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

A Bayesian-Based Controller for Snake Robot Locomotion in Unstructured Environments

Yuanyuan Jia[1] and Shugen Ma[2]

Abstract— **This paper presents a novel Bayesian-based controller for snake robots in cluttered environment. It extends the conventional shape-based compliant control into statistical field providing an explicit mathematical formulation with Bayesian network. A sequential density propagation rule is derived by introducing several probability densities in a unified framework. Specifically, two input influence densities are proposed to model the cumulative effect of various external forces that the snake robot undergoes. Moreover, the measurement likelihood model is exploited to give a more robust closed-loop feedback. Overall, the proposed approach provides an innovative way to handle challenging tasks of snake robot control in complicated environment. Experimental results have been demonstrated for both simulation and real-world data.**

I. INTRODUCTION

Snake robot received a significant amount of attention in recent years motivated by its wide applications [1]. However, its high redundant structure makes snake robot control still very challenging [2]. Many approaches have been studied to circumvent the problems inherent in this task. Most earlier efforts studied the biomechanics of biological snakes. Hirose proposed the serpenoid curve in [1]. Ma studied the serpentine curve in [3]. Sophisticated control methods have been adopted for snake robot applications using techniques such as Central Pattern Generator (CPG) [2] and model-based methods [4]. However, these approaches usually fail when snake robots move in unstructured environment or present collisions with obstacles.

Different methods have been studied for the locomotion of snake robots exploiting contact with obstacles. A hybrid model is proposed by Transeth et al. using the dynamics of snake robots and the contact force with obstacles in [5]. Kano et al. [6] investigated a decentralized control method with local reflexes from contact sensors. Many methods for obstacle-aided snake robots rely on accurate modeling and sensing assumptions, where the number, convexity of collision points and the friction models are usually predefined. These assumptions limit their performance in unstructured environments where constraints are commonly hard to foreknow and difficult to model. In order to simulate the biologic interaction strategy of snakes with obstacles and exploit machine learning techniques, much research considers neural network based methods [2]. This notion is carried further in the work of Sartoretti et al. [7] where the A3C algorithm is used by a trained agent. Although this approach provides a promising direction, it does not give an explicit mathematical model of the interaction with environment yet. Moreover, due to the limitation of supervised learning method, it highly depends on training data and may not deal with unlearned situation well.

Various approaches have been proposed for obstacle avoidance of snake robots. Wu and Ma presented a neurally controlled steering method for collision free behavior of a snake robot in [8]. Tanaka et al. [4] presented a range-sensor-based method for semiautonomous whole body collision avoidance and discussed the self-collision avoidance. Recently, an intuitive and computationally effective approach to snake robots addressing several difficulties inherent in this complex task is based on a shape-based compliant controller [9][10]. Although this kind of methods can handle the external interaction in principle, they require to precisely measure the forces or torques and have a good estimation of the controller's gain matrices, which limits their application especially for unknown clutter environment where the dynamics may be varying and hard to model accurately.

In this paper, we propose a Bayesian network based controller for snake robots. It unifies several existing state-of-the-art approaches into a consistent formulation. Specifically, a novel conditional density propagation scheme has been derived for snake robot control in unstructured environment with perturbations. Moreover, by estimating the input influence density, measurement likelihood, and the state transition in a sequential importance sampling implementation, we explicitly handle the obstacle interaction and shape variation problems in an innovative way.

II. RELATED WORK

We first briefly review the related work of shape-based compliance control and statistical process control.

A. Shape-Based Compliant Control for Snake Robot

Shape-based compliant control adopts an admittance controller to change the snake robot's shape parameters by the external force F_t [10]. For example, if we define $\beta_t = (A_t, \gamma_t)^T$ where A and γ are the amplitude and angular offset respectively in the serpenoid curve model [1], the admittance controller is given by

$$M\ddot{\beta}_t + B\dot{\beta}_t + K\beta_t = F_t, \tag{1}$$

where $M, B, K \in \mathbb{R}^{2\times 2}$ represent the effective mass, damping, and spring constant matrices, respectively. By embedding the serpenoid model inside the shape function and adjusting shape parameters through the above controller, this

[1,2]Yuanyuan Jia and Shugen Ma are with Department of Robotics, Ritsumeikan University, 1 Chome-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan jiayuanusa@gmail.com, Shugen.Ma@ieee.org

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

method can effectively change different portions of the snake robot's body responding to meeting obstacles.

