Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Spatiotemporal Light Navigation System for Tracking Control of Mobile Robots

Takahiro Okuda * **Yuki Minami** * **Masato Ishikawa** *

* *Osaka University, 2-1, Yamadaoka, Suita, Osaka 565-0871, JAPAN*
(e-mail: okuda@eom.mech.eng.osaka-u.ac.jp,
{minami, ishikawa}@mech.eng.osaka-u.ac.jp).

Abstract: In this paper, we propose a spatiotemporal light navigation system for tracking control of a low-performance robot. The spatiotemporal light navigation system is composed of a projector and a mobile robot equipped with a light sensor. The projector casts a binary image on the field, and the mobile robot detects the information of the image via the light sensor and is supposed to decide its action by means of a simple embedded logic. First, we formulated the tracking control problem as a problem to design the casting image and sensor position of the robot. Then, we proposed a design method of casting images based on halftone image processing and analyzed the stability of the tracking control system. Finally, we experimented with verifying the effectiveness of the proposed method.

Keywords: Mobile robots, Light, Navigation, Image processing, Quantization

1. INTRODUCTION

There have been intensive attentions to the possibility of applying micro-robots and nano-robots to robotics or medical fields. Usually, those robots have only low-performance sensors and actuators such as low-resolution sensors and ON/OFF actuators, so they lack sufficient control and communication capabilities. Therefore, the challenge here is about how to achieve intelligent autonomous behavior in such low-function robots.

Some researchers attempted to control microbes or small-sized robots by using phototaxis Aaron et al. (2013); Shahrokhi et al. (2017); Morse et al. (1998); Frangipane et al. (2018). For example, previous studies in Aaron et al. (2013); Shahrokhi et al. (2017); Morse et al. (1998) investigated a method in which small robots equipped with light sensors achieved several tasks by changing the light distribution. In addition, in Frangipane et al. (2018), they controlled the formation of *E. coli* by changing the light distribution, exploiting the behavior *E. coli* gathers in bright spots. In these studies, it was important to change the light distribution appropriately to achieve target tasks such as formation, object manipulation, and so on.

However, it has not been deeply explored how exactly light distribution should be changed in order to achieve a specified target task. It is likely that different target tasks will require appropriately different light distributions. In addition, the properties of robots that can easily achieve the target tasks vary depending on the light distribution. Therefore, it is necessary to consider how to change the light distribution according to the desired target task, and how to design the robot such that it can easily achieve the target task in the given light distribution.

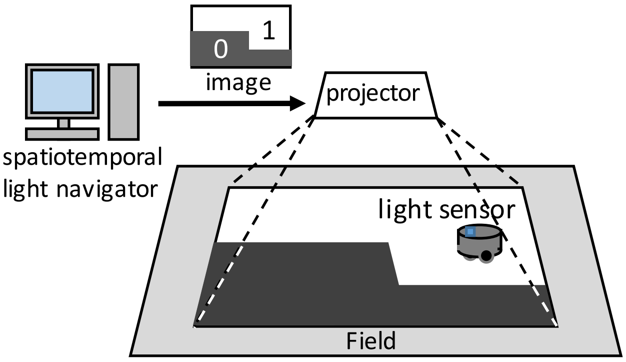


Fig. 1. Conceptual diagram of spatiotemporal light navigation system.

In this study, we develop a *spatiotemporal light navigation system*, which is composed of a projector and a mobile robot equipped with a light sensor, shown in Fig. 1. For the system, we propose a method to achieve a target task by controlling light distribution. As a first step, we consider a case where the robot can take two actions: turning left and right, as an example of simple robots. The robot turns right when it detects light; otherwise, it turns left. First, we propose a virtual control law that can be easily realized even with such a simple robot. Next, we design the sensor position of the robot and the light distribution to realize the virtual control law. The proposed method is based on the Lyapunov stability analysis and halftone image processing algorithm Robert (1987). Then, we verify the effectiveness and robustness of the proposed method through several experiments.

This paper has the following two contributions. One is to formulate the tracking control problem of a mobile robot with a simple embedded logic as a problem to design light distribution and sensor position of the robot. The other is

⋆ This research was partly supported by JSPS KAKENHI (16H06094) and HAYAO NAKAYAMA Foundation for Science & Technology and Culture.

to propose a design algorithm aided by Lyapunov stability analysis and halftone image processing.

