Kinematics of Nonzero Constant Mean Curvature Surface Models using a Kenmotsu-type Representation Formula

Noriyasu Iwamoto[1]

***Abstract*— A simple functional expression giving multiple geometries is important for the kinematics of a smooth-surfaced robot. Such a representation gives the functions of tangent vectors and Jacobian matrices and reduces the computational time of forward and inverse kinematics. In this paper, we construct the kinematics of robotic surfaces using an equation for rotational surfaces called the Kenmotsu-type representation formula. The formulas in this paper switch between various nonzero constant mean curvature surfaces, namely, spheres, cylinders, unduloids, and nodoids. Forward kinematics and numerical simulations confirm the short computation time of these formulas; moreover, the local coordinate system on the robot is a Cartesian coordinate system. We also present an iterative algorithm for inverse kinematics based on the Newton–Raphson method.**

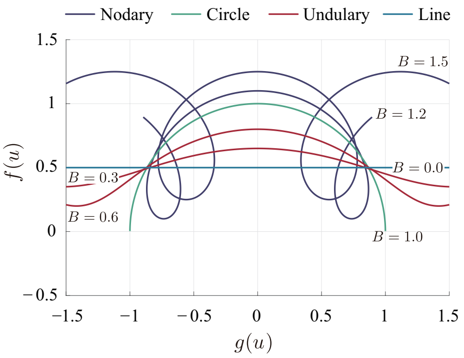


Fig. 1. Generatrix for $H = 1$.

I. INTRODUCTION

For many years, scientists have mathematically and physically investigated the shape of living organisms and physical phenomena and their deformations. Recently, many attempts have been made to control the deformation of an object's shape. For example, words such as "active printed material" [1] and "morphing" [2] are trending. Research on so-called robotic surfaces has also emerged in robotics [3][4]. Different control targets in different fields must solve the same problems related to surface-control technology; therefore, constructing such technology is important. The present study focuses on robotic surfaces.

The space composed of variables controlled by a robotic surface is called the actuator space, whereas the space composed of the realizable positions and postures on a point or shape of the robot to be controlled is called the task space. Robotic control requires transformation between the two spaces, but a direct transformation does not find the commonalities between robots with different structures. A configuration space consisting of geometric variables of surfaces should be established as an intermediate space between the two spaces. The transformation between the configuration and task spaces is considered to be common to any robotic surface.

However, when a robotic surface deforms under shear, two major problems prevent the construction of control theory. First, a surface representation requires six geometric parameters, which are the first and second fundamental quantities [5], and they must satisfy the integrability condition. Second, a shear deformation transforms the local coordinate system of the robot to an oblique coordinate system, which cannot be directly treated with conventional robotics theory based on the Cartesian coordinate system. To solve these problems, we must find a surface model with the following characteristics: (i) describable with few parameters, (ii) can represent multiple geometrical shapes, and (iii) ensures that the local coordinate system on the model is an orthogonal coordinate system.

Here, we propose a surface model possessing with these three characteristics and consider its forward and inverse kinematics. The kinematics have a short computational time and can directly utilize the formulation of rigid-body robot kinematics.

II. Equation of Rotational Surfaces

The Kenmotsu-type representation formula, first derived by Kenmotsu [6], is an equation for rotational surfaces with a mean curvature of a given function. After applying a coordinate transformation to this formula, we obtain an equation for a surface with isothermic coordinates and constant mean curvature.

A. Kenmotsu-type Representation Formula

A rotational surface is described by $\boldsymbol{p}(\bar{u}, \bar{v}) = [f(\bar{u}) \cos \bar{v}, f(\bar{u}) \sin \bar{v}, g(\bar{u})]^T$, where $\bar{u}$ is the arc length parameter of a generatrix $(f(\bar{u}), g(\bar{u}))$ of the rotational surface. For a function $H(\bar{u})$ defined on an interval, the generatrix with mean curvature $H(\bar{u})$ is given by

$$\begin{aligned} f(\bar{u}) &= \sqrt{(V - c_1)^2 + (W + c_2)^2} \\ g(\bar{u}) &= \int_0^{\bar{u}} \frac{(W + c_2)V' - (V - c_1)W'}{\sqrt{(V - c_1)^2 + (W + c_2)^2}} du, \end{aligned} \tag{1}$$

