Forward/Inverse Kinematics Modeling for Tensegrity Manipulator based on Goal-conditioned Variational Autoencoder

Yuhei Yoshimitsu[1], Takayuki Osa[2], and Shuhei Ikemoto[1]

***Abstract*— This paper uses a data-driven approach to model a highly redundantly driven tensegrity manipulator's forward and inverse kinematics. The tensegrity manipulator is based on a class-1 tensegrity with 20 struts and bends by 40 pneumatic actuators whose internal pressures are independently controlled. Based on the data obtained through random trials with the robot, a VAE-based kinematics model is trained. The forward model, inverse model, and null space of kinematics are simultaneously acquired as subnetworks of the VAE-based kinematics model. Experiments confirmed that the subnetworks representing forward and inverse kinematics could be used for the end position estimation and control, respectively. In addition, the subnetwork representing null space can generate different target pressures that achieve the same end position, which was confirmed to mean variable stiffness properties similar to musculoskeletal robots.**

I. INTRODUCTION

The bodies of living organisms are not only soft but also driven in a highly redundant manner. For example, muscles are soft, and bones, which are rigid links, are connected by multi-degree-of-freedom joints and driven by even more muscles than joint degrees of freedom [1]. Biomimetics is one of the promising approaches to understanding function, and many musculoskeletal robots have been developed [2]–[6]. On the other hand, softness is being pursued in soft robotics in ways not limited to mimicking living organisms to elucidate various ways of using softness [7], [8]. Similarly, the extremely high redundancy is worth investigating in approaches not limited to biomimetics. We have developed a tensegrity manipulator to employ this idea, using tensegrity as a design that allows for increased degrees of freedom and the number of actuators while providing softness [9], [10] as shown in Fig.1.

Tensegrity is a stable three-dimensional structure formed by multiple rigid bodies under tension between each other. When all rigid bodies can be represented by at least N simple compressive members, we call it a class-N tensegrity [11]. The most common tensegrity robot rolls and/or crawls by deforming the structure. This type of tensegrity robot typically uses class-1 tensegrity [12]–[17]. Recently, there has also been active research on the use of tensegrity in manipulators, which typically use class-$(N \geq 2)$ tensegrity in this case [18]–[22]. In contrast, our tensegrity manipulator employs class-1 tensegrity, where all members do not interfere except for 3DOF pin joints. This contributes to increasing the degrees of freedom and the number of actuators while avoiding mechanical interference, which suffers the decay of motion performance and hardware failures. In addition, the use of pneumatic cylinders as actuators provides mechanical softness. On the other hand, the complexity of the structure makes it difficult to analytically model and investigate the functionality of the extremely high redundancy.

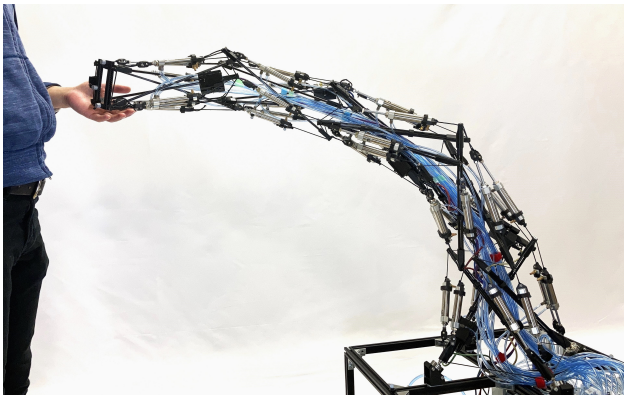

Fig. 1. The developed tensegrity manipulator. A class-1 tensegrity is employed for increasing degrees of freedom and number of actuators while providing softness.

Applying machine learning has been promising in the control of musculoskeletal and tensegrity robots and actively studied in recent years [23]–[26]. This study uses a data-driven approach to model the forward and inverse kinematics of our class-1 tensegrity manipulator. If there is redundancy in the kinematics, the inverse kinematics model needs auxiliary codes to generate a variety of outputs within the same input. At the same time, in the forward kinematics model, the auxiliary code means the information which was not needed for the modeling. To capture this nature, this paper employs a kinematics model based on Variational Aturoencoder [27] (VAE) to represent null space as a latent space by simultaneously modeling forward and inverse mappings [28], [29]. By training the VAE model based on data obtained through experiments, we evaluated the accuracy of end position estimation and control and investigated the characteristics of the null space. The results confirmed that the developed tensegrity manipulator has variable stiffness similar to that of the musculoskeletal system and that can be modeled and used with a data-driven approach.