2. SPATIOTEMPORAL LIGHT NAVIGATION SYSTEM

In this study, we focus on a spatiotemporal light navigation system, as illustrated in Fig. 1. The system is composed of a projector and a mobile robot equipped with a light sensor which can detect only “bright” or “dark.” In this system, time-series of black and white images are cast onto the field via the projector. In other words, the system changes the light distribution on the field temporally and spatially. Then, the mobile robot detects the information of the binary image via the light sensor and determines its action using an extremely simple embedded logic. Therefore, the action of the robot is determined by the behavior rule of the robot and the light distribution. The system has the following advantages:

- Even though the robot has no *localization* (to estimate its position) capability, it can perform position-dependent actions – owing to the fact that the detected light carries position-related information.
- Various tasks are possible with a fixed robot rule, by simply changing the projected image.

This study has two aims: One is to design a *spatiotemporal light navigator*, which generates binary images according to a specified task. The other is to design a simple robot that can easily perform tasks under this system.

3. PROBLEM FORMULATION

In this section, we formulate the spatiotemporal light navigation problem, which determines the binary image to cast and the required sensor position of the robot.

3.1 Model of the mobile robot

In this study, we employ a differential two-wheeled robot, illustrated in Fig. 2. We define the position and the orientation of the robot as (x, y) and θ, respectively. Moreover, we define the translational speed and rotational speed of the robot as v and ω, respectively. Then, the relationship between the state $[\dot{x}\ \dot{y}\ \dot{\theta}]^\top$ and input $[v\ \omega]^\top$ is expressed by:

$$G: \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{1}$$

In addition, the mobile robot has only one light sensor. We locate the light sensor ahead of the center of the robot by distance d, which is a designable parameter. We define the position as (x_s, y_s), then the following equations hold:

$$\begin{aligned} x_s &= x + d\cos\theta, \\ y_s &= y + d\sin\theta. \end{aligned} \tag{2}$$

Furthermore, we assume that the sensor has limited accuracy. Specifically, we assume that the sensor returns 1 when it encounters light and 0 otherwise. Here, we consider a two-dimensional bounded region $\boldsymbol{P}_x \times \boldsymbol{P}_y \subset \boldsymbol{R}^2$ and assume that a binary image $\phi : \boldsymbol{P}_x \times \boldsymbol{P}_y \to \{1, 0\}$ is cast on the plane $\boldsymbol{P}_x \times \boldsymbol{P}_y$. Note that $\phi(x, y) = 1$ means white pixel and $\phi(x, y) = 0$ means black pixel. Then, the sensor value $s \in \{1, 0\}$ is expressed by

$$\begin{aligned} s &= S(d; \phi, x, y, \theta) \\ &:= \phi(x_s, y_s) \\ &= \phi(x + d\cos\theta, y + d\sin\theta). \end{aligned} \tag{3}$$

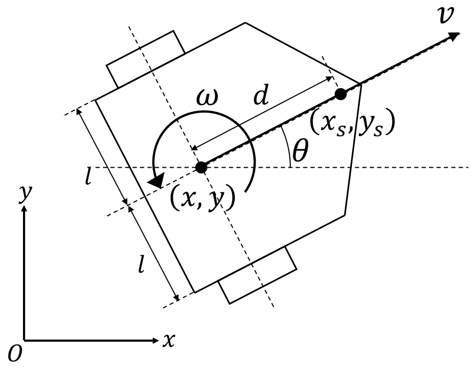


Fig. 2. Model of the mobile robot.

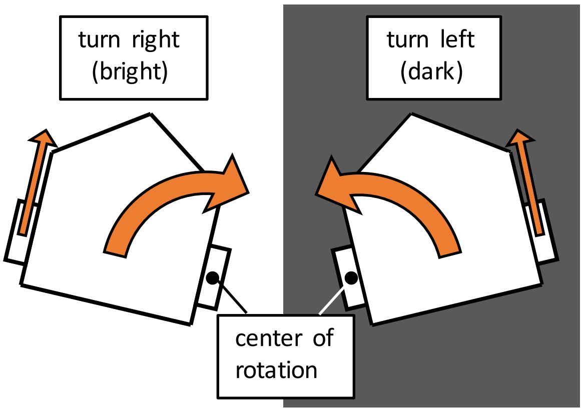


Fig. 3. Behavior rule of the mobile robot.

The controller K_{robo} of the robot determines the input $[v\ \omega]^\top$ depending on the sensor value s according to the following equation:

$$K_{\mathrm{robo}}: \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{cases} \begin{bmatrix} V \\ -\dfrac{V}{l} \end{bmatrix} & \text{if } s = 1, \\ \begin{bmatrix} V \\ \dfrac{V}{l} \end{bmatrix} & \text{if } s = 0, \end{cases} \tag{4}$$

where l is the half distance between the two wheels and V is the constant speed with which the robot moves forward. This equation indicates that the left wheel rotates if $s = 1$, and the right wheel rotates if $s = 0$, as illustrated in Fig. 3.

3.2 Formulation of design problem

We consider the target path $y = f(x)$ that the robot tracks, where $f : \boldsymbol{P}_x \to \boldsymbol{P}_y$ is a continuous function. We aim to generate a binary image ϕ on the bounded region $\boldsymbol{P}_x \times \boldsymbol{P}_y$ to track the reference curve specified by f. In other words, we design a spatiotemporal light navigator K_{navi}, which converts f to a binary image ϕ. Moreover,

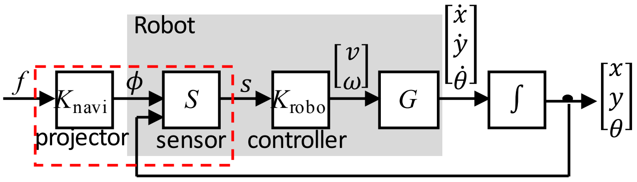

Robot
f
K_{navi}
ϕ
S
s
K_{robo}
$\begin{bmatrix} v \\ \omega \end{bmatrix}$
G
$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$
$\int$
$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$
projector
sensor
controller

Fig. 4. Block diagram of the system.

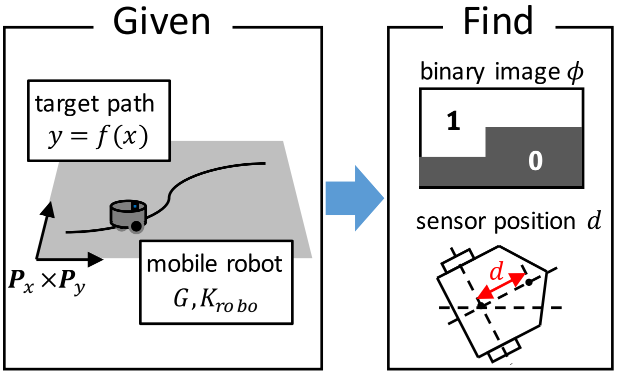


Fig. 5. Conceptual scheme of design problem.

we aim to design d, the distance between the robot and the sensor. The block diagram of this system is shown in Fig. 4, and we design K_{navi} and a part of S. In summary, we consider the following problem as shown in Fig. 5.

Design problem of spatiotemporal light navigator

Suppose a mobile robot (G, K_{robo}) and a target path f are given. Find an image ϕ and sensor position d to make the robot track the given path:

Originally, we can design K_{robo}, but we design only ϕ and d in this study. This is because we would like to simplify the robot controller to the limit. Then, we focus on how to make the light distribution, that is, the design of ϕ, and how it affects the robot, that is, the design of d.

4. DESIGN METHOD

In this section, we propose the design method of the sensor position and the cast image to follow the target path by limiting the target path to a straight line. We determine the target path as a straight line $f(x) = p_y$, where $p_y \in \boldsymbol{P}_y$ is a constant. Then we find image ϕ and sensor position d which will make the robot follow the straight line.

4.1 Design of sensor position d

Now we consider the case where an input according to the following equation is applied to the robot:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} V \\ -k(y_s - p_y) \end{bmatrix}, \tag{5}$$

where $k \in \boldsymbol{R}$ is a positive constant. According to (1) and (2), the motion of the robot can be expressed by

$$\begin{cases} \dot{x} = V\cos\theta, \\ \dot{y} = V\sin\theta, \\ \dot{\theta} = -k(y + d\sin\theta - p_y). \end{cases} \tag{6}$$

Then, the following theorem holds:

Theorem 4.1. Suppose that the motion of the mobile robot is expressed by (1), target path $y = P_y$ is given, and the sensor position is expressed by (2), where $k > 0, d > 0$ and $V > 0$. If the input is expressed by (5), then the system has a global asymptotic stable equilibrium at $(y, \theta) = (p_y, 0)$.

proof. The equilibrium point of (6), which satisfies $(\dot{y}, \dot{\theta}) = (0, 0)$ is $(y, \theta) = (p_y, 0)$ or $(p_y, \pi)(-\pi < \theta \leq \pi)$. If (6) is linearized at equilibrium point $(y, \theta) = (p_y, 0)$, the eigenvalues λ at the equilibrium point are given by

$$\lambda = \frac{kd}{2}\left(-1 \pm \sqrt{1 - \frac{4V}{kd^2}}\right). \tag{7}$$

Since $k > 0$, $d > 0$, and $V > 0$, the real part of the eigenvalues is negative; therefore, the equilibrium point $(y, \theta) = (p_y, 0)$ is stable. Similarly, if (6) is linearized at the equilibrium point $(y, \theta) = (p_y, \pi)$, the eigenvalues λ' are as follows:

$$\lambda' = \frac{kd}{2}\left(1 \pm \sqrt{1 + \frac{4V}{kd^2}}\right). \tag{8}$$

Since $k > 0$, $d > 0$, and $V > 0$, the equilibrium point $(y, \theta) = (p_y, \pi)$ is not stable. In addition, if we assume the following equation as a candidate for the Lyapunov function:

$$U(y, \theta) = V(1 - \cos\theta) + \frac{1}{2}k(y - p_y)^2. \tag{9}$$

it becomes minimum at $(y, \theta) = (p_y, 0)$. Then, the time differentiation of this function is shown by the following equation:

$$\frac{d}{dt}U(y, \theta) = -kdV\sin^2\theta \leq 0. \tag{10}$$

In this equation, $\frac{d}{dt}U(y, \theta)$ always equals 0 at equilibrium points only. In real systems, disturbance is involved; thus, it will not stop at an unstable equilibrium point. Therefore, the point $(p_y, 0)$ is a globally asymptotically stable equilibrium of the system (6). □

That is, if we can realize the input of (5) under the controller in (4) and set $d > 0$, (y, θ) will be globally asymptotically stable at $(p_y, 0)$, which means that the robot can track the target path.

We verified theorem 4.1 by simulation. We set $k = 6.25[\text{rad}/(\text{m} \cdot \text{s})]$, $d = 0.06[\text{m}]$, $V = 0.1[\text{m/s}]$, $p_y = 0[\text{m}]$, and we calculated the routes of the robot with (6) at some initial position. Note that the value of k is determined based on the number of pixels in the projected image described later. The routes are shown in Fig. 6. From these results, we observed that the robot could successfully track the target path, at least roughly.

4.2 Design of binary image ϕ

We attempt to generate an image which realizes the input of (5) under the controller (4). First, we assume that the value of the sensor is continuous from 0 to 1, rather than

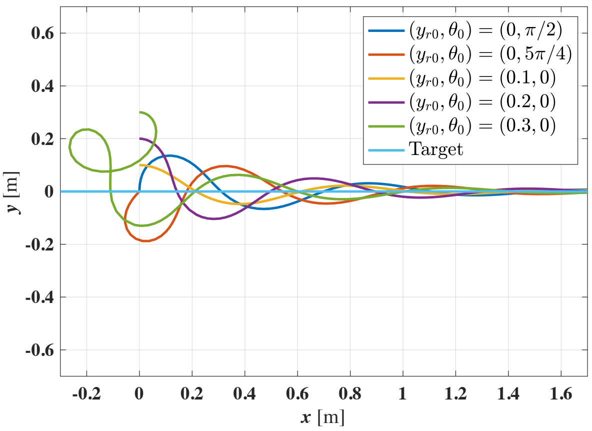


Fig. 6. Motion of the robot in the simulation.

being discrete (0 or 1), and we define the continuous value as $\sigma \in [0, 1]$. Then, we consider a virtual controller

$$\overline{K}_{\rm robo} : \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} V \\ -\dfrac{2V}{l}\left\{\sigma - \dfrac{1}{2}\right\} \end{bmatrix} \qquad (11)$$

in this situation. This controller is generated by linear interpolation of $K_{\rm robo}$. To realize (5) under the controller in (11), σ is expressed by the following equation:

$$\sigma = \frac{kl}{2V}(y_s - p_y) + \frac{1}{2}. \qquad (12)$$

Based on (12), we make $\sigma = 0$ correspond to black and make $\sigma = 1$ correspond to white, and make the intermediate values correspond to grayscale. Then, we can generate an image $\bar{\phi}$, which is graduated in the y direction and has a constant color in the x direction, as illustrated in Fig. 7. If we use the grayscale image $\bar{\phi}$, we can realize (5) with controller $\overline{K}_{\rm robo}$.

However, the grayscale information is lost when passing through $K_{\rm robo}$, which is actually used. Therefore, as shown in Fig. 8, we generate a binary image ϕ based on the grayscale image $\bar{\phi}$ by using halftone processing Robert (1987). Halftone processing is a technique for simulating grayscale images by varying the proportion of black and white pixels. There are several methods for doing this; however, in this study, we use the error diffusion method. In the error diffusion method, the noise generated by binarization is dispersed to nearby pixels. The ratio of diffusing is determined by using an error diffusion filter, and in this study, we use the Floyd and Steinberg Filter Robert (1987). An example of the image is shown in Fig. 9. Using this method, we can expect to realize (5) in a pseudo manner with controller $K_{\rm robo}$ in (4).

5. VERIFICATION

We verified the theory with an actual machine. We employed a robot shown in Fig. 10. The color sensor only perceives light intensity. The microcontroller generates the input of (4) according to the value determined by the sensor. The parameters are $d = 0.06$[m], $V = 0.1$[m/s], and $l = 0.04$[m].

First, we set the pixel size as 1mm × 1mm and $k = 6.25[\mathrm{rad}/(\mathrm{m}\cdot\mathrm{s})]$, then the image illustrated in Fig. 9 was

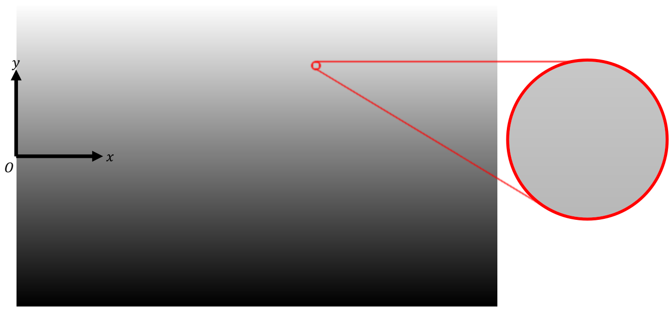


Fig. 7. Grayscale image $\bar{\phi}$.

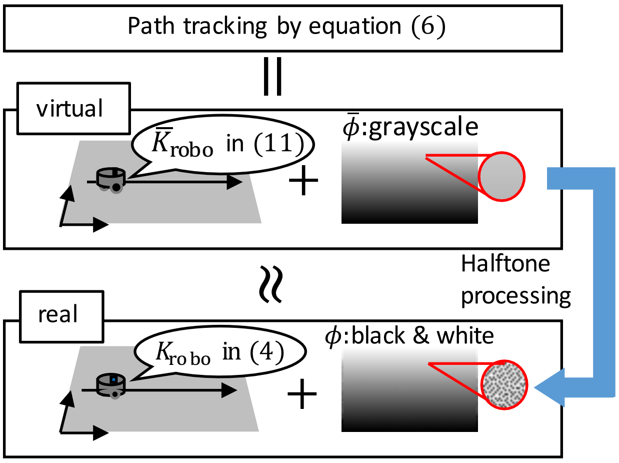


Fig. 8. Step of image design.

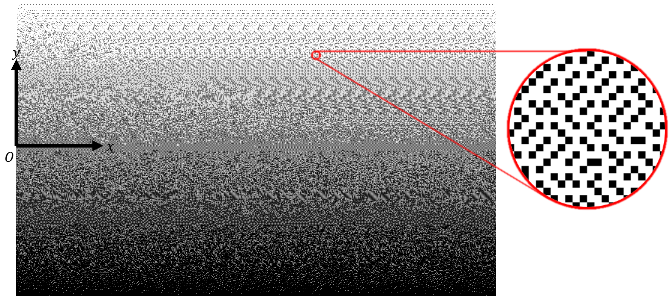


Fig. 9. Halftone image ϕ.

