2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Phase Synthesis for Spatial Locomotion Control of Retractable Worm Robots

Zhongcheng Wang[1], Shiwei Yuan[1], Manfeng Dou[1], Jianhua Yang[1], and Bin Liang[2]

***Abstract*— Retractable worm robots possess hyper-flexibility, allowing them to work in confined spaces that are difficult for humans. However, the spatial locomotion control of these robots remains challenging due to the robots' large degrees of freedom. To address this challenge, we propose a phase synthesis (PS) scheme for retractable worm robots. The scheme combines an undulating gait inspired by caterpillars with three-dimensional movement commands. We first introduce the kinematics model and real-world prototype of our retractable worm robot, called RW-Robot, and then we introduce footstep phases to express the timing of segments' spatial movement. According to the length of movement periods, we classify the movement into short-term movements and long-term movements and compress their patterns in the frequency domain. Our PS scheme aligns the patterns according to the footstep phases to generate new gaits of spatial locomotion. We evaluate the scheme in real-world experiments, including steering and climbing a slope. The experimental results indicate that our scheme allows the RW-Robot to perform flexible spatial locomotion from simple user input.**

I. Introduction

Natural biological organisms have inspired researchers to design and control robots with hyper degrees of freedom (DoFs). These robots, inspired by snakes or worm-like organisms and benefiting from their exceptional flexibility, can work in confined spaces, e.g., underground tubes [1], earthquake ruins [2], space stations [3], [4], and nuclear power plants [5], [6], which can be rather difficult for humans. With in-depth research into the mechanics and locomotion principles of these creatures [7], [8], plenty of novel robots have been developed in recent time [9], [10]. Compared to snake robots, which lack the retract ability, worm robots adopt a distinguished principle that moves by retracting and elongating their body alternately, increasing the movement efficiency in confined spaces [11]. Robots with both translational and rotational DoFs are developed to combine the reachability of snake robots and the movement efficiency of worm robots [12], [13]. On the other hand, the hyper DoFs of these robots have been raising an extreme challenge for locomotion control. Distinct from the control of snake robots, the retractable worm robots move based on an undulating gait known as the *reverse peristalsis wave*. The undulating gait synthesized with the steering and climbing movement requires extra spatial decoupling of the kinematics to generate joint trajectories for stable locomotion efficiently.

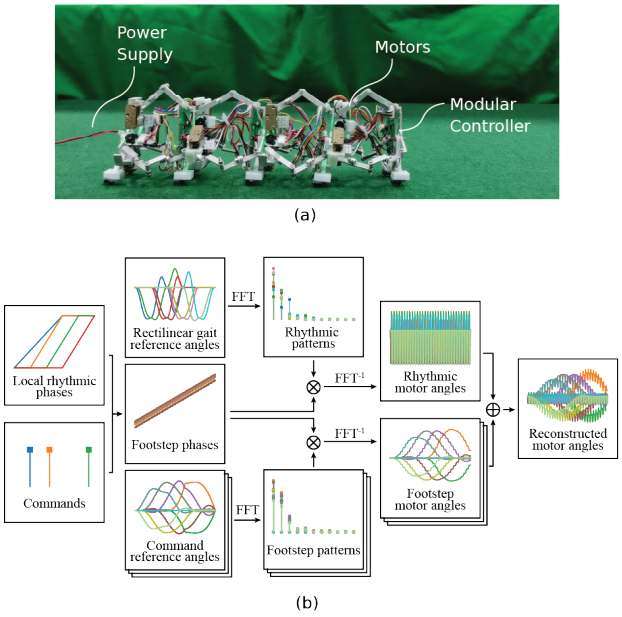

Fig. 1. Overview of our scheme. (a) The prototype of the RW-Robot with 4 modular segments. (b) The architecture of the gait generator with phase synthesis scheme. With the given manipulation commands, the gait generator synthesizes the different gait patterns by phases and outputs the motor angles in the next several steps.