*This work was supported by JSPS KAKENHI Grant Numbers 19K0285, 19H01122, and 21H03524.

[1]Y. Yoshimitsu and S. Ikemoto is with Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu, Kitakyushu, Fukuoka, Japan. ikemoto@brain.kyutech.ac.jp

[2]T. Osa is with Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan.

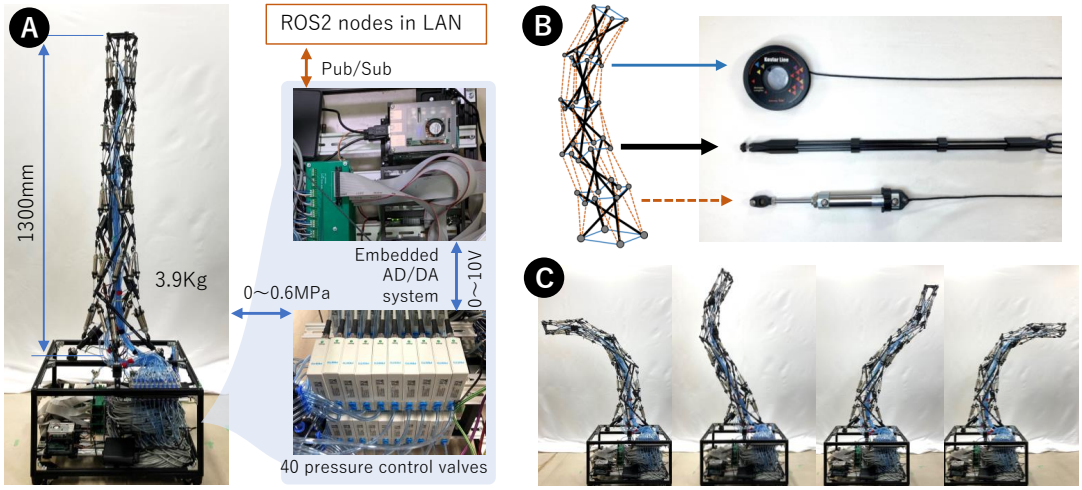

Fig. 2. The architecture and features of the developed tensegrity manipulator. A) The tensegrity manipulator's height and mass are 1300 [mm] and 3.9[Kg]. 40 pneumatic control valves regulate internal pressures of 40 pneumatic cylinders independently. The control system is compatible with ROS2 Foxy. B) The structure is based on a class-1 tensegrity. It can mainly consist of three parts: stiff tensile cables (blue) , stiff compressive struts (black), and tensile actuators (orange). C) The developed tensegrity manipulator can exhibit various bending postures.

II. DEVELOPED TENSEGRITY MANIPULATOR

Fig.2 shows the architecture and features of the developed tensegrity manipulator. The developed tensegrity manipulator has 1300 [mm] height and 3.9 [Kg] mass. The structure is based on a class-1 tensegrity in which five 4-struts tensegrity prisms are vertically stacked and connected. Of the total of 80 tensile members, 40 are replaced by pneumatic cylinders, which allow for actively bending. In the developed tensegrity manipulator, two types of pneumatic cylinders with an equal stroke length of 45 [mm] and different diameters are used. Specifically, the lower 24 cylinders have a diameter of 16 [mm] (MSPCN16-45, Misumi), and the upper 16 cylinders have a diameter of 10 [mm] (MSPCN10-45, Misumi). To facilitate the bending, the bottom of the structure (i.e. four different struts' ends) is connected to the base via sliders and ball joints. The base contains 40 pressure control valves (VEAB, FEST Inc.) so that internal pressures of 40 pneumatic cylinders can be independently controlled. The target pressure values are fed from the embedded AD/DA system that is compatible with ROS2 Foxy.

Employing a class-1 tensegrity contributes to the use of 40 pneumatic cylinders without mechanical interference. In addition, the class-1 tensegrity contributes to a robot with fewer parts because of the repetitive connections of compressive and tensile members. In the developed tensegrity manipulator, two types of struts with an equal length of 300 [mm] are used. Specifically, the lower 12 struts consist of three CFRP pipes of 5 [mm] in outer diameter, and the upper 8 struts employ only one CFRP pipe which is one of three CFRP pipes. Therefore, despite the entire structure being complex, the developed tensegrity manipulator comprises from few parts: two types of struts, two types of pneumatic cylinders, and the same type of stiff cables.