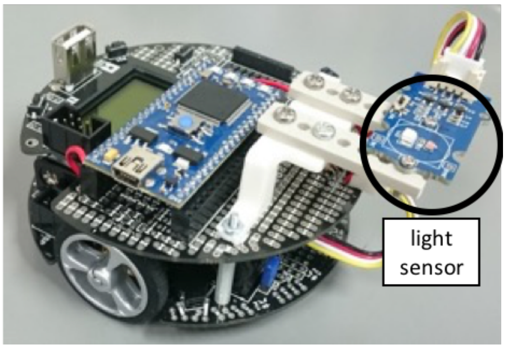


Fig. 10. Overview of the robot.

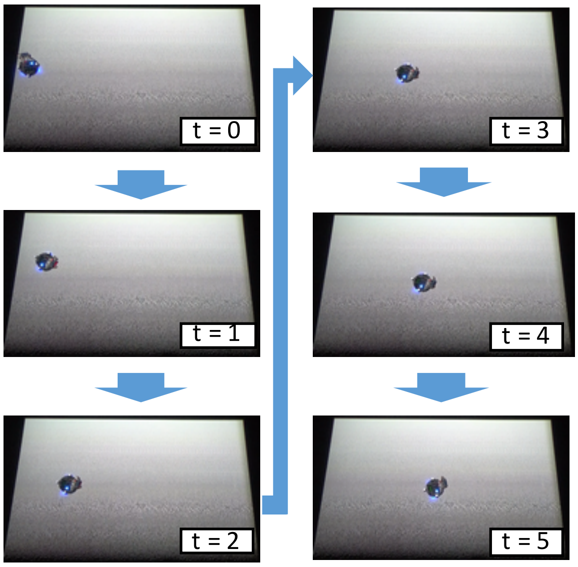


Fig. 11. Snapshots of the experiment.

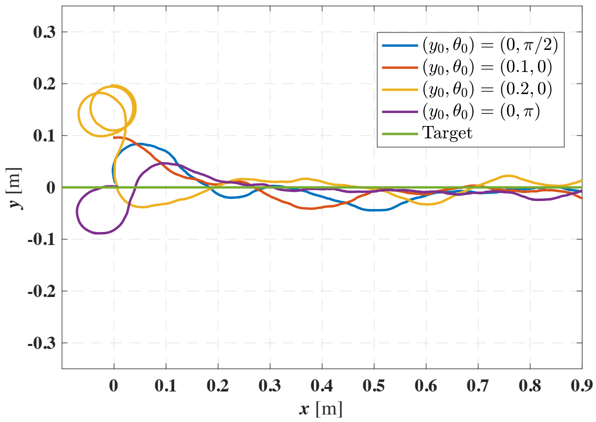


Fig. 12. Motion of the robot for the image in Fig. 9.

generated. Note that the image consists of 1280×800 pixels and a value of $k = 6.25[\text{rad}/(\text{m} \cdot \text{s})]$ corresponds to the image being completely white at the top and completely black at the bottom, which satisfies the stability condition $k > 0$. We defined the coordinate axis as Fig. 9 and set $p_y = 0$.

Subsequently, the image was cast onto the field perpendicular to the direction of gravity. The robot illustrated in Fig. 10 moved in a field of Fig. 9 with some initial positions and initial angles. The snapshots of the experiment are shown in Fig. 11. The routes are shown in Fig. 12. From these results, it can be seen that the robot succeeded in roughly tracking the target path.

Further, we generated the image illustrated in Fig. 13. The robot illustrated in Fig. 10 moved on a field of Fig. 13 with some initial positions and initial angles. The routes are shown in Fig. 14. From these results, it can be confirmed that the range of initial conditions to track the path is very small for a simple binary image. In this sense, the proposed method can be said to be suitable for tracking control of a low-performance robot.

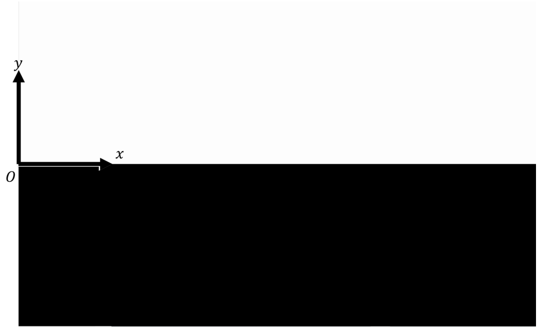


Fig. 13. Binary image $\bar{\phi}$.

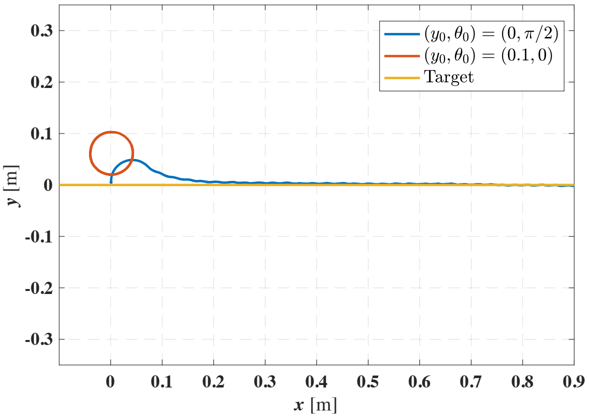


Fig. 14. Motion of the robot for the image in Fig. 13.

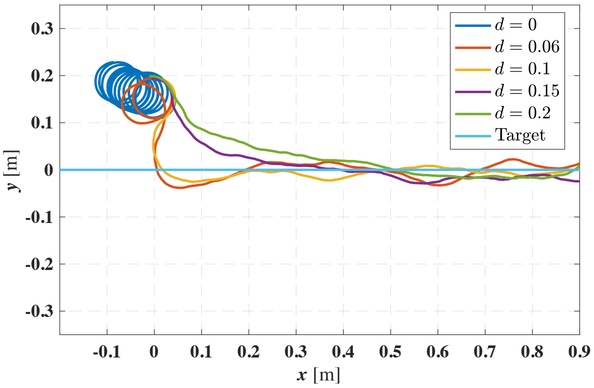
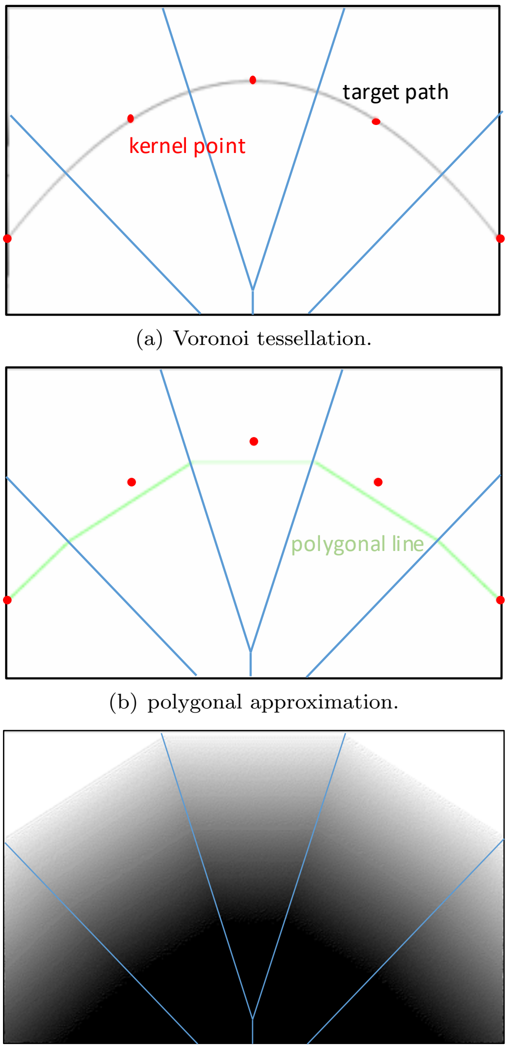


Fig. 15. Influence of the sensor position d on the resulting routes.

In addition, we observed the route with change in the sensor position. The route was as shown in Fig. 15. From these results, it was noted that the robot could not track the path when the sensor was at the center of the robot. Furthermore, when the position of the sensor was located more towards the front of the robot, the motion of the robot was less oscillatory.

