Range-sensor-based Semi-autonomous Whole-body Collision Avoidance of a Snake Robot

Motoyasu Tanaka, *Member, IEEE,* Kazuyuki Kon, *Member, IEEE,* and Kazuo Tanaka, *Fellow, IEEE*

***Abstract*—The paper presents a control system for a snake robot based on range sensor data that semi-autonomously aids the robot in avoiding collisions with obstacles. In the proposed system, an operator indicates the desired velocity of the first link of the robot using a joystick, and the joint input which accomplishes both the desired velocity of the first link and collision avoidance between subsequent links and obstacles is automatically calculated by the controller, which selects the links needed to be grounded and exploits redundancy. The controller uses real-time data from range sensors for obstacle positions. The experimental system which has range sensors and the function generating environmental map using simultaneous localization and mapping (SLAM) was developed with decreasing calculation cost, and experiments were performed to verify the effectiveness of the proposed system on unknown environment.**

***Index Terms*—Snake robot, Inspection in narrow spaces, Collision avoidance, Switching constraints, Redundancy.**

I. Introduction

SEARCHING along narrow paths without any damage to surrounding rubble is needed in rescuing survivors in collapsed buildings following a disaster or accident to prevent further disasters. Because of difficulty and danger, rescue robots replace people in such activities. An articulated mobile robot is suitable for searching along narrow paths because of its slender body. Many articulated mobile robots have been developed in [1]–[7]. These robots generate propulsion forces using active wheels [1]–[3] or tracks [4]–[7].

A snake robot is an articulated mobile robot. As real snakes have no limbs, maneuvering is performed by bending their bodies regardless of the simple shape. A snake robot, which is intended to mimic a real snake, also maneuvers by bending its body. The robot can be downsized compared with other articulated mobile robots because the drive part of the robot is only a joint requiring no mechanism to directly generate propulsion forces, e.g., active wheels and tracks. This paper focuses on snake robot locomotion.

Along narrow paths, the problem is understanding the interaction between the robot and its surroundings as these separations can be small. If an obstacle prevents the robot from advancing, the robot should change motion to avoid the obstacle or be able to advance using it.

Various studies exist on the locomotion of snake robots using contact with obstacles [8]–[13]. Hirose [8] proposed "lateral inhibition" in which the snake robot adapts the shape of the obstacle during contact with it. Kamegawa [9] proposed an extended "asymmetrical reverse lateral inhibition" during contact. Using curvature derivative control, Date [10] achieved an adaptive locomotion along narrow corridors surrounded by smoothly curved walls. Transeth [11] derived a hybrid model which included the dynamics of the snake robot and the contact force between robot and obstacles, verifying the model by experiments. Liljebäck [12] derived a hybrid model based on event tracking and proposed the hybrid controller for the obstacle-aided locomotion. The work in [11] and [12] considered only circular obstacles. Kano [13] proposed a decentralized control scheme with local reflexive mechanisms and verified its effectiveness in pegged and smooth terrains.

However, joints of the articulated mobile robot become stuck against thin, nobbled, and non-smooth obstacles. Moreover, fragile, high-heat, electrically-charged or sticky obstacles should be avoided because damage to the robot could ensue if contact is made. Thus, it is necessary to use both obstacle-aided locomotion and obstacle avoidance in actual narrow environments.

Methods of obstacle avoidance for a snake robot have been proposed such as the central pattern generator with range sensors [14], the artificial potential field and simulated annealing [15], and the tuned serpenoid curve [16]. These methods program the motion of the robot's head to avoid obstacles. However, the robot's body (the following links) do not always pass through the path followed by the robot's head and it is possible that these following links collide with the obstacles.

In [17], we proposed a controller which accomplishes moving obstacle avoidance for the body of the snake robot using redundancy and grounded/lifted switching of wheels. The snake robot has passive wheels causing a sideways constraint force enabling forward movements. Therefore, it is difficult for a robot to move in a direction normal to the wheels to avoid obstacles. If the robot could force the wheels to slip in the normal direction, an avoidance motion can be accomplished. However, the needed joint torque for the wheels to slip is too much and it is necessary to control the robot considering the constraint force on the wheels. The previously proposed controller [17] accomplishes an avoidance motion by ignoring some of the constraints on the wheels. This is performed by infinitesimally lifting some wheels. The controller is suitable for an operator to remotely control the snake robot because its head can track a desired trajectory set by the operator and the avoidance motion of the body is automatically generated. However, only a single obstacle was present and the controller

M. Tanaka and K. Tanaka are with the Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Tokyo 182-8585 Japan. (e-mail: mtanaka@uec.ac.jp).

K. Kon is with the Graduate School of Engineering, Kyoto University, Kyoto 615-8540, Japan. (email: xeno1983@gmail.com).

Manuscript received Month Day, 20XX; revised Month Day, 20XX.

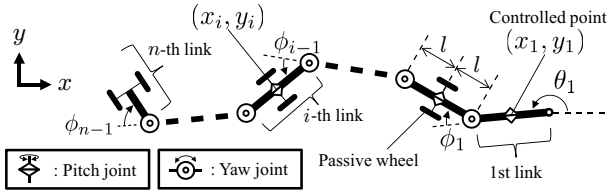


Fig. 1. Parameters of the snake robot.

cannot be applied in unfamiliar environments because the positions of the robot and obstacle are tracked by a motion capture system.

For obstacle avoidance along narrow paths, it is necessary for the controller to gather information of the surroundings, and to calculate an appropriate motion of the robot using the information. The problem is the simultaneous pursuit of the bending motion and obstacle avoidance because wheels are passive and a robot needs to undulate appropriately to generate propulsion forces.