Several challenges remain open for the above controller: 1) The optimal gain matrices M, B, K are commonly difficult to be estimated for a snake robot; 2) The external force F_t is hard to be accurately measured in collisions since it is fluctuating in terms of force direction, angle, magnitude, etc.; 3) It does not have an observer design even though the external force may implicitly change the snake robot's posture. Therefore, the controller state could not be estimated accurately. This inspires us to propose a more sophisticated shape-based control formulation.

B. Statistical Process Modeling for Snake Robot

Many reported controllers such as in [4][9][10] are all deterministic systems. When sensors provide noisy stochastic measurements and input signal has disturbance, the state can only be observed partially. Generally speaking, the control problem under uncertainties has to be formulated as a stochastic problem with noisy observations. Deterministic mathematical modeling of dynamic systems is usually imperfect, and a statistical approach is necessary to estimate unknown parameters and to evaluate their accuracies.

Kalman filtering has been widely used for solving stochastic problems [11]. Rollinson et al. [12] applied Kalman filter to state estimation of snake robots. However, conventional Kalman filters take a Gaussian assumption for the distribution of noise, which is not held in many applications. More sophisticated statistical methods such as Bayesian network [13] have been studied for different aspects of dynamic robot systems such as navigation and control. Nevertheless, to the best of our knowledge, not enough work has been done for snake robot using Bayesian network in stochastic field yet.

III. STOCHASTIC SHAPE-BASED BAYESIAN CONTROL

In this section, we propose a novel Bayesian formulation for snake robot control.

A. Shape-Based Bayesian Network Modeling for Snake Robot

An articulated snake robot with six modules can be represented by a Bayesian network [13] such as shown in Fig. 1. It has two layers: the hidden state layer (circle nodes) and the measurement layer (square nodes). Each circle node corresponds to a *module* of the snake. The undirected links represent physical constraints among different modules. Each individual module is associated with its measurement. The directed link from a module's state to its associated measurement represents the local measurement likelihood.

In order to keep the serpenoid shape [1] for the snake robot, a joint state representation is adopted. We denote the state at time t by $\mathbf{x}_t = (\mathbf{x}_t^1, \ldots, \mathbf{x}_t^i, \ldots, \mathbf{x}_t^I)$, where I is the total number of modules, i the module index. Specifically, since the snake robot implements the serpenoid shape function similar as [9] for a better comparison, the state is chosen as $\mathbf{x}_t^i = (A_t^i, \gamma_t^i)$ where $i = 1, \ldots, I$ is the module index, A is the amplitude, γ is the angular offset of serpenoid curve [1]. When these parameters are changing, different shapes would be generated. Moreover, we denote the measurement of $\mathbf{x}_t$ by $\mathbf{q}_t = (\widetilde{A}_t, \widetilde{\gamma}_t)$ which is estimated by a camera based object detector in this work, the set of all history states up to time t by $\mathbf{x}_{0:t}$ where $\mathbf{x}_0$ is the initialization prior, the set of all measurements up to time t by $\mathbf{q}_{1:t}$.

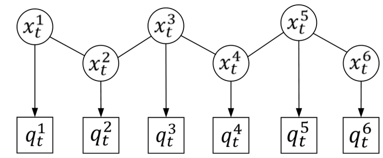

Fig. 1. Bayesian network for a six-modular snake robot.

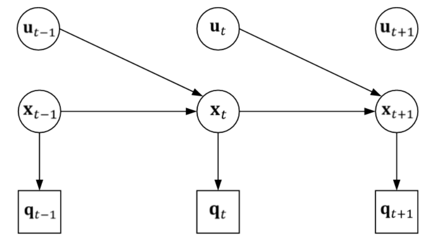

Fig. 2. Dynamical modeling for snake robot locomotion.

