Learning Global Inverse Kinematics Solutions for a Continuum Robot

Chapter · January 2016

DOI: 10.1007/978-3-319-33714-2_6

4 authors:

Thomas George Thuruthel
Scuola Superiore Sant'Anna

Egidio Falotico
Scuola Superiore Sant'Anna

Matteo Cianchetti
Scuola Superiore Sant'Anna

Cecilia Laschi
Scuola Superiore Sant'Anna

Learning Global Inverse Kinematics Solutions for a Continuum Robot

Thomas George Thuruthel[1], Egidio Falotico[1], Matteo Cianchetti[1] and Cecilia Laschi[1]

[1] The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, Italy

Abstract. This paper presents a learning based approach for obtaining the inverse kinematics (IK) solution for continuum robots. The proposed model learns a particular global solution for IK problem by supervised learning without any prior knowledge about the system. We have developed an approach that solely relies on the sampling method and a unique IK formulation. The convergence of the solution, practically feasible sample data requirements and adaptability of the model is shown with simulations of a redundant continuum robot.

1. Introduction

Most robotic applications rely on task space controllers. The primary objective of such controllers is to guide the end effector trajectory, in case of manipulators, or the center of mass trajectory for legged locomotion. However, since these controllers can only act directly on the actuator space, a causal mapping between the task space and actuator space is required. Inverse Kinematic mappings are used to derive the configuration space coordinates given the task space coordinates.

Inverse kinematic models have been extensively studied for rigid bodied robots by using analytical and machine learning methods. Although analytical approaches, like iterative methods and inverse transform methods, have been fairly successful, they have fared poorly for complex systems. As expected, soft robots have been even more difficult to model analytically. Owing to their infinite degrees of freedom and elastic nature, even developing forward models have turned out to be difficult. There have been few developments in developing forward models based on piecewise constant curvature models [1] and non-constant curvature models [2]. However, developing the inverse model would be even more challenging, although, there have been recent successes [3]. We believe that a machine learning approach would be better for complex soft robotic systems not only because of their ease of application and generalization ability but also because of their ability to adapt to changes in the starting system.

Like their rigid counterpart, learning the inverse kinematics model of soft bodies offers two major problems. Firstly, there exists an infinite number of possible solutions to the non-homogenous inverse kinematics equation for redundant systems. The other is related to the non-convexity of the solution set. A path based sampling approach for learning the inverse kinematics was used in [4]. Their approach of using goal babbling for sampling enabled efficient detection and resolution of inconsistent samples. Further in [5], they have used this approach to learn the inverse kinematics of the bionic handling assistant. However, this approach still does not guarantee a global solution to the IK problem and the goal directed exploration leads to highly redundant learning data. Another approach is to develop multiple locally learned models to estimate the inverse kinematics [6, 7, 8, 9]. The idea behind these approaches is the fact that IK solutions form a convex set in a spatially local set. In [6], they have augmented the input representation to the inverse model for learning an appropriate particular solution to the IK problem by adding additional cost functions. Nonetheless, their method involves learning the differential inverse kinematics which leads to drifting of error during integration. Distal supervised learning approaches have also been applied for the IK learning problem in [10, 11]. However, indirect training of the inverse model in this way is difficult due to local minima problems, instability and presence of inaccuracies in the forward model. Another interesting research is the use of structured prediction for resolving the ill-posed IK problem [12]. But this method would not adapt well to changes in the original system and would not scale well with higher dimensional systems.

In this paper we try to address the shortcomings of the existing approaches. We propose a global solution to learning the IK problem by appropriately biasing and selecting the input/output representation to the learning system, similar to [6].However, unlike [6], we learn the inverse kinematics on the position level .We are using a simple multi-layer perceptron as a function approximator, although, the method is independent of the learning architecture. We have tested the method on a kinematic simulator of the bionic handling assistant (BHA) [13]. We demonstrate the adaptable nature of the learned IK model and efficiency of the algorithm with respect to the amount of sample data required.

2. Formulation of the Inverse Kinematics Learning Problem

A straightforward representation for learning the IK of a system would be to generate samples of $(\boldsymbol{q}, \boldsymbol{x})$, where, $\boldsymbol{x} \in \Re^m$, is the vector containing the coordinates of the end effector and $\boldsymbol{q} \in \Re^n$, is the vector containing the joint space configuration vector, and to learn the mapping $\boldsymbol{x} \to \boldsymbol{q}$. Since there are infinite number of such functions, the learning system will learn a particular function depending on the sample data, averaging over samples that map to the same output $(\boldsymbol{x})$. However, naively learning such a system will lead to errors due to the non-convexity of the solution set. As mentioned in [6], the differential IK formulation (1) forms a convex solution set locally.

$$\dot{\boldsymbol{x}} = J(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{1}$$

Where, J is the Jacobian matrix. A differential IK learning system would approximate the mapping $(\dot{\boldsymbol{x}}, \boldsymbol{q}) \rightarrow (\dot{\boldsymbol{q}})$. The limitations of this approach have been discussed before.The proposed method can be understood more clearly from the Taylor series expansion of the forward kinematics equation given in (2)

$$\Delta\boldsymbol{x} = J(\boldsymbol{q})\Delta\boldsymbol{q} + \Delta\boldsymbol{q}^T H(\boldsymbol{q})\Delta\boldsymbol{q} + \ \dots \tag{2}$$

Here H is the Hessian matrix. Now, for small changes in $\boldsymbol{q}$ $(\Delta\boldsymbol{q})$, equation (2) can be approximated by (3).Note that this approximation becomes more erroneous as the value of the $\Delta\boldsymbol{q}$ increases.

$$\Delta\boldsymbol{x} \approx J(\boldsymbol{q})\Delta\boldsymbol{q} \tag{3}$$

Learning the mapping $(\Delta\boldsymbol{x}, \boldsymbol{q}) \rightarrow (\Delta\boldsymbol{q})$ would be very similar to learning the differential inverse kinematics. However, now we are no longer dealing with IK problem in the velocity level. The non-convexity of the solution set is still maintained, at least reasonably, as long as $\Delta\boldsymbol{x}$ is small (spatially localized). The non-convex properties break down if we consider the higher order elements, hence the need for the approximation. Once the mapping is learned, we can incrementally add up local solutions to reach a desired end effector point. However, this method will still have the same shortcomings of the differential IK approach. Additionally, since the value of $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{q}$ is bounded during the learning process, using input values which are greater than that used during learning while testing will lead to undesirable behavior. Consequently, we modify the learning variables to develop a more practical and effective mapping. Equation (3) can be expanded and represented as shown below:

$$J(\boldsymbol{q_i})\boldsymbol{q}_{i+1} = \ \boldsymbol{x}_{i+1} - f(\boldsymbol{q_i}) + J(\boldsymbol{q_i})\boldsymbol{q_i} \tag{4}$$

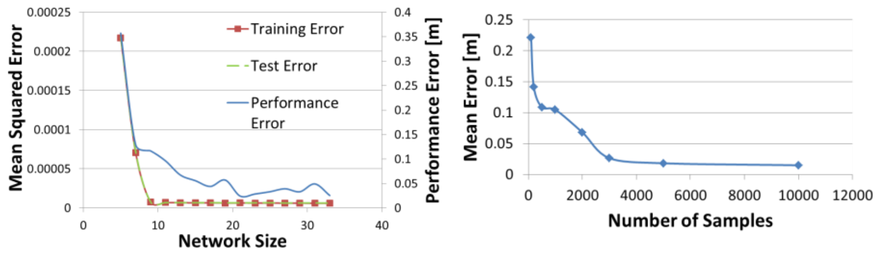
This simple rearrangement now presents the learning algorithm with a new mapping; $(\boldsymbol{x}_{i+1}, \boldsymbol{q_i}) \rightarrow (\boldsymbol{q}_{i+1})$. Where, $\boldsymbol{q}_{i+1}$ is the joint configuration that archives the end effector position $\boldsymbol{x}_{i+1}$.Now if a neural network is trained for approximating this mapping while ensuring the spatial locality in the sample data (by keeping $|\ \boldsymbol{q}_{i+1} - \boldsymbol{q_i}\ | < \epsilon$), it will have all properties of the mapping learned based on equation (3). In addition, and more importantly, now the inputs and outputs are bounded only by the kinematic constraints of the system; i.e. the inputs and outputs belongs to target space and configuration space and not a subset of it. This enables us to provide target positions for the end effector which is much farther away from local region and still obtain a corresponding joint configuration that brings the end effector closer to the target position. After each step, the end effector position will come closer to the target position as the accuracy of the IK approximator increases as $|\ \boldsymbol{x}_{i+1} - \boldsymbol{x_i}\ |$ decreases, until convergence. Therefore, because of our unique formulation, we can get IK solutions for any point in the task space irrespective of its locality from the starting point, unlike the case of differential IK solvers.

3. Training the Neural Network

For approximating the function that maps the current joint configuration and next end effector to the next joint configuration we are using a multilayer perceptron with Tan-sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. Bayesian regularization backpropagation method is used for training the neural network. The inputs and outputs are normalized in the pre-processing stage and divided randomly into sets for training and testing in the ratio 70:30 respectively. A validation test is generally not used as the algorithm prevents over-fitting by Bayesian regularization [14].

Robot Model. The training data is obtained from a kinematic model of the BHA [13]. The model uses a constant curvature approximation for modelling the continuum kinematics of the manipulator. The robot is composed of three segments; each segment is actuated by the three pneumatic actuators. The kinematic model takes in as input the length of each actuator and outputs the three dimensional coordinates of the end effector with respect to a reference frame fixed at the origin. Fig. 2a shows the schematic of the BHA. The model is a very good representation of the real BHA with a relative error of 1%.

Training data. Sample data are obtained by continuous motor babbling. The soft and continuous nature of soft robots makes it a safe exploration strategy. This would be the easiest way to ensure that all the solution space of the IK problem is reached, although, probably not the most efficient. Nevertheless, motor babbling would provide more information than goal babbling. The input/output pairs are generated ensuring that $|\ \boldsymbol{q}_{i+1} - \boldsymbol{q}_i\ | < \epsilon$, where ϵ is decided by trial and error. Keeping ϵ too low will increase the amount of samples required for learning and keeping it high will affect the convex nature of the solution set. Also, random noise, ranging from $\pm 2\%$ of the soft manipulator length ($\pm 17\text{mm}$) is added to the actual end effector position to simulate real world conditions [5]. The artificially added noise also helps in avoiding overfitting while learning. The size of the neural network is determined by training a fixed sample data with increasing network size. The training and testing errors are recorded along with the performance error of the IK solver, which is obtained by checking the solution provided by the IK solver for a fixed fifty end effector targets (Fig. 1a). Finally, the lowest network size which provides the preferred behavior is selected (20, in our case). Fig. 1b shows the performance of the IK solver with respect to the number of required samples, again, for fifty random target points. These tests are performed as follows: the manipulator starts at an upright home position [0, 0, 0.9m] and then the IK solver estimates the solutions for reaching each target point. The error shown in Fig.1b is the final distance between the target position and end effector position. The proposed method requires significantly less samples for learning the IK when compared to [5], in spite of the higher noise added.



(a) (b)

Fig. 1 a Neural network selection, **b** Performance of IK solver with number of samples

4. Simulations and Analysis

Three types of experiments are conducted on the learned IK model to validate and analyze the proposed approach. The first one is a simple ‘reaching a point’ simulation. Fifty points are randomly chosen (Fig. 2a) and the IK solver gives its estimate of the joint configurations. The forward model is used to compare the desired positions and the IK estimates. The errors along with the mean and standard deviation are shown in Fig. 2b. Since the points are not near to the starting point, the solver takes an average of 5.68 steps to converge to a value within a range of 1mm with a standard deviation of 1.504.

The second experiment is a circular trajectory tracking simulation. The target trajectory is a circle centered at [0, 0, 0.7m] with a radius of 0.2 meters and the end effector starting at the home position [0, 0, 0.9m]. The same path is followed twice. The path is discretized into 200 individual points, 100 points for one rotation. Fig. 2c shows a comparison of the path derived from the IK solver with the target path. Fig. 2d shows the magnitude of error in following the path along with the mean and standard deviation for each rotation. The large error in the beginning is because the manipulator starts from the home position which is away from the target path.