This paper presents a method based on range sensors of semi-autonomous obstacle avoidance between the body of the snake robot and its surroundings. In the proposed method, a controller selects the optimal allocation of grounded wheels and using all range sensor data calculates the joint input to avoid colliding with obstacles. Note that there are multiple obstacles and this controller is different from that used in [17] as many obstacles are considered. The main contribution of the paper is to provide a novel cost function in which multiple obstacles are considered in the control approach of [17], and to accomplish a semi-autonomous whole-body collision avoidance of a snake robot based on range sensors using the approach. Moreover, it is to accomplish real-time obstacle avoidance by decreasing calculation cost using a heuristic reduction at the selection of the optimal allocation of grounded wheels. The features of both a joint torque for lifting links and a motion for avoiding obstacles are considered in the reduction. Furthermore, it is also to develop a real system which has range sensors on the robot's body and the function of generation of environmental map using SLAM, and to verify the effectiveness of the proposed method on unknown environment. Using the proposed method, the collision avoidance motion of the body links of the robot is semi-autonomously generated. Therefore, when the robot is remotely controlled, all an operator has to do is to steer the head of the robot to avoid collisions; whole body collision avoidance is accomplished by the controller.

II. Model

The study uses the snake robot shown in Fig.1. A link comprises a pitch rotational joint and a pair of wheels, and is connected to an adjoining link via a yaw rotational joint. The wheels are mounted coaxially with the pitch joint on each link except the first link. All wheels are passive and all joints are active. Instead of passive wheels, a ball caster is mounted under the first link; the first link makes contact with the ground through the ball caster. The range sensors mounted on the robot record distances between link and obstacles. The controlled point is set at the center of the first link. We assume that the passive wheels do not slide sideways and the environment is flat. The robot can maneuver by appropriately rotating the yaw joints using velocity constraints from the passive wheels. If the pitch joints are appropriately rotated, the robot can ascend and descend a step [18]. Nevertheless, we only use the pitch joints to lift up some of the wheels. The desired value of the pitch angle to lift up the wheels is set to a minute angle and we assume that the motion of the robot is affected by only the motion of the yaw joints. Note that the pitch joint angles are only used for grounded/lifted switching of the links, and are not explicitly included in the model.

As shown in Fig.1, let n be the number of links, ϕ_i be the i-th yaw angle, ψ_i be the i-th pitch angle, $2l$ be the length of each link, (x_i, y_i) be the center position of the i-th link, and θ_i be the absolute angle of the i-th link. We set $\boldsymbol{\phi} = [\phi_1, \cdots, \phi_{n-1}]^T$, $\boldsymbol{\psi} = [\psi_1, \cdots, \psi_{n-1}]^T$, $\boldsymbol{w} = [x_1, y_1, \theta_1]^T$, and $\boldsymbol{q} = [\boldsymbol{w}^T, \boldsymbol{\phi}^T]^T$. We derive the kinematic model of the robot. The difference between the model of this study and that of [17] is the position of the controlled point. If we set the controlled point to the front end of the first link as in [17], the back end of the first link (the position of the first yaw joint) passes laterally to the path of the front end while pivoting and the robot is at an increased risk of collision with surrounding obstacles. Thus, we set the center position and attitude of the first link as the controlled variable $\boldsymbol{w}$.

The robot becomes a hybrid system switched by whether the wheels are grounded or lifted in response to the motion of the pitch joints. The kinematic model is different depending on the grounded/lifted status of the wheels. Thus, we allocate a unique integer number, called the mode, to represent the overall status of the wheels. Let σ denote the discrete mode number and ΔT be the switching time period. As shown in [17], the system of the snake robot is then expressed as

$$\begin{aligned} \tilde{\boldsymbol{A}}_{\sigma(t)}\dot{\boldsymbol{w}} &= \tilde{\boldsymbol{B}}_{\sigma(t)}\boldsymbol{u} \\ \sigma(t) &= \sigma_k, \quad \forall t \in [t_k, t_{k+1}) \end{aligned} \tag{1}$$

where $\boldsymbol{u} = \dot{\boldsymbol{\phi}}$, $\sigma \in M$, $M = \{1, 2, \cdots, N_m\}$, N_m the number of modes, and $t_k = k\Delta t$ $(k = 0, 1, 2, \cdots)$ is the switching time. The mode instantaneously switches to σ_k at $t = t_k$ and holds the mode number σ_k at $t_k \leq t < t_{k+1}$. Next, the mode switches at $t = t_{k+1}$. (1) is derived by velocity constraints which means that the grounded passive wheel does not slide in the sideways direction and each row of matrices $\tilde{\boldsymbol{A}}_\sigma$ and $\tilde{\boldsymbol{B}}_\sigma$ means that velocity constraint caused by each grounded wheel in mode σ. The paper treats (1) as the kinematic model of the robot.

III. Control system

Fig. 2 outlines our proposal for a remote-control system. Obstacle positions are estimated using the data from range sensors. An operator provides the controller with the desired velocity setting for the controlled point using a joystick while viewing the real-time image from the robot's camera. The controller selects an optimal mode and calculates a yaw joint input and pitch joint reference based on the velocity setting, obstacle positions, joint angles, and the cost function $V(\boldsymbol{q})$,

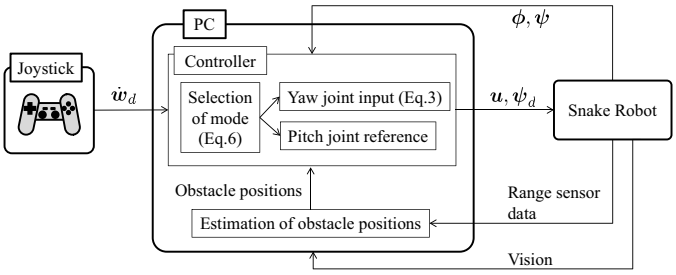


Fig. 2. Remote-control system for the semi-autonomous collision avoidance.

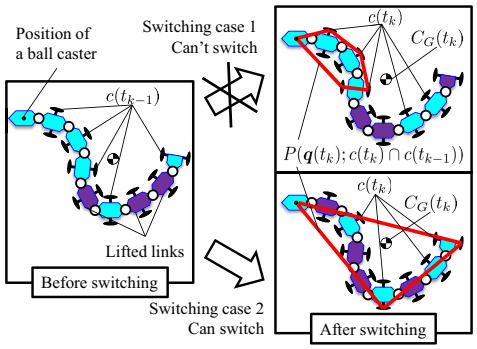


Fig. 3. Snake robot examples of the switching condition (7): in case 1, the condition is not satisfied whereas in case 2, the condition is satisfied.

and sends the joint input and reference to the robot. The cost function V depends on avoidance of singular configuration, joint-limit, and collision between the robot and obstacles, and the robot can avoid them by decreasing V. Collision avoidance of the whole body of the snake robot is implemented by the selection of mode and the use of controller redundancy so as to decrease V.

A. Joint input

We consider the joint input which accomplishes the desired motion of controlled point and decrease of V. Trajectory tracking control [17] is not suitable for this system unless $\boldsymbol{w}$ is directly measured by an external sensor as a motion capture system. Hence, we use a feed-forward controller for the velocity of the controlled point in calculating the joint input. The values with which an operator using a joystick provides a controller are the translation velocity $v_t \geq 0$ and the rotation velocity v_r of the controlled point. The desired velocity of the controlled point $\dot{\boldsymbol{w}}_d$ is then obtained as

$$\dot{\boldsymbol{w}}_d = \begin{bmatrix} v_t \cos(\theta_1 - \pi) \\ v_t \sin(\theta_1 - \pi) \\ v_r \end{bmatrix}, \tag{2}$$

and given mode σ, the yaw joint velocity is determined from

$$\boldsymbol{u} = \tilde{\boldsymbol{B}}_\sigma^\dagger \tilde{\boldsymbol{A}}_\sigma \dot{\boldsymbol{w}}_d + (\boldsymbol{I} - \tilde{\boldsymbol{B}}_\sigma^\dagger \tilde{\boldsymbol{B}}_\sigma)\alpha\boldsymbol{\eta}, \tag{3}$$

where $\tilde{\boldsymbol{B}}_\sigma^\dagger$ is a pseudo-inverse matrix of $\tilde{\boldsymbol{B}}_\sigma$, $\boldsymbol{I}$ an identity matrix, $\alpha < 0$ the gain for the cost function V, and $\boldsymbol{\eta} = [\partial V/\partial\phi_1, \cdots, \partial V/\partial\phi_{n-1}]^T$. (3) is the feed-forward input for the model (1). The first term on the right-hand side of (3) contributes the motion of the controlled point $\dot{\boldsymbol{w}}$, and the second term represents the kinematic redundancy and does not affect the motion of $\dot{\boldsymbol{w}}$. The closed loop system is obtained as

$$\tilde{\boldsymbol{A}}_\sigma(\dot{\boldsymbol{w}} - \dot{\boldsymbol{w}}_d) = \boldsymbol{0}. \tag{4}$$

Thus, if $\tilde{\boldsymbol{A}}_\sigma$ has a full column rank (i.e., the status of the robot is not a singular configuration), $\dot{\boldsymbol{w}} = \dot{\boldsymbol{w}}_d$ is satisfied.

The derivation of the cost function V is derived as

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial \boldsymbol{w}}\dot{\boldsymbol{w}} + \frac{\partial V}{\partial \boldsymbol{\phi}}\dot{\boldsymbol{\phi}} \\ &= \frac{\partial V}{\partial \boldsymbol{w}}\dot{\boldsymbol{w}} + \boldsymbol{\eta}^T \tilde{\boldsymbol{B}}_\sigma^\dagger \tilde{\boldsymbol{A}}_\sigma \dot{\boldsymbol{w}}_d + \alpha\boldsymbol{\eta}^T(\boldsymbol{I} - \tilde{\boldsymbol{B}}_\sigma^\dagger \tilde{\boldsymbol{B}}_\sigma)\boldsymbol{\eta}. \end{aligned} \tag{5}$$

The first and second terms on the right-hand side of (5) represent effects caused by the motion of the controlled point, and third term represents the effect caused by redundancy. From $\alpha < 0$ and $(\boldsymbol{I} - \tilde{\boldsymbol{B}}_\sigma^\dagger \tilde{\boldsymbol{B}}_\sigma) \geq 0$, the third term is not positive and contributes to a decrease in V.

In contrast, the pitch joints are used to determine the grounded/lifted status of the wheels corresponding to σ. As the actuator of the robot has a PID position-control facility, we set the appropriate values as the gains for the actuator of the pitch joints and use the facility as a PID controller. The desired angle of pitch joints $\boldsymbol{\psi}_d$ corresponding to the grounded/lifted status of the wheels are equal to that of [17].

B. Selection of mode

The robot selects the optimal mode that decreases V because the derivation of the cost function (5) is different depending on the mode. We set V_σ to the value of V in mode σ, At $t = t_k$, we formulate V optimizing as a finite time optimal control problem:

$$\min_{\sigma_k} \int_{t_k}^{t_{k+1}} V_{\sigma_k}(\hat{\boldsymbol{q}})\ \mathrm{d}t, \tag{6}$$

where $\hat{\boldsymbol{q}}$ is the estimate for vector $\boldsymbol{q}$ calculated by the numerical integration of (4) and (3). The optimal solution σ_k is calculated based on a full search of all combinations of switching mode and σ_k is selected as $\sigma(t)$ at $t < t_{k+1}$. As proposed in [17], we use the following constraint conditions in solving (6),