Considering the locomotion problem for a snake robot, we further accommodate the state transition by a dynamical Bayesian network such as shown in Fig. 2. It contains three consecutive time slices. The directed link between consecutive states represents the state transition which is assumed as a *Markov* chain. An additional layer is introduced to represent the input $\mathbf{u}_t$, which is the virtual accumulative force generated by environmental objects such as targets and obstacles during interaction. Similarly, we denote the set of all inputs up to time t by $\mathbf{u}_{0:t}$. By applying the *Separation Theorem* [13], the following *Markov Properties*, i.e., conditional independence properties can be easily verified from Fig. 2:

$$p(\mathbf{u}_t|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}, \mathbf{q}_{1:t}) = p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) \tag{2a}$$

$$p(\mathbf{q}_t|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}, \mathbf{q}_{1:t-1}) = p(\mathbf{q}_t|\mathbf{x}_t) \tag{2b}$$

$$p(\mathbf{x}_{t+1}|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}) = p(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{2c}$$

B. Bayesian Conditional Density Propagation

Dynamic Bayesian network shown above has advantages for statistical analysis comparing with Kalman filters and difference equations. Specifically, it provides more flexible potential for modeling stochastic process. In order to design a controller, we investigate the prediction problem in this section. This can be achieved by inferring the joint posterior $p(\mathbf{u}_{0:t}, \mathbf{x}_{0:t+1}|\mathbf{q}_{1:t})$, which captures all the history of evolution of uncertainties:

$$
\begin{aligned}
& p(\mathbf{u}_{0:t}, \mathbf{x}_{0:t+1}|\mathbf{q}_{1:t}) \\
&= p(\mathbf{u}_t|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}, \mathbf{q}_{1:t}) p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}|\mathbf{q}_{1:t}) \quad (3)\\
&= p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}|\mathbf{q}_{1:t}) \quad (4)\\
&= \frac{p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) p(\mathbf{q}_t|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}, \mathbf{q}_{1:t-1})}{p(\mathbf{q}_t|\mathbf{q}_{1:t-1})} \\
&\quad \cdot p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t+1}|\mathbf{q}_{1:t-1}) \quad (5)\\
&= \frac{p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) p(\mathbf{q}_t|\mathbf{x}_t) p(\mathbf{x}_{t+1}|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t})}{p(\mathbf{q}_t|\mathbf{q}_{1:t-1})} \\
&\quad \cdot p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t-1}) \quad (6)\\
&= \frac{p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) p(\mathbf{q}_t|\mathbf{x}_t) p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p(\mathbf{q}_t|\mathbf{q}_{1:t-1})} \\
&\quad \cdot p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t-1}) \quad (7)\\
&= c_t p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) p(\mathbf{q}_t|\mathbf{x}_t) p(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&\quad \cdot p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t-1}). \quad (8)
\end{aligned}
$$

In (4), the *Markov* property (2a) is used. In (6), we apply the *Markov* property (2b). In (7), we adopt the property (2c). The denominator in (7) can be regarded as a normalization constant since it is not related to the state.

The above Bayesian stochastic formulation explicitly models the physical interaction between the snake robot and its surrounding environment through observation measurement. Clearly, the posterior at time t is affected by four factors: (1) the input influence density $p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t)$; (2) the measurement likelihood $p(\mathbf{q}_t|\mathbf{x}_t)$; (3) the state transition density $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$; (4) the posterior $p(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t-1})$ at the previous time $t-1$.

IV. DENSITY MODELING AND SEQUENTIAL IMPORTANCE SAMPLING IMPLEMENTATION

In order to estimate the posterior derived in the above section, we use the *sequential importance sampling* (SIS) [14] as a paradigm. Other density estimation methods can be used instead. The basic idea of SIS approximation [14] is to use a weighted sample set $\{\mathbf{x}_{0:t}^n, w_t^n\}_{n=1}^{N_s}$ to estimate the posterior density, where $\{\mathbf{x}_{0:t}^n, n = 1, \ldots, n_s, \ldots, N_s\}$ are the samples, $\{w_t^n, n = 1, \ldots, n_s, \ldots, N_s\}$ the associated normalized weights, and $\sum_n w_t^n = 1$. According to the *importance sampling theory* [14], the samples $\mathbf{x}_{0:t}^n$ can be generated from an importance density $f(\cdot)$ with associated importance weights:

$$
w_t^n \propto \frac{p(\mathbf{u}_{0:t}, \mathbf{x}_{0:t+1}^n|\mathbf{q}_{1:t})}{f(\cdot)}. \quad (9)
$$

