2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Kinematics Constraint Modeling for Flexible Robots based on Deep Learning[1]

Olatunji Mumini OMISORE, *Member, IEEE*, and Lei WANG, *Senior Member, IEEE*,

***Abstract*— Application of flexible robotic systems and teleoperated control recently used in minimally invasive surgery have introduced paradigm shift in interventional surgery. While Prototypes of flexible robots have been proposed for surgical diagnostic and treatments, precise constraint control models are still needed for flexible pathway navigation. In this paper, a deep learning based kinematics model is proposed for motion control of flexible robots. Unlike previous approach, this study utilized the different layers of deep learning system for learning the best features to predict the damping value for each point in the robot's workspace. The method uses differential Jacobian to solve IK for given targets. Optimal damping factor that converges precisely around given target is rapidly predicted by a DNN. Simulation of the robot and implementation of the proposed control models are done in V-rep and Python. Validation with arbitrary points shows the deep-learning approach requires an average of 26.50 iterations, a mean error of 0.838, and an execution time of 3.6 ms for IK of single point; and converges faster than other existing methods.**

Keywords*— *snake-like robots, inverse kinematics, motion control, trajectory planning, minimally invasive surgery;

I. Introduction

Open surgery, a traditional technique for interventional surgery, has been remarked with limitations that galvanized the development of minimally invasive surgery (MIS). The latter has developed better in recent time that interventional surgery has reached a tipping point with minimized hospital stays, perioperative and recovery times, and post-surgical pains in patients [1]. MIS procedures involve manipulating surgical end-effectors with flexible navigators through minimal ports incised on patient's body for interventional procedures. This modern approach has been reported to aid surgeons in carrying out a range of diagnostic, therapeutic, and rehabilitative procedures. Needs of real-time navigation and vision during cardiac and abdominal interventions have been an important domain with research focus since onset of MIS. While, efficient and ergonomic use considerations have been addressed by distally attached surgical and visualization tool appendages on flexible endoscopic or endovascular tools, however highly dexterous robotic mechanism with enhanced constraints control model and feedback systems are required to access core-hidden organs in human cardio-thoracic areas during teleoperated MIS. This can enhance clearer views of the internal environment where surgery is intended to be performed [2].

Continuum and snake-like robotic systems have recently appeared as a perfect fit for carrying out interventional surgery through flexible anatomical pathways. Some studies have reported the application of flexible link-based robots for intraluminal procedures on internal organs in both upper- and lower torso of human abdomen. Ota *et al.* designed a highly articulated robotic surgical system for pericardial therapeutic delivery with minimal invasion [3]. Salle *et al.* proposed a highly modular MIS for coronary artery bypass grafting [4]. In pre-clinical study, a related robotic system was used to navigate and visualize some landmarks encountered during oropharyngeal surgery in cadavers [5]. These snake-like mechanisms are made-up with connected serial links for minimal invasion procedures. Each link has a small diameter (Ø≤2 cm) and rotates around unique axes [6]. Dexterity are enhanced by each added link introducing extra degrees of freedom (DoF) while designs with limited or no-offset joints further enhances the dexterity of the robots for MIS [7]. Constraints modeling of kinematics and dynamics systems for motion and teleoperation control of the robotic systems require costly and complex computations with stable guarantee for existence of constraints resolution [8].

Recent survey of studies on constraints modeling and control of MIS robots shows [6], kinematics and dynamics constraints resolutions for articulation of link-based robots are classified as closed-form and iterative methods. This is somewhat different bio-inspired continuum and soft robots also involve segmental-based curvature computations. Both are based on Jacobian analysis and segment mapping between robots' Cartesian and joint spaces. Intelligent mappings modeled on Denavit Hartenberg (DH) notations have been proposed for solving control problems in flexible robotics. Agarwal [9] applied AI-based Fuzzy C-means approach for planning trajectory of a 4-DoF redundant manipulator. The clustering model was based on weighted within-scatter and between-cluster metrics of robot, while manipulability index was used as performance criterion. Chen *et al.* [10] proposed a 3D neural model for safety-enhanced trajectory navigation with minimum sweeping area factor in robot's workspace. In surgical domain, genetic algorithm was proposed for 4-DoF i^2-Snake robot based on pre-recorded suturing and anatomical data [11]. Damped least-squares (DLS) approach, a known stabilizer of pseudoinverse at near-singular points, has been reclined to introduce deep learning for constraints control in flexible robots. Use of reinforcement learning and policy optimization was investigated for kinematics of 5-DoF robot with joint, velocity, and pose limitations [12]. Omisore *et al.* [13] proposed a deeply-learnt model for kinematics resolution of surgical snake-like robots. Wang *et al* [14] also implemented the model to optimize kinematics solution of 7-DoF Kuka robot. While this model is capable of predicting unique damping factors required for accurate kinematics of most target points in the robots workspace, effective deep learning network model with deep convolution, optimization, and regularization layers are deemed for improved prediction.

In this study, kinematics model is proposed based on deep learning for motion control of flexible robots used in MIS. Remainder of this paper is organized that. Section II presents kinematics structure and modeling of n-DoF snake-like robot and an encoder-decoder deep learning model proposed for kinematics resolution; model implementation, experiments and validation results are discussed in Section III; lastly, the study conclusion and future works are stated in Section IV.

[1]This work was supported by National Natural Science Foundation of China (#U1713219, #61950410618); Shenzhen Natural Science Foundation (#JCYJ20190812173205538); and CAS PIFI for postdoctoral study.

The authors are with Research Center for Medical Robotics and MIS Devices, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China; {omisore@siat.ac.cn, wang.lei@siat.ac.cn}.

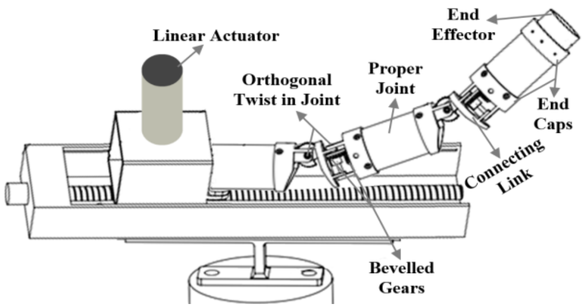

Fig. 1: CAD Model of the Snake-like Robot with 4-DoF.

II. Deep Learning Model for Kinematics Control

The targeted robot is a snake-like mechanism designed for MIS in human abdomen (Fig. 1). Details of the robot are in previous studies [6, 13-15]. This paper focuses on improving existing inverse kinematics approaches for constraints control and teleoperation of the snake-like robot. This includes enhancing Jacobian DLS method with deep learning for fast and accurate kinematics and dynamics resolution.

A. Kinematics Modeling

DLS approaches are renowned stabilizer of pseudoinverse of near-singular points in a robot's workspace. Specifically, it has been used for numerical kinematics resolution [13-15]. This involves selection of constant damping factor that can be used to approximate solution of the Jacobian near-singular points. Starting from the base link of the model in Fig. 1, cumulative transformation of the coordinate systems of reference frame attached to each link are taken in forward direction. For an assumed pose ($\hat{P}$) of the last link (given as Fig. 1), a referenced transformation matrix $({}_{i=2,\ldots,n+1}^{\;i-1}T)$ with position (${}^{j}P \in \Re^{3\times1}$) and orientation (${}_{i}^{j}R \in \Re^{3\times3}$) can be derived based on DH parameters of the consecutive link [13]. The relationship between any frame $\{j\}$ from an initial frame $\{i\}$ for a given pose is defined with respect to origin $\hat{O}$, at the base. Thus, for a given joint vector (θ), DH parameters of the links are substituted directly to compute the single transformation matrix for a desired configuration; this can be evaluated to determine the final pose of the robot. The direct transformation in Eq. 1 computes pose of the last link which can be used for both analytical closed-form and iterative kinematics modeling [8]. Direct frame transformation is used in this study for learning-based kinematics modeling by integrating Jacobian damped least squares (DLS) method with deep learning system. Each transformation operation can be split as given in Eq. 2, showing the relationships between the rates of change of the final pose and joint [16].