[1] Noriyasu Iwamoto is with Faculty of Textile Science and Technology, Shinshu University, 3-15-1, Tokida, Ueda City, Nagano 386-8567 Japan iwamoto@shinshu-u.ac.jp

where the functions V and W are respectively defined as

$$\begin{aligned} V(\bar{u}) &= \int_0^{\bar{u}} \sin\left(2\int_0^u H(t)dt\right) du \\ W(\bar{u}) &= \int_0^{\bar{u}} \cos\left(2\int_0^u H(t)dt\right) du. \end{aligned}$$

The prime symbol denotes partial differentiation by $\bar{u}$. Eq. (1) is called the Kenmotsu-type representation formula and c_1 and c_2 are real numbers.

Assuming $H \neq 0$, the functions V and W are

$$V(\bar{u}) = \frac{1}{2H}(1 - \cos(2H\bar{u})), \quad W(\bar{u}) = \frac{1}{2H}\sin(2H\bar{u}),$$

respectively. Using the above expressions and the incomplete elliptic integrals of Types 1 and 2, given by

$$\begin{aligned} F(k,\varphi) &= \int_0^{\varphi} \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} \\ E(k,\varphi) &= \int_0^{\varphi} \sqrt{1-k^2\sin^2\theta}\; d\theta, \end{aligned}$$

respectively, we obtain the following equation [7]:

$$\begin{aligned} f(\bar{u}) &= \beta\sqrt{1-\gamma^2\sin^2(H\bar{u})} \\ g(\bar{u}) &= \alpha F(\gamma, H\bar{u}) + \beta E(\gamma, H\bar{u}), \end{aligned} \tag{2}$$

where

$$\alpha = \frac{1-B}{2H}, \quad \beta = \frac{1+B}{2H}, \quad \text{and} \quad \gamma = \frac{2\sqrt{B}}{1+B}.$$

Here, the variables B determines the geometric shape.

$$B = \sqrt{(2c_1H-1)^2 + (2c_2H)^2}$$

$\bar{u}$ in Eq. (2) is the shift of $\bar{u}$ in Eq. (1) by a specific value. The curves for $H = 1$ generated by Eq. (2) are shown in Fig. 1.

B. Coordinate Transformation

A rotational surface $\boldsymbol{p}(\bar{u},\bar{v})$ obtained using Eq. (2) is not an isothermal coordinate system because $\|\partial\boldsymbol{p}/\partial\bar{u}\| \neq \|\partial\boldsymbol{p}/\partial\bar{v}\|$. If a local coordinate system is a Cartesian coordinate system at all points of a robot, it must be an isothermal coordinate system. Therefore, we apply the following coordinate transformation:

$$u = \frac{1}{\ell}\int_0^{\bar{u}} \frac{\sqrt{f'(u)^2 + g'(u)^2}}{f(u)} du, \quad v = \frac{1}{\ell}\bar{v}$$

when ℓ represents the elongation of the robot. Substituting the above inverse function into Eq. (2), we obtain

$$\begin{aligned} f(\ell u) &= \beta\,\mathrm{dn}(\gamma, \beta H\,\ell u) \\ g(\ell u) &= \alpha\beta H\,\ell u + \beta\mathscr{E}(\gamma, \beta H\,\ell u), \end{aligned}$$

where the functions dn and $\mathscr{E}$ respectively denote the dn function (a Jacobi elliptic function) and the Jacobi epsilon function. If the amplitude of $u = F(k,\varphi)$, defining the incomplete integral of the 1st kind, is $\varphi = \mathrm{am}(k,u) = F^{-1}(k,u)$, then dn and $\mathscr{E}$ are respectively given by:

$$\begin{aligned} \mathrm{dn}(k,u) &= \sqrt{1-k^2\sin^2(\mathrm{am}(k,u))} \\ \mathscr{E}(k,u) &= E(k,\mathrm{am}(k,u)). \end{aligned}$$

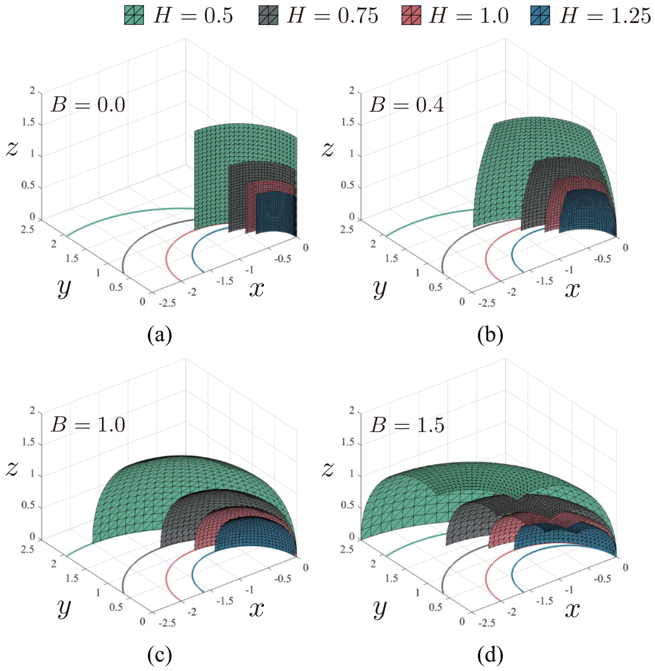


Fig. 2. Surface shapes when $\ell = 1$ in Eq. (3).

Finally, setting

$$\boldsymbol{p} = \begin{bmatrix} f(\ell u)\cos(\ell v) - f(0) \\ \mathrm{sgn}(H)\,f(\ell u)\sin(\ell v) \\ \mathrm{sgn}(H)\,(g(\ell u) - g(0)) \end{bmatrix}, \tag{3}$$

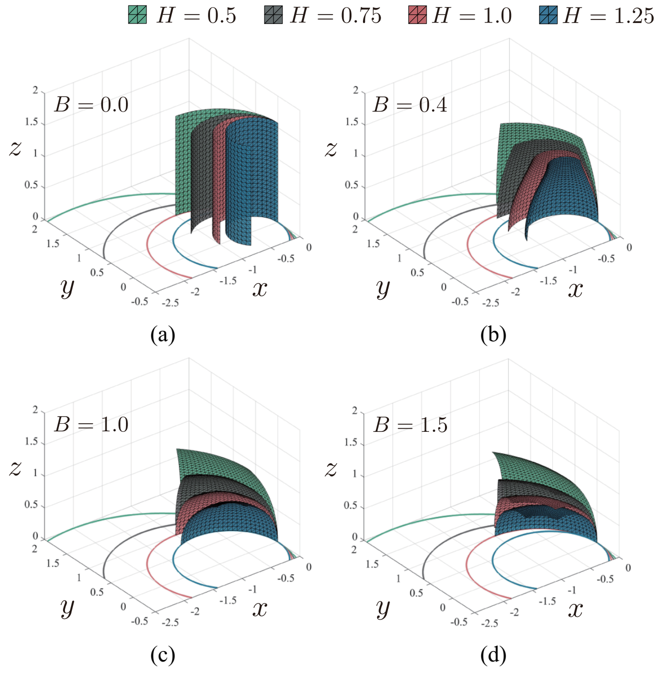
we have $\|\partial\boldsymbol{p}/\partial u\| = \|\partial\boldsymbol{p}/\partial v\| = \beta^2\,\mathrm{dn}^2(\gamma, \beta H\ell u)$; that is, $\boldsymbol{p}$ is an isothermal coordinate system. Note that $\partial\boldsymbol{p}/\partial v$ has no value in the $z-$direction.

In the following, we assume that the mean curvature is positive ($H > 0$).

When $B = 0$, $\boldsymbol{p} = 1/(2H)[\cos(\ell v), \sin(\ell v), \ell u]^T$, which describes a cylindrical surface. When $B = 1$, $\boldsymbol{p} = 1/H[\cos(\ell v)\,\mathrm{sech}(\ell u), \sin(\ell v)\,\mathrm{sech}(\ell u), \tanh(\ell u)]^T$, which describes a spherical surface. When $0 < B < 1$ and $B > 1$, the surface is called an unduloid and a nodoid, respectively. Fig. 2 shows the surface shapes given by Eq. (3) with $B = 0$ (cylinder), 0.4 (unduloid), 1.0 (sphere), and 1.5 (nodoid). In these drawings, the range of (u,v) was set to $u = [0, \pi/2], v = [0, \pi/2]$ and ℓ was set to 1.