The features of the developed tensegrity manipulator, such as the flexibility of the pneumatic drive, the large number of actuators, the small number of mechanical interferences, and the small number of component parts, make analytical modeling difficult. On the other hand, even though the hardware is complex, it can reduce hardware failures. Therefore, it is suitable for collecting data through random movements without prior knowledge. These features indicates that the developed tensegrity manipulator is suitable as a platform to investigate the functions of flexibility and redundancy from a constructivist perspective.

III. VAE-BASED KINEMATICS MODEL

Fig.3 shows the VAE-based forward/inverse kinematics model employed in this study. This model is basically the goal-conditioned VAE model [29] which is an extension of the Joint VAE model [30], but no discontinuous latent variable is used. The central idea is to enforce a part of latent variables to output goals (e.g. end position/orientation) in the encoding/decoding of motion commands (e.g. target pressure) by VAE so that the remaining latent variables represent null space.

Let $\boldsymbol{u} \in \mathbb{R}^n$ denotes motor commands fed to a robot. Considering the VAE framework, the posterior/encoder $q_{\boldsymbol{\phi}}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{u})$ and the likelihood/decoder $p_{\boldsymbol{\theta}}(\boldsymbol{u}|\boldsymbol{y}, \boldsymbol{z})$ are represented by neural networks parameterized by $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ where $\boldsymbol{y}$ and $\boldsymbol{z}$ both denote latent vectors. The Joint VAE model minimizes the objective function as following:

$$\mathcal{J} = \mathbb{E}_{\boldsymbol{u}\sim\mathcal{D}}\left[\mathcal{L}_{\text{joint}}(\boldsymbol{\theta}, \boldsymbol{\phi})\right]. \tag{1}$$

The $\mathcal{L}_{\text{joint}}(\boldsymbol{\theta}, \boldsymbol{\phi})$ is defined as following:

$$\begin{aligned}\mathcal{L}_{\text{joint}}(\boldsymbol{\theta}, \boldsymbol{\phi}) \quad = \quad & \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{u})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{u}|\boldsymbol{y}, \boldsymbol{z})\right] \\ & -\gamma|D_{\text{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{y}|\boldsymbol{u})||p(\boldsymbol{y}) - C_y| \\ & -\gamma|D_{\text{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{u})||p(\boldsymbol{z}) - C_z|, \end{aligned} \tag{2}$$

where C_y and C_z represent the information capacities that are gradually increased during training, and γ is a coefficient.

In the VAE-based forward/inverse kinematics model, we explicitly train a part of latent variables $\boldsymbol{y}$ in a supervised

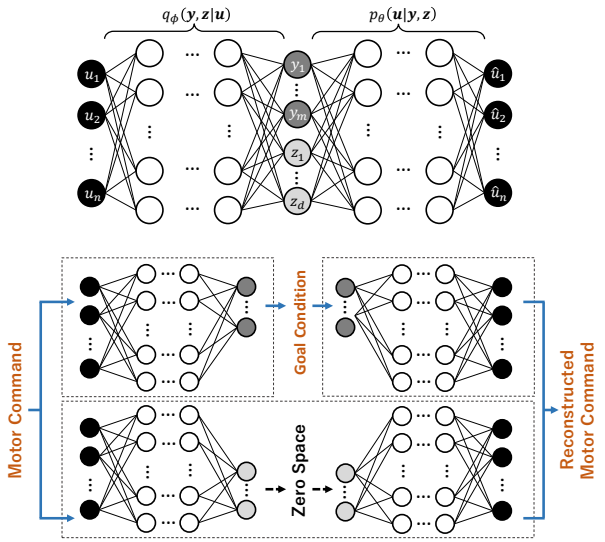

Fig. 3. The schematic of the model for forward/inverse kinematics modeling learning system. A VAE is trained by data of motor commands, and a part of the latent space is trained to be corresponding goal conditions. The rest of the latent space is not trained explicitly but will express the null space of the mapping between motor commands and goal conditions through the training of VAE.

manner. For the encoder, this auxiliary supervised learning enables forward kinematics modeling. On the other hand, for the decoder, this auxiliary supervised learning forces a one-to-many mapping from $\boldsymbol{y}$ to $\boldsymbol{u}$, which induces $\boldsymbol{z}$ to have a complementary representation, i.e., null-space modeling, enabling inverse kinematics modeling as the entire decoder. The objective function can be written as:

$$\mathcal{J} = \mathbb{E}_{\{\boldsymbol{u},\boldsymbol{y}\}\sim\mathcal{D}}\left[\mathcal{L}_{\rm gc}(\boldsymbol{\theta},\boldsymbol{\phi}) + \log q_{\boldsymbol{\phi}}(\boldsymbol{y}|\boldsymbol{u})\right]. \tag{3}$$

where

$$\begin{aligned}\mathcal{L}_{\rm gc}(\boldsymbol{\theta},\boldsymbol{\phi}) \quad = \quad & \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{u})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{u}|\boldsymbol{y},\boldsymbol{z})\right] \\ & -\gamma|D_{\rm KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{u})||p(\boldsymbol{z}) - C_z|. \end{aligned} \tag{4}$$

The second term in Eq.3 is the likelihood function to represent the goal conditions given in the data $\mathcal{D}$. We use the reparametrization trick in [27] for the continuous latent variable $\boldsymbol{z}$.