$$C_G(t_k) \in P(\boldsymbol{q}(t_k); c(t_k) \cap c(t_{k-1})), \tag{7}$$

$$\hat{C}_G(t) \in P(\hat{\boldsymbol{q}}(t); c(t_k)), t_k \leq t < t_{k+1}, \tag{8}$$

where $c(t_k)$ is the set of grounded links, $C_G(t_k)$ the center of gravity of the whole body of the snake robot at $t = t_k$, $P(\boldsymbol{q}(t_k); c(t_k))$ the supporting polygon area determined from all the wheels of grounded links and from the ball caster of the first link if grounded, and $\hat{C}_G$ is the estimate value of C_G calculated using $\hat{\boldsymbol{q}}$. The condition (7) signifies that the center of gravity is contained in the supporting polygon constructed by the common grounded links before and after switching (Fig. 3). The condition (8) if satisfied indicates the system is statically stable at $t_k \leq t < t_{k+1}$. These conditions confirm the robot is statically stable ensuring the controller avoids impractical switching: e.g., where all grounded links are switched to lifted links.

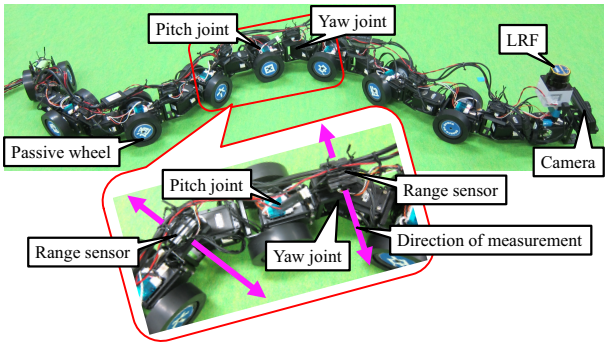


Fig. 4. Snake robot "T^2 Snake-2s."

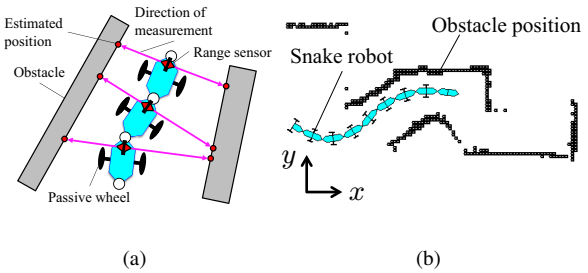


Fig. 5. Estimated obstacle positions from the range sensors on each link (a) and from SLAM (b).

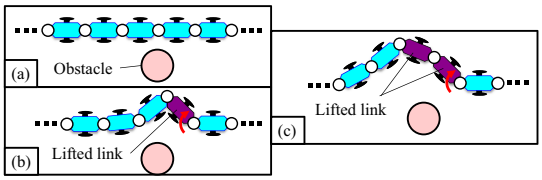


Fig. 6. Lifted links and motion avoidance; before avoiding motion (a), avoiding motion by one lifted link (b), and by two lifted links (c).

C. Cost function

We propose the cost function to accomplish subtasks. The subtasks perform the analysis to avoid collisions between the robot and obstacles, the singular configuration, and rotating limits for each joint. Let the position of obstacles be $(x_{Oj}, y_{Oj}), j = 1, \cdots, n_o$, and the distance between the center position of the i-th link (x_i, y_i) and the nearest obstacle be d_i. The cost function for collision avoidance V_o is expressed as

$$V_o(\boldsymbol{q}) = \sum_{i=2}^{n} f(d_i), \;\; f(x) = \begin{cases} (x - d_0)^2, & (x < d_0) \\ 0, & (otherwise) \end{cases} \tag{9}$$

where $d_0 > 0$ is the constant which defines the border point in the curve of $f(x)$. The curve of $f(x)$ approximates a quadratic. V_o is the summation of $f(d_i)$ corresponding to each link center, except for the first link. $f(d_1)$ is not contained in V_o because the obstacle avoidance of the first link only depends on the motion of the controlled point. The robot can reduce the risk of collision with obstacles if the robot moves by decreasing V. Note that (9) does not depend on the number of obstacles because the nearest point between the robot and obstacles are used.

Additionally, the robot cannot maneuver if it encounters itself in a singular configuration [19]. Moreover, because the angular range of joints is limited, the robot must avoid rotations that exceed joint limits. We designed a cost function related to both the singularity V_s and the joint limit V_l as

$$V_s(\boldsymbol{q}) = \frac{1}{\det \tilde{\boldsymbol{A}}_\sigma^T \tilde{\boldsymbol{A}}_\sigma}, \quad V_l(\boldsymbol{q}) = \sum_{i=1}^{n-1} \phi_i^4. \tag{10}$$

If the robot is in a singular configuration, $\tilde{\boldsymbol{A}}_\sigma$ does not have a full column rank and $\det \tilde{\boldsymbol{A}}_\sigma^T \tilde{\boldsymbol{A}}_\sigma$ becomes zero. Thus, the robot can curtail singular configurations by selecting modes that decrease V_s. V_l is the summation of the quartic functions of the joint angles and rapidly increases if the angles become large. Thus, if the robot moves with decreasing V_l, the magnitudes of the joint angles reduce which curb the likelihood that the robot reaches joint limits.

We then set the cost function V as

$$V(\boldsymbol{q}) = a_s V_s + a_l V_l + a_o V_o, \tag{11}$$

where a_s, a_l, and $a_o > 0$ are weight constants. If the robot moves with decreasing V using the input (3), the singular configuration, rotating over the joint limit or colliding with obstacles, is less likely to be reached.

If the robot configuration approaches a singular configuration, angles approach the joint limit, and links approach obstacles, the respective V_s, V_l, and V_o in V rapidly increase. If the number of modes is large enough so that many candidate σ_k are possible, it is hard to select the mode at which V_s, V_l, and V_o rapidly increase in (6). Therefore, combining the subtasks is a reasonable and viable option.