The mean curvature H controls the bending quantity of the surface. In Fig. 2, the mean curvatures H of the green, black, red, and blue surfaces are 0.5, 0.75, 1.0, and 1.25, respectively. For reference, a circle with radius $1/H$ is shown in the $x-y$ plane of each case. Note that the mean curvature of a sphere is the reciprocal of the sphere's radius and the edges of the surface are circles (Fig. 2(c)). The mean curvature of a cylinder is one-half the inverse of the radius, so the surface edge of $H = 0.5$ in Fig. 2(a) follows a circle of radius 1.

From Fig. 2, we observe that the surface size increases with decreasing mean curvature. As the mean curvature



(a) (b)

(c) (d)

Fig. 3. Surface shapes when ℓ is set to give $\pi/2$ as the edge length of $u = 0$.

approaches zero, the surface becomes enormously large, which is inappropriate for a bending robot. Therefore, the elongation rate ℓ is controlled to prevent such expansion. As an example, Fig. 3 shows the results of setting the elongation rate as

$$\ell = \frac{\pi}{2} \Bigg/ \int_0^{\pi/2} \|\boldsymbol{p}_v\| dv \tag{4}$$

which constrains the length of the $u = 0$ edge of the surface to $\pi/2$. In all geometries, the shape approaches a plane as H decreases, preventing excessive expansion. This construct appears to represent the natural deformation of a bending robot.

III. Forward Kinematics

Forward kinematics gives the position and orientation of an arbitrary point on a surface in the base coordinate system. Both the position and orientation can be obtained from the homogeneous transformation matrix between the base coordinate system and a coordinate system described at each point. Let the tangent vectors be $\boldsymbol{p}_u = \partial \boldsymbol{p}/\partial u$ and $\boldsymbol{p}_v = \partial \boldsymbol{p}/\partial v$, and the unit normal vector be $\boldsymbol{n} = (\boldsymbol{p}_u \times \boldsymbol{p}_v)/\|\boldsymbol{p}_u \times \boldsymbol{p}_v\|$. Now place a coordinate system consisting of two unit tangent vectors and a normal vector on the surface. As $\boldsymbol{p}$ is a function, both $\boldsymbol{p}_u$ and $\boldsymbol{p}_v$ can also be expressed as functions. The homogeneous transformation matrix between the base coordinate system and the coordinate system at a point can be expressed as

$${}^{b}T(u, v) = \begin{bmatrix} \mathrm{sgn}(H)\,\frac{\boldsymbol{p}_u}{\|\boldsymbol{p}_u\|} & \mathrm{sgn}(H)\,\frac{\boldsymbol{p}_v}{\|\boldsymbol{p}_v\|} & \boldsymbol{n} & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

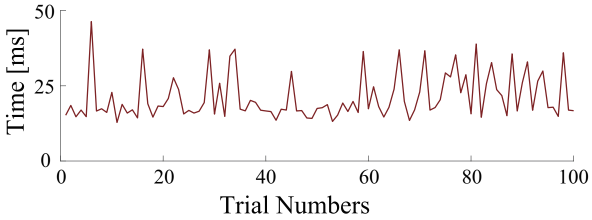


Fig. 4. Computational times of forward kinematics of 625 points on the surface, performed for 100 (B, H) pairs.

A. Numerical Simulations

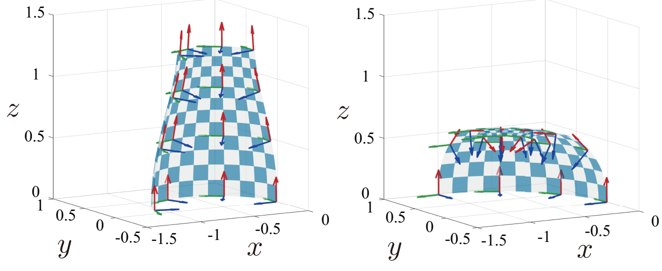
The forward kinematics include approximate computations of elliptic integrals and elliptic functions. As inverse kinematics require the repeated computation of forward kinematics, their computational time depends on that of the forward kinematics. Additionally, forward kinematics must realize a Cartesian coordinate system at any point on the robot. Hence, we investigated the time requirements of forward kinematics and the realization of an orthogonal coordinate systems.

The numerical simulations were conducted in MATLAB R2019a. The dn function was computed using MATLAB's ellipj function. To compute the Jacobi epsilon function, we require an am function and an incomplete integral of the 2nd kind. The am function was computed as $\mathrm{am} = \tan^{-1}(\mathrm{sn}/\mathrm{cn})$ using the values of the sn and cn functions obtained by ellipj. In the following numerical simulations, the values of the am function lie within $[-\pi, \pi]$ to avoid self-intersections of the surface, and the discontinuities resulting from the function atan2 are ignorable. The tolerance of ellipj was set to 2.2204×10^{-16}. The incomplete integrals of the 2nd kind were solved by trapezoidal integration using the trapz function.

We randomly generated 100 pairs of B and H, and determined ℓ by Eq. (4). For each pair of B and H, we computed the homogeneous transformation matrices of 625 points on the surface. The measured computational times of the forward kinematics calculation are plotted in Fig. 4. Recovering a surface from the first and second fundamental quantities requires solving a large-scale simultaneous equation, which typically takes more than one second. All trials in the present case were completed within 50 milliseconds, implying that one forward-kinematics calculation was computed within a short time. Fig. 5 shows the surface shapes at $(B, H, \ell) = (0.243, 1.14, 1.84)$ and $(B, H, \ell) = (1.49, 1.28, 1.03)$ and the local coordinate systems of 16 points. The maximum value of $(\boldsymbol{p}_u/\|\boldsymbol{p}_u\|) \cdot (\boldsymbol{p}_v/\|\boldsymbol{p}_v\|)$ was of the order of 10^{-16} and the two tangent vectors can be considered orthogonal.

IV. Inverse Kinematics

Let the robot's configuration parameters be $\boldsymbol{q} = [B, H, \ell]^T$. The inverse-kinematics problem is defined as follows: Find $\boldsymbol{q}$ such that a given point (u_s, v_s) on the robot coincides with a given position $\boldsymbol{p}^t \in \mathbb{R}^3$. The error function is denoted by $\boldsymbol{e}_{\mathrm{pos}} = \boldsymbol{p}^t - \boldsymbol{p}(u_s, v_s)$, and a point for which $\boldsymbol{e}_{\mathrm{pos}} = \boldsymbol{0}$ is explored by iterative computation. Let



(i)

(ii)

Fig. 5. Shapes and local coordinate systems of two surfaces obtained by forward kinematics.

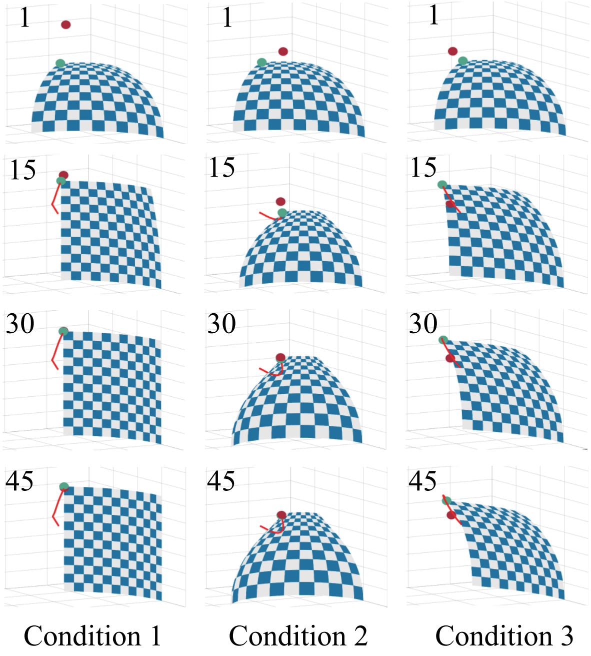


Fig. 6. Deformations of a surface after 1, 15, 35, and 45 steps of inverse kinematics.