In this study, the $\boldsymbol{u}$ means target pressures of 40 pneumatic cylinders of the tensegrity manipulator, and the $\boldsymbol{y}$ means three-dimensional positions of the tip. Once the model in Fig.3 is trained, the following functions become available:

1) By the decoder, from the target end position and an adequate $\boldsymbol{z}$, the target pressure to achieve the given target is obtained (the inverse kinematics model).
2) By the encoder, from the actual current pressure, the estimated end position that converges at equilibrium is obtained (the forward kinematics model).
3) By the decoder, from the target end position and a variety of latent code candidates $\boldsymbol{z}$, a variety of target pressures to achieve the given target with different pressures is obtained (change in vectors belonging to the null space).

TABLE I

THE NETWORK STRUCTURE.

Layer name	Output dim	Activation function
Input	96 (motor command dim)	ReLU
Hidden 1	520	ReLU
Hidden 2	1080	ReLU
Hidden 3	520	ReLU
Hidden 4	96	ReLU
Hidden 5	3 (task dim) + 37 (latent dim)	Linear
Middle	96	ReLU
Hidden 6	520	ReLU
Hidden 7	1080	ReLU
Hidden 8	520	ReLU
Hidden 9	96	ReLU
Output	40 (motor command dim)	Linear

IV. EXPERIMENTAL SETUP

This section presents specific information on the experimental setup, including network structure, training method, and data sampling.

A. Network structure and training method

Table.I shows the specific information on the network structure of the VAE-base kinematics model. Because the developed class-1 tensegrity manipulator has 40 pneumatic cylinders, the input/output dimension is set to 40. The latent space appears on the middle layer where has 40 dimensions. Because the input/output and middle layers' dimensions are identical and the other hidden layers have bigger dimensions, no information is lost due to dimensional compression. Three out of the 40 dimensions in the middle layer are given supervised data corresponding to the motor commands. Therefore, the latent space that will express the null space has 37 dimensions (i.e. $u \in \mathbb{R}^{40}$, $y \in \mathbb{R}^3$, and $z \in \mathbb{R}^{37}$ hold).

The layers are fully connected. These weights are optimized upon the objective function defined in Eq.3 and 4. The C_z is scheduled to increase linearly from 0 to 5.0 from the start of the training until the 20000 iteration and to remain unchanged thereafter. For the training, the ADAM optimizer is employed.

B. Data sampling

The training data consists of pairs of target pressures and corresponding end positions in the equilibrium posture $\mathcal{D} = \{\boldsymbol{u}_{(t)}, \boldsymbol{y}_{(t)}\}$ where $t = 1, 2, \cdots, N$. The training data were collected using the following procedure:

1) Generate a random target pressure $u_i \sim U(0.1, 0.6)$ [MPa] where u_i indicates the i-th element of the $\boldsymbol{u}$.
2) Feed the random target pressure, and wait 3 seconds for the transient response to subside.
3) Store the target pressure $\boldsymbol{u}_{(t)}$ and the corresponding end position $\boldsymbol{y}_{(t)}$ as a record of $\mathcal{D}$.
4) Return to 1).

To measure the end position, we used the Motion Capture System (8 Optitrack Prime X13 cameras with Motive software. NatNet SDK is used for ROS2 compatible data provision). On the tensegrity manipulator's side, a thin, light

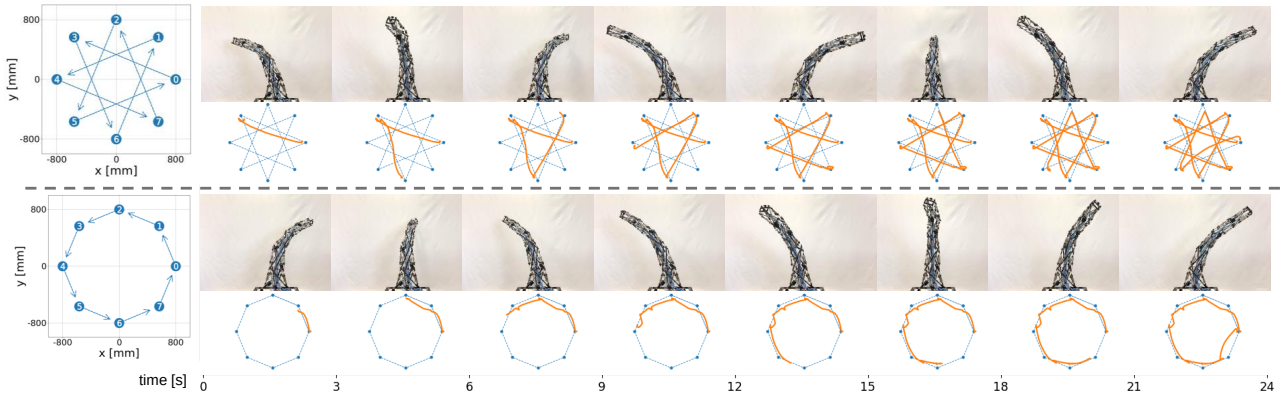

Fig. 4. Sequential snapshots and the end trajectories when given two different orders via eight points on a circle. The eight points are equally spaced on a horizontal circle with a diameter of 1600[mm] with a height of 900[mm]. These points are fed to the trained VAE-based kinematics model to obtain corresponding target pressures. The latent variables $\boldsymbol{z} = [z_i]$, $i = 1 \cdots 37$ were set to $z_i = 3.0$ and $z_i = 0.0$ at the top and bottom results, respectively.

plastic plate with three reflecting markers was firmly attached to the rectangular cable loop at the tip. We sampled $N = 38,800$ records in total by actually moving the developed tensegrity manipulator.

V. RESULTS

To verify the trained VAE-based kinematics model, we separately tested the encoder and decoder as the forward and inverse kinematics models, respectively. In this section, we will explain in the following order: A) feedforward control of the end position using the obtained inverse kinematics model, B) the end position estimation using the obtained forward kinematics model, and C) null space representation of the redundancy obtained in the latent space.

A. Feedforward control of the end position using the obtained inverse kinematics model

Fig.4 shows sequential snapshots and trajectories of the end position control by the inverse kinematics model (the decoder of trained VAE). To verify that the decoder represents the inverse kinematics model, we gave target end positions $\boldsymbol{y}$ with two different patterns, i.e. star (top) and circular (bottom) shapes, from eight points on a horizontal circle with a diameter of 1600[mm] with a height of 900[mm]. All elements of the latent variables $\boldsymbol{z} = [z_i]$, $i = 1 \cdots 37$ were set to $z_i = 3.0$ and $z_i = 0.0$ for the star and circular shapes, respectively. The outputs were directly used for the target pressure values and fed to the embedded AD/DA system of the tensegrity manipulator. As with the data sampling process, it waited 3 seconds for each new target pressure given.

In Fig.4, the given target end positions are roughly realized for both transitions of target end positions. The three-dimensional average reaching error in Fig.4 is approximately 90[mm]. Note that this control is feedforward, and there is no reaching error feedback loop. Therefore, it confirmed that the inverse kinematics model was acquired as the decoder part of the trained VAE.

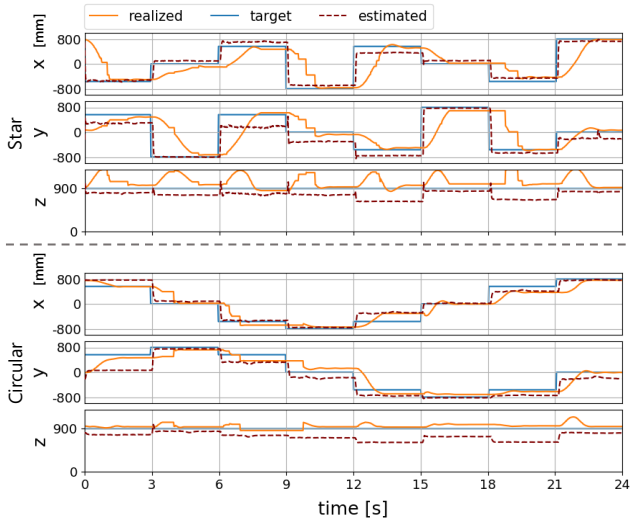

Fig. 5. Control and estimation results of the end position. The horizontal and vertical axes indicate the elapsed time and positions in each direction of the Cartesian coordinate system. The top/bottom are the results when 8 points are passed through star/circular shapes, respectively.

B. End position estimation using the obtained forward kinematics model

Fig.5 shows time-series data of the tensegrity manipulator's end positions in each direction of the Cartesian coordinate system during its control. These graphs show target/realized/estimated end positions. The target and realized positions are corresponding to movements shown in Fig.4. On the other hand, the estimated positions were obtained by feeding realized (measured) pressure values to the encoder part of the trained VAE. It is noteworthy that the VAE was trained based on the desired pressure values, but the encoder received the realized pressure values for the position estimation.

In Fig.5, the estimated end position quickly followed changes in the target end position. It would be counter-intuitive that the encoder can immediately estimate where the end will go despite the realized end position being delayed.

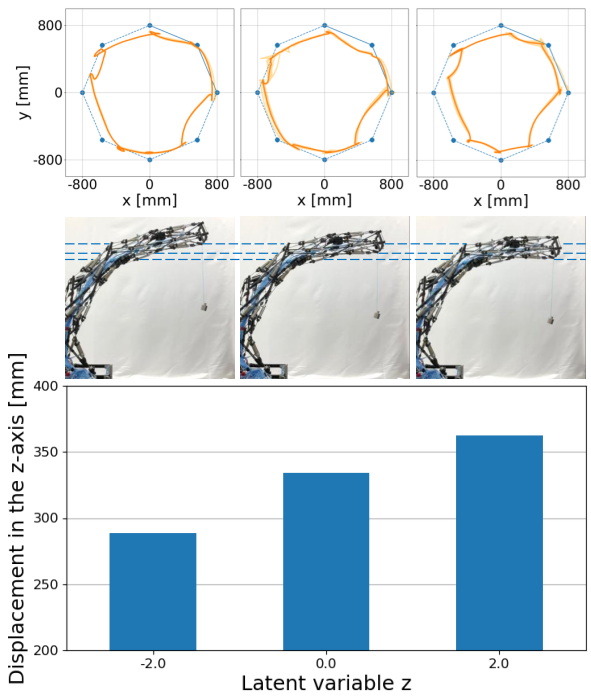

Fig. 6. Verification of change in stiffness due to change in latent variables. Top) Results of controlling the end position with different latent variables. The left, middle, and right are given the latent variables $z_i = -2.0$, 0, and 2.0, respectively. Middle) Posture change of tensegrity manipulator at the same posture, different latent variables, and the same vertical downward load. The load is applied by hanging a 200 [g] weight. The initial posture is made by giving $\boldsymbol{y} = [600, 0, 1100]$. Bottom) Vertical displacement of the end position.

It implies that pressure control valves can quickly realize target pressure values even if the tensegrity manipulator is moving. This point merits further investigation, but at least we have confirmed that the encoder part of the trained VAE acquired the forward kinematics model.

C. null space representation of the redundancy obtained in the latent space

Fig.6 shows the effect of varying the latent variables during control by the decoder part. In the experiment, at first, we examined the reaching behavior with three different latent variables $z_i = \{-2, 0, 2\}$. The result appears in the top row of Fig.6. It shows that changes in the latent variables $\boldsymbol{z}$ do not impact the accuracy of the end position control. Therefore, it means that the latent variables $\boldsymbol{z}$ represent the null space of the kinematics that the VAE-based model learned.

Next, we examined the physical meaning of the change in the latent variables. For this purpose, we fed three different latent variables $z_i = \{-2, 0, 2\}$ and the same target end position $\boldsymbol{y} = [600, 0, 1100]$ to the decoder to obtain target pressure values to make the same initial end position with different internal states. Then, a 200 g weight was hung from the end to measure the amount of vertical displacement. Three loaded postures with three different internal states are depicted in the middle row of Fig.6, and the amount of vertical displacement is shown in the bottom row of Fig.6. These results indicate that changes in the latent variables

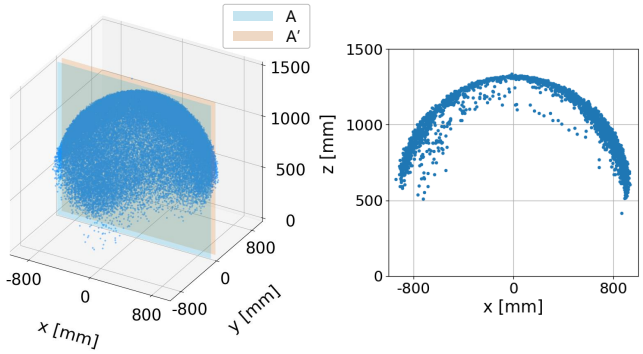

Fig. 7. The distribution of $\boldsymbol{y}$ stored in the supervised data. The left figure shows the 3D scatter plot of $\boldsymbol{y}$ in the Cartesian coordinate system. The right figure shows the points contained between sections A and A'.