$$ {}_{n}^{0}T = {}_{1}^{0}T \times {}_{2}^{1}T \times \ldots \times {}_{n}^{n-1}T = \begin{pmatrix} {}_{i}^{j}R(\theta) & {}^{j}P(\theta)^{T} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix} \quad (1) $$

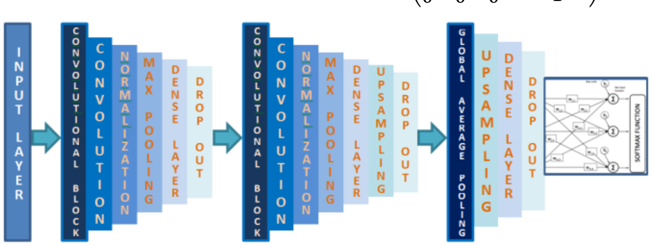

Fig. 2: Deep learning model for predicting damping factor

$$ {}_{i}^{i-1}T = \begin{bmatrix} r_{1x} = c\theta_i & r_{2x} = -s\theta_i c\alpha_i & r_{3x} = s\theta_i s\alpha_i & p_x = a_i c\theta_i \\ r_{1y} = s\theta_i & r_{2y} = c\theta_i c\alpha_i & r_{3y} = -c\theta_i s\alpha_i & p_y = a_i s\theta_i \\ r_{1z} = 0 & r_{2z} = s\alpha_i & r_{3z} = c\alpha_i & p_z = d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

B. Jacobian-based DLS Modeling

To model the nonlinear relationship, a damped least-squares inverse of the robot's Jacobian, with varying damping factor, is adopted. 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, to avoid numerical instabilities and kinematic singularities in the robot's workspace. With first-order partial derivatives, IK of the snake-like robot with n links can be approximated as $J(\hat{\theta}) = \left(\frac{\partial \hat{P}_i}{\partial \theta_j}\right)$; where $\frac{\partial \hat{P}_i}{\partial \theta_j}$ is the linear velocity of the frame pose in each i^{th}-axis (that is x-, y-, and z-axis) with respect to change in j^{th} joint.

Hence, the kinematics problem becomes finding the best joint vector $\theta = [\theta_1, \theta_2, \ldots, \theta_n]^T$ in $\left(J(\hat{\theta})\Delta\theta - \vec{e} = 0\right)$ such that P_e circa a desired target point, P_d. The Jacobian matrix in Eq. (3) is iteratively updated with angles in the robot's joints until the kinematic error ($\vec{e}$) is approximately equal to zero, in Eq. 4. However, to solve the equation, we seek for an optimal change ($\Delta\theta$) that minimizes $\|J\Delta\theta - \vec{e}\|^2$ such that the matrix $J(I - J^{\dagger}J)$ is a projector on the null space of J. Therefore, the solution vector $\Delta\theta$ can be evaluated Eq. 3. Following Buss & Kim [17], since JJ^T is computationally inexpensive and guaranteed to be invertible when J has a full row rank, the objective becomes finding the minimum-norm of the joint speed that minimizes $\|J\Delta\theta - \vec{e}\|^2$ such that the end-effector can be numerically stable near a singular point. With the DLS model in Eq. 4, the kinematics problem is reduced to finding suitable damping values (λ) that steadily minimize the sum of norms of the solution vector and joint-vector of the robot for any given point in a given workspace.