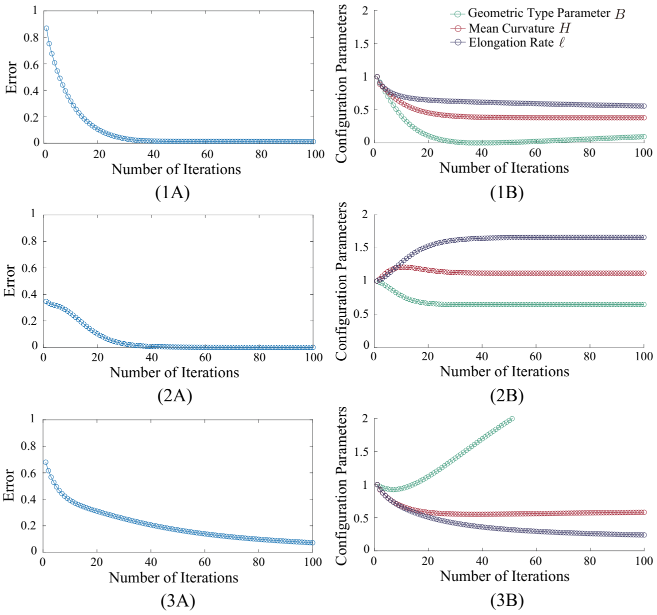


Fig. 7. Stepwise residuals (1A)(2A)(3A) and configuration parameters (1B)(2B)(3B) of inverse kinematics.

the Jacobian matrix be

$$J(\boldsymbol{q}) = \frac{\partial \boldsymbol{e}_{\text{pos}}}{\partial \boldsymbol{q}} = -\begin{bmatrix} \frac{\partial \boldsymbol{p}}{\partial B} & \frac{\partial \boldsymbol{p}}{\partial H} & \frac{\partial \boldsymbol{p}}{\partial \ell} \end{bmatrix} \tag{6}$$

and apply the Newton–Raphson method to the update law:

$$\boldsymbol{q} := \boldsymbol{q} - \delta J^{-1} \boldsymbol{e}_{\text{pos}}, \tag{7}$$

where δ is the step width.

The inverse kinematics algorithm first obtains $\boldsymbol{p}(u_s, v_s)$ using forward kinematics. It then computes the residual $\boldsymbol{e}$. Next, the Jacobian matrix is obtained by Eq. (6) and Eq. (7) is calculated. We set $\delta = 0.1$. The above process is repeated until $\boldsymbol{q}$ satisfies $\boldsymbol{e} = \boldsymbol{0}$. As $\sqrt{B}$ in γ and $H \neq 0$ (see Sec. IIA), $B \geq 0$ and $H > 0$ must be satisfied. For each step in this simulation, we enforced the correction $B = 0$ when $B < 0$ and $H = 0.0001$ when $H \leq 0$.

A. Numerical Simulations

For the point $(u_s, v_s) = (\pi/2, \pi/2)$, we set the target point as $\boldsymbol{p}^t = [-0.547, 1.07, 1.23]^T$ (Condition 1), $\boldsymbol{p}^t = [-1.23, 0.480, 1.48]^T$ (Condition 2), and $\boldsymbol{p}^t = [-0.755, 1.03, 0.91]^T$ (Condition 3). In all cases, the initial value of the configuration parameter was $(1, 1, 1)^T$. The deformation results of the three inverse kinematics simulations are shown in Fig. 6. At the upper left corner of each result is the number of steps. The green and red spheres represent $\boldsymbol{p}(u_s, v_s)$ and $\boldsymbol{p}^t$, respectively, and the red line describes the trajectory of $\boldsymbol{p}(u_s, v_s)$. In all simulations, $\boldsymbol{p}(u_s, v_s)$ approached $\boldsymbol{p}^t$ as the steps progressed. Panels (1A), (2A), and (3A) of Fig. 7 show the stepwise residuals, and panels (1B), (2B), and (3B) show the variations in the configuration parameters under Conditions 1, 2, and 3, respectively. In Fig. 7(1A) and (2A), the error norm gradually decreases and $\boldsymbol{p}(u_s, v_s)$ reaches the target position at approximately 50 steps under Conditions 1 and 2. Under both of these conditions, the surface geometry at the 200th step was unduloid. Under Condition 1 (Fig. 7(1B)), the geometry became a cylinder ($B = 0$) before transitioning to unduloid ($B > 0$). Under Condition 3, the variable B increased as the steps progressed, converging to 4.13 after approximately 500 steps and becoming nodoid. At that time, $\boldsymbol{p}(u_s, v_s)$ reached the target position. Empirically, the convergence speed appeared to slow down when the final geometry was a nodoid. The times of computing 100 steps were 0.0921 s under Condition 1 and 0.0775 s under Condition 2. Under Condition 3, the time of computing 500 steps was 0.153 s. These times are within the practical range if the robot does not require high-speed control

