2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Autonomous Navigation and Obstacle Avoidance of a Snake Robot with Combined Velocity-Heading Control

Mahdi Haghshenas-Jaryani[1] and Hakki Erhan Sevil[2]

Abstract— **This paper presents combined velocity-heading control of a planar snake robot for the autonomous navigation and obstacle avoidance in a simulation environment. The kinematics and dynamics of the snake robot were derived using the articulated-body algorithm without considering the non-holonomic constraints. A double-layer controller was designed to control both heading direction and average velocity through joint motion control. We adopted a rule-based expert system for autonomous navigation while avoiding obstacles/restricted-areas. The guidance commands were realized by two proportional controllers that use feedback of the estimated speed and heading of the robot. To validate the combined velocity-heading controller, a series of simulations were carried out for a snake robot with 6 links (8 DOF). The autonomous navigation and obstacle-avoidance algorithms provided the commands to follow the desired trajectories. The simulation results showed the effectiveness of the controller in following the desired heading directions and achieving targeted velocities with small errors to reach the goal position by avoiding obstacles.**

I. INTRODUCTION

The demand for developing autonomous robotic systems with capability to adapt and operate in unknown and dynamic environments has recently emerged with potential for a variety of applications such as space exploration, search-and-rescue, monitoring and inspection, and agriculture. Biologically inspired robots have shown greater potential for these applications. Particularly, biological snakes have fascinated roboticists over the past five decades due to their versatile limbless locomotion that adapt easily to unstructured and unknown environments [1]. Ever since preliminary study of biological snake locomotion by Gray in 1946 [2] and development of the first snake robot by Hirose [3] in early 70's, enormous amount of work have been focused on developing dynamic models and control algorithms for modeling and controlling these complex robotic systems [4].

Most snake robots studied over the past 48 years were considered as a serial kinematic chain with nonholonomic constraint. This constraint was explicitly imposed to the model of the snake robot for avoiding lateral slip (side-slip constraints) which is a crucial factor for generating the lateral undulatory locomotion (the most common locomotion gait between biological snakes) in snake robots [4]. The nonholonomic constraints were implemented in the physical robot platforms by adding passive wheels under each link of snake robots [3]. However, biological snakes rely on the interaction of their body with the surrounding environment based on anisotropic friction properties to generate these motion constraints and consequently the progressive motion. There have been fewer works on modeling and analysis of snake robots without the side-slip constraint [4], [5].

From control point of view, many research efforts were dedicated to the pattern gait control of snake robots [5], [6] with some work studied the position/heading direction with side-slip constraints [7]. The only work by Hicks and Ito [8] has studied the determination of optimal gaits for control of position/heading direction for the locomotion of snake robots *without* side-slip constraints on a flat surface. However, the velocity control was not considered.

In our earlier work [9], [10], the dynamic model of a four-link snake robot was derived using the Kane method without the side-slip condition. That leads to an under-actuated dynamic system (i.e. 6 DOF with only 3 control inputs) where the internal shape motion is not directly related to the robot's external locomotion. A robust joint controller based on sliding-mode control (SMC) technique was developed to modulate the internal body motion and to generate the required serpentine locomotion by attenuating lateral slip at each link. The controller was able to compensate for uncertainties regarding the model and the environment parameters such as mass and friction coefficients [9]. Additionally, a double-layer SMC was developed to control the under-actuated four-link snake robot in the Cartesian space for following specific paths without velocity control [10].

This paper presents a combined velocity-heading direction control of snake robots without side-slip constraints. The dynamics of a snake robot with *N*-link (i.e. *N+2* DOF and *N-1* control inputs) was derived using the articulated-body algorithm [11]. A waypoint navigation algorithm was used to navigate the surrounding environment. A double-layer controller was designed to control both the heading direction and the average velocity of a snake robot in following desired trajectories. The outer control layer modulates the serpentine curve motion parameters (angular frequency and bias) for controlling the heading direction and velocity, respectively. The inner-layer controls the joints' motion of the snake robot to follow the desired serpentine motion.

II. KINEMATICS-DYNAMICS OF SNAKE ROBOTS

A planar model of a snake-robot with *N* links connecting through *N-1* revolute joints without side-slip constraints (nonholonomic) was considered as shown in Fig 1. We assumed anisotropic friction applied to snake robot with

*This work was not supported by any organization

[1]Mahdi Haghshenas-Jaryani is with the Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM 88003, USA mahdihj@nmsu.edu

[2]Hakki Erhan Sevil is with the Department of Intelligent Systems & Robotics, University of West Florida, Pensacola, FL 32514, USA hsevil@uwf.edu

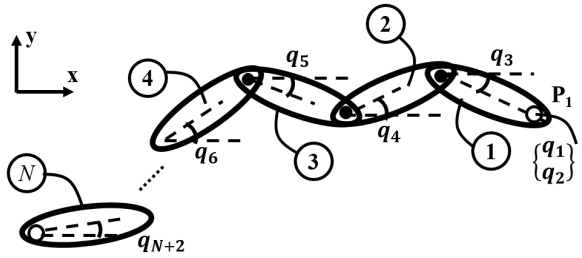

Fig. 1. Overall Model of a Snake Robot With *N* Links and *N-1* Joints.