Since Chirikjian and Burdick proposed the backbone curve model for hyper-redundant robots [14], there have been various schemes for the locomotion control of retractable worm robots based on the backbone curves. The configurations of the retractable robots and their deformation during locomotion are described as model functions. The finite state machine is a commonly utilized method to generate planar or spatial locomotion schedules [15], [16]. By manually assigning the patterns of the gaits, controllers provide diverse locomotion skills rapidly. However, the analysis and design of the gait procedure require considerable effort. Central Pattern Generators (CPGs) imitate the motion control mechanism in natural creatures [17], [18]. The generators produce various rhythmic movements for legged robots by modulating the coupled oscillators. Yet, the complex and coupled kinematics of retractable worm robots remain challenging for these methods. We refer to the concepts of phase and patterns while deploying them on a simplified control architecture. Another approach to deal with the kinematics

*Corresponding authors: Jianhua Yang and Bin Liang.
[1]Zhongcheng Wang, Shiwei Yuan, Manfeng Dou, and Jianhua Yang are with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China. {wangzhongcheng, yuanshiwei}@mail.nwpu.edu.cn, {yangjianhua, doumf}@nwpu.edu.cn
[2]Bin Liang is with the Department of Automation, Tsinghua University, Beijing 100084, China, and also with the School of Automation, Northwestern Polytechnical University, Xi'an 710129, China. bliang@tsinghua.edu.cn

is to use optimizers [19], [20]. The optimizers allow the generation of rich gaits, promising obstacle avoidance and stable locomotion, but consume a lot of time to calculate the optimal solutions once the gait parameters change. We design optimizers to generate reference gaits. By extracting the patterns, we eliminate the need for optimizers in gait synthesis and reconstruction.

In this work, we propose a Phase Synthesis (PS) scheme for spatial locomotion control of retractable worm robots. The scheme generates an instant solution combining an undulating gait inspired by caterpillars in nature and three-dimensional movement commands. We first introduce the kinematics model and real-world prototype of our retractable worm robot, called RW-Robot. We then express the timing of segments' spatial movement using several footstep phases. We compress the basic rectilinear gait patterns along with spatial movement patterns in the frequency domain and align them according to the footstep phases to generate new gaits of spatial locomotion. Finally, we evaluate the scheme in real-world experiments containing steering and climbing a slope.

Our contributions include: (i) We formulate the rectilinear and spatial moving models based on a backbone-line-based RW-Robot and express the patterns in the frequency domain to compress their dimensions. The patterns are fundamental to spatial locomotion and also have reference values for similar robots. (ii) We introduce local and footstep phases to indicate the segments' short-term and long-term movement timing, respectively. The footstep phases align the patterns to reproduce rich spatial gaits without the need for the solution of inverse kinematics. (iii) We show experimental validation on the prototype of RW-Robot.

II. Robot Design

A. Backbone-line-based Model of RW-Robot

We use a backbone line to describe the kinematics of the retractable worm robots (RW-Robot). The soft and flexible trunk of caterpillars, which can be imitated by a chain of rotational and translational joints, inspires our design of RW-Robot, as shown in Fig. 2 (a). The Backbone-Line-Based Model (BLBM) of our 4-segment prototype is shown in Fig. 2 (b) and (c). The BLBM consists of a series of key points and straight lines connecting the adjacent key points. The solid points on the BLBM represent the fixed joints without DoFs. The hollow points are virtual rotational joints on the robot, each of which has two DoFs, i.e., pitch and yaw. The lines on both sides of the hollow points represent a pair of virtual translational joints. They are additionally assumed to have the same length. As a result, each paired translational joint shares a single DoF. The rotational and translational joints are virtual because we implement our RW-Robot using Canfield Mechanisms (CMs) [21], a 3-DoF parallel robot. The Canfield mechanisms have the equivalent kinematics of the virtual joints, though not based on the real mechanical structure.

The coordinate systems of the joints of our RW-Robot with key points s_i, where $i = 0, 1, 2, \ldots, 16$ from the tail segment