$$ \Delta\theta = J^T(JJ^T)^{-1}\vec{e} \quad (3) $$

$$ \Delta\theta_\lambda = \underset{\Delta\theta}{argmin}\{\|J\Delta\theta - \vec{e}\,\|^2 + \lambda^2\|\Delta\theta\|^2\}. \quad (4) $$

C. Prediction of damping factor with deep learning

Since the differential kinematics solution is ill-conditioned in neighborhood of singular points, and this is experienced in the form of high joint velocities. Stability can be enhanced with prediction of appropriate damping factor. For this, a deep learning network (Fig. 2) with separate encoding and decoding modules is developed for predicting apt λ values.

i) ***Encoding Module****:* this includes convolutional blocks and softmax dense layers that multiplex features of given target points and re-construction of this information into salient values based on topography of the given point in the workspace. Unlike in our previous architecture [14], the input layer of the proposed deep learning model accepts inputs via seven units. The neurons accept features of a given point (P_t) defined as its axial coordinates (P_t^x, P_t^y, P_t^z) and tangential norms ($\|P_t^x\|$, $\|P_t^y\|$, $\|P_t^z\|$, $\|P_t\|$) from an initial point. The variables are added to get the network aware of the robot's workspace topography; such that relationships between initial and target points in the workspace could be quantified. Thus, input maps of 1×7 feature vectors were produced and passed across to the two convolutional blocks. Each block performs convolution operations (8 filters, kernels size of 3, stride of 1, and same length padding), followed by batch normalization to ensure the network processes tensors faster, and initial max pooling operations. Instead of having predetermined slope, a rectified linear unit is used to allow a small gradient for inactive cases. A final block with global average pooling is added for smoothening. A dense with dropout layer is added to regularize extracted features and reduce overfitting.

ii) ***Decoder Module:*** a fully-connected dense layer with Softmax activation was taken to extract the network's weights and biases and predicts apt λ values for given target points. The input of the decoder module is a 64-sized feature vector map. The Softmax layer generalizes the binary form of logistic regression by mapping the feature–sets extracted from the input given target points $X = [x_1, x_2, x_3, \dots, x_n]$ to fit λ values via a linear function f. The operation involves dot product of x_i and the weight matrix W and produces the actual probability scores, as given in Eq. 3. If the probability score is construed as unnormalized log equivalent, the loss function is swapped with Softmax regression function without the negative log likelihood of predicted λ. Thus, the loss function in Eq. 5 is used in the Softmax layer to predict the value of λ for a given target position. Worth mentioning, $e^{f(x_i,W)}$ is a natural logarithm used for inverse of the exponentiation. The actual exponentiation and normalization via sum of exponents is the Softmax function.

$$f(x_i, W) = W x_i \tag{5}$$

$$P\big(y = f(x_i, W)\big) = \left(e^{f(x_i,W)_{y_i}} \Big/ \textstyle\sum_i e^{f(x_i,W)}\right) \tag{6}$$

D. *Kinematics Resolution*

Finally, the optimal joint-vector ($\Delta\theta$) is determined with the predicted λ value, unique minimizer for norm of the damped joints' velocities for the given target point in the workspace. For inverse kinematics, $\Delta\theta$ in the DLS model is obtained as given in Eq. 6, and can as well be supplied into the transformation matrix (Eq. 1) based on the DH notation for forward kinematics solution. The fast kinematics solution can be utilized for dynamics analysis described in *Ref.* [15].

$$\Delta\theta_\lambda = \underset{\Delta\theta}{argmin} \left\| \binom{J}{\lambda I} \Delta\theta - \binom{\vec{e}}{0} \right\| \tag{6}$$

III. Model Implementation and Validation

A. *Robot Simulation*

The flexible snake-like robot model in Fig. 1 was designed in Solidworks® (Dassault Systems Solidworks Corp., USA), exported via unified robot description format for real-time simulation within V-Rep (Coppelia Robotics, Switzerland). For interactive operations, the robot's joints and links were imported separately, assembled and parameterized in V-Rep. The components were manipulated with custom Lua scripts embedded in Python implementation of the network model via dynamic link library to achieve threaded communication.