Remark 1: The proposed controller can directly be used with the same parameters if the number of sensors changes because the proposed cost function does not depend on the number of obstacles and range sensors.

IV. Experimental Setup

The study uses the snake robot "T^2 Snake-2s" (Fig. 4) in experiments. The camera and LRF are mounted on the first link; the robot obtains forward visual images with its head and range data from its surroundings using the LRF. The LRF is mounted at the center of the first link. Moreover, the range sensors mounted on each link record distances between link and obstacles. These sensors detect the distances between the robot and surrounding objects. The robot obtains an estimated position of itself and map data using simultaneous localization and mapping (SLAM) from the LRF. The data from the range sensors of all links are used for adapting to the moving surroundings and allowances for the error in SLAM.

A. Estimation of obstacle positions

The map data and the estimated value of $\boldsymbol{w}$ are obtained using Hector SLAM software [20] with the range data of the

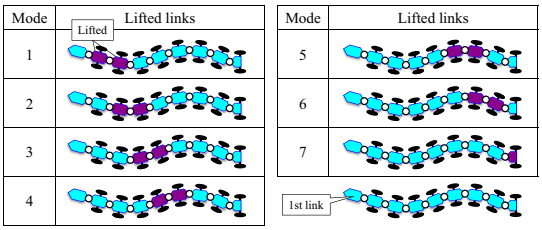


Fig. 7. Relationship between the mode and lifted links.

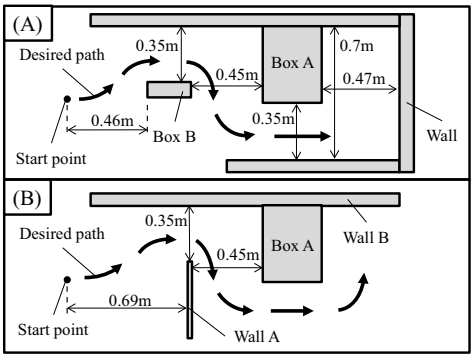


Fig. 8. Experimental fields where arrows show the desired path of the robot's head.

LRF. Obstacle positions are estimated using the map data and the data from the range sensors in each link. The obstacle positions determined by the range sensor of each link can be calculated using sensor data, joint angles, and the mounting location of the sensors (Fig. 5(a)). We then use the map data (an example is given in Fig. 5(b)) calculated by Hector SLAM as the obstacle positions determined by LRF because the map data contains the obstacle positions.

B. Mode reduction

The snake robot of Fig. 4 has nine links ($n = 9$) and the number of modes N_m calculated by the related equation in [17] is 219. Hence, because the number of modes is large, using the proposed controller for experiments with actual robots is difficult as the calculation cost in performing (6) is also quite large. Moreover, simultaneously lifting many adjacent links is impractical for an actual robot from the point of view of the required torque from actuators.

We thus reduce the number of modes with the purpose to avoid any lack in torque and to decrease calculation cost. The snake robot has the following features related to lifted links.

- The more adjacent lifted links, the larger the avoiding motion is during obstacle avoidance. (Fig. 6)
- The more lifted links, the easier the robot falls as a result of a lack in grounded wheels.
- If there is no lifted link, the robot does not have kinematic redundancy and cannot avoid singular configurations.

Moreover, the snake robot shown in Fig. 4 has the following features.

- The robot cannot simultaneously lift three adjacent links because of lack in torque.

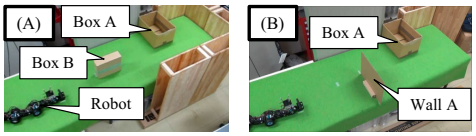


Fig. 9. Initial location of the robot and obstacles in each field.

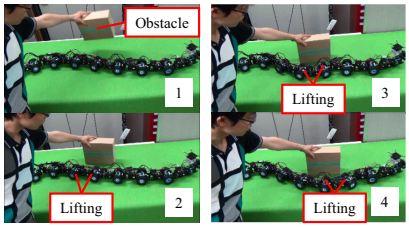


Fig. 10. Moving obstacle avoidance in case 1. The robot lifts some links and automatically generates the avoiding motion.

- The robot cannot simultaneously lift the penultimate and ultimate links as this produces a cantilever.

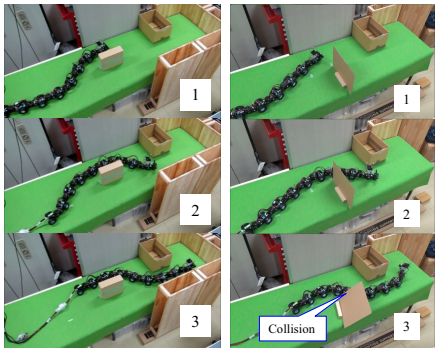
Considering the above features, we reduce the modes to seven modes, illustrated in Fig. 7. In modes 1–6, only two links are lifted and joined. In mode 7, only the ninth link is lifted. Using these modes, all links can avoid obstacles using large motion as depicted in Fig. 6(c). Additionally, these modes are usually statically stable because they have many grounded points, and are easily selected in (6).

V. Experiments

Experiments were performed to verify the effectiveness of the proposed control system. We use the controlled variable $\boldsymbol{w}_{SLAM}$ estimated by Hector SLAM [20] as $\boldsymbol{w}$. The actual value of $\boldsymbol{w}$ is measured by the motion capture system to analyze the results considering the error from SLAM. Note that the measured $\boldsymbol{w}$ is only used for the analysis of the experimental results and is never used by the control system. Experimental fields are shown in Fig. 8. Field A is a narrow path surrounded by walls and boxes as obstacles whereas field B is surrounded by obstacles including a thin wall (wall A). Fig. 9 shows the initial location of the robot and obstacles in each field. The operator provides the controller with the desired velocities v_t and v_r using a joystick and the robot accomplishes the desired velocity of the controlled point by rotating its joints as determined using (3). We set $\Delta T = 1$s, $\alpha = -1$, $a_s = 0.3$, $a_l = 0.001$, $a_o = 30$, $d_0 = 0.3$m.

