DATA-ORIENTED INVERSE KINEMATICS USING THREE CAMERAS' POINTS OF VIEW

Matheus Carvalho Nascimento de Souza [a], *Jessica Duarte Cardoso Nascimento* [a], *Carlos Alberto Campos da Purificação* [a], *Taniel Silva Franklin* [a]

[a] *Software, SENAI CIMATEC, Brazil.*

Abstract: Robots have advantages in operating in hard-to-reach environments for humans, but the modeling of their inverse kinematics is a complex task. Therefore, this article addresses inverse kinematics in soft robots, which present modeling and control challenges due to the non-linear properties of materials. The aim of this work was to present a data-driven inverse kinematics method using three-camera viewpoints to build a robotic skeleton. For the development of the model, three neural network topologies were used: Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP), and Transformer, with the last one presenting a better performance.

Keywords: soft robots; inverse kinematics; robot manipulator; artificial intelligence.

CINEMÁTICA INVERSA ORIENTADA A DADOS UTILIZANDO OS PONTOS DE VISTA DE TRÊS CÂMERAS

Resumo: Os robôs apresentam vantagens em operar em ambientes de difícil acesso para humanos, mas a modelagem de sua cinemática inversa é complexa. Sendo assim, este artigo aborda a cinemática inversa em robôs macios, que apresentam desafios de modelagem e controle devido às propriedades não lineares dos materiais. O objetivo desse trabalho é apresentar um método de cinemática inversa orientado a dados usando os pontos de vistas de três câmeras para construir um esqueleto robótico. Para o desenvolvimento do modelo foram utilizadas três topologias de redes neurais: Long Short-Term Memory (LSTM), Multilayer Perceptron (MLP) e Transformer, com o último apresentando um melhor desempenho.

Palavras-chave: robôs macios; cinemática inversa; manipulador de robôs; inteligência artificial.

1. INTRODUCTION

Robots have become increasingly important in commercial and industrial applications due to their ability to assist humans with dangerous and repetitive tasks. They help provide valuable support in environments that are difficult to access or require specialized personnel, reduce the risk of human injury and minimize logistical challenges. Unlike rigid manipulators, soft robots feature compliant bodies that can adapt their shapes to contact surfaces, designed for reduced impact forces and allowing for more safety with humans. The flexibility of soft robotic manipulators enables a wide range of motion, virtually unlimited in terms of degrees-of-freedom (DOF), encompassing bending, extension, contraction, twisting and bending. However, these soft materials exhibit non-linear characteristics, making it challenging to model, control and calibrate soft robots.

Furthermore, these manipulators present challenges related to modeling, control and calibration due to the peculiar properties of soft materials, including non-linearities and the presence of multiple degrees of freedom [1], [2], [3]. Another relevant aspect is that these materials have a remarkable level of stochasticity, which varies according to the specific properties of each material considered [4].

The kinematics of robot manipulators is explored through two different models: forward kinematics and inverse kinematics. Direct kinematics aims to determine the position and orientation of the robot manipulator based on the angle vector of the joints and the geometric patterns of the model. In other words, the spatial location of the final manipulator is found based on information about the joint configurations and grain characteristics of the system. Inverse kinematics, the object of study in this article, can be considered a fundamental area of robotics, as it deals with the sovereignty of the joint positions of a robotic manipulator, starting from a desired final position for its tool. It is a challenging task, as a robotic manipulator usually has multiple solutions for a single desired final position. The correct solution depends on a number of factors, such as physical limitations of the manipulator and constraints of the environment [5].

It is worth mentioning that soft robots do not control joints equipped with encoders or potentiometers, as rigid robots do, and their body deformations cannot be easily measured, which introduces complexities in the expected shape [6]. Many soft sensors have been combined with different material technologies to improve perception. Fiber optic shape detection can be considered as a promising method as it can detect stress, bending and strength, but equipment and constraints included characterization and complex strength [2], [7]. This approach can provide real-time visual shape detection using a three-camera setup and a markerless tracking algorithm to build a cloud of points that represent the shape [8].

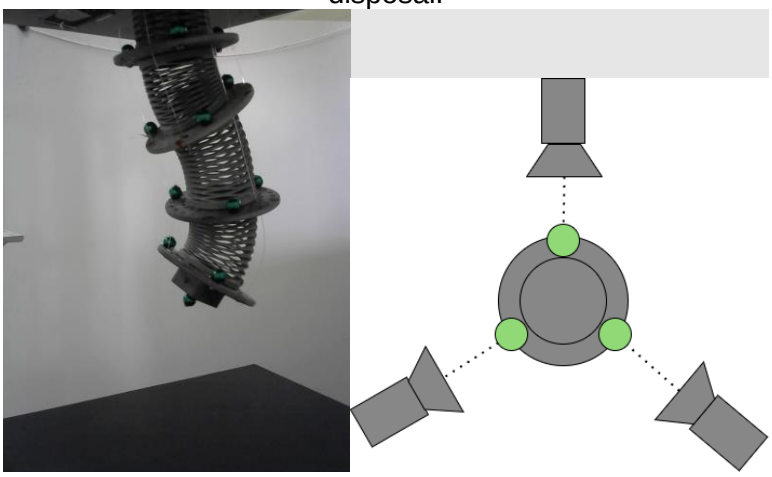
In this context, this work aims to present a data-oriented inverse kinematics method using three-camera viewpoints that estimate the shape through visual annotation to build a skeleton, using the OpenCV library. This article is organized as follows: the second topic describes which methodology was used and how the dataset was constructed. The analysis of the applied models and the main results obtained is made in the third topic. Finally, the last topic presents the conclusion, followed by the references used in the article.

2. METHODOLOGY

2.1. DATA COLLECTION SETUP

For the dataset construction, the manipulator was set up as shown in Figure 1 (a), in front of a white background and with thirteen spherical markers. Twelve markers were distributed along the manipulator's body on different sides of the manipulator at the top of each disk, 120º from one another, and a thirteenth marker in the tip of the manipulator.

Figure 1: (a) Image of the manipulator from one single camera used for the dataset, (b) Diagram of the top-down view of the cameras' setup, the manipulator, and the markers disposal.



(a) (b)

Three Logitech C270 webcams with a 720p resolution were set up facing each sequence of markers and calibrated to correct distortions as shown in Figure 1 (b). Each motor simultaneously received a signal input, and each resulting pose was captured by all cameras at the same time so that each pose would be recorded from three different angles. The signal sent to the motors is called amplitude modulated pseudo-random bit sequences (APRBS) which provide a suited nonlinear system excitation signal [9] given its square-based form, random amplitude and random time length.

2.2. DATASET ANNOTATION

For the annotation, the images obtained from each camera were put together considering the same timestamp, and each marker's x and y positions were manually annotated using a Python script. The markers of the sequences facing each camera were annotated with ids 1-4, 6-9, and 11-14, and the markers on the tip were annotated with ids 5, 10, and 15. When a marker disappeared from human sight or was too blurred to be recognized, it would be considered as missing data and its coordinates would be set to (nan, nan). In total, 3762 images were annotated. During the

exploratory data analysis, the coordinates of the tip marker were discarded from the dataset as they presented a huge number of missing values.

2.3. NEURAL NETWORK ARCHITECTURES

Three architectures for deep learning neural networks were used: Multilayer perceptron (MLP), Long Short-Term Memory (LSTM) and Transformer. The robot's pose *s* is defined through the annotated markers' coordinates, while the available motor movements are represented by *u*. The prediction of the next pose relies on both the current position of each marker and the chosen control action. The desired output is the appropriate set of movements for the manipulator in its current pose to achieve the desired pose. Being θ^* the network parameters desired, the inverse kinematics can be represented by $N(\cdot)$, as follows:

$$u_k = \mathcal{N}_{\theta^*}(s_{k+1}, s_k, u_{k-1}) \tag{1}$$

Where u_k (6 elements) is the desired output defining the motors' actions for a current state s_k (12 elements) and u_{k-1} (6 elements), that make the robot achieve the desired pose s_{k+1} (12 elements). Therefore, the neural network equation defined by (1) has 30 elements as input, and 6 elements as output representing the motors' actions.

2.3.1. MLP

According to [10] it is composed of interconnecting neurons, also called nodes, which are defined by the modeling of nonlinear mapping between an output vector and an input vector. These interconnections form a system with a combination of weights that can be trained and fine-tuned.

2.3.2. LSTM

The LSTM [11] is a type of recurrent neural network (RNN) that has the addition of mechanics for selectively forgetting and remembering information as it is trained, such as a forget gate. This capability allows it to be sensible to long-term dependencies in sequential data, for as each feature is analyzed by the model, its memory state is updated, generating a new output at each step. In this way, this type of model is especially suited for tasks that require a structured dataset, given the sequential nature of how the data is arranged.

2.3.3. Transformer

The transformer architecture, developed for solving problems related to natural language [12], is designed to process structured data as sequences. It follows a layered encoder-decoder framework, possessing self-attention procedures that allow the model to capture dependencies between inputs during each step of its training. In this case, the positional encodings convey the order of the elements, which correspond to different features during the processing of the model.

2.4. TRAINING OVERVIEW

The AI models were trained on a simple Intel (R) Core (TM) i5-3470 CPU @ 3.20GHz with 4 processors and 8GB of RAM. Different hyperparameter settings were tried and not only were shallower architectures with fewer parameters to train sufficient to model the data but of course the models were trained faster. It might have been due to the small amount of available data (only 3762 data points). The training parameters used in all three models are presented in Table 1.

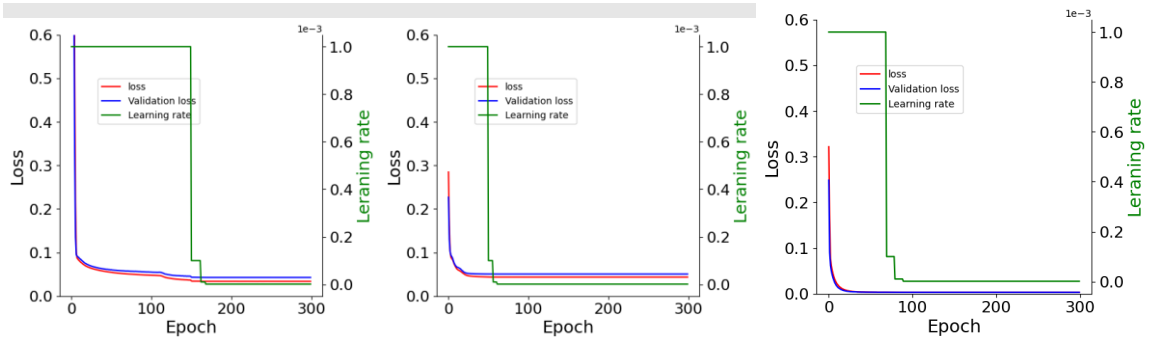
Table 1. Model evaluation metrics

Hyperparameter	Value
Batch size	32
Learning rate	10^{-3}

3. RESULTS AND DISCUSSION

The training and validation MSE results through the epochs of each model are depicted in Figure 2, illustrating the convergence time of each one.

Figure 2: MSE training and validation losses for MLP (a) LSTM (b) and Transformer (c).



(a) (b) (c)

It is noteworthy that the Transformer model converges faster than MLP and LSTM models, reaching the lowest MSE among the three. Also, none of the models were early stopped with the Keras' early stopping strategy, taking the whole number of epochs previously set to train, which was 300. Besides a faster convergence of the Transformer model during training and validation steps, it also outperformed MLP and LSTM by far on the test set. A visual comparison of the annotated vs predicted values can be seen in Figures 3, 4, and 5, while the quantitative results are shown in Table 2.

Table 2. Model evaluation metrics

	MAE	RMSE
MLP	0.14	0.17

LSTM	0.16	0.19
Transformer	0.04	0.05

Figure 3: Annotated and predicted values for MLP neural network model.

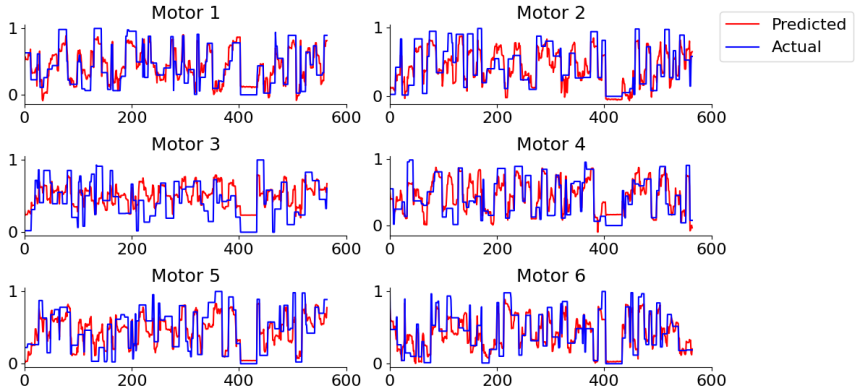


Figure 4: Annotated and predicted values for LSTM neural network model.

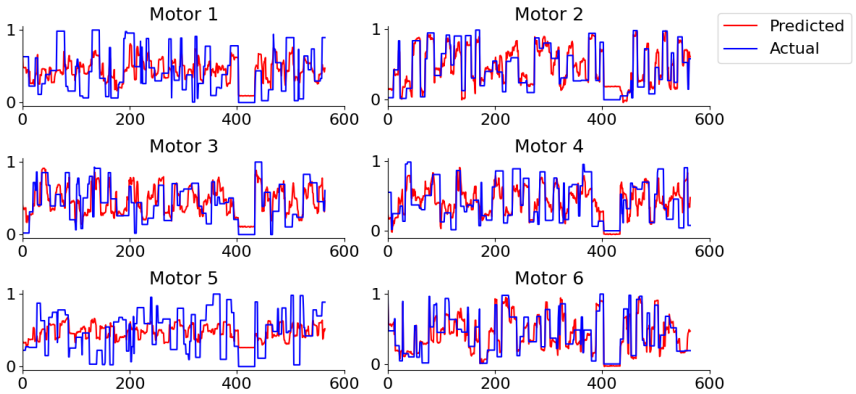
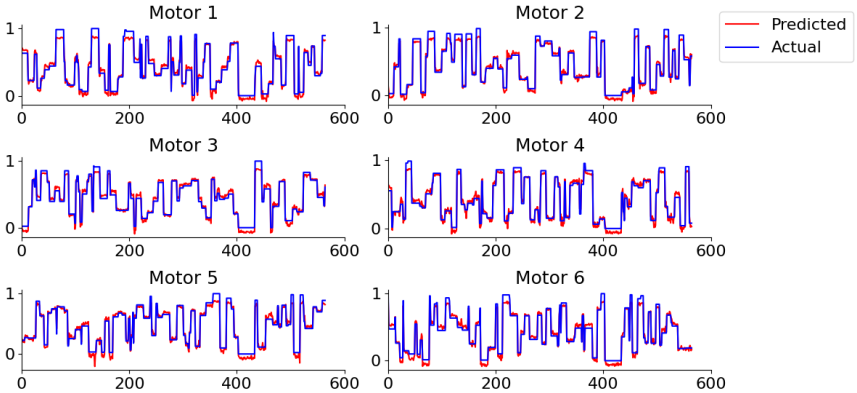


Figure 5: Annotated and predicted values for Transformer neural network model.