The final simulation has the same circular target trajectory, but with the last three joints locked at a fixed value. The IK solver is given no knowledge about the freezing of the three joints. Therefore, the IK solver still outputs the next configuration for all the joints in each iteration, however, only the first six joints will be modified accordingly and the last three joints will maintain their initial configuration. The trajectory followed in this new setting is shown in Fig. 2e. The corresponding absolute errors with mean and standard deviation for each rotation are shown in Fig. 2f. Note that for the last two experiments only one step is needed to get the appropriate IK solution as the target points are nearby.

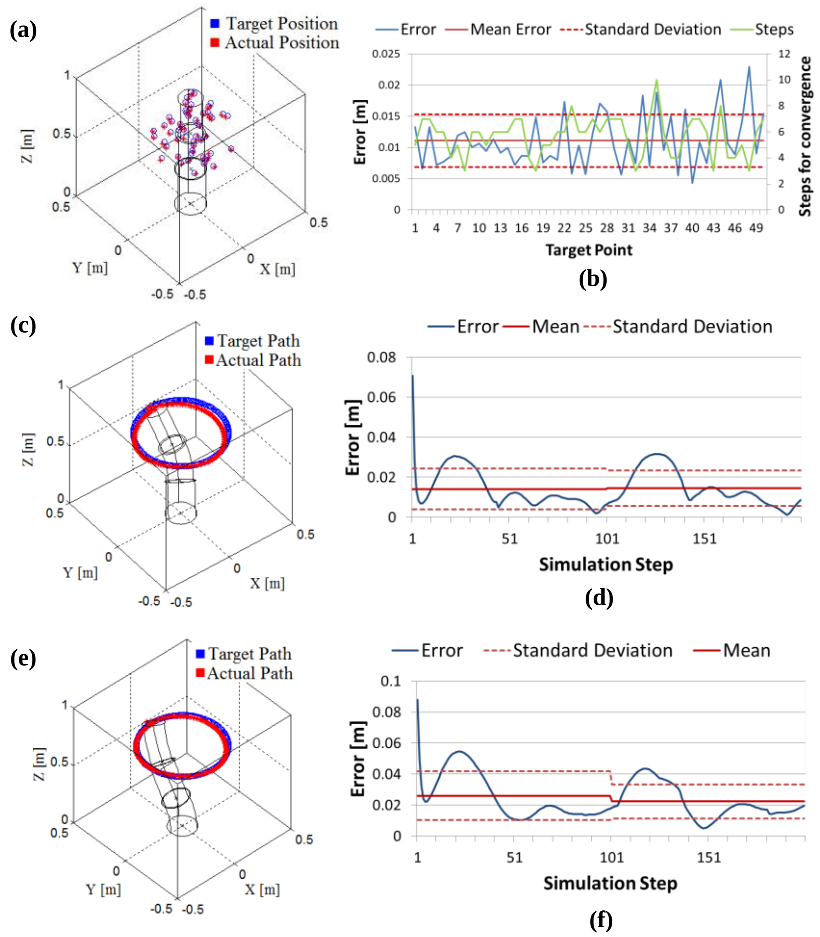


Fig. 2 a Fifty randomly selected target points and their IK solutions, **b** Error values for each target point with their steps for convergence, **c** Trajectory tracking experiment and results, **d** Error values with mean and standard deviation for each rotation (100 steps), **e** Trajectory tracking experiment with last three joints fixed, **f** Error values with mean and standard deviation for each rotation (100 steps)

The results show that solutions provided by the IK solver are still good despite losing three degrees of actuation. This shows that the IK solver is able to provide a meaningful solution at any joint configuration. However, note that this is only possible because of the redundancy present in the system. Interestingly, this means that we could exploit the redundancy of the system for executing secondary tasks, by compromising correspondingly on the primary task.