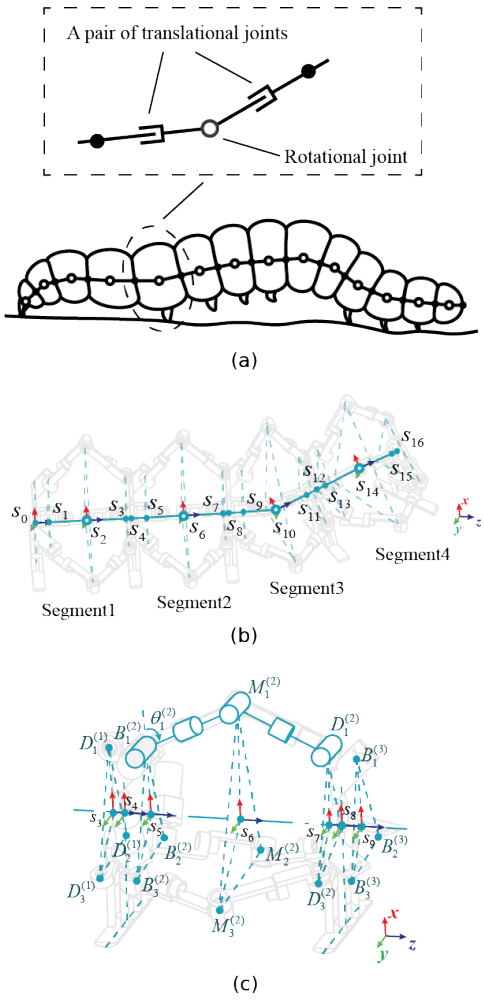

Fig. 2. The BLBM of our 4-segment RW-Robot. (a) The RW-Robot over the sketch of a caterpillar. The RW-Robot uses a backbone line to express the deformation of a caterpillar's trunk. (b) The BLBM over the mechanical design of our RW-Robot. (c) Mechanical details of segment 2.

to the head segment, are shown in Fig. 2 (b). The position of s_i is denoted as $\boldsymbol{q}(s_i, t)$, which is a function of both s_i and time t. The orientations of the key points are represented by coordinate systems $\{s_i\}$ whose y-axis is along the backbone line. The position of the key point s_i is expressed as follows:

$$\boldsymbol{q}(s_i, t) = \sum_{j=0}^{i} l(s_j, t)\boldsymbol{u}(s_j, t) \tag{1}$$

where, $l(s_i, t)$ denotes the length between point s_{i-1} (or point O when $i = 0$) and s_i, and $\boldsymbol{u}(s_i, t)$ denotes the unit direction vector connecting the point s_{i-1} (or O when $i = 0$) to s_i.

To express the configuration of the backbone line, we define three sets of parameters, i.e., plunge distances $\boldsymbol{p}_d := \left\{p_d^{(i)} \mid i = 1, 2, 3, 4\right\}$, pitch angles $\boldsymbol{\tau} := \left\{\tau^{(i)} \mid i = 0, 1, 2, \ldots, 5\right\}$, and yaw angles $\boldsymbol{\psi} := \left\{\psi^{(i)} \mid i = 0, 1, 2, \ldots, 5\right\}$. For $i = 1, 2, 3, 4$, the parameters describe the configuration of the segment i, and $\tau^{(0)}(t)$, $\psi^{(0)}(t)$ describe the direction of s_0 to s_1. Besides,

we set $\boldsymbol{q}(s_0,t)$ directly for a complete representation of all key points' position trajectories. The line length and direction of the backbone line are thus expressed as follows:

$$l(s_{4i-2},t) = l(s_{4i-1},t) = p_d^{(i)}(t) \tag{2}$$

$$\boldsymbol{u}(s_{2i+k},t) = \begin{bmatrix} \sin\sum_{j=0}^{i}\psi^{(j)} \cdot \cos\sum_{j=0}^{i}\tau^{(j)} \\ -\sin\sum_{j=0}^{i}\tau^{(j)} \\ \cos\sum_{j=0}^{i}\psi^{(j)} \cdot \cos\sum_{j=0}^{i}\tau^{(j)} \end{bmatrix} \tag{3}$$

where, $i = 1,2,3,4$, and $k = -1,0,1,2$ if $2i+k \leq 16$. We would like to remark that the equations in (2) hold due to the assumption that the translational joints in a segment have the same length. Additionally, $\boldsymbol{u}(s_1,t)$ and $\boldsymbol{u}(s_2,t)$ are calculated assuming $i = 0$ and $k = 1,2$ according to (3). Due to $l(s_0,t)$ and $\boldsymbol{u}(s_0,t)$ are the modulus and unit vector of $\boldsymbol{q}(s_0,t)$, respectively, both are assigned directly by $\boldsymbol{q}(s_0,t)$. Except for $l(s_0,t)$ and the lengths defined in (2), the remaining lