable near the singular points. Using DLS in IK problems involves determining a damping value (λ) that minimizes the sum of norms of the solution vector and joint-vector:

$$\Delta\theta_{\lambda} = \underset{\Delta\theta}{argmin}\left\{\left\|J\Delta\theta - \vec{e}\right\|^2 + \lambda^2\left\|\Delta\theta\right\|^2\right\}. \tag{10}$$

For simplicity, the minimization problem can be rewritten as the system of equation given in Eq. (11). Hence, the optimal joint-vector ($\Delta\theta$) can be determined as the unique minimizer for norm of the damped joints' velocities with Eq. (12). Finally, $\Delta\theta$ for the joint-vector in the DLS solution, given in Eq. (12), can be re-expressed with Eq. (13).

$$\Delta\theta_{\lambda} = \underset{\Delta\theta}{argmin}\left\|\begin{pmatrix} J \\ \lambda I \end{pmatrix}\Delta\theta - \begin{pmatrix} \vec{e} \\ 0 \end{pmatrix}\right\| \tag{11}$$

$$\Delta\theta = \begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} \vec{e} \\ 0 \end{pmatrix} \tag{12}$$

$$\Delta\theta = J^T(JJ^T + \lambda^2 I)^{-1}\vec{e} \tag{13}$$

4.2. Prediction of damping factor by deep learning

Optimal convergence of the minimization problem in Section 4.1.2 is very important for effective operations (surgery or radiotherapy) of the robot. A major challenge that hinders the applications of conventional DLS approaches is the determination of optimal value for λ. This weighing factor stipulates the best trade-off between norms of solution vector and the joint vector which can be used to manage the kinematic error such that the robot can simultaneously track given target points in fast and precise manner. Thus, λ must be carefully chosen in Eq. (13). Conventionally, common rules for choosing λ indicate that it is a non-zero small constant value. Moderately, lower magnitude of λ lead to accurate solutions, but often fails around singular points in the workspace. Sometimes, it should be relatively high to obtain feasible solutions. Alternatively, λ can be chosen as an estimate of minimum singular values from SVD of the robot's Jacobian or as a function of its workspace manipulability. However, constant damping factor and existing selective filtering methods are not effective in terms of tracking precision and speed for all data points in the robot's workspace. Therefore, in this study, DNN is designed for predicting optimal damping factor for any target point in the workspace of the snake-like robot.

The architecture of our deeply-learnt feed-forward network is shown in Fig. 2. The network adopts Bayesian regularization back-propagation learning algorithm which uses Levenberg–Marquardt optimization method for updating weight and bias values. This method minimizes combinations of squared errors and weights to produce an optimal model that efficiently generalizes the network. Similarly, it stores a large amount of input–output mapping relationships for efficient prediction.

The DNN architecture has an input layer with four neurons. These neurons accept coordinates ($P_t^x,\ P_t^y,\ P_t^z$) of a target point (P_t) and the norm ($\|P_t\|$) of P_t from an initial point. The four input variables ($P_t^x,\ P_t^y,\ P_t^z,\ \|P_t\|$) are passed through the four neurons at the input layer such that each variable is handled by a unique neuron. The input neurons are connected to the hidden layer so that all possible combinations of the input variables can be utilized in the network. The four input variables can form a maximum of fifteen different combinations which represent the number of neurons at the hidden layers. With this structure, each combination is processed in a unique neuron of the hidden layer. To better handle the non-linear relationship between the input variables ($P_t^x,\ P_t^y,\ P_t^z,\ \|P_t\|$) and the output variable (λ_t), all neurons at each lth hidden layer are *fully-connected* to those in the $l+1$th layer. Since the network would be built with a huge, highly diverse, and nonlinear dataset, a deeply-learnt network with several hidden layers is required for training process. This deep and fully-connected structure at the hidden layer is adopted to enhance the network's learning capability. To achieve this, the input data received at each jth neuron of an $l-1$th hidden layer is processed

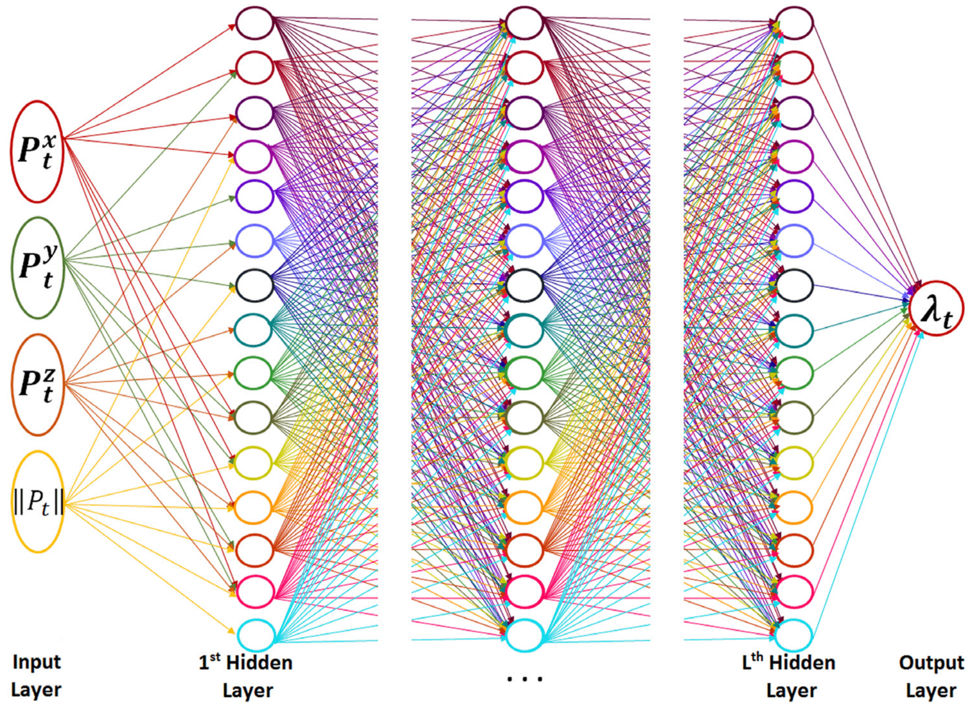

Fig. 2. DNN architecture for prediction of damping factor.

by computing the weighted sum of all connections to the neuron added with the bias, as expressed in Eq. (14).

$$\aleph_j^l = \sum_j w_{jk}^l \times x_j^l + \beta_j^l \tag{14}$$

where w_{jk}^l is the connection weight from kth neuron in the $l-1$th layer to the jth neuron in lth layer; x_j^l is the input data received by jth neuron of lth layer; β_j^l is the bias of jth neuron in lth layer.

Activation of each neuron is calculated by using the sigmoid transfer function defined in Eq. (15). This is done successively in all hidden layers until weighted values get to the output layer. The single neuron in this layer emits an optimal λ value as the network's prediction for a target point. This deep structure is designed to ensure consistent and optimal prediction of λ value which is an important factor for precise and fast IK computation of the snake-like robot.

$$\sigma\left(\aleph_j^l\right) = \frac{1}{1 - e^{-\aleph_j^l}} \tag{15}$$

To train the deep network, data points in the snake-like robot's workspace are computed using Eq. (3) and set into Eq. (13) to generate appropriate input–output dataset. During training, the backpropagation algorithm iteratively process tuples of the dataset and compares the predicted damping value with the actual value in the dataset. For each training tuple, the weight and bias at each neuron are modified to minimize the network's prediction error. This is done by propagating the error values from the output layer backward to the first hidden layer. Error at the output layer (ε^o) is calculated using:

$$\varepsilon_k^o = -\eta \times \frac{\partial \zeta}{\partial\left(\Sigma\right)} \tag{16}$$

where ε_k^o is the error propagated from output layer, η is the learning rate, a positive constant value, $\partial\zeta$ is the cost function at the layer, and $\partial\left(\Sigma\right)$ is measure of error that is computed as $\Sigma = \sigma\left(\aleph_k^o\right)$ at the output layer.

Finally, connection weights in the network are updated base on value of ε_k^o while the values predicted at the output layer are evaluated with new weights. These processes continue until the difference between the actual and predicted damping values in the training dataset is insignificant. Our design of the DNN has two essential advantages. First, the non-fully connection between the input layer and the first hidden layer can ensure that all possible combinations of the input variables are utilized in deciding the damping factor for any given target point in the robot's workspace. Activations with full connection between these two layers may act as a sufficient statistic, but it will require additional computational cost and representation power. Furthermore, this will lead to information overload which will affect the generalization of the network during training, and thereby, causing training step to take longer time. Second, although full connection is avoided between the input layer and the first hidden layer, but it is adopted between all hidden layers, and between the last hidden and output layers. This is to better handle the non-linear relationships that could exist between the input variables (P_t^x, P_t^y, P_t^z, $\|P_t\|$) and the output variable (λ_t). Furthermore, this can enhance the deep network with learning capability to suitably predict unique damping factors for different regions of the robot's workspace.

5. Experimental results

The performance of the proposed IK method is validated on the eight-link robotic model shown in Fig. 1. The snake-like robot is simulated using the Matlab Robotics Toolbox (Corke, 2011). Actual values of the robot's prototype, explained in Section 3, are used for the simulation. We assess the proposed IK method using a simulation of the robot so as to separate the effects of backlashes in the beveled gear at joints of the actual robot from the kinematics error inherent with the IK method. The method is also implemented in Simulink (MathWorks® Inc.) for easy communication with the Robotics Toolbox. The graphical implementation of the proposed IK method is shown as Fig. 3. It should be noted that the eight-link module is capable of spatial rotations in eight orthogonal planes

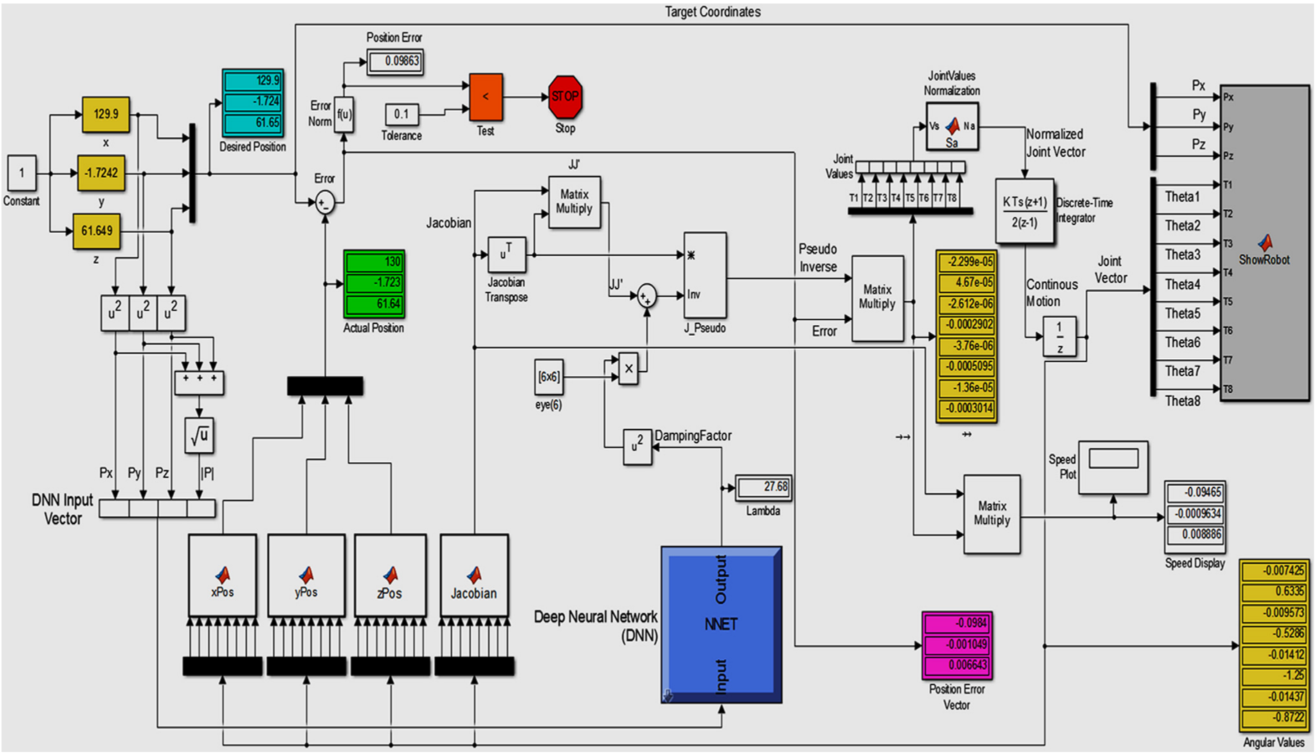

Fig. 3. Graphical implementation of the proposed DL-DLS IK method in Simulink.

without offset at any joint. To the best of our knowledge, there is no method that can accurately solve the IK of such models for the purpose of real-time operations required for all data points in the workspace of such robots. Upon successive computations, the joint vectors obtained are passed to the robotic model for driving tip of the last link to desired targets.

5.1. Implementation of DNN

The deep network, described in Section 4, is implemented with Neural Network Toolbox in Matlab 8.3 on a Lenovo M4380 desktop Computer with Intel® duo processor of Core i3-3420 (2.40 GHz each). The manipulable workspace of the eight-link robot is generated by combining possible values of the joint-vector at an angular interval of 0.32 rad. On the other hand, the joint-vectors are taken as input to Eq. (3). To obtain the best damping factors for training the network, lambda in Eq. (13) is manually adjusted, at a step of 0.1, for possible data points in the robot's workspace. This is done to determine unique damping factor for each point in the workspace. The unique factors are chosen as best minimizer of Eq. (13). That is, the factors that require least iteration for each data point and ensure convergence. Arbitrary trials with several data points in the workspace show that effective damping factors are between 0.1 and 50. Coordinates of each data point, norm of the data point from an initial point and the unique damping factor are collected as dataset for the DNN. With the angular interval of 0.32 rad, a total of 1,171,875 data points are generated as the robot's workspace to model the DNN.

The entire dataset has a dimension of 5×1171875 which includes coordinates of each data point (P_t^x, P_t^y, P_t^z), norm from origin ($\|P_t\|$), and the best damping factor (λ_t). However, during preprocessing, damping factor of 6.4% of the entire dataset (75,000 data points) are not well represented in the entire dataset, so they were removed to enhance the behavioral cloning of the network. The remaining 80% of the dataset (877,500 data points) is randomly partitioned into 70% training set, 15% validation set, and 15% testing set, respectively. Structures of the original, pruned, and final workspaces considered during the network setup are shown in Fig. 4. Due to high diverse and non-linear nature of the huge dataset, the neural network consists of 11 hidden layers, and each layer has 15 neurons. The 11×15 architecture is chosen after several arbitrary trials made to define the best hidden structure that can well-generalize the network and achieve stable performance for the remaining data.

5.2. Implementation results

Operation of the proposed IK method starts with receiving coordinates of a target point or consecutive via points in a desired trajectory. Euclidean norm of the target (from origin) alongside with the coordinates of the target point are set as input of the DNN. The network in turn predicts the best damping factor needed to precisely solve IK of the point for real-time operation of the robot. Subsequently, the predicted value is applied to solve Eq. (13) for the resulting angular values. Finally, the values obtained from the proposed DL-DLS IK method are set to the corresponding joints of the simulated robot using *fkine()* — a predefined function in Robotics Toolbox. This is done to show the final pose of the robotic model. Similarly, joints of the actual robot are rotated to these angles with the help of a customized user interface designed in LabVIEW® (National Instruments). This programming environment is chosen because of its built-in supports for NI CompactRIO (cRIO-9118) with FPGA reconfigurable chassis and digital I/O modules which are used for controlling motor drives at the robot's joints. Table 1 shows results of the IK method for four data points randomly chosen in the robot's workspace. The predicted damping factor, joint angles, and actual tip points achieved for the data points. Postures of each data point in both Matlab and prototype of the eight-link robot are shown in Fig. 5.

The robotics toolbox is employed to verify the actual points achieved for each data point. Coordinates of the initial position of last link's tip in the Matlab plot is (256.48, 0, 0) for each figure.

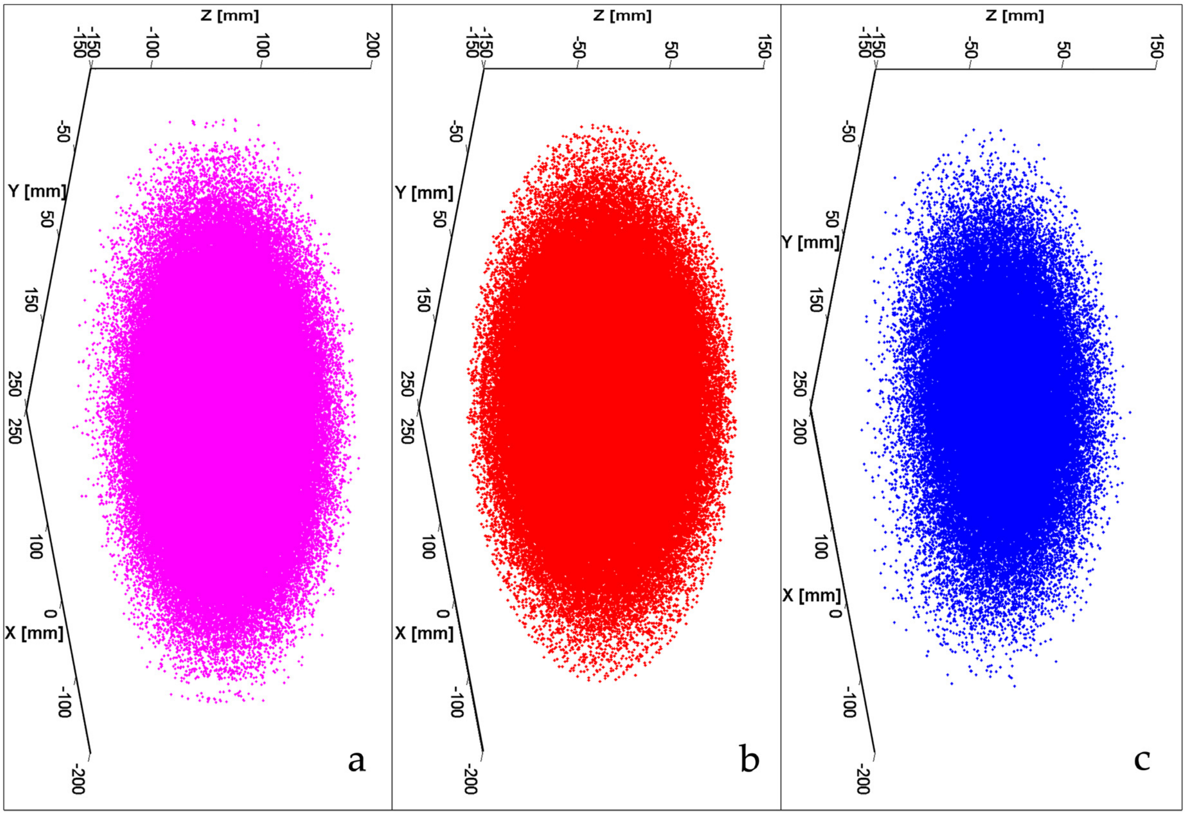

Fig. 4. Stages of workspaces used: (a) Original workspace, (b) Pruned workspace, and (c) Final workspace.

Table 1
IK result from DL-DLS based on 4 arbitrary target points.

Id	Target position (mm)			Predicted λ-value	Joint angles (deg)								Actual tip point (mm)			$\|\vec{e}\|$ (mm)
	X	Y	Z		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	X	Y	Z	
1	−33.109	−52.032	−125.433	14.36	−84.68	−43.54	104.93	−113.21	113.83	−114.99	113.83	−112.91	−31.821	−51.431	−125.58	0.304
2	183.870	−88.379	−95.767	41.66	0.63	−1.43	−17.59	19.08	−17.59	19.14	−17.59	19.14	183.892	−88.391	−95.791	0.035
3	172.910	152.120	101.260	29.97	28.53	−18.45	7.85	−4.70	7.79	−4.70	7.79	−4.70	172.913	152.066	101.244	0.056
4	−30.658	−17.77	−91.959	25.81	114.45	113.10	−69.73	−114.92	−69.67	−114.45	9.57	−113.92	−30.729	−17.832	−91.962	0.094

For every plot, the IK method uses coordinates of the given target point and its Euclidian distance from the initial point to determine an appropriate damping factor that could make the end-effector converge faster to the target point. It can be clearly seen from the plots of Fig. 5 that postures from both Matlab plot and actual robot look similar for each data point in Table 1. Similarly, the error offset between target and actual points are less than 0.5 mm for all the data points. This indicates that the accuracy of the proposed IK method is superior. Furthermore, both simulated and actual robots in Fig. 5 show different snake poses. Hence, the proposed snake-like model is capable of accessing target points irrespective of being farther or closer to the robot's base in one or more of the three axes. For instance, it can be seen that the final tip position closely reached the target points in Fig. 5a, b, and c despite the points are farther away from the robot's base. Links of the robotic model do not overlap for the specific IK solution of the three target points. Similarly, the proposed IK method can work for data points closer to base of the snake-like robot as the case of Fig. 5d. These results show that, the method is capable of steering the snake-like robot via both free and confined areas. Therefore, the proposed snake-like robotic can be capable of serpentine movements via both minimal incision in the abdomen or through natural orifices such as mouth or anus.

6. Performance evaluations

In this section, evaluations carried out to validate the proposed IK method are reported.

6.1. Evaluation based on DNN's prediction

Typically, accuracy and convergence time of the proposed DL-DLS IK method depends on the damping factor predicted by the deep network for any given target point in the robot's workspace. As a result, we evaluate the performance of the deep network based on its prediction accuracy (Vargas et al., 2014), which measures network's performance according to its mean squared of error (MSE). As shown in Fig. 6, MSE of the deep network drops rapidly through the training process. It can be observed from the figure that, both validation and training error values are roughly similar throughout the network training process. Hence, prediction accuracy of the deep network is very good.

With the training parameters in Table 2, an MSE value of 0.31085 is achieved at 343rd epoch of the training period, exactly after ten consecutive validation checks. At this point, the training performance cannot improve further. After several arbitrary trials, the training parameters in Table 2 produce the best performance with least percent errors of 14.05% for training, 10.93%

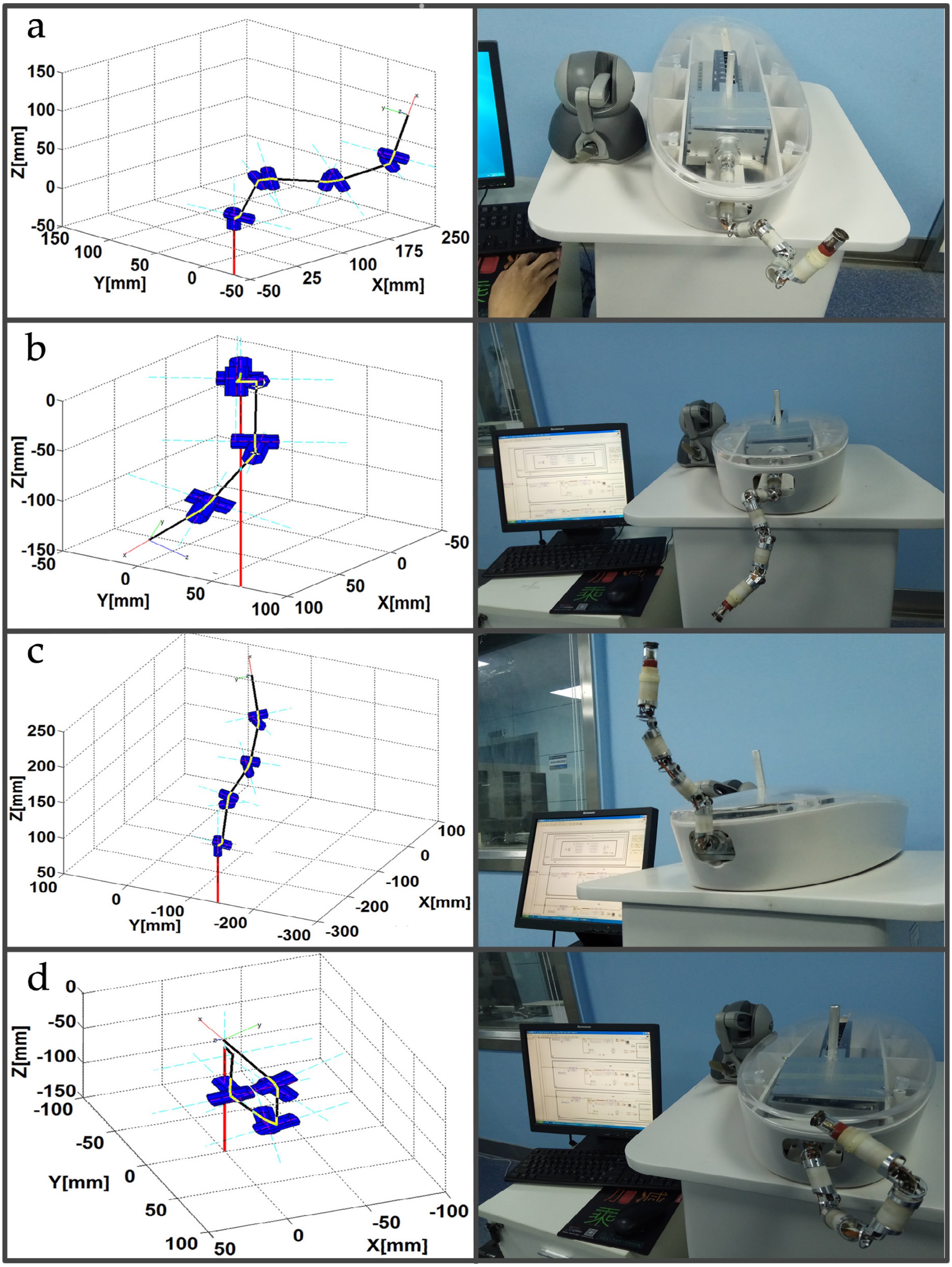

Fig. 5. IK results for arbitrary target points in Table 1 based on predicted λ-value.

for validating, and 7.49% for testing the network. As in Fig. 7a–c, regression plots of the three phases show there is a consistent correlation between the actual and predicted damping factors. In essence, the deep network can facilitate accurate and fast computation of appropriate joint angles for any data point in the robot's workspace.

6.2. Evaluation based on existing IK methods

Performance of the proposed method is also evaluated against eight existing IK methods. These include four popularly used IK methods that are based on error damping with least-squares approach (DLS-based), and another four methods which are not based

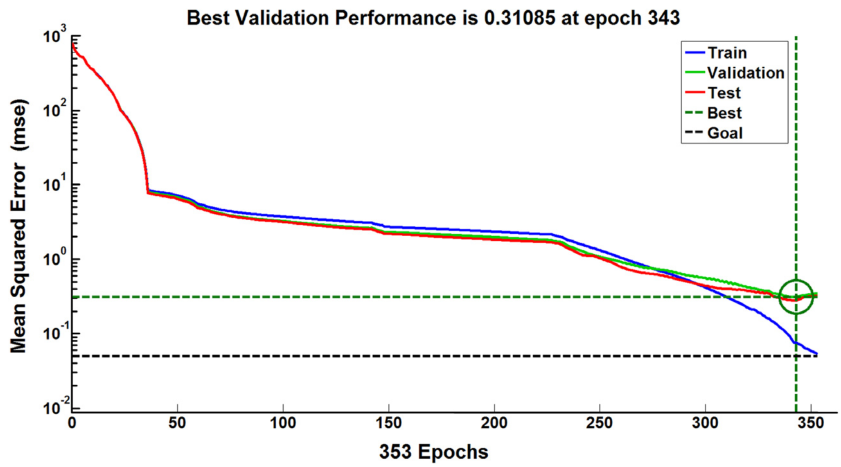

Fig. 6. Performance plots of the deep network.

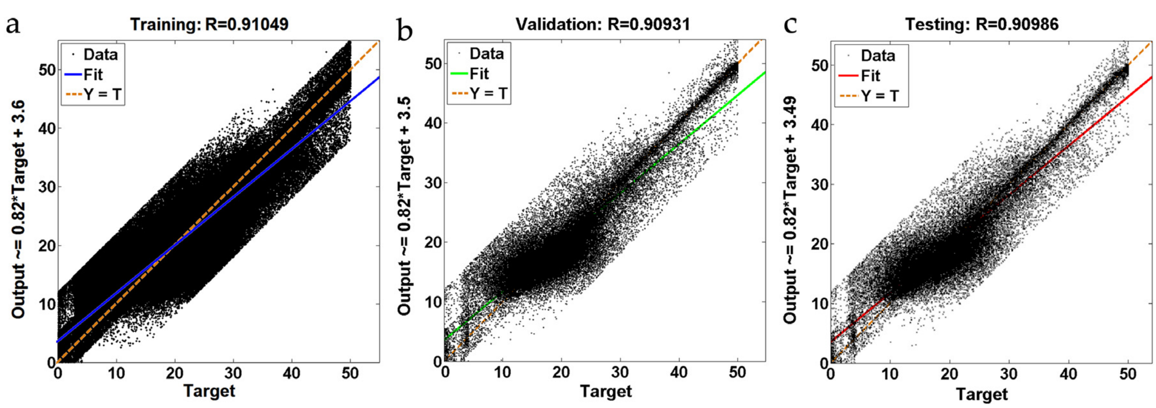

Fig. 7. Regression plots of the deep network: (a) Training phase; (b) Validation phase; (c) Testing phase.

Table 2
Optimal parameters for training the deep network.

Id	Parameter	Value	Id	Parameter	Value
1	Training mode	*trainbr*	4	Marquardt adjustment	1^{100}
2	Performance goal	0.05	5	Maximum validation failures	10
3	Learning rate	0.01	6	Minimum performance gradient	1^{-10}

on DLS. The considered DLS-based approaches are Jacobian with basic damped least squares (J-DLS), singular value decomposition of Jacobian with damped least squares (J-SVD-DLS), adaptive nonlinear least-squares (AN-DLS), and selectively damped least squares (SDLS). These methods are chosen because of their close similarity with the proposed DL-DLS IK method. For each method, analytic Jacobian of the robot is modeled and the IK solutions are obtained based on the inverse Jacobian algorithm. To avoid biasness towards any of the methods, the damping value that provides the best result for each method is used. For J-DLS and J-SVD-DLS, a constant damping factor ($\boldsymbol{\lambda_c}$) of 1.1 is used as in Buss and Kim (2005). This value ($\boldsymbol{\lambda_c} = 1.1$) also gives the best results for both J-DLS and J-SVD-DLS in this study. However, AN-DLS and SDLS are implemented as described in Deo and Walker (1992) and Nakamura and Hanafusa (1986), respectively. The considered non-DLS methods for this evaluation are the non-iterative geometric approach (Omisore, Han, Ren, Zhang and Ivanov, 2017), cyclic coordinate descent method (Wang & Chen, 1991), fast iterative IK solver (FABRIK) (Aristidou & Lasenby, 2011), and Jacobian Transpose (J-Trans) (Wolovich & Elliott, 1984), respectively. The non-iterative geometric method is applied in Omisore, Han, Ren, Zhang and Ivanov (2017) to solve IK of the snake robot-like robot while the remaining three methods have been widely used to resolve IK problems in robotics.

Both categories of the IK methods were implemented for eight-link model of the simulated snake-like robot. For this evaluation, 1487 data points are randomly selected from the remaining 20% of the entire dataset that makes up the robot's workspace. These include the data points that are not pre-known to the network as they are neither part of the datasets used for training, validating, nor testing the network. Finally, each point in the evaluation dataset is set as target point for the IK methods and the obtained results are stored for further analyses. The evaluation results and threshold values used to control the iterative methods are presented.

Table 3
Evaluation of results of IK methods for the snake-like robot with eight links.

IK methods	Reachability measure	Number of iterations	Execution time (ms)	Iterations per ms
Proposed DL-DLS Method	91.59%	33.02 ± 39.60	9.20 ± 16.92	3.59 ± 1.62
J-SVD-DLS (Nakamura & Hanafusa, 1986)	42.97%	87.27 ± 82.18	48.18 ± 29.89	1.81 ± 1.74
SDLS (Buss & Kim, 2005)	82.53%	75.99 ± 92.01	50.26 ± 30.79	1.51 ± 1.17
AN-DLS (Deo & Walker, 1993)	75.04%	94.12 ± 98.02	36.81 ± 17.94	2.56 ± 1.65
J-DLS (Wampler & Leifer, 1988)	41.89%	115.94 ± 95.58	43.93 ± 26.80	2.64 ± 0.92
Non-iterative approach (Omisore et al., 2017)	19.57%	1.00	3.95 ± 1.37	–
Fast iterative IK solver (Aristidou & Lasenby, 2011)	8.16%	348.63 ± 92.01	368.92 ± 135.24	0.95 ± 0.77
Cyclic coordinate descent (Wang & Chen, 1991)	3.87%	482.51 ± 98.02	719.29 ± 186.06	0.67 ± 0.48
J-Trans (Wolovich & Elliott, 1984)	0.00%	500.00	1007.36 ± 13.58	0.49 ± 0.13

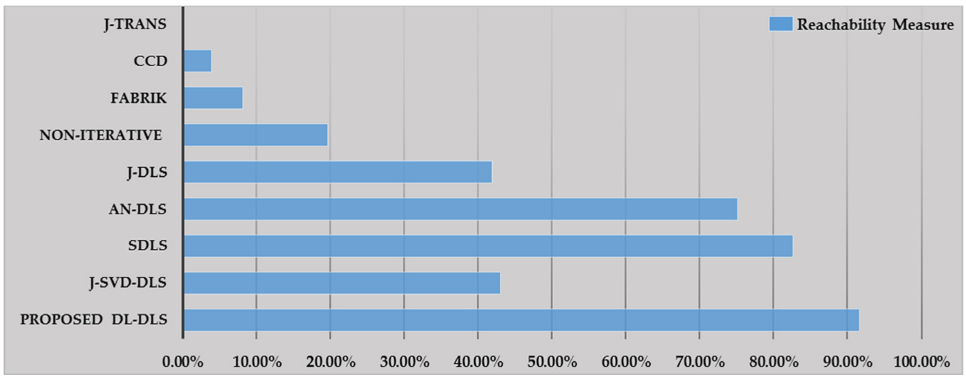

Fig. 8. Evaluation of IK methods based on their reachability measure.

6.2.1. Analysis of the evaluation results

Performances of the IK methods are evaluated based on four metrics namely; reachability measure, execution time, number of iterations required for convergence, and number of iterations executed per ms. The results obtained for each metric are given in Table 3. All methods are implemented as described herein or in the referred studies. Aside from the non-iterative approach (Omisore, Han, Ren, Zhang and Ivanov, 2017), other methods are iterative based. Hence, an admissive error of 0.1 mm is set as threshold for the kinematic error. Similarly, the maximum iterations allowed for each data point is set as 500. The latter is used to avoid endlessness in cases where a given data point is not reachable by an IK method or such method requires excessive iterations before eventual convergence.

A. Reachability measure

Reachability measure is the percentage ratio of data points the robot can reach with an IK method upon the thresholds for maximum kinematic error and iterations. The proposed IK method is compared with existing ones based on their reachability measures in Fig. 8. It can be seen that DL-DLS method achieves the highest reachability measure with value of 91.59%. Analytically, it shows that our method can resolve IK of around 92% of points in the evaluation dataset. This high accuracy value can be attributed to performance of the deep network in predicting appropriate damping factor for each data point. Next to the proposed method are SDLS and AN-DLS which also achieve reachability measures of 82.53% and 75.04%, respectively. These values can be attributed to the bounding and switching strategies used when determining damping factors in the two approaches, respectively. Generally, analysis of the reachability measures shows that the DLS-based methods outperform the non-DLS ones.

The non-DLS based methods perform badly in terms of the reachability measure due to different reasons. Despite the fact that the non-iterative geometric approach requires only a single iteration to compute the IK, it is only suited for snake-like robots with reduced number of links (Omisore, Han, Ren, Zhang and Ivanov, 2017, Omisore, Han, Ren, Zhang, and Wang, 2017). Also, FABRIK and CCD, which are iterative geometric methods, result in low reachability measure due to hard-joint limits of the snake-like robot. They can only perform better when one of the axial values of a given target point is close to zero, thereby having the snake-like robot operating similar to a planar arm robot. Lastly, the Jacobian transpose method requires more iterations for convergence. In this evaluation, the 500 iterations set are not sufficient for the non-DLS methods to minimize the error to the set tolerance value. In fact, J-Trans method makes the robot to start oscillation earlier and more frequently due to consistent orthogonal joints that connect the robot's links (Omisore, Han, Ren, Zhang and Ivanov, 2017).

B. Average iteration and execution time

Performances of the IK methods are analyzed based on average number of iterations and execution time required by each method to converge. These metrics are inspected to determine the oscillation and tracking qualities of the IK methods. The non-iterative geometric method (Omisore, Han, Ren, Zhang and Ivanov, 2017) is exempted because it does not require multiple iterations, and it only converges for less than 20% of data points in the dataset. Similarly, FABRIK, CCD, and J-Trans are exempted since they achieve very low reachability measure due to joint limits set for the snake-like robot. As presented in Table 3, the proposed IK method converges faster than other methods and requires an average of 33.02 (±39.60) iterations to solve IK of a single data point. Compared with other DLS-based methods, the proposed method requires the lowest iterations for convergence. With admissible kinematic error of 0.1 mm and maximum iteration of 500, the proposed method requires a mean execution time of 9.20 (±16.92) ms to converge at reachable data points in the evaluation dataset.

Combining the mean execution time with the average number of iterations required for convergence, more than 3.6 iterations of the proposed DL-DLS method can be completed in 1 ms, and an average execution time of 0.24 ms to complete an iteration. On contrary, many iterations are required by other DLS-based methods before eventual convergence for some data points. It is important to stress that J-DLS and AN-DLS require more iterations to get the robot's end-effector to converge at coordinates of the

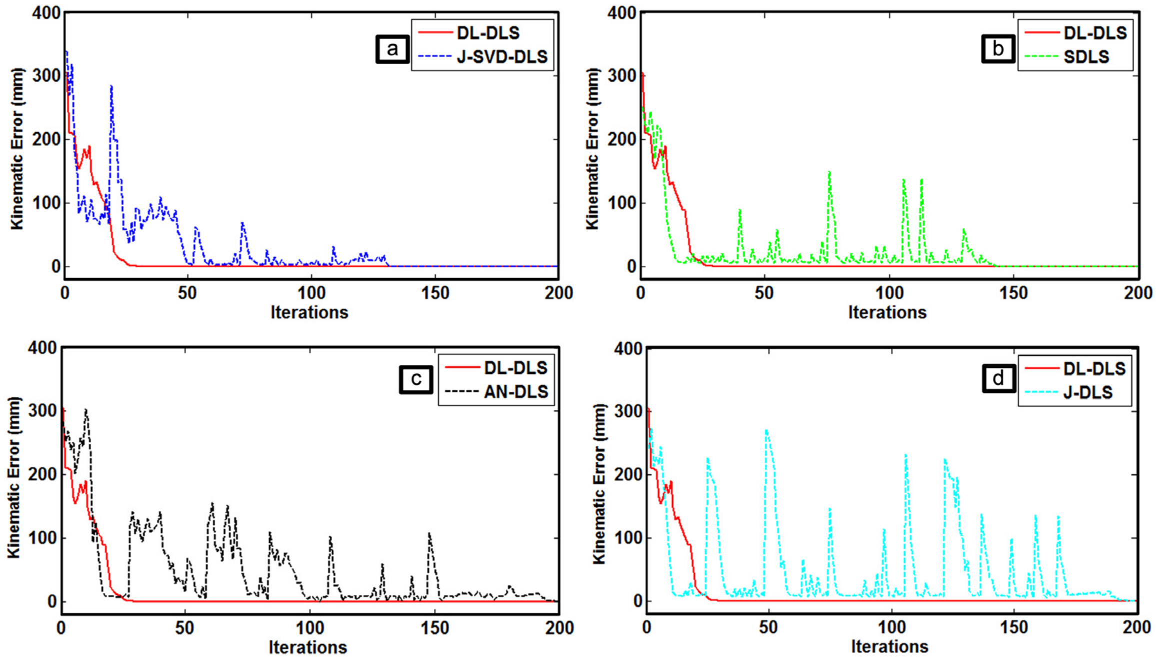

Fig. 9. Average iterations required and error damping between the proposed and existing DLS-based methods.

target point for the given threshold value of 0.1 mm of kinematic error. In addition, the proposed DLS-based method is compared with the existing DLS-based methods with respect to the error damping progress along the iterations. For this purpose, the mean values of the kinematic error achieved during the damping process of the proposed method are plotted against those of the existing DLS-based methods. As shown in Fig. 9, kinematic error of the DL-DLS method damps steadily at each iteration unlike what was obtained from other competing methods. The existing DLS methods produce unpredicted instabilities along the increasing order of iteration, even though convergence is eventually achieved. Thus, the proposed method can drive the robot to successfully track points in reachable area of the robot's workspace with smooth motion and insignificant oscillations of the robot's links.

6.2.2. Analysis of variations in kinematic error

To further describe the robustness of the proposed method over iterative based ones, different values are set as error thresholds while solving the robot's kinematics. The unique tolerance values used for the controlled trials are [1e−4, 5e−4, 1e−3, 5e−3, 1e−2, 5e−2, 1e−1, 5e−1, 1, 3, 5] mm, while the maximum iteration is set as 500 for each tolerance value. The non-iterative approach (Omisore, Han, Ren, Zhang and Ivanov, 2017) is not considered because it requires only one iteration and error minimization is not possible. Average execution times for the iterative IK methods are computed and plotted with respect to the corresponding threshold values. As shown in Fig. 10, the proposed DL-DLS method tracks and converges to targets faster than other iterative methods for the different threshold values set as the admissive kinematic error.

In a moderate case, with tolerance value of 0.1 mm, an average of 9 ms is required for convergence, while a strict case with tolerance value of 1e−4 mm requires an average execution time of 0.28 s. Next to the proposed method, J-SVD-DLS requires approximately 48.2 ms and 4.02 s in similar situations, respectively. With maximum iterations of 500 and error tolerance of 5 mm, J-Trans method cannot converge the robot's end-effector below the admissive kinematic error. The mean execution time required to solve IK of a single data point remains, approximately, 10.1 s for all the admissive error values. Overall, it can be seen that the DLS-based methods perform better than the non-DLS methods. The best cases of the latter are obtained for FABRIK, which can keep the robot's end-effector below the admissive error within 4 s for tolerance values of [1e−1, 5e−1, 1, 3, 5] mm.

7. Conclusions and future works

A major challenge in radiosurgery remains precise delivery of pre-planned treatment doses on cancerous tissues in the abdominal areas. To minimize the aftermath effects of external radiation systems, we present a new snake-like robot for minimal invasive radiosurgery of gastrointestinal cancer, and proposed a deeply-learnt DL-DLS method for IK of the robot. The proposed IK method uses DLS of analytical Jacobian of the robot's end-effector to determine appropriate joint-vector for desired points. To predict optimal damping factor for given data points, a DNN was trained with a huge dataset of 877,500 data points. Implementation of the proposed method with an eight-link model of the snake-like robot shows the method has a very high reachability and convergence measures. Furthermore, evaluation results show that, the proposed method is better than existing IK methods. A major contribution of this approach is total avoidance of singular configurations in the robot's workspace, and yet, the end-effector converges to the desired data points in a very fast and precise manner. These can be attributed to the adeptness of the DNN's structure built for prediction of optimal damping factors.

In this study, DL is employed for prediction of appropriate damping factor needed for solving IK of spatially snake-like robots. The proposed DL-DLS method gains prediction knowledge by deeply learning nonlinear functions that can be defined with the four features of data points in the robot's workspace. Hence, predictions were based on knowledge discovery through supervised learning. This behavioral cloning approach cannot adapt to completely new situations with surmountable deviations from the knowledge gained during the supervised learning. Dynamic model of the snake-like model shall be considered along with important design constraints, such as gear error and torques, acting at each joint of the robot. Finally, the models will be extended for path planning and collision avoidance during navigation of the robot along specified trajectories.

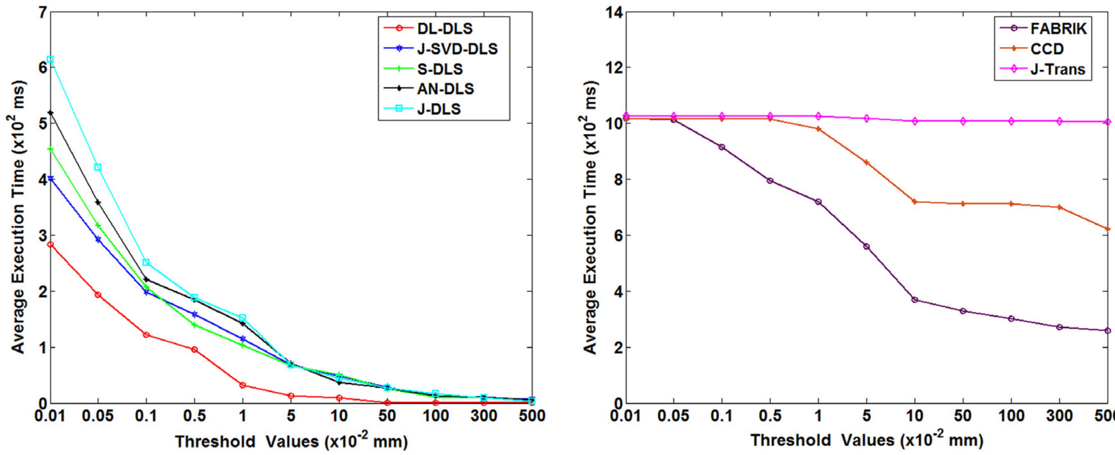

Fig. 10. Analysis of mean execution time for different threshold values.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under the Grants #U1713219, #U1505251, #71531004; and the Guangdong Innovation Research Team Funds for Image-Guided Therapy under the Grant #2011S013. Omisore O. M. acknowledges CAS-TWAS President's Fellowship for sponsoring his Ph.D at the University of Chinese Academy of Sciences, Beijing, 100049, China.

References

Aristidou, A., & Lasenby, J. (2011). FABRIK: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, *73*, 243–260.

Buss, S., & Kim, J. (2005). Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, *10*(3), 37–49.

Chung, W., Youm, Y., & Chung, W. (1994). Inverse kinematics of planar redundant manipulators via virtual links with configuration index. *Journal of Robotic Systems*, *11*(2), 117–128.

Corke, P. Robotics, vision and control, springer tracts in advanced robotics, (p. 73).

Degani, A., Choset, H., Wolf, A., Ota, T., & Zenati, M. (2006). Percutaneousintrapericardial interventions using a highly articulated robotic probe.In: *IEEE/RAS-EMBS international conference on biomedical robotics and biomechatronics*, Pisa, Italy, February 20-22, 2006, (pp. 7-12).

Deo, A., & Walker, I. (1992). Robot subtask performance with singularity robustness using optimal damped least-squares. In: *IEEE international conference on robotics and automation*, Nice, France, May 12-14, 1992, (pp. 434-441).

Deo, A., & Walker, I. (1993). Adaptive non-linear least squares for inverse kinematics. In: *IEEE international conference on robotics and automation*, Atlanta, USA, May 2-6, 1993, (pp. 186-139).

Elgazzar, S. (1985). Efficient kinematic transformations for the PUMA 560 Robot. *IEEE Journal on Robotics and Automation*, *1*(3), 142–151.

Flacco, F., & de Luca, A. (2013). Optimal redundancy resolution with task scaling under hard bounds in the robot joint space, 2013. In: *IEEE international conference on robotics and automation* Karlsruhe, Germany, May 6-10, 2013.

Flacco, F., de Luca, A., & Khatib, O. (2012). Motion control of redundant robots under joint constraints: saturation in the null space. In: *IEEE international conference on robotics and automation*, Minnesota, USA, May 14-18, 2012.

Hadjerci, O., Hafiane, A., Morette, N., Novales, C., Vieyres, P., Delbos, A., et al. (2016). Assistive system based on nerve detection and needle navigation in ultrasound images for regional anesthesia. *Expert Systems with Applications*, *61*, 64–77.

Hayashi, M., Chernov, M., Tamura, N., Izawa, M., Muragaki, Y., et al. (2013). Concept of robotic gamma knife micro-radiosurgery and results of its clinical application in benign skull base tumors. *Acta Neurochirurgica Supplement*, *116*, 5–15.

Jamali, A., Khan, R., & Rahman, M. (2011).A new geometrical approach to solve inverse kinematics of hyper redundant robots with variable link length,4th. In:*international conference on mechatronics*, Kuala Lumpur, Malaysia, May 17-19.

Jayarao, M., & Chin, L. (2007). Robotics and its applications in stereotactic radiosurgery. *Neurosurgery Focus*, *23*(6), 1–7.

Kenwright, B. (2011). Real-time character inverse kinematics using the gauss–seidel iterative approximation method. In: *fourth international conference on creative content technologies*, Nice, France, July 22-27, 2011, (pp. 63-68).

Kostic, D., Hensen, R., de Jager, B., & Steinbuch, M. (2002).Closed-form kinematic and dynamic models of an industrial-like rrr robot. In: *IEEE international conference on robotics and automation* Washington, USA, May, (pp. 1309-1314).

Kucuk, S., & Bingul, Z. (2014). Inverse kinematics solutions for industrial robot manipulators with offset wrists. *Applied Mathematical Modelling*, *38*, 1983–1999.

Kuo, J., Yu, C., Petrovich, Z., & Apuzzo, M. (2003). The cyberknife stereotactic radiosurgery system: description, installation, and an initial evaluation of use and functionality. *Neurosurgery*, *53*(5), 1235–1239.

Lenz, I., Lee, H., & Saxena, A. (2015). Deep learning for detecting robotic grasps. *International Journal of Robotics Research*, *34*(4–5), 705–724.

Levine, S., Wagener, N., & Abbeel, P. (2015).Learning contact-rich manipulation skills with guided policy search. In: *IEEE international conference on robotics and automation* Seattle, USA, May 26-30, (pp. 156-163).

Li, Y., Zou, L., Li, Y., Xie, Y., & Xiong, J. (2017).A software platform of treatment planning and navigation for radiotherapy snake robot, 59th. In: *meeting of american association of physicists in medicine* July 30.

Liu, H., Liu, H., Sun, F., & Fang, B. (2017a). Kernel regularized nonlinear dictionary learning for sparse coding. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *PP*(99), 1–10. http://dx.doi.org/10.1109/TSMC.2017.2736248.

Liu, H., Qin, J., Sun, F., & Guo, D. (2017b). Extreme kernel sparse learning for tactile object recognition. *IEEE Transactions on Cybernetics*, *47*(12), 4509–4520.

Makondo, N., Claassens, J., Tlale, N., & Braae, M. (2012).Geometric technique for the kinematic modeling of a 5 dof redundant manipulator, 5th. In: *robotics and mechatronics conference of south africa*, Gauteng, South Africa, November 26-27.

Nakamura, Y., & Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control. *ASME Journal of Dynamic Systems, Measurement, and Control*, *108*(9), 163–171.

Omisore, O., Han, S., Ren, L., & Wang, L. (2016).A fuzzy-pd model for master-slave tracking in teleoperated robotic surgery, 12th. In: *IEEE biomedical circuits and systems conference*, ShangHai, China, October. 17-19.

Omisore, O., Han, S., Ren, L., Zhang, N., & Ivanov, K., et al. (2017). Non-iterative geometric approach for inverse kinematics of redundant lead-module. In;*A radiosurgical snake-like robot, biomedical engineering*, OnLine 2017, 16:93, 1-25.

Omisore, O., Han, S., Ren, L., Zhang, N., & Wang, L. (2017). A new geometric solution for inverse kinematics of hyper-redundant snake robot used in real-time teleoperated surgery. In: *18th IEEE international conference on industrial technology*. Toronto, Canada, March 22-25, 2017, (pp. 710-714).

Parsa, S., Daniali, H., & Ghaderi, R. (2010). Optimization of parallel manipulator trajectory for obstacles and singularity avoidances based on neural network. *International Journal of Advanced Manufacturing Technology*, *51*(5), 811–816.

Pechev, A. (2008). Inverse kinematics without matrix inversion. In:*IEEE international conference on robotics and automation*. Pasadena, CA, USA, May 19-23, 2008, (pp. 2005-2012).

Pieters, B., De Back, D., Koning, C., & Zwinderman, A. (2009). Comparison of three radiotherapy modalities on biochemical control and overall survival for the treatment of prostate cancer: a systematic review. *Radiotherapy and Oncology*, *93*(2), 168–173.

Ren, L., Omisore, O., Han, S., & Wang, L. (2017). A master-slave control system with workspaces isomerism for teleoperation of a snake robot. In: *39th international conference of the ieee engineering in medicine and biology society*, Seogwipo, Republic of Korea, July 11-15, 2017.

Sardana, L., Sutar, M., & Pathak, P. (2013). A geometric approach for inverse kinematics of a 4-link redundant in-vivo robot for biopsy. *Robotics and Autonomous Systems*, *61*, 1306–1313.

Schieda, N., Malone, S., Al-Dandan, O., Ramchandani, P., & Siegelman, E. (2014). Multi-modality organ-based approach to expected imaging findings, complications and recurrent tumour in the genitourinary tract after radiotherapy. *Insights Imaging*, *5*, 25–40.

Shang, J., Noonan, D., Payne, C., Clark, J., & Sodergren, M., et al. (2011).An articulated universal joint based flexible access robot for minimally invasive surgery. In: *IEEE international conference on robotics and automation*,Shanghai, China. May 9-13, 2011, 114, (pp.7-1152).

Sheng, L., Yiqing, W., Qingwei, C., & Weili, H. (2006). A new geometrical method for the inverse kinematics of the hyper-redundant manipulators. In: *IEEE international conference on robotics and biomimetics*, December 17-20, 2006, Kunming, China, 1356-1359, (pp. 1356-1359).

Srinivasa, Narayan, Bhattacharyya, Rajan, Sundareswara, Rashmi, Lee, Craig, & Grossberg, Stephen (2012). A bio-inspired kinematic controller for obstacle avoidance during reaching tasks with real robots. *Neural Networks*, *35*(2012), 54–69.

Tchon, K. (2008). Optimal extended jacobian inverse kinematics algorithms for robotic manipulators. *IEEE Transactions on Robotics*, *24*(6), 1440–1445.

Toshani, H., & Farrokhi, M. (2014). Real-time inverse kinematics of redundant manipulators using neural networks and quadratic programming: a lyapunov-based approach. *Robotics and Autonomous Systems*, *62*, 766–781.

Vargas, L., Leite, A., & Costa, R. (2014).Overcoming kinematic singularities with the filtered inverse approach.In:*international federation of automatic control*, Cape Town, South Africa, August 24-29.

Wampler, C. (1986). Manipulator inverse kinematic solution based on vector formulations and damped least squares methods. *IEEE Transactions on System Man and Cybernetics*, *16*(1), 93–101.

Wampler, C., & Leifer, L. (1988). Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators. *Journal of Dynamic Systems, Measurement & Control*, *110*, 32–38.

Wang, L., & Chen, C. (1991). A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, *7*(4), 489–499.

Whitney, D. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, *10*(2), 47–53.

Wolovich, W., & Elliott, H. (1984). A computational technique for inverse kinematics. In: *23rd IEEE conference on decision and control*. Las Vegas, USA, December 12-14, 1984, (pp. 1359-1363).

Yahya, S., Moghavvemi, M., & Mohamed, H. (2011). Geometrical approach of planar hyper-redundant manipulators: inverse kinematics ,path planning and workspace. *Simulation Modelling Practice and Theory*, *19*, 406–422.

Yahya, S., Mohamed, H., Moghavvemi, M., & Yang, S. (2009).A new geometrical inverse kinematics method for planar hyper redundant manipulators, conference on innovative technologies. In: *intelligent systems and industrial applications*, Sunway, Malaysia, July 25-26, (pp. 20-22).