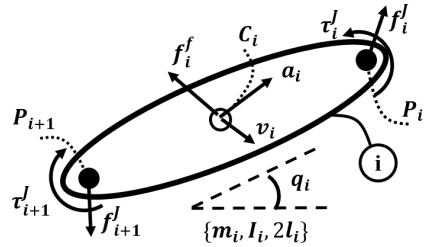

Fig. 2. Free Body Diagram of a Snake Robot Link.

a friction larger in the normal than tangential directions. Therefore, the robot has *N*+2 degrees-of-freedom (DOF) and is an under-actuated dynamic system where the internal shape motion is not anymore directly related to the overall displacement of the snake robot. Forward dynamics of the snake-robot, shown in Fig. 1, was derived using the articulated-body algorithm [11]. The first joint is considered as a floating planar joint (two translational and one rotational DOF) and the rest of the joints are one rotational DOF. The articulated-body algorithm calculates the dynamics of an N-link kinematic chain with *O(N)* arithmetic operations through a 3-pass procedure. The first pass, from base (tail) to the last link (head) calculates the velocities and bias terms, the second pass, from (head) to (tail) computes the articulated-body inertia and bias forces and the third pass from tail-to-head calculates the accelerations. Spatial vector algebra (6D) was employed to derive the equations. The spatial velocity, $\mathbf{v} = [\underline{\boldsymbol{\omega}}, \underline{\mathbf{v}}_C]^T \in \mathrm{M}^6$ and spatial force, $\mathbf{f} = [\underline{\mathbf{n}}_C, \underline{\mathbf{f}}]^T \in \mathrm{F}^6$ are combined the rotational and translational terms in one 6×1 vector representation where $\underline{\boldsymbol{\omega}} = [\omega_x \ \omega_y \ \omega_z]^T$ and $\underline{\mathbf{v}}_C = [v_{Cx} \ v_{Cy} \ v_{Cz}]^T$ in the motion domain as well as corresponding terms, $\underline{\mathbf{n}}_C = [n_{Cx} \ n_{Cy} \ n_{Cz}]^T$ and $\underline{\mathbf{f}} = [f_x \ f_y \ f_z]^T$, in the force domain.

$$\begin{cases} \mathbf{f}_i^J = I_i^A \mathbf{a}_i + \mathbf{p}_i^A \\ \mathbf{v}_i^J \in S \quad \mathbf{f}_i^J \in S^\perp \end{cases} \tag{1}$$

where,

$$\begin{aligned} I_i^A &= I_i + I_{i-1}^A - I_{i-1}^A S_{i-1} \Phi_i S_{i-1}^T I_{i-1}^A \\ \mathbf{p}_i^A &= \mathbf{p}_i + \mathbf{p}_{i-1}^A - I_{i-1}^A S_{i-1} \Phi_i \left(\tau_i - S_{i-1}^T \mathbf{p}_{i-1}^A\right) \\ \mathbf{p}_i &= \mathbf{v}_i \times^* I_i \mathbf{v}_i - \mathbf{f}_i^{ext} \\ \mathbf{v}_i &= \mathbf{v}_{i-1} + \mathbf{v}_i^J \qquad \mathbf{v}_0 = \mathbf{0} \\ \mathbf{v}_i^J &= S \dot{\mathbf{q}}_i \quad S^T \mathbf{f}_i^J = \tau_i \\ \mathbf{v}_i \times^* &= \begin{bmatrix} \underline{\boldsymbol{\omega}} & \underline{\mathbf{v}}_i \\ \mathbf{0} & \underline{\boldsymbol{\omega}} \end{bmatrix} \text{(Special cross-product in force domain)} \end{aligned} \tag{2}$$

and, $S_{6 \times N_f} \subseteq \mathrm{M}^6$ denotes a subspace of motion at a joint. S describes the constraint motion with $N_f \leq 6$ DOF and $S^\perp$ is the orthogonal complement of S includes the constraint forces that impose those motion constrains at the joint.

The generalized coordinates, $\mathbf{q} = [q_1, q_2, \cdots, q_{N+2}]^T$, describe the snake robot motion in 2D space where q_1 and q_2 are the coordinates of the tip of the snake robot (point P_1) and the rest are the absolute angle of the links with respect to the inertial frame *O* as shown in Fig.1. The $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the first and second derivatives of the generalized coordinates.

The free body diagram of a link of the snake-robot is shown in Fig. 2. The $\mathbf{f}_i^J$ and $\mathbf{f}_{i+1}^J$ are the joint forces, and torques at the proximal and distal joints associated with the link, are identified as τ_i and τ_{i+1}. The external friction force is applied with components in normal and tangential directions, represented here in a spatial force vector form $\mathbf{f}_i^{ext} = [0, 0, 0, \mathbf{f}_i^t, \mathbf{f}_i^n, 0]^T$. The friction between the snake robot and the ground was modeled as the Coulomb friction with anisotropic properties, $\mu_t \ll \mu_n$.

$$\begin{aligned} \mathbf{f}_i^n &= -\mu_n m_i g \ \mathrm{sgn}(\mathbf{v}_i) \cos(q_i) \\ \mathbf{f}_i^t &= -\mu_t m_i g \ \mathrm{sgn}(\mathbf{v}_i) \sin(q_i) \end{aligned} \tag{3}$$

where m_i and g are the mass of the ith link and the gravitational acceleration, respectively. Solve for the body acceleration $\mathbf{a}_i$ yields;

$$\begin{aligned} \mathbf{a}_i &= \mathbf{a}_{i-1} + S_i \ddot{\mathbf{q}}_i \qquad \mathbf{a}_0 = \mathbf{0} \\ \ddot{\mathbf{q}}_i &= \Phi_i \left(\tau_i - \mathbf{b}_i\right) \\ \Phi_i &= \left(S_i^T I_i^A S_i\right)^{-1} \\ \mathbf{b}_i &= S_i^T I_i^A \mathbf{a}_{i-1} - S_i^T \mathbf{p}_i^A \end{aligned} \tag{4}$$

Solving (1)-(4) in the described forward and backward passes will provide the equations of motion govern the locomotion of the snake robot on a flat surface.

III. CARTESIAN CONTROL OF SNAKE ROBOTS

The goal of control here is to follow the desired trajectories defined by the path planner. Thus, a double-layer controller, as shown in Fig. 3, was designed for snake robots to follow a specific direction with a desired linear velocity. Figure 3(a) shows the outer layer of the controller that modulates the parameters of the snake-robot's gait (in terms of relative joint angles) to control the overall locomotion of the robot. On the other hand, the inner layer, shown in Fig. 3(b), is calculating the required joint torque to generate the desired joint motion.

A. Outer Control Layer: Heading and Velocity Control

The outer-layer modulates the parameters of the sinusoidal function in order to change the direction and the linear velocity in that direction. The parameter γ_c is related to the heading direction control. The overall direction of the snake robot, $\bar{\theta}$, is defined as the average of absolute links' angle (orientation of the link). The γ_c will be determined based on a proportional controller:

$$\begin{aligned} \gamma_c &= k_\gamma (\bar{\theta}_d - \bar{\theta}) \\ \bar{\theta} &= \frac{1}{N} \sum_{i=1}^{N} q_i \end{aligned} \tag{5}$$

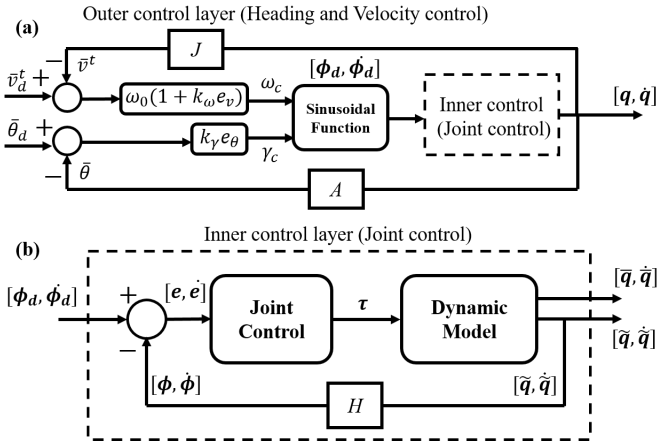

Fig. 3. a) Heading-Velocity Control Diagram and b) the Inner Joint Control.

where, k_γ is the control gain, $\bar{\theta}_d$ and $\bar{\theta}$ are the desired and actual heading directions of the snake robot, respectively. The angular frequency of the periodic motion, ω_c, will be determined, (6), to modulate the linear forward velocity.

$$\omega_c = \omega_0 \left(1 + k_\omega(\bar{v}_d^t - \bar{v}^t)\right) \tag{6}$$

where ω_0 is the base angular frequency, k_ω is the control gain, as well as $\bar{v}^t = J^t\dot{\mathbf{q}}$ and $\bar{v}_d^t$ are the actual and desired average velocity of the snake-robot's links in the tangential direction. Notice that $\bar{v}_d^t$ and $\bar{\theta}_d$ are specified by the path planner algorithm. Then, γ_c and ω_c are fed into the sinusoidal equation to define the desired joint angles and their derivative, $[\phi_d, \dot{\phi}_d]$.

B. Inner Control Layer: Joint Control

To generate the serpentine gait by the snake robot, the joints angle must be varying in a sinusoidal motion with amplitude α, phase shift of β, and bias γ as shown in (7).

$$\phi_i = \alpha \sin(\omega t + (i-1)\beta) + \gamma \tag{7}$$

To achieve this goal, a proportional-derivative (PD) controller is developed for tracking the snake robot's joint motion in following the serpentine curve as shown in Fig. 3(b). The forward dynamics of the snake robot provides the generalized coordinates and their derivatives after the numerical integration. The dynamic system's output, i.e. generalized coordinates, were split into the translational and rotational terms described by $\bar{\mathbf{q}}$ and $\tilde{\mathbf{q}}$, respectively. The rotational part was used to determine the joint motion defined by the joint angles between two adjacent links as follows;

$$\phi = H\tilde{\mathbf{q}} \quad \text{and} \quad \dot{\phi} = H\dot{\tilde{\mathbf{q}}} \tag{8}$$

where $H_{n-1\times n}$, $H_{ij,i=j} = 1, H_{ij,j=i+1} = -1, \text{and} H_{ij} = 0$

The goal here is to determine the vector of joint control action, $\boldsymbol{\tau}$, in order the relative joint angles to follow the desired sinusoidal trajectories presented in (7). The control action is defined as follows using the PD controller. The control gains will be determined in order to achieve the desired dynamic performance by the robot.

$$\boldsymbol{\tau} = K_p\left(\phi_d - \phi\right) + K_d\left(\dot{\phi}_d - \dot{\phi}\right) \tag{9}$$

where $\boldsymbol{\tau} \in \mathbb{R}^{(N-1)}$ is the vector of joint torque, K_p and K_d are matrices of proportional and derivative gain control. The calculated joint torque at each instance will be fed back into (3) to simulate the controlled locomotion of the snake-robot, which satisfies both heading direction and linear velocity.

IV. PROBABILISTIC THREAT EXPOSURE MAP (PTEM) BASED NAVIGATION

In this study, we adopt a rule-based autonomous navigation system that is previously presented by co-author Sevil and his colleagues [12]–[17]. The navigation consists of a set of "decision states", which contain rules to determine how robotic platform should move by generating heading and speed signals [12]. The autonomous guidance algorithms operate on the Probabilistic Threat Exposure Map (PTEM) and the implementation of these algorithms in simulation assumes the availability of the PTEM. Our previous research [13]–[17] has developed an obstacle mapping system that is capable of extracting the information needed to construct the PTEM directly from real-time or simulated sensor data. Further details of this approach can be found in [13]–[17].

A. Probabilistic Threat Exposure Map (PTEM)

The PTEM is used to identify restricted areas for robots based on quantifying the level of risk, including running into or getting very close to restricted regions at a given position in the area of operation [13]–[17]. PTEM is a continuous probabilistic map consisting of the sum of Multidimensional Gaussian PDFs (Probability Density Functions), formulated as

$$f(r) = \sum_{i=1}^{N} \frac{1}{2\pi\sqrt{det(K_i)}} \exp[-\frac{1}{2}(r-\mu_i)^T K_i^{-1}(r-\mu_i)] \tag{10}$$

where $\mu_i = [\mu_{x_i} \;\; \mu_{y_i}]^T$ and $K_i = diag\{\sigma_{x_i}, \sigma_{y_i}\}$ are the mean vector and the covariance matrix of the ith threat and r is the position of a point of interest. We assumed that the obstacles are circular shape in this study, thus the variances along the X and Y directions are assumed to be the same. Two parameters that are needed for specifying a Gaussian PDF are the mean value (location) and the variance (radius). In (10), PTEM is calculated for a given position in the area of operation (Fig. 4(a)). We use a threshold value, $f(r)$, to quantify the location and size of the obstacles within the area of operation. Then, applying the threshold value we can find locations of the restricted areas on the map, which are obstacles in this study, using the following equation

$$A_r \quad = \quad \{\underline{r} = [xy]^T | f(r) \geq f_r\} \tag{11}$$

which identifies the set of positions where the PTEM value is greater than or equal to the threshold, f_r (Fig. 4(b)). We keep the threshold value, $f(r)$, as constant in this study.