B. *Deep Learning Model Training*

The encoder-decoder model was implemented with Keras interface and Tensorflow framework on a desktop computer with Intel® Core i7 processor (3.4 GHz each), 48 GB RAM and Nvidia GTX 1080 graphics card. End-to-end network training was implemented for appropriate feature extraction and prediction of damping (λ) values for given target points.

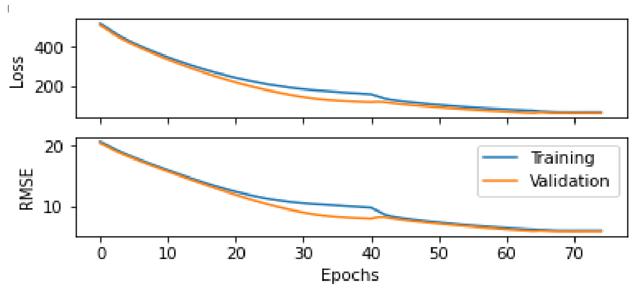

Fig. 3: Training performance of the proposed network

Complete workspace of an 8-DoF robot, simulated in Matlab, was used to generate a huge training dataset which has a dimension of 485315×8 [13]. It includes the axial coordinates (P_t^x, P_t^y, P_t^z), norm ($\|P_t^x\|$, $\|P_t^y\|$, $\|P_t^z\|$, $\|P_t\|$) and best damping factor (λ_t) for all points in the workspace. The dataset was randomly partitioned to 60%, 20%, and 20% for training, validation, and testing the network, respectively. Network tuning was done randomly during training to obtain the best hyperparameters. The convolutional blocks in the network were activated on ReLU with constant kernel size of 3, and the respective filter numbers were 64. Fig. 3 shows the network has a suitable regularizing effect with reduced generalization error of 50% layer dropout rate. Adaptive moment (adam) optimizer was used for network training. Parameters set were dynamic learning rate initialized of 10^{-3}, a decay value of 10^{-6}, and recurrent dropout of 10% added for regularization. Mini-batches of size 128 were employed to avoid GPU-CPU memory transfers while the initial split layers were run in 400 epochs to extract features.

C. *Experimental Results*

To assess the proposed deep learning model for kinematics constraint control, target positions were selected in the 8-DoF robot's workspace and set as the model's input. The desired operation is to obtain joint angles that can set the robot tip to each target position in a move. Hence, a circular path with desired trajectory was set and its consecutive points were taken as input of a custom function in Python. Coordinates of the points and axial Euclidean norm values, computed from the former, are taken by the function. The trained network model employs these parameters to predict the best damping factor (λ) needed to precisely solve Eq. 13 for the angular values. Finally, the joint values obtained are communicated back to the simulated robot via the remote API client script. Thus, the simulated snake-like robot reflects every point-to-point motion command issued while the kinematics accuracy and response time can be calculated for evaluation purposes.

For each point, each joint of the robot is rotated based on angles computed with the proposed method and displayed with the customized user interface designed in V-Rep. Results obtained for six arbitrary points in the robot's workspace are shown in Fig. 4. It can be observed from the figure that the proposed method could determine appropriate angular joint values for each target point. Qualitatively, this indicates that the proposed method exhibits useful accuracy for kinematics resolution needed in flexible robotics control.