6. EXTENSION TO THE GENERAL CASE

We investigated not only straight-line tracking but also curved-line tracking of the robot. We hypothesized that we can approximate the curved-line as a polygonal line to realize curved-line tracking. We show the method to



(a) Voronoi tessellation.

(b) polygonal approximation.

(c) generating image for straight-line tracking in each Voronoi area.

Fig. 16. Image generation algorithm for curve-line tracking

approximate a curved line as a polygonal line below and in Fig. 16:

Image design method for curved-line tracking

(i) Select several points on the curved line as kernel points, and divide into Voronoi area by the kernel points as shown in Fig. 16(a).

(ii) Create straight lines in each area by connecting midpoints of adjacent kernel points as shown in Fig. 16(b).

(iii)Design the image to track the straight line in each area as shown in the Fig. 16(c).

As an example, we set the target path as $y = 0.2 \sin 2\pi x$ and set $k = 12.5[\mathrm{rad}/(\mathrm{m} \cdot \mathrm{s})]$. Then, we created Fig. 17. Incidentally, we selected 64 kernel points evenly in the x-direction. The robot illustrated in Fig. 10 moved on a field of Fig. 17. The route is shown in Fig. 18. From this result, it can be seen that the robot succeeded in roughly tracking the path.

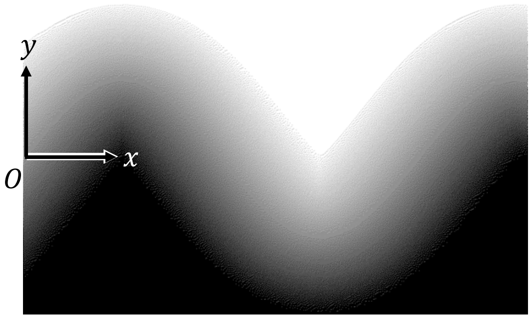


Fig. 17. Image for sine-curve tracking.

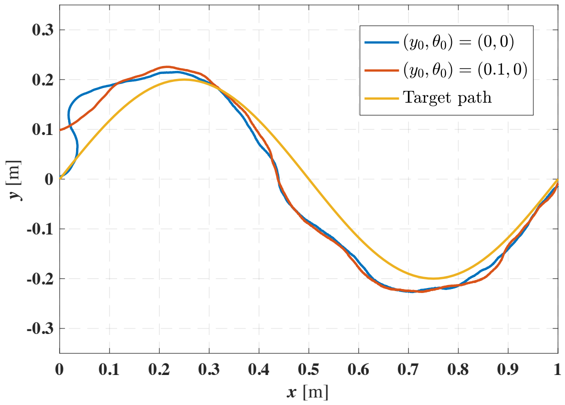


Fig. 18. Motion of the robot for the image in Fig. 17.

7. CONCLUSION

In this study, we aimed to successfully cause a mobile robot equipped with a light sensor to track a target path on a two-dimensional plane via spatiotemporal light navigation. We designed both the sensor position and the image to be cast. The design method was based on the Lyapunov stability analysis and the halftone image processing algorithm. In addition, we experimentally verified with an actual machine that the robot can achieve path tracking.

Future work could include achieving other tasks such as object manipulation and formation by using other images or by temporal changes of images.

REFERENCES

Aaron, B., Golnaz, H., Justin, W., Michael, R., and James, M. (2013). Massive uniform manipulation: Controlling large populations of simple robots with a common input signal. *IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Frangipane, G., Dell'Arciprete, D., Petracchini, S., Maggi, C., Saglimbeni, F., Bianchi, S., Vizsnyiczai, G., Bernardini, M.L., and Di Leonardo, R. (2018). Dynamic density shaping of photokinetic e. coli. *Elife*, 7, e36608.

Morse, T.M., Lockery, S.R., and Ferrée, T.C. (1998). Robust spatial navigation in a robot inspired by chemotaxis in caenorhabditis elegans. *Adaptive Behavior*, 6(3-4), 393–410.

Robert, U. (1987). *Digital halftoning.* MIT press.

Shahrokhi, S., Lin, L., Ertel, C., Wan, M., and Becker, A.T. (2017). Steering a swarm of particles using global inputs and swarm statistics. *IEEE Transactions on Robotics*, 34(1), 207–219.