B. Construction of PTEM based on Simulated Sensor Data

Simulated sensor data are first clustered into regions [13]–[17]. Using squared sum errors, the centroid of the data points is calculated. The maximum distance to centroid point, then, is selected to be the radius of the cluster. The PTEM

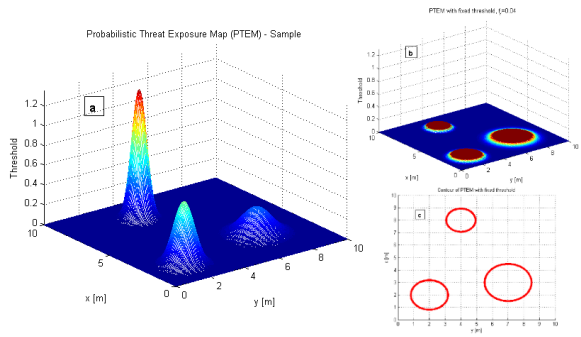

Fig. 4. Construction of Probabilistic Threat Exposure Map Sample a) Sample PTEM without Applying the Threshold, b) PTEM after Applying Threshold, and c) Restricted areas [16]

is constructed upon the cluster center (centroid) and radius, followed by calculation of the value of the PTEM that is greater than the threshold as formulated in (11). This is done by adding a Gaussian PDF for each cluster, which requires the determination of its mean values, μ_{x_i} and μ_{y_i}, as well as variances, σ_{x_i} and σ_{y_i}. The mean values are determined by assigning centroid of the respective cluster, the variance value is calculated by solving of the zero-mean Gaussian PDF for a given threshold value f_r and the radius of the respective cluster, r_{ca}.

$$f_r = 1/(\sqrt{2\pi}\sigma)\exp[-r_{ca}^2/(2\sigma^2)] \tag{12}$$

C. Autonomous Navigation

The algorithm used in this study for waypoint navigation of the snake robot utilizes the concept of a virtual target. The main goal of this approach is to generate control commands for the snake robot to follow a moving virtual target. The virtual target navigates between waypoints while avoiding the obstacles, and creates a trajectory for the actual platform to follow through. The virtual target is assumed to be ahead of the snake robot with pre-defined time step, Δt. As virtual target moves along the trajectory, the navigation algorithm uses current position of the snake robot and position of the virtual target to generate commanded heading and velocity signals for the snake robot. The virtual target's motion is defined in such a way that it moves connecting the assigned waypoints while considering PTEM information. It considers, also, the dynamical constraint of the snake robot, so that the snake robot can end up in the correct waypoint location.

V. RESULTS

Simulation results for the control of the snake-robot are presented here. A planar 6-link snake-robot (N = 6), Fig. 1, with the mass, length, and width of each link are set to $m = 0.1kg$, $2l = 0.16m$, and $2w = 0.05m$ is considered. The normal and tangential friction coefficients are set to $\mu_n = 0.5$ and $\mu_t = 0.05$, respectively. The joint parameters and control gains are set as $\alpha = \pi/12$, $\omega_0 = 2\pi rad/s$, $\beta = 2\pi/5$, $k_\omega = 6$, $k_\gamma = 0.05$, $k_p = 1.25$, and $k_d = 0.1$.

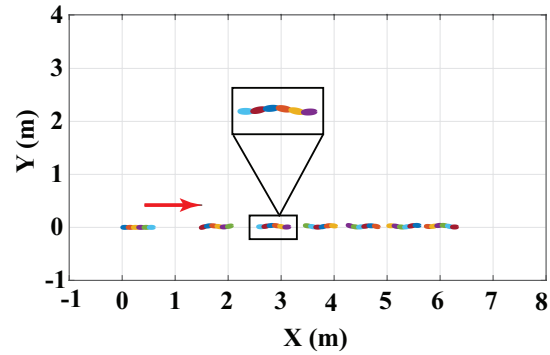

Fig. 5. Snapshots of Robot Locomotion with 0° Heading Direction.

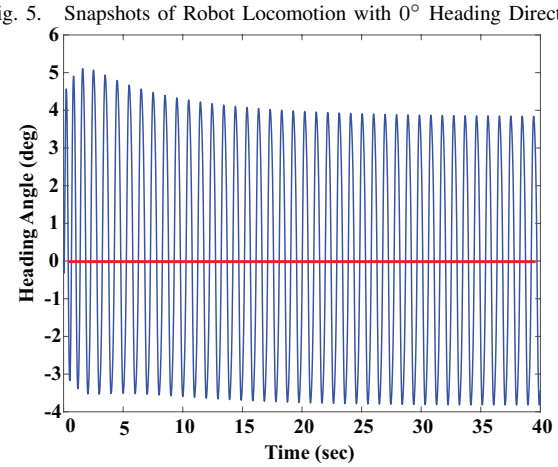

Fig. 6. Actual and Desired Heading Direction of the Snake Robot 0°.

A. Heading Direction Control

The simulation results for following two heading directions of 0° and 45° with a set velocity of $0.17m/s$ are presented. The snapshots of the snake robot locomotion in the Cartesian space are shown in Fig. 5 for the case 0°. The robot started from initial pose and followed a straight line. Figure 6 shows the time evolution of the heading angle recorded over the simulated time. The result indicates a small oscillation of ±4° around the targeted heading angle of 0° which is due to the oscillatory nature of the snake robot motion and is negligible.

Figure 7 shows the snapshots of the robot locomotion for the desired heading direction of 45°. The robot started from initial pose and then turned towards left to follow the direction. The time evolution of the heading is shown in Fig. 8. These results show the effectiveness of the controller to guide the robot for following the desired trajectories in the Cartesian and Joint spaces. Additionally, the required distance and time for the snake robot to achieve the desired 45° direction were identified as the length of the snake robot (1.0 m) and 10 seconds, respectively.

B. Velocity Control

The velocity controller was examined for achieving the desired velocity for the set heading direction of zero degrees. Figure 9 shows the snapshots of the snake robot locomotion for two targeted average velocity of $0.14m/s$ and $0.17m/s$. The time evolution of the average velocity for these two cases are shown in Fig. 10 (a) and (b). In both cases, the targeted

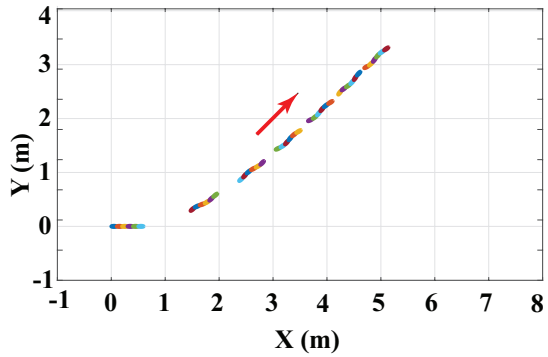

Fig. 7. Snapshots of Robot Locomotion with 45° Heading Direction.

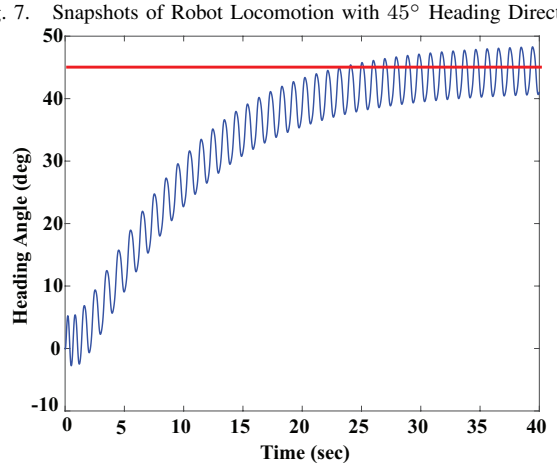

Fig. 8. Actual and Desired Heading Direction of the Snake Robot 45°.

velocity is achieved with a negligible error. The rise-time (10-90%) of the velocity responses are about 5 seconds for both cases. This rise-time is about half of the rise-time for the heading responses. The control gains can be adjusted to achieve the desired dynamic responses from the snake robot.

C. Autonomous Navigation Results

We performed simple autonomous navigation simulation experiments as proof-of-concept results. We define waypoints as $(2, 0), (3, 0.4), (4.2, 1.3)$ for the snake robot (Fig. 11), and the robot's maximum speed is set as $0.15\ m/s$. The waypoints in this study are defined with a proximity circle values. Those values define a circular area around the waypoint which snake robot determines whether it reaches to waypoint or not by entering in that circle. In the simulations, proximity circle radius is defined as $0.1\ m$. In Fig. 11, the green dotted line indicates the trajectory of the snake robot, blue line shows the trajectory of the virtual target, red marker shows the position of the waypoint, and cyan circle represents the waypoint proximity circle. The result shows a successful implementation of the snake robot navigation toward the waypoints. The actual and desired velocity and heading values are depicted in Fig. 12.

In the second case, same initial point and same waypoints are used, and an obstacle with a radius ($r = 0.075\ m$) is added at the location $(2.5, 0.27)$, in order to test obstacle avoidance capability of the proposed approach. In the Fig. 13, the black circle and markers indicate the actual size and position of the obstacle, magenta circle indicates the size and

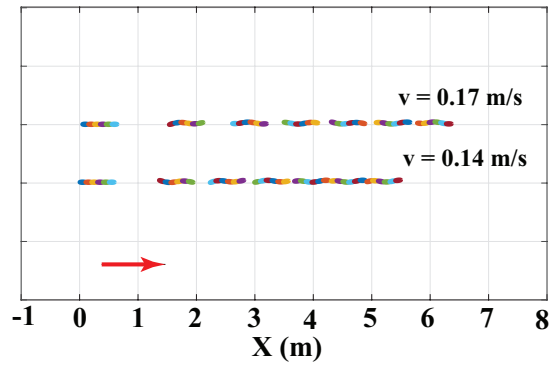

Fig. 9. Snapshots of Snake Robot Locomotion with 0° Heading Direction for the Targeted Velocities of $0.17m/s$ and $0.14m/s$.

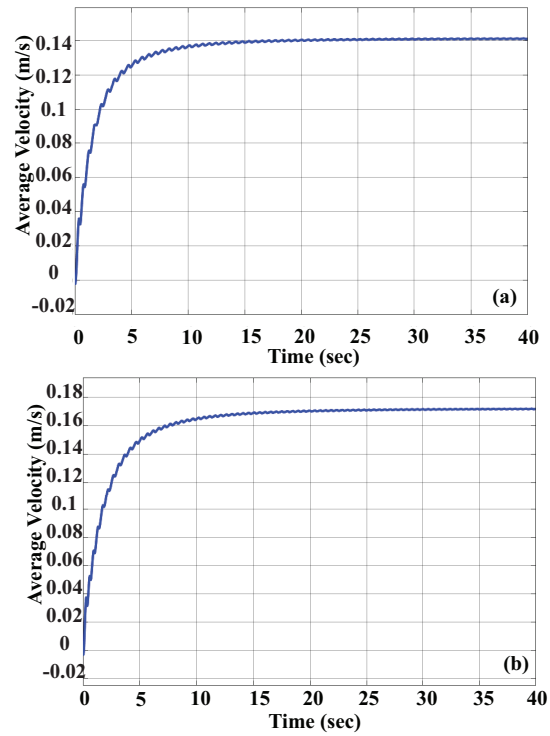

Fig. 10. Actual Average Velocity of the Snake Robot with Desired Velocity of a) $0.14m/s$ and b) $0.17m/s$.

position of the obstacle identified by the PTEM algorithm. According to the results, the snake robot successfully detects, avoids the obstacle while it navigates through the waypoint. The actual and desired velocity and heading values are depicted in Fig. 14 for the simulations with obstacle. It should be noted that for both simulation experiments the virtual target creates its trajectory considering the obstacle, waypoints, as well as the dynamic constraints of the snake robot. The obstacle avoidance effect, in fact, can be seen in the virtual target's resulting trajectory, which allows snake robot to navigates successfully between waypoints.

CONCLUSIONS

This paper presents the autonomous navigation and obstacle avoidance of a planar snake robot locomotion with combined velocity-heading control in a simulation environment. The kinematics, dynamics, and control of the snake robot were developed. Additionally, a rule-based expert system for autonomous navigation while avoiding obstacles/restricted-

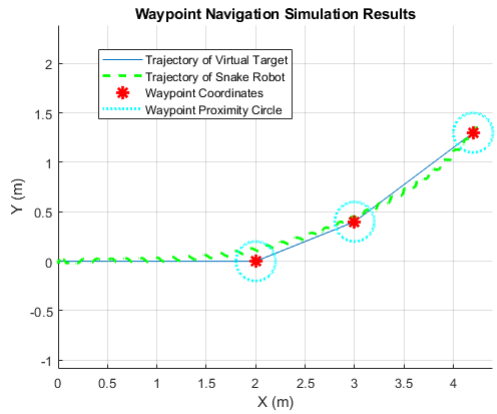

Fig. 11. Waypoint Navigation Simulation Result

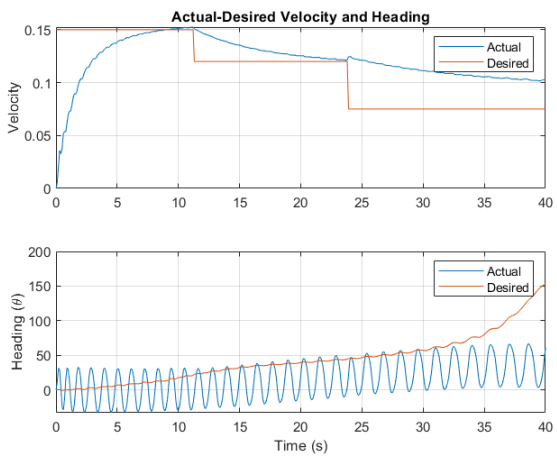

Fig. 12. Actual and Desired Velocity and Heading - no Obstacle

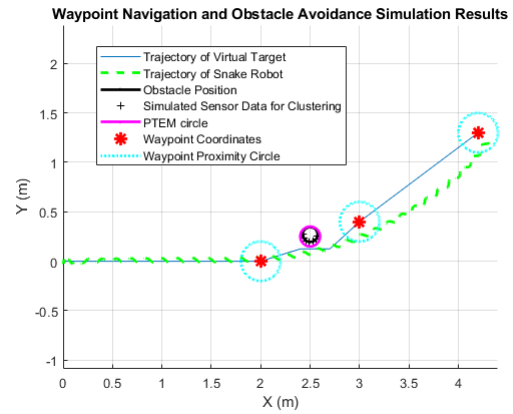

Fig. 13. Waypoint Navigation Simulation Result - with Obstacle

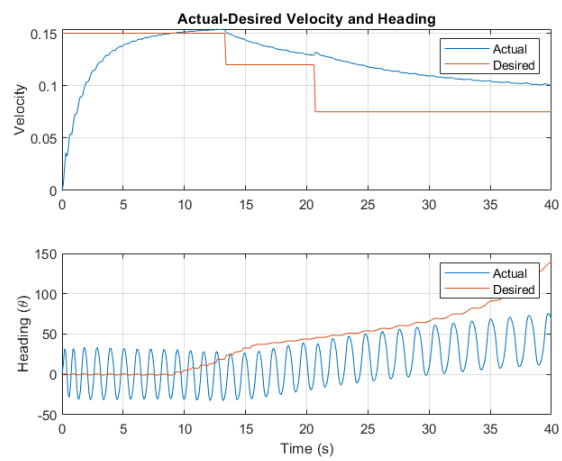

Fig. 14. Actual and Desired Velocity and Heading - with Obstacle

areas was adapted. These control, navigation, and obstacle avoidance algorithms were examined on a planar 6-link snake robot where the results indicated their effective performance. The future work will study the effect of the robot's parameters variation on the overall performance, consider the locomotion of snake robots in a cluttered and dynamic environment, and the algorithms will be tested on an actual snake robot for comparison.

References

[1] K. Dowling, "Limbless locomotion: learning to crawl," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 4. IEEE, 1999, pp. 3001–3006.

[2] J. Gray, "The mechanism of locomotion in snakes," *Journal of experimental biology*, vol. 23, no. 2, pp. 101–120, 1946.

[3] S. Hirose, *Biologically inspired robots: snake-like locomotors and manipulators*. Oxford university press Oxford, 1993, vol. 1093.

[4] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, *Snake robots: modelling, mechatronics, and control*. Springer Science & Business Media, 2012.

[5] S. Ma, H. Araya, and L. Li, "Development of a creeping snake-robot," in *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No. 01EX515)*. IEEE, 2001, pp. 77–82.

[6] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE transactions on robotics and automation*, vol. 11, no. 6, pp. 781–793, 1995.

[7] M. Ishikawa, "Iterative feedback control of snake-like robot based on principal fibre bundle modelling," *International Journal of Advanced Mechatronic Systems*, vol. 1, no. 3, pp. 175–182, 2009.

[8] G. Hicks and K. Ito, "A method for determination of optimal gaits with application to a snake-like serial-link structure," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1291–1306, 2005.

[9] M. Haghshenas-Jaryani and G. Vossoughi, "Modeling and sliding mode control of a snake-like robot with holonomic constraints," in *2008 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2009, pp. 454–461.

[10] ——, "Trajectory control of snake-like robots in operational space using a double layer sliding mode controller," ser. (IDETC/CIE), vol. 5A: 39th Mechanisms and Robotics Conference, 08 2015, dETC2015-46480. [Online]. Available: https://doi.org/10.1115/DETC2015-46480

[11] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[12] A. Dogan and U. Zengin, "Unmanned aerial vehicle dynamic-target pursuit by using probabilistic threat exposure map," *AIAA Journal of Guidance, Control and Dynamics*, vol. 29, no. 4, pp. 944–954, 2006.

[13] P. Desai, H. E. Sevil, A. Dogan, and B. Huff, "Construction of an obstacle map and its real-time implementation on an unmanned ground vehicle," in *Proc. of TePRA 2011, IEEE International Conference on Technologies for Practical Robot Applications*, Massachusetts, USA, 11-12 April 2011, pp. 139–144.

[14] H. E. Sevil, P. Desai, A. Dogan, and B. Huff, "Modeling of an unmanned ground vehicle for autonomous navigation and obstacle avoidance simulations," in *Proc. of DSCC 2012, ASME Dynamic Systems and Control Conference*, Ft. Lauderdale, USA, 17-19 October 2012, pp. 529–534, DSCC2012-8867.

[15] ——, "Real-time obstacle avoidance and waypoint navigation of an unmanned ground vehicle," in *Proc. of DSCC 2012, ASME Dynamic Systems and Control Conference*, Ft. Lauderdale, USA, 17-19 October 2012, pp. 263–271, DSCC2012-8843.

[16] O. Daskiran, H. E. Sevil, A. Dogan, and B. Huff, "UGV and UAV co-operation for constructing probabilistic threat exposure map (PTEM)," in *Proc. of 15th AIAA Aviation Conference*, Dallas, USA, 22-26 June 2015, AIAA 2015-2740.

[17] G. Rajashekaraiah, H. E. Sevil, and A. Dogan, "PTEM based moving obstacle detection and avoidance for an unmanned ground vehicle," in *Proc. of ASME 2017 Dynamic Systems and Control Conference (DSCC 2017)*, Virginia, USA, 11-13 October 2017, DSCC2017-5330.