$\boldsymbol{z}$ modify the stiffness of the tensegrity manipulator. Therefore, it suggests that the developed tensegrity manipulator has variable stiffness properties similar to musculoskeletal robots.

VI. DISCUSSION

Fig.7 shows the distribution of target end positions y stored in the supervised data $\mathcal{D}$ sampled in the procedure explained in sectionIV-B. In this figure, the end of the tensegrity manipulator looks almost uniformly distributed on a certain sphere surface. The variation in the orthogonal direction (the thickness direction of the sphere surface) confirms that the end moves three-dimensionally. However, the distribution along the orthogonal direction is highly biased toward the sphere surface. This shows the bias in motions that the developed tensegrity manipulator innately has.

This feature of $\mathcal{D}$ implies that the kinematic relationship along the orthogonal direction is more challenging to model than on the sphere surface. Because movements along the orthogonal direction can be achieved by taking "C" or "S" shapes, this also implies that generating different pressures, which realize the same end position but different bending postures, is more challenging than varying the stiffness. In fact, this paper successfully showed the variable stiffness but not postural variation with the same end position. Focusing on incremental data sampling procedure to improve the null space representation is an important future work for this research.

Although Fig.4 showed that the inverse kinematics model could control the end position, the positional error showed different tendencies for two different patterns. It assumes that pneumatic cylinder hysteresis induced this problem. Therefore, considering the previous target pressure values in the VAE-based model will improve the control accuracy, and this will also be future work.

VII. CONCLUSION

In this study, we used the VAE-based kinematics model for expressing the redundant forward/inverse kinematics of the developed class-1 tensegrity manipulator. The VAE-based kinematics model aims to represent null space with a part

of latent variables by training the rest part with supervised data. Experiments using the trained VAE-based kinematics model confirmed that the forward/inverse kinematics were acquired in its encoder/decoder subnetworks. In addition, by modulating the latent variables which had not been trained in a supervised manner, different pressures that conduce the same end position were obtained. The loading experiment confirmed that these pressure values have different stiffness, thereby showing that the developed tensegrity manipulator has variable stiffness properties similar to musculoskeletal robots.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers 19K0285, 19H01122, 19K20370, and 21H03524.

REFERENCES

[1] D. A. Neumann, *Kinesiology of the musculoskeletal system-e-book: foundations for rehabilitation*. Elsevier Health Sciences, 2016.

[2] A. Hitzmann, H. Masuda, S. Ikemoto, and K. Hosoda, "Anthropomorphic musculoskeletal 10 degrees-of-freedom robot arm driven by pneumatic artificial muscles," *Advanced Robotics*, vol. 32, no. 15, pp. 865–878, 2018. [Online]. Available: https://doi.org/10.1080/01691864.2018.1494040

[3] S. Kurumaya, K. Suzumori, H. Nabae, and S. Wakimoto, "Musculoskeletal lower-limb robot driven by multifilament muscles," *ROBOMECH Journal*, vol. 3, no. 1, p. 18, 2016. [Online]. Available: https://doi.org/10.1186/s40648-016-0061-3

[4] T. Kozuki, H. Toshinori, T. Shirai, S. Nakashima, Y. Asano, Y. Kakiuchi, K. Okada, and M. Inaba, "Skeletal structure with artificial perspiration for cooling by latent heat for musculoskeletal humanoid kengoro," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 2135–2140.

[5] G. Martius, R. Hostettler, A. Knoll, and R. Der, "Compliant control for soft robots: Emergent behavior of a tendon driven anthropomorphic arm," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 767–773.

[6] Y. Nakanishi, Y. Asano, T. Kozuki, H. Mizoguchi, Y. Motegi, M. Osada, T. Shirai, J. Urata, K. Okada, and M. Inaba, "Design concept of detail musculoskeletal humanoid "kenshiro" - toward a real human body musculoskeletal simulator," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, Conference Proceedings, pp. 1–6.

[7] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, p. 467, 2015. [Online]. Available: https://doi.org/10.1038/nature14543

[8] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Science Robotics*, vol. 1, no. 1, 2016. [Online]. Available: http://robotics.sciencemag.org/content/1/1/eaah3690.abstract

[9] S. Ikemoto, K. Tsukamoto, and Y. Yoshimitsu, "Development of a modular tensegrity robot arm capable of continuous bending," *Frontiers in Robotics and AI*, vol. 8, p. 347, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2021.774253

[10] Y. Yoshimitsu, K. Tsukamoto, and S. Ikemoto, "Development of pneumatically driven tensegrity manipulator without mechanical springs," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 3145–3150.