A. Case 1: using only the range sensors on the links

We performed experiments using only obstacle positions from the range sensors on each link to calculate (9). For case 1, the robot can avoid obstacles without considering the error in SLAM because the map data is not used. Moreover, the robot can avoid moving obstacles as demonstrated in Fig. 10 because real time data was used from the range sensors on the links. In field A, the robot maneuvered without colliding with surroundings [Fig. 11(a)]. In contrast, in field B, a collision between the body of the robot and wall A occurred



(a) (b)

Fig. 11. Motion of the robot for case 1: (a) in field (A) and (b) field (B).

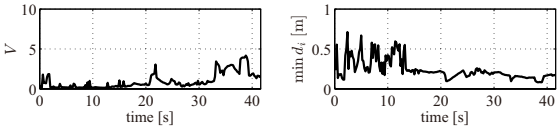


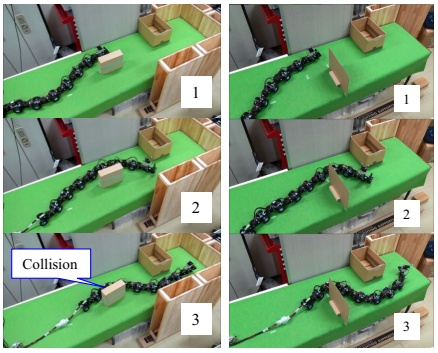
Fig. 12. Time response of V and $\min d_i$ in field (B) for case 1. The wheel of the sixth link of the robot collides with wall A at $t = 39$s.

[Fig. 11(b)]. This is because the range sensor on each link could not detect the nearest point between the robot and the thin wall A. From Fig. 12, $\min d_i$, the minimum value of d_i, was 0.08m at $t = 39$s just before the collision. However, the actual distance was 0.05m and therefore the collision occurred. Thus, depending on the number of sensors, collisions can happen because of undetected obstacles, the manner of sensing, and the body shape of the robot in case 1.

B. Case 2: using only the map data

Next, we used only obstacle positions based on the map data. In field B, the robot maneuvered without collision [Fig. 13(b)]. Although real-time detection of the wall A could not be executed, collision avoidance was accomplished because the map contained the position of wall A.

In field A, the collision between the rear links of the robot and the box B occurs in the bottom photo of Fig. 13(a). The estimated position of the robot and the map data are shown in Fig. 14 at the time of collision, and we can find that the robot does not collide with obstacles in the map. From Fig. 15, the error of estimation of y_1 and θ_1 by SLAM were relatively large. Let the error between the actual position of the i-th link and the i-th position estimated by using $\boldsymbol{w}_{SLAM}$ be e_i. The farther the link from the robot's head, the larger the error e_i of the link (Fig. 16). The collision happened because the gap between the actual obstacle positions and estimated positions in the map was caused by the error from SLAM. The estimated position of the rear links of the robot is especially affected by the error of θ_1 because the snake robot has a long body. Therefore, collisions may happen depending on the errors from SLAM in case 2.



(a) (b)

Fig. 13. Motion of the robot for case 2, showing the situation (a) in field (A) at $t = 18$, 36, 54 s, and (b) in field (B).

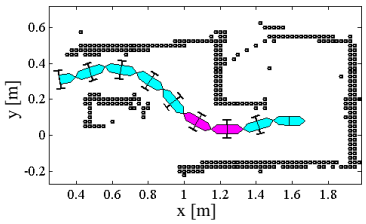


Fig. 14. Result of the estimated position of the robot and the map in field (A) for case 2 at $t = 54$s. Black circles mark obstacle positions based on map data; lifted links are colored magenta.

C. Case 3: using both the range sensors and the map

We used the obstacle positions from both the range sensors on each link and the map. In this case, the nearest point from each link is selected in the obstacle positions based on the range sensors and the map, and V_o is calculated using the nearest point. For case 3, the robot can avoid moving obstacles as well as in case 1.

In field B, the robot maneuvered without collision with its surroundings, including the thin wall A [Fig. 17(b)]. In field A, the robot similarly maneuvered without collision [Fig. 17(a)]. The minimum distance $\min d_i$ between the link and obstacle was kept at more than 0.06 m [Fig. 19]. From Figs. 20 and 21, the error in the estimated position of the robot was of the same order as for case 2. Fig. 18 shows the estimated position of the robot and the obstacle positions at $t = 57$s based on the map and range sensor data from the links. The relative positions between the robot and obstacles estimated by SLAM are different from the actual relative positions [bottom photo of Fig. 17(a)] and we find also that the map does not show the actual obstacle positions. However, box B was detected by the range sensors on the seventh and eighth link, and the robot avoided collision using the position of this box. Thus, it was confirmed that range sensors could cover a certain level of error from SLAM for case 3. Additionally, we consider field C in which there are many obstacles (boxes, walls, circular objects, a bucket, and a crumpled newspaper) to verify the effectiveness of the proposed method. In field C, the robot

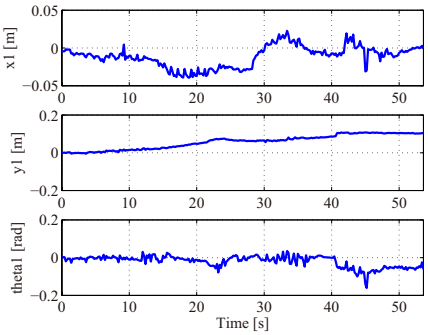


Fig. 15. Error in the estimated position by SLAM ($\boldsymbol{w}_{SLAM} - \boldsymbol{w}$) in field (A) of case 2.