V. Discussion

The proposed forward kinematics is a way to avoid the integrability condition for surfaces. The important geometric quantities of a surface are six variables. The first fundamental quantities E, F, and G and the second fundamental quantities

L, M, and N. The surfaces targeted herein can be represented by three variables E, L, and N out of these six variables. Given these three variables, the shape of the surface can be reconstructed by integrating the Gauss–Weingarten formula along the coordinate curve. The integration corresponds to solving a system of simultaneous equations and generally takes more than one second of computation time. However, to integrate the Gauss–Weingarten formula, we must explore E, L, and N, satisfying the integrability conditions, Gauss equations and Codazzi–Mainardi equations. The work is not trivial, and inverse kinematics based on fundamental quantities is impractical. The proposed forward kinematics requires Jacobi elliptic functions and incomplete integrals of the 2nd kind, that is, integrals along curves only. If a generatrix for a surface of revolution is obtained, then we are guaranteed to construct a surface, allowing us to compute fast forward and inverse kinematics.

Unfortunately, the proposed representation does not allow for inverse kinematics on the attitude of a robot's point. When the rotation matrix of the attitude on point (u, v) is

$$R(u,v) = \begin{bmatrix} \frac{\boldsymbol{p}_u}{\|\boldsymbol{p}_u\|} & \frac{\boldsymbol{p}_v}{\|\boldsymbol{p}_v\|} & \boldsymbol{n} \end{bmatrix}, \tag{8}$$

and the rotation matrix of the target attitude is R_d, the error vector is taken as follows:

$$\boldsymbol{e}_{\mathrm{att}}(\boldsymbol{q}) = \boldsymbol{a}(R_d R^T), \tag{9}$$

where $\boldsymbol{a}(R)$ is a function to compute the angle-axis vector of a rotation matrix R. In this case, the partial derivative of the error vector with respect to the mean curvature H is a zero vector.

$$\frac{\partial \boldsymbol{e}_{\mathrm{att}}}{\partial H} = \boldsymbol{0}. \tag{10}$$

Note that H does not control the surface shape but only its size. This indicates that only two of the three variables for attitude can be controlled. Controlling the attitude requires the addition of a new variable to Eq. （3）.

VI. CONCLUSIONS

This paper presented the forward and inverse kinematics of a surface model using functions of rotational surfaces that can unify the representations of cylinders, spheres, unduloids, and nodoids. These kinematics provide a mechanism-independent theory between the configuration space and task space of robotic surfaces. In future work, the proposed inverse-kinematics algorithm should be studied in further detail. For example, the update laws of the inverse-kinematics algorithm for rigid-body robots use the Gauss–Newton or Levenberg–Marquadt method [8]. By examining the details of the algorithm, we can expect to reduce the solution time.

This work was supported by Grant for Scientific Research from TAKEUCHI Scholarship Foundation and JSPS KAKENHI Grant Numbers JP21K1427.

REFERENCES

[1] D. Raviv et al. "Active Printed Materials for Complex Self-Evolving Deformations", Smart Materials and Structures, vol. 31, no. 5, 2022. Scientific Reported, vol. 4, no. 7422, 2014.

[2] R. Guseinov et al. "Programming temporal morphing of self-actuated shells", Nature Communications vol. 11, no. 237, 2020.

[3] Wei Xiao et al., "Modeling and analysis of soft robotic surfaces actuated by pneumatic network bending actuators", Smart Materials and Structures, vol. 31, no. 5, 2022.

[4] N. Iwamoto, H. Arai, and A. Nishikawa. "Surface Robots based on S-Isothermic Surfaces." 2021 IEEE International Conference on Robotics and Automation, 2021.

[5] M. P. do Carmo, "Differential Geometry of Curves & Surfaces", Dover Publications, 2016.

[6] K. Kenmotsu, "Surfaces of Revolution with Prescribed Mean Curvature," Tohoku Mathematical Journal, Second Series, vol. 32, no. 1, pp. 147-153, 1980.

[7] J. Inoguchi, "Surface Geometry and Integrable Systems," Asakura Publishing, 2015 (in Japanese).

[8] T. Sugihara, "Solvability-unconcerned inverse kinematics by the Levenberg-Marquadt method", IEEE Transactions on Robotics, vol. 27, no. 5, pp. 984-991, 2011.