D. *Evaluation Results*

Accuracy and execution time are two main metrics that are usually used for evaluating kinematics control methods. While most existing methods are unable to achieve both high kinematics accuracy and fast response time simultaneously, a few existing models were rated very high in previous study. Thus, performance of the proposed kinematics resolution method is compared with those of some existing Jacobian-based methods. These include the deeply-learnt damped least squares (DL-DLS), singular value decomposition of Jacobian with damped least squares (SVD-DLS), and the conventional Jacobian damped least squares (J-DLS). These methods are proposed towards fast and accurate kinematics resolution in robotics. Since SVD-DLS and J-DLS lack a learning system of choosing damping factor, a unique $\lambda = 0.01$ was used in this study. Design and implementation details of the methods have been reported [13]. In this study, performances of the methods are evaluated with kinematics accuracy, reachability measure, number of iterations, and execution time. The evaluation results obtained for each performance metric, based on threshold values; maximum iteration of 500 and an admissive kinematic error of 1 mm, are presented in Table 1. The proposed IK method converges faster than other existing DLS-based methods and requires a mean of 26.50 iterations and execution time of 3.6 *ms* to solve IK of a data point. This is slightly lower that the number of iterations required by the other methods. Performance of J-DLS can be improved by increasing the maximum number of iterations.

IV. Conclusion and Future Works

Recent developments in robotic surgery include design of flexible mechanisms to enhance surgical interventions such as suturing, tumor resection, and radiosurgery. Nonetheless, precise constraint control models are still lacking for flexible pathway navigation. In this paper, a deep learning based kinematics model is proposed for motion control of flexible robots. Unlike previous approach, this study utilized the different layers of deep learning system for learning the best features to predict the damping value for each point in the robot's workspace. The proposed study is targeted towards controlling flexible robotic system used in MIS. Thus, effective computation of the constraints control is necessary for motion and trajectory tracking. Case study of flexible snake-like robot with 8 redundant links was carried out. The simulation results shows better kinematics solution can be obtained with deep learning model compared to using conventional machine learning and mathematical approaches. With admissible kinematic error of 1 mm and maximum iteration of 500, the proposed method converged faster to the given target points in the evaluation dataset.

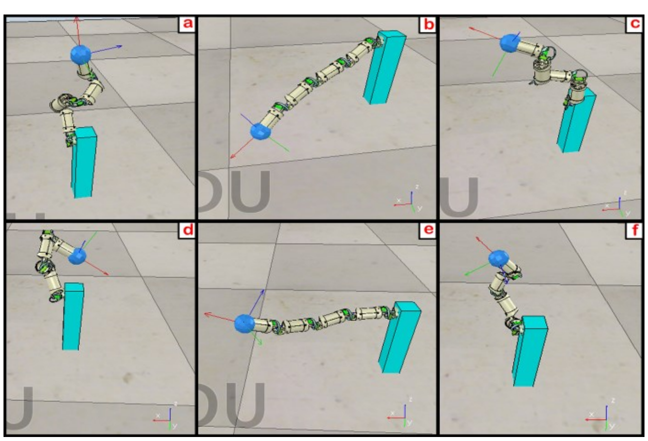

Fig 4: Results obtained for arbitrary points in the robot's workspace

Table 1: Performance evaluation of proposed and existing DLS methods*

Sample Target Position			Kinematics Error (mm)			
			Proposed	**DL-DLS**	**SVD-DLS**	**J-DLS**
-33.109	-52.032	-125.433	**0.974**	0.977	0.995	90.812
172.910	152.120	101.260	0.887	**0.881**	**0.881**	90.812
90.0630	-25.077	-110.417	0.888	0.759	**0.726**	90.812
19.630	-87.720	-11.540	**0.900**	0.920	0.940	90.812
-23.416	-54.531	131.820	**0.422**	0.944	0.944	90.812
214.920	-1.4307	-107.750	0.957	**0.804**	0.845	90.812
Average Iteration Taken			**26.500**	35.667	86.500	500

* Results were obtained at 1 mm admissible error and 500 iterations for

Finally, further studies are aimed by modifying the network structure to improve its performance, and to further lower the kinematics error for precise sub-millimetre robotic navigation.