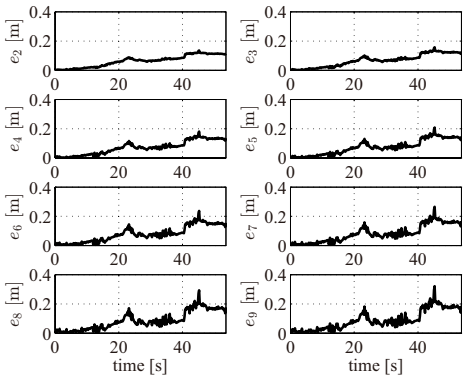
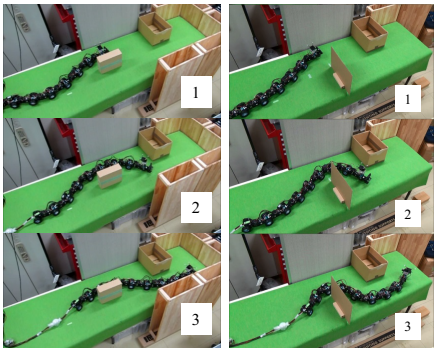


Fig. 16. Time responses of e_i in field (A) of case 2.



(a) (b)

Fig. 17. Motion of the robot for case 3, showing positioning (a) in field (A) at $t = 19$, 38, 57 s, and (b) in field (B).

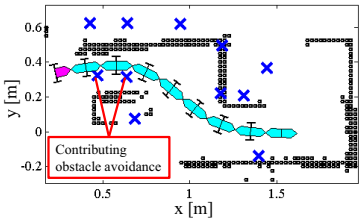


Fig. 18. Result of the estimated position of the robot and obstacles in field (A) for case 3 at $t = 57$s. The blue crosses mark obstacle positions based on the range sensor on each link. The black circles indicate obstacle positions based on the map data. The lifted link is colored magenta.

maneuvered without collision with its surroundings as well as in fields A and B [Fig. 22].

Note that singularity avoidance and joint-limit avoidance were achieved in all three cases but the figure showing results is omitted owing to space limitations. The jerky movement of the robot was generated because of the jump of joint input depending on the switch of the mode. However, the motion speed of the robot was slow and the robot could keep moving without serious problem such as rollover depending on the switch.

SLAM-MOT (moving object tracking) which accomplishes the detection of the moving object using a LRF was proposed in [21]. However, because the snake robot has a long body, some parts of the body of the robot may fall outside the measuring area of a LRF, and SLAM-MOT is not applied in collision avoidance between the body and obstacles in such cases. Therefore, for moving-obstacle avoidance, the use of range sensors on each link is appropriate.

If the number of range sensors is increased and can prevent missing detection of obstacles in case 1, the robot is expected to adaptively accomplish collision avoidance in narrow spaces with large changes because the data is obtained in real time. However, a significant increase in the number of sensors is unsuitable for a robot from the view point of size and complexity of circuitry within the robot. Case 2 is suitable for cases where the surroundings do not change and the error from SLAM is small. Case 3 can be used for cases where surroundings change and the error from SLAM is small because the range sensors on each link cover a certain degree of error from SLAM. The merit of cases 2 and 3 is its suitability in fully autonomous control because the navigation of the robot's head can be achieved using SLAM. As a result, we conclude that the realistic solution in estimating the obstacle positions is to use both the range sensors on each link and the LRF as in case 3.

VI. Conclusion

We presented a control system which accomplishes semi-autonomous whole-body collision avoidance of a snake robot based on range sensor data. An operator provides a controller with the desired velocity of the first link, and the controller autonomously calculates the joint input which accomplishes the desired velocity of the first link and avoidance of singularity, joint-limit, and collision by using a novel cost function. The controller selects the optimal location of grounded/lifted links and the lifting motion for the pitch joint switching of grounded links. The proposed controller was implemented in an actual robotic system with reducing the modes considering the required torque and calculation cost incurred by the controller, and experiments were performed. Experimental results demonstrated the effectiveness of the control system in determining a course along narrow paths.

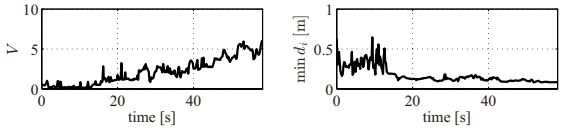


Fig. 19. Time response of V and $\min d_i$ in field (A) of case 3.

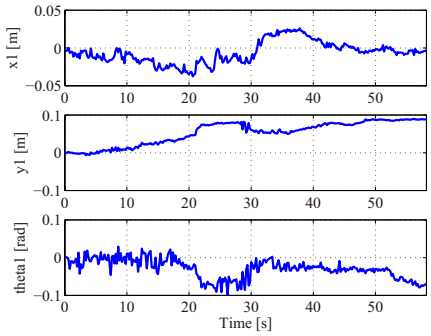


Fig. 20. Error in the estimated positions using SLAM ($\boldsymbol{w}_{SLAM} - \boldsymbol{w}$) in field (A) for case 3.

Future work includes combining the proposed collision avoidance with obstacle-aided locomotion. A novel method for obstacle-aided locomotion needs to be considered based on the proposed control framework which includes redundancy and selection of modes. Although at present the motion of the robot's head is provided by an operator, we will present a fully autonomous snake robot that would include trajectory generation of the robot's head.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 26870198.

REFERENCES

[1] K. Kouno, H. Yamada, and S. Hirose, "Development of Active-Joint Active-Wheel High Traversability Snake-Like Robot ACM-R4.2," *J. of Robotics and Mechatronics*, vol.25, no.3, pp.559-566, 2013.

[2] H. Kimura and S. Hirose, "Development of Genbu: Active wheel passive joint articulated mobile robot," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.823-828, 2002.