[11] R. E. Skelton and M. C. Oliveira, *Tensegrity Systems*. Springer Nature, 2009.

[12] C. Paul, F. J. Valero-Cuevas, and H. Lipson, "Design and control of tensegrity robots for locomotion," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 944–957, 2006.

[13] M. Shibata and S. Hirai, "Rolling locomotion of deformable tensegrity structure," *Mobile Robotics*, pp. 479–486, 2009. [Online]. Available: https://doi.org/10.1142/9789814291279_0059

[14] K. Kim, D. Moon, J. Y. Bin, and A. M. Agogino, "Design of a spherical tensegrity robot for dynamic locomotion," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 450–455.

[15] M. Vespignani, J. M. Friesen, V. SunSpiral, and J. Bruce, "Design of superball v2, a compliant tensegrity robot for absorbing large impacts," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 2865–2871.

[16] J. Rieffel and J.-B. Mouret, "Adaptive and resilient soft tensegrity robots," *Soft Robotics*, vol. 5, no. 3, pp. 318–329, 2018. [Online]. Available: https://doi.org/10.1089/soro.2017.0066

[17] K. Kim, A. K. Agogino, and A. M. Agogino, "Rolling locomotion of cable-driven soft spherical tensegrity robots," *Soft Robotics*, vol. 7, no. 3, pp. 346–361, 2020. [Online]. Available: https://doi.org/10.1089/soro.2019.0056

[18] S. Lessard, D. Castro, W. Asper, S. D. Chopra, L. B. Baltaxe-Admony, M. Teodorescu, V. SunSpiral, and A. Agogino, "A bio-inspired tensegrity manipulator with multi-dof, structurally compliant joints," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 5515–5520.

[19] E. Jung, V. Ly, N. Cessna, M. L. Ngo, D. Castro, V. SunSpiral, and M. Teodorescu, "Bio-inspired tensegrity flexural joints," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Conference Proceedings, pp. 5561–5566.

[20] D. Fadeyev, A. Zhakatayev, A. Kuzdeuov, and H. A. Varol, "Generalized dynamics of stacked tensegrity manipulators," *IEEE Access*, vol. 7, pp. 63 472–63 484, 2019.

[21] D. Wei, T. Gao, X. Mo, R. Xi, and C. Zhou, "Flexible bio-tensegrity manipulator with multi-degree of freedom and variable structure," *Chinese Journal of Mechanical Engineering*, vol. 33, no. 1, p. 3, 2020. [Online]. Available: https://doi.org/10.1186/s10033-019-0426-7

[22] V. Ramadoss, K. Sagar, M. S. Ikbal, J. H. L. Calles, R. Siddaraboina, and M. Zoppi, "Hedra: A bio-inspired modular tensegrity robot with polyhedral parallel modules," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, Conference Proceedings, pp. 559–564.

[23] K. Kawaharazuka, K. Tsuzuki, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Musculoskeletal autoencoder: A unified online acquisition method of intersensory networks for state estimation, control, and simulation of musculoskeletal humanoids," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2411–2418, 2020.

[24] J. Campbell, A. Hitzmann, S. Stepputtis, S. Ikemoto, K. Hosoda, and H. B. Amor, "Learning interactive behaviors for musculoskeletal robots using bayesian interaction primitives," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 5071–5078.

[25] D. Surovik, K. Wang, M. Vespignani, J. Bruce, and K. E. Bekris, "Adaptive tensegrity locomotion: Controlling a compliant icosahedron with symmetry-reduced reinforcement learning," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 375–396, 2019. [Online]. Available: https://doi.org/10.1177/0278364919859443

[26] M. Zhang, X. Geng, J. Bruce, K. Caluwaerts, M. Vespignani, V. SunSpiral, P. Abbeel, and S. Levine, "Deep reinforcement learning for tensegrity robot locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Conference Proceedings, pp. 634–641.

[27] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[28] H. Masuda, A. Hitzmann, K. Hosoda, and S. Ikemoto, "Common dimensional autoencoder for learning redundant muscle-posture mappings of complex musculoskeletal robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 2545–2550.

[29] T. Osa and S. Ikemoto, "Goal-conditioned variational autoencoder trajectory primitives with continuous and discrete latent codes," *SN Computer Science*, vol. 1, no. 5, p. 303, 2020. [Online]. Available: https://doi.org/10.1007/s42979-020-00324-7

[30] E. Dupont, "Learning disentangled joint continuous and discrete representations," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 708–718.