5. Conclusion

In this paper, we have proposed an approach for learning a particular global IK solution for a redundant continuum robot. Our method differs from the countless others solely based on the sampling approach and the unique IK formulation. The particular solution selected is dependent on the sample data and the generalization properties of the neural network. We have proved with the help of realistic simulations, that the learned IK solution is truly robust, stable and global. This learned IK solutions can now be used for control. Since the IK solution does not depend on the load acting on the system and other parameters like friction, this can be directly used for controlling the position of a soft robot, provided that the forward kinematics is well defined. Simple PID controllers can be used to control the joint configuration as prescribed by the IK solver. The joint dynamics can be considered independent of each other, making it easy to control each of the joint parameters separately. Moreover, the same approach can be used for learning the inverse statics of continuum robots. Researchers have been able to learn the inverse statics of continuum robots for a non-redundant case [15]. We believe that our approach can be easily extended for learning the inverse statics of a redundant continuum robot also.

6. Acknowledgement

The authors would like to acknowledge the support by the European Commission through the I-SUPPORT project (HORIZON 2020 PHC-19, #643666).The authors would like to thank Italian Ministry of Foreign Affairs, General Directorate for the Promotion of the ”Country System”, Bilateral and Multilateral Scientific and Technological Cooperation Unit, for the support through the Joint Laboratory on Biorobotics Engineering project.

Bibliography

1. Webster, R.J., Jones, B.A.: Design and kinematic modeling of constant curvature continuum robots: a review. Int. J. Robot. Res. 29, 1661–1683 (2010)
2. Renda, F., Giorelli, M., Calisti, M., Cianchetti, M. & Laschi, C.: Dynamic model of a multi-bending soft robot arm driven by cables. IEEE Trans. Robot. 30, 1109–1122 (2014)

3. Marchese, A. D., Komorowski, K., Onal, C. D., Rus, D.: Design and control of a soft and continuously deformable 2D robotic manipulation system. In Proc. IEEE International Conference on Robotics and Automation, 2189–2196 (2014)
4. Rolf, M., Steil, J.J., Gienger M.: Goal Babbling Permits Direct Learning of Inverse Kinematics. IEEE Transactions on Autonomous Mental Development, Volume 2 Issue 3, 216-229 (2010)
5. Rolf, M., Steil J.J.: Efficient Exploratory Learning of Inverse Kinematics on a Bionic Elephant Trunk. Neural Networks and Learning Systems, IEEE Transactions on, Volume 25, Issue 6, 1147 – 1160 (2013)
6. D'Souza, A., Vijayakumar, S., Schaal, S.: Learning Inverse Kinematics. Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems, Volume 1, 298 – 303 (2001)
7. Vannucci, L., Falotico, E., Di Lecce, N., Dario, P., Laschi, C.: Integrating feedback and predictive control in a Bio-inspired model of visual pursuit implemented on a humanoid robot. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9222, 256-267 (2015)
8. Susumu, E.O., Tachi, S.: Inverse kinematics learning by modular architecture neural networks. Proc. IEEE International Conference on Robotics and Automation, 1006–1012 (2001)
9. Vannucci, L., Cauli, N., Falotico, E., Bernardino, A., Laschi, C.: Adaptive visual pursuit involving eye-head coordination and prediction of the target motion. In Proceedings of the 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2014), IEEE, 541–546 (2014)
10. Melingui, A., Merzouki, R., Mbede, J.B., Escande, C., Benoudjit, N.: Neural Networks based approach for inverse kinematic modeling of a Compact Bionic Handling Assistant trunk. IEEE 23rd Int. Symposium on Industrial Electronic, 1239-1244 (2014)
11. M. Jordan and D. Rumelhart (1992). Forward models: supervised learning with distal teacher. *Cognitive Science*, vol. 16, 307–354.
12. Botond, B., Nguyen-Tuong, D., Csato, L., Bernhard, S., Peters, J.: Learning Inverse Kinematics with Structured Prediction. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 698-703 (2011)
13. Rolf, M., Steil, J.J.: Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3440 – 3446 (2012)
14. Foresee, F.D., Hagan, M.T.: Gauss-Newton approximation to Bayesian regularization. Proceedings of the 1997 International Joint Conference on Neural Networks, 1930–1935 (1997)
15. Giorelli, M., Renda, F., Calisti, M., Arienti, A., Ferri, G., Laschi, C.: Neural Network and Jacobian Method for Solving the Inverse Statics of a Cable -Driven Soft Arm with Nonconstant Curvature. IEEE Transactions on Robotics, Volume 31, Issue 4,823-834 (2015)