Abnormally, the LSTM model returned the worst performance, since this architecture is theoretically known to model sequential data better than MLP and Transformer. It is visible from the comparisons that the Transformer network generated

results much closer to the ground truth, with the other two showing clearer divergences, as well as demonstrating an especially faster convergence during the training and validation steps, standing out even more when evaluating its performance on the test set, where it significantly outperformed both MLP and LSTM. This corroborates with the results already discussed.

4. CONCLUSION

In this paper, we present a data-driven inverse kinematics approach, using three-camera viewpoints to estimate the shape of a soft robot. The objective was to overcome the inherent limitations of soft robots, which present significant challenges in terms of modeling, control and calibration due to the peculiar characteristics of their materials, such as non-linearities and the presence of multiple degrees of freedom. The OpenCV library was used as a tool for the practical implementation of the method, providing flexibility and important resources for processing images from the three cameras.

The use of a markerless tracking algorithm and the construction of a cloud of points representing the shape of the soft robot allowed us to obtain accurate information about the position and orientation of the manipulator's joints. This approach was shown to be promising for real-time visual shape detection and offers significant potential for improving inverse kinematics in soft robots.

Data-driven inverse kinematics using three-camera viewpoints could represent a significant advance in the field since it provides a shape reconstruction that could be analyzed from different points of view. With the continuous development of this method and its application in other scenarios and contexts, it is expected that the field of robotics using soft robots can reach even higher levels of accuracy and efficiency.

Acknowledgments

This research was executed in partnership between SENAI CIMATEC and Shell Brasil. The authors would like to acknowledge Shell Brasil Petróleo LTDA, the Brazilian Company for Industrial Research and Innovation (EMBRAPII), and Brazilian National Agency for Petroleum, Natural Gas and Biofuels (ANP) for the support and investments in RD\&I.

5. REFERENCES (ARIAL 12)

[1] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, “Control Strategies for Soft Robotic Manipulators: A Survey,” Soft Robotics, vol. 5, no. 2, pp. 149–163, 2018.

[2] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee et al., “Review of machine learning methods in soft robotics,” Plos one, vol. 16, no. 2, p. e0246102, 2021.

[3] H. Zhang, Y. Li, Y. Guo, X. Chen, and Q. Ren, “Control of Pneumatic Artificial Muscles with SNN-based Cerebellar-Like Model,” in International Conference on Social Robotics, 2021, pp. 824–82.

[4] F. Pique, H. T. Kalidindi, L. Fruzzetti, C. Laschi, A. Menciassi, and E. Falotico, “Controlling Soft Robotic Arms Using Continual Learning,” IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 5469–5476, 2022.

[5] R. F. Nunes, S. C. A. Mantovani, “Mapeamento da cinemática inversa de manipuladores robóticos usando RNAs configuradas em paralelo aplicado a um manipulador de 5 GDL controlado pela placa Intel® Galileo Gen 2”, pp. 1-2, 2018.

[6] F. Iida and C. Laschi, “Soft robotics: Challenges and perspectives,” Procedia Computer Science, vol. 7, pp. 99–102, 2011, proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11).

[7] I. Floris, J. M. Adam, P. A. Calderon, and S. Sales, “Fiber Optic Shape ´ Sensors: A comprehensive review,” Optics and Lasers in Engineering, vol. 139, no. September 2020, p. 106508, 2021.

[8] D. B. Camarillo, K. E. Loewke, C. R. Carlson, and J. K. Salisbury, “Vision based 3-D shape sensing of flexible manipulators,” Proceedings - IEEE International Conference on Robotics and Automation, pp. 2940– 2947, 2008.

[9] S. A. Billings, Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains. John Wiley & Sons, 2013.

[10] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” Atmospheric environment, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.