For the sequential case, if the importance density $f(\cdot)$ is chosen to factorize as,

$$
\begin{aligned}
& f(\mathbf{u}_{0:t}, \mathbf{x}_{0:t+1}|\mathbf{q}_{1:t}) \\
&= f(\mathbf{u}_t, \mathbf{x}_{t+1}|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}, \mathbf{q}_{1:t}) f(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t}) \\
&= f(\mathbf{u}_t, \mathbf{x}_{t+1}|\mathbf{x}_t) f(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t-1}). \quad (10)
\end{aligned}
$$

where the *Markov* properties $f(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t}) = f(\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}|\mathbf{q}_{1:t-1})$ and $f(\mathbf{u}_t, \mathbf{x}_{t+1}|\mathbf{u}_{0:t-1}, \mathbf{x}_{0:t}, \mathbf{q}_{1:t}) = f(\mathbf{u}_t, \mathbf{x}_{t+1}|\mathbf{x}_t)$ from Fig. 2 are applied. Then by substituting (8) and (10) into (9), we have

$$
w_t^n \propto w_{t-1}^n \frac{p(\mathbf{u}_t|\mathbf{x}_{t+1}^n, \mathbf{x}_t^n) p(\mathbf{q}_t|\mathbf{x}_t^n) p(\mathbf{x}_{t+1}^n|\mathbf{x}_t^n)}{f(\cdot)}. \quad (11)
$$

The dynamics $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ can be estimated using a random walk model. Estimation of the input influence density $p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t)$, the measurement likelihood $p(\mathbf{q}_t|\mathbf{x}_t)$ and the choice of the importance density $f(\mathbf{u}_t, \mathbf{x}_{t+1}|\mathbf{x}_t)$ are not trivial and will be discussed in the following subsections.

A. *Input Influence Model*

The input influence density $p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t)$ models the interaction between environmental inputs and predicted states. Estimation of this density should consider different applications and is usually critical in practical implementation. Kelasidi et al. [15] adopted an Artificial Potential Field (APF) model for snake robot locomotion. Inspired by this method, we propose two efficient input influence models dealing with target seeking and obstacle interaction. Since these two different interactions are independent, we have,

$$
p(\mathbf{u}_t|\mathbf{x}_{t+1}, \mathbf{x}_t) = p(\mathbf{u}_{a,t}|\mathbf{x}_{t+1}, \mathbf{x}_t) p(\mathbf{u}_{r,t}|\mathbf{x}_{t+1}, \mathbf{x}_t) \quad (12)
$$

where $p(\mathbf{u}_{a,t}|\mathbf{x}_{t+1}, \mathbf{x}_t)$ models the influence by virtual attraction force from the mission target, $p(\mathbf{u}_{r,t}|\mathbf{x}_{t+1}, \mathbf{x}_t)$ formulates influence of the accumulative repulsion force incurred by obstacles. The formulation of these densities is presented as follows.

1) Target Influence Model: In a target searching task, the objective is to reach the goal of mission. Therefore, an attraction potential model can be assumed to direct the robot for searching an optimal path [15]. Similarly, we define the attraction potential function for sample $\mathbf{x}_{t+1}^n$ as,

$$
p(\mathbf{u}_{a,t}|\mathbf{x}_{t+1}^n, \mathbf{x}_t^n) = \frac{1}{\alpha_a} \exp\left\{-\frac{d_{a,n,t}^2}{\sigma_a^2}\right\} \quad (13)
$$

where α_a is a constant, σ_a is a prior constant characterizing the maximal distance for attraction. $d_{a,n,t}$ denotes the distance to the goal, for instance, it can be a Euclidean distance $d_{a,n,t} = \|\mathbf{u}_{a,t} - \Delta\mathbf{x}_{t+1}^n\|$, where $\Delta\mathbf{x}_{t+1}^n = \mathbf{x}_{t+1}^n - \mathbf{x}_t^n$. Different with the conventional APF model [15] where the goal is static, the target in the proposed model can change or move during the snake robot's locomotion.

2) Obstacle Influence Model: A repulsion potential model is commonly adopted for snake robot control with environment interactions [15]. Similarly, we denote the repulsive potential function for sample $\mathbf{x}_{t+1}^n$ by,

$$
p(\mathbf{u}_{r,t}|\mathbf{x}_{t+1}^n, \mathbf{x}_t^n) = 1 - \frac{1}{\alpha_r} \exp\left\{-\frac{d_{r,n,t}^2}{\sigma_r^2}\right\} \quad (14)
$$

where α_r is a normalization constant, σ_r is a prior constant characterizing the allowed maximal repulsion distance, $d_{r,n,t}$ is the distance between $\mathbf{x}_{t+1}^n$ and the obstacle $\mathbf{u}_{r,t}$, for example, it can be a Euclidean distance $d_{r,n,t} = \|\mathbf{u}_{r,t} - \Delta\mathbf{x}_{t+1}^n\|$ where $\Delta\mathbf{x}_{t+1}^n = \mathbf{x}_{t+1}^n - \mathbf{x}_t^n$. Different with the existing APF method in [15] where all obstacles are handled together, only

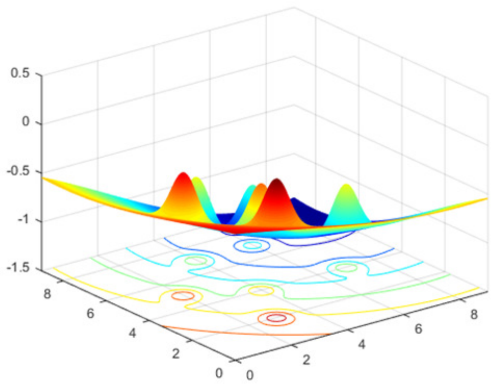

Fig. 3. The input influence potential field with obstacles and one target.

neighboring obstacles are computed based on necessities for the sample $\mathbf{x}_{t+1}^n$, which makes the model more practical for real-world applications since the total number of obstacles and their distribution are hard to predefine in advance and may be dynamically changing. Moreover, this scheme is also effective to handle the *local optimum problem* when computing the potential field since only neighbouring obstacles are considered.

During the navigation in a cluttered environment, the snake robot moves in an area with potentials created by the target and obstacles. Fig. 3 illustrates an example of such kind of input influence potential field for unstructured terrain with one target and six obstacles. As we can see, it looks like a big slope where the target locates at the bottom. The influence of target attraction is global while the obstacles' influence is local. Each obstacle plays a role as a *hill*. Then the locomotion of snake robot is like passing through a hilly land until reaching the target. Such kind of potential field is not static but dynamically updating during the whole journey.

B. Measurement Likelihood Model

Benefiting from the serpenoid function embedded in the state, the snake robot can generate a cyclic gait and thus move forward with the state transition. Exploiting the interaction models discussed above, the snake robot can respond to the external forces in cluttered terrain and change its motion status. However, without an observer design, the snake robot has no estimation of states, making the system reliability low. The measurement likelihood density $p(\mathbf{q}_t|\mathbf{x}_t)$ in the proposed framework estimates the uncertainties between snake robot's current state and measurement, which can serve as a feedback and help to solve the above problem. However, how to model this density is critical but also challenging. The inspiration for developing this model was based on graceful shape deformation of biological snakes moving through unstructured terrains. When encountering environmental disturbances, snakes can make fast transient responses but seldom overreact mainly because they know their internal shape status and tend to make any change smoothly. The objective of an effective observation feedback model should be similar to those observed in the biological organisms, which can be achieved by a measurement likelihood function for sample $\mathbf{x}_t^n$,

$$p(\mathbf{q}_t|\mathbf{x}_t^n) = \frac{1}{\alpha_m}\exp\left\{-\frac{d_{m,n,t}^2}{\sigma_m^2}\right\} \tag{15}$$

where α_m is a normalization constant, σ_m is a prior constant characterizing the maximal effective distance, $d_{m,n,t}$ is the distance between $\mathbf{x}_{t+1}^n$ and the corresponding measurement $\mathbf{q}_t$, for example, an Bhattacharyya distance is accepted in our implementation. This model provides the system with an ability to simulate the shape deformation observed in real snakes when responding to environmental perturbations.

Overall, intuitively speaking, the proposed framework can achieve robust control for snake robot by four reasons: 1) The serpenoid curve model guarantees the snake robot moving in a rhythmic gait; 2) The state transition density introduces motion randomness and thus endows the robot's locomotion with more possibilities; 3) The input influence stimulates the motion variation and shape deformation; 4) The measurement likelihood provides a closed-loop feedback and thus retains the transformation in a fast and smooth way. All of these four factors are indispensable to achieve a robust control for snake robots. When no target or obstacle appears, the input influence density will be uniformly distributed. If the state transition also adopts a uniform distribution, the controller will degrade to a simple serpenoid curve model.

C. Importance Density

The efficiency of a sequential importance sampling based approach is strongly dependent on the selected importance density $f(\cdot)$. When $f(\cdot)$ is close to the true posterior, the samples are more effective. A natural choice of the importance density is the state dynamics $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$. In our experiments, we choose this common choice for simplicity.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed Bayesian-Network-based Controller (BNC) was compared with the Shape-based Compliant Controller (SCC) [10] on both simulation and real world data. Table I presents a brief description of the proposed algorithm for one time slot. Five hundred samples were used to predict the joint state density in our experiments.

TABLE I
BRIEF DESCRIPTION OF BNC ALGORITHM

- ∘ $j = 0$, Sampling $\mathbf{x}_{t+1}^n$ from $f(\mathbf{u}_t, \mathbf{x}_{t+1}|\mathbf{x}_t)$
- ∘ Initial Weighting, $w_{t+1}^n \sim \mathbf{x}_{t+1}^n$
- ∘ Initial Predication, $\hat{\mathbf{x}}_{t+1,j}=\sum_{n=1}^{N_s} w_{t+1}^n \cdot \mathbf{x}_{t+1}^n$
- ∘ Loop $j = 1 : J$ # Re-sampling Scheme
  - – Input Influence Weighting, $p_1(\cdot)$
  - – Measurement Likelihood Weighting, $p_2(\cdot)$
  - – Updating Weights, $w_{t+1}^n = w_{t+1}^n \cdot p_1(\cdot) \cdot p_2(\cdot)$
  - – Normalizing Weights
  - – Updating Predication, $\hat{\mathbf{x}}_{t+1,j}=\sum_{n=1}^{N_s} w_{t+1}^n \cdot \mathbf{x}_{t+1}^n$
- ∘ End Loop j

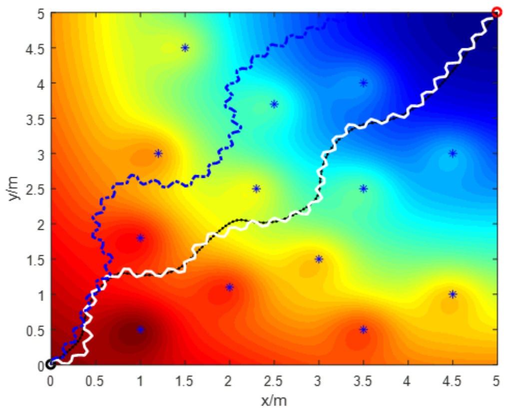

Fig. 4. Simulation results. The black curve is the optimal path. The white one is achieved by BNC. The blue dash curve is using SCC.

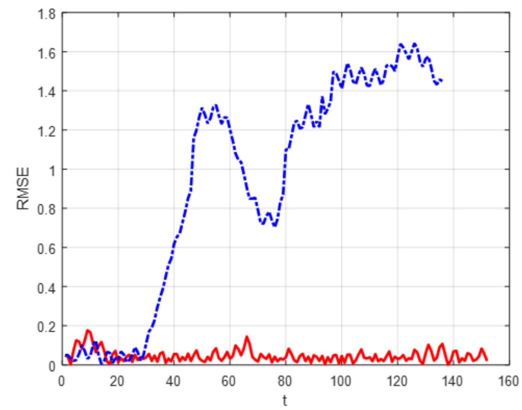

Fig. 5. Comparison of Root-Mean-Square-Error (RMSE) using SCC (blue dash curve) and BNC (red solid curve).

A. Simulation Results

We designed a simulation in Matlab for a thorough comparison. Fig. 4 shows the experimental scenario with one target and thirteen obstacles, which are all randomly arranged. A potential field is generated by calculating the proposed input influence models. Although sharing similarities with APF [15], we only consider the obstacle in the neighbor of snake robot during the locomotion. By doing this, the local optimum problem commonly annoying in APF-type methods could be successfully avoided. Three trajectories are illustrated in the Fig. 4. The blue dash curve is generated by the head of a snake robot using SCC. The white solid curve shows the results of BNC. The black solid curve is an optimal path computed by a gradient-descent algorithm based on the input influence potential field. In experiments, we find that the performance of SCC is sensitive to the initialization status. Different initial direction and position may lead to various trajectories. As illustrated in Fig. 4, the SCC method falsely misses the target direction in the middle and runs outside the field. Although the external forces are partially modeled, the collision is still uncontrollable in terms of contacting direction, degree, and the phase of snake robot's serpenoid gait. However, benefiting from the closed-loop design, BNC can achieve much more robust performance. As long as the target can be detected, the snake robot tends to move for it. When it approaches to an obstacle, the effect of repulsion force will be triggered. The sample close to the obstacle will have a smaller weight while the sample away to the obstacle will be given a larger weight. Such a *reward-far-punish-close* scheme will deform the snake robot's shape and motion curve.

Fig. 5 shows the RSME of SCC and BNC on ten tests with the same target and similar initial position. The curves are calculated by computing summation of the difference between each method's trajectory and the optimal path. It can be seen that BNC is more robust. The errors of BNC are caused by three factors: 1) Due to the different initial status, it may need an adjustment process at the beginning; 2) The uncertainty may cause a bias during sharp turnings; 3) The environmental friction and interference may further generate additional deviations. These kinds of uncertainties are all unpredictable in the real-world applications but need to be considered. This is why a stochastic framework like the proposed BNC is desirable.

B. Real-World Data

A snake robot similar as reported in [16] was used to implement the proposed BNC method. It has five actuators chained together in a serial configuration. The links are composed of 3D-printed connection blocks and separated by DYNAMIXEL RX-24F servo motors. A single free wheel with a rubber cover is mounted at the bottom of each link. The robot was controlled by signals sent from a Linux computer through an RS-485 communication link. Similar as [10], an overhead camera was used to monitor the scene. Different colorful tags were used to label objects for simplicity, specifically, red for target, blue for obstacle, yellow for robot head, and green for body module. A learning based video detector was exploited to find the position of these objects in real-time. Then, the input influence density and measurement likelihood could be dynamically calculated, specifically, measurement $\mathbf{q}_t$ was estimated by a link model [2] between centers of adjacent modules.

Fig. 6 illustrates the performance using both SCC and the proposed BNC in a real-world test, where one target and more than twenty obstacles are randomly set in the scene. It challenges many algorithms because of the presence of moving targets and obstacles. SCC (1st row) suffered from the *unexpected collision* problem during the interaction with obstacles. The reason is mainly because of the complexity of environmental interactions. For example, the friction situation, collision angle, and strength of contacting force are all not handled explicitly in the controller. In the experiments, we found that the controller was sensitive to the initial pose. Moreover, although the shape was deformed by portions, it could not respond to moving obstacles in time due to lack of a closed-loop design. Furthermore, without a target influence model, it usually missed the desirable direction after

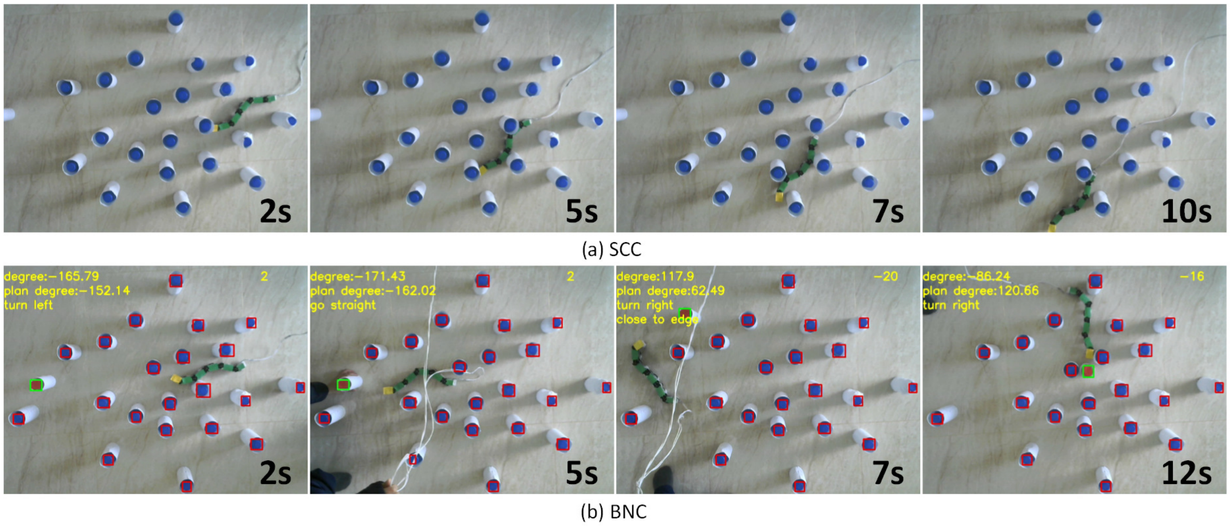

Fig. 6. Comparison of SCC (1st row), and the proposed BNC (2nd row) for unstructured environments.