References

[1] R. Beasley, *Medical Robots: Current Systems and Research Directions*, Journal of Robotics, vol. 2012, pp.1-14, 2012.
[2] O. Omisore *et al*, *Towards Characterization and Adaptive Compensation of Backlash in a Novel Robotic Catheter System for Cardiovascular Interventions*, IEEE TBioCAS, 12(4):824-838, 2018.
[3] T. Ota, A. Degani, D. Schwartzman, B. Zubiate, J. McGarvey, H. Choset, and M. A. Zenati, "A Highly Articulated Robotic Surgical System for Minimally Invasive Surgery", Annals of Thoracic Surgery 2009, 87:1253-1256
[4] D. Salle, P. Bidaud and G. Morel, "Optimal Design of High Dexterity Modular MIS Instrument for Coronary Artery Bypass Grafting", Proceedings of IEEE ICRA, 1276-1281, New Orleans, USA 2004.
[5] C. Rivera-Serrano, P. Johnson, B. Zubiate, R. Kuenzler, H. Choset, M. Zenati, S. Tully, and U. Duvvuri, "A Transoral Highly Flexible Robot: Novel Technology & Application", American Laryngological, 2012, 122:1067-1071.
[6] O. M. Omisore, S. P. Han, X. Jing, H. Li, Z. Li, and L. Wang, "A Review on Flexible Robotic Systems for Minimally Invasive Surgery", IEEE Trans on Systems Man Cybernetics-Systems, 2020.
[7] P. Anderson, R. Lathrop and R. Webster, "Robot-like dexterity without computers and motors: a review of hand-held laparoscopic instruments with wrist-like tip articulation", Expert Review of Medical Devices, 2016, 13(7): 661-672.
[8] M. Han, L. Zhang, J. Wang and W. Pan, "Actor-Critic Reinforcement Learning for Control With Stability Guarantee", IEEE Robotics and Automation Letters, 5(4):6217-6224, 2020.
[9] V. Agarwal *Trajectory Planning of Redundant Manipulator using fuzzy Clustering Method,* International Journal of Advanced Manufacturing Technology. 2012, 61, 727–744
[10] Y. Chen, W. Xu, Z. Li, S. Song, C. Lim, Y. Wang, H. Ren, *Safety-Enhanced Motion Planning for Flexible Surgical Manipulator Using Neural Dynamics*, IEEE Transactions on Control Systems Technology,25(5):1711-1723, 2017.
[11] A. Schmitz, P. Berthet-Rayne, and G-Z. Yang, *Endoscopic Bi-Manual Robotic Instrument Design Using a Genetic Algorithm*, IEEE IROS, Macau, China, Nov 2019.
[12] P. Berthet-Rayne and A. Schmitz, "Using Deep-Learning Proximal Policy Optimization to Solve the Inverse Kinematics of Endoscopic Instruments", IEEE Transactions on Medical Robotics and Bionics, 3(1):273-276, 2021
[13] O. Omisore, S. Han, L Ren, A. Elazab, N. Azeez, T. Talaat, L. Wang, *Deeply-Learnt Damped Least-Squares Method for Inverse Kinematics of Snake-like Robots*, Neural Networks, 107:34-47, 2018.
[14] X. Wang, X. Liu, L. Chen, and H. Hu, "Deep-learning damped least squares method for inverse kinematics of redundant robots", Measurement 171 (2021) 108821.
[15] O. Omisore *et al*, "A teleoperated snake-like robot for minimally invasive radiosurgery of gastrointestinal tumors", proceedings of the 18th IEEE ICARSC, Torres Vedras, Portugal, April, 2018.
[16] W. Wolovich and H. Elliott (1984). A Computational Technique for Inverse Kinematics. 23rd IEEE Conference on Decision and Control. Las Vegas, USA, December 12-14, 1984, p. 1359-1363.
[17] S. Buss and J. Kim (2005). Selectively Damped Least Squares for Inverse Kinematics, Journal of Graphics Tools, 10(3), 37-49.