[3] K. Suzuki, A. Nakano, G. Endo, and S. Hirose, "Development of Multi-wheeled Snake-like Rescue Robots with Active Elastic Trunk," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.4602-4607, 2012.

[4] T. Kamegawa, T. Yamasaki, H. Igarashi, and F. Matsuno, "Development of the Snake-like Rescue Robot 'KOHGA'," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.5081-5086, 2004.

[5] K. Ohsuka and H. Kitajima, "Development of Mobile Inspection Robot for Resue Activities: MOIRA," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.3373-3377, 2003.

[6] J. Borenstein, M. Hansen, and A. Borrell, "The OmniTread OT-4 Serpentine Robot – Design and Performance," *J. of Field Robotics*, vol.24, no.7, pp.601-621, 2007.

[7] M. Neumann, T. Predki, L. Heckes, and P. Labenda, "Snake-like, tracked, mobile robot with active flippers for urban search-and-rescue tasks," *Industrial Robot: An International Journal*, vol.40 no.3, pp.246-250, 2013.

[8] S. Hirose, Biologically Inspired Robots (Snake-like Locomotor and Manipulator), Oxford University Press, 1993.

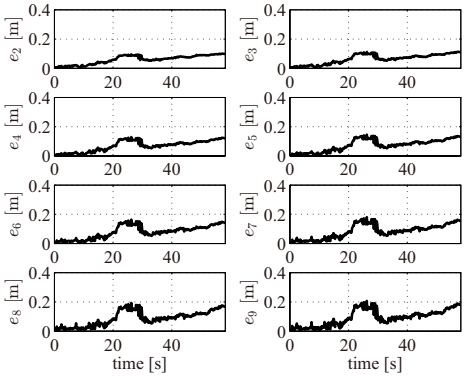


Fig. 21. Time responses of e_i in field (A) of case 3.

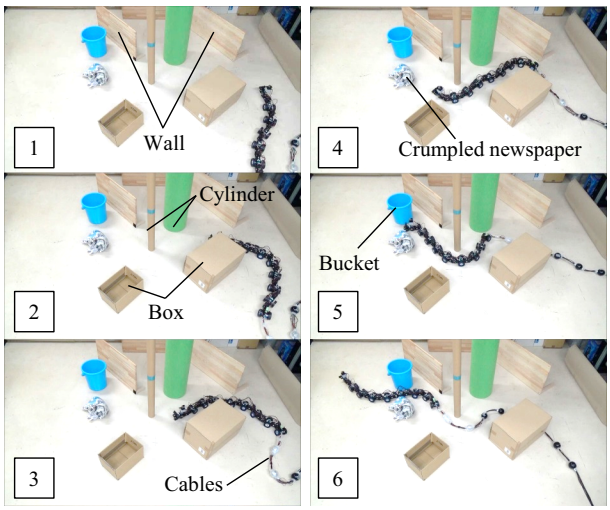


Fig. 22. Motion of the robot for case 3 in field (C).

[9] T. Kamegawa, R. Kuroki, M. Travers, and H. Choset, "Proposal of EARLI for the snake robot's obstacle aided locomotion," *IEEE Int. Symp. on Safety, Security, and Rescue Robotics*, pp.1-6, 2012.

[10] H. Date and Y. Takita, "Adaptive Locomotion of a Snake Like Robot Based on Curvature Derivatives," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.3554-3559, 2007.

[11] A. Transeth, R. Leine, C. Glocker, K. Pettersen, and P. Liljebäck, "Snake Robot Obstacle-Aided Locomotion: Modeling, Simulations, and Experiments," *IEEE Trans. on Robotics*, vol.24, no.1, pp.88-104, 2008.

[12] P. Liljebäck, K. Pettersen, O. Stavdahl, and J. Gravdahl, "Hybrid Modelling and Control of Obstacle-Aided Snake Robot Locomotion," *IEEE Trans. on Robotics*, vol.26, no.5, pp.781-799, 2010.

[13] T. Kano, T. Sato, R. Kobayashi, and A. Ishiguro, "Local reflexive mechanisms essential for snakes' scaffold-based locomotion," *Bioinspiration& Biomimetics*, vol. 7, 046008, pp.1-11, 2012.

[14] X. Wu and S. Ma, "Neurally Controlled Steering for Collision-Free Behavior of a Snake Robot," *IEEE Trans. on Control Systems Technology*, vol.21, no.6, pp.2443-2449, 2013.

[15] D. Yagnik, J. Ren, and R. Liscano, "Motion Planning for Multi-Link Robots Using Artificial Potential Fields and Modified Simulated Annealing," *Proc. IEEE/ASME Int. Conf. on Mechatronics and Embedded Systems and Applications*, pp.421-427, 2010.

[16] Y. Hitaka and M. Yokomichi, "Obstacle Avoidance of a Snake Robot in Narrow Hallway," *Proc. IEEE Int. Conf. on Mechatronics and Automation*, pp.214-219, 2012.

[17] M. Tanaka and F. Matsuno, "Control of Snake Robots with Switching Constraints: trajectory tracking with moving obstacle," *Advanced Robotics*, vol.28, no.6, pp.415-429, 2014.

[18] M. Tanaka and K. Tanaka, "Climbing and Descending Control of

a Snake Robot on Step Environments based on Kinematics," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.3285-3290, 2013.

[19] P. Prautsch, T. Mita and T. Iwasaki, "Analysis and Control of a Gait of Snake Robot," *Trans. of IEEJ*, vol.120-D, pp.372-381, 2000.

[20] S. Kohlbrecher, O. Stryk, J. Meyer, and U. Klingauf, "A Flexible and Scalable SLAM System with Full 3D Motion Estimation," *Proc. IEEE Int. Symp. on Safety, Security and Rescue Robotics*, pp.155-160, 2011.

[21] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The Int. J. of Robotics Research*, vol.26, no.9, pp.889-916, 2007.