spontaneous collision with obstacles and finally went out of the scene quickly. However, the proposed BNC (2nd row) performed superiorly even with moving obstacles and targets. In most situations with different initialization status, the proposed BNC method presented an impressive performance. Although a joint state space is adopted, the snake robot's shape is not exclusively changed as a whole. On the contrary, different portions can be flexibly stimulated by external inputs similar as [7]. Specifically, due to the various position along a snake robot, each module may receive different interaction forces even when reaching to the same obstacle. In such a situation, the distances defined in input influence models are quite different among individual modules and thus play an important role to change the snake robot's shape. Thus, the proposed framework endows the entire body with a particular ability to deform locally by environmental features. Compared with the deterministic controller such as SCC, the snake robot's locomotion using BNC is more *active* and *agile*. The reason may be the stochastic modeling inherent in the non-Gaussian density propagation. Benefiting from keeping multi-hypotheses of both motion and shape variations, the snake robot can respond to instant moving target and obstacles quickly.

VI. CONCLUSIONS

We have presented a novel Bayesian-based controller for snake robots by modeling the interaction with environmental objects using probability density propagation. Preliminary experimental results have demonstrated promising performance in unstructured circumstances. For the future work, we would like to investigate more sophisticated interaction models and extend the proposed framework to more problems such as simultaneous localization and mapping.

REFERENCES

[1] S. Hirose, P. Cave, and C. Goulden, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. Oxford University Press, 1993.

[2] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A review on modelling, implementation, and control of snake robots," *Robotics and Autonomous Systems*, vol. 60, pp. 29–40, 2012.

[3] S. Ma, "Analysis of snake movement forms for realization of snake-like robots," *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3007–3013, 1999.

[4] M. Tanaka and K. Tanaka, "Shape control of a snake robot with joint limit and self-collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 25, pp. 1441–1448, 2017.

[5] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments," *IEEE Transactions on Robotics*, vol. 24, pp. 88–104, 2008.

[6] T. Kano, T. Sato, R. Kobayashi, and A. Ishiguro, "Local reflexive mechanisms essential for snakes' scaffold-based locomotion." *Bioinspiration biomimetics*, vol. 7, no. 4, p. 046008, 2012.

[7] G. Sartoretti, W. Paivine, Y. Shi, Y. Wu, and H. Choset, "Distributed learning of decentralized control policies for articulated mobile robots," *IEEE Transactions on Robotics*, vol. 35, no. 5, 2019.

[8] X. Wu and S. Ma, "Neurally controlled steering for collision-free behavior of a snake robot," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 2443–2449, 2013.

[9] J. Whitman, F. Ruscelli, M. J. Travers, and H. Choset, "Shape-based compliant control with variable coordination centralization on a snake robot," *IEEE Conf. on Decision and Control (CDC)*, pp. 5165–5170, 2016.

[10] M. J. Travers, J. Whitman, and H. Choset, "Shape-based coordination in locomotion control," *The International Journal of Robotics Research*, vol. 37, pp. 1253 – 1268, 2018.

[11] Y. Pei, S. Biswas, D. S. Fussell, and K. Pingali, "An elementary introduction to kalman filtering," *Commun. ACM*, vol. 62, pp. 122–133, 2017.

[12] D. Rollinson, H. Choset, and S. Tully, "Robust state estimation with redundant proprioceptive sensors," in *Proceedings of the ASME 2013 Dynamic Systems and Control Conference*, 2013.

[13] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. Wiley, 1990.

[14] A. Gelman and X.-L. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Statistical Science*, vol. 13, no. 2, pp. 163–185, 1998.

[15] E. Kelasidi, K. Y. Pettersen, and J. T. Gravdahl, "A waypoint guidance strategy for underwater snake robots," *22nd Mediterranean Conference on Control and Automation*, pp. 1512–1519, 2014.

[16] A. Kakogawa, S. Jeon, and S. Ma, "Stiffness design of a resonance-based planar snake robot with parallel elastic actuators," *IEEE Robotics and Automation Letters*, vol. 3, pp. 1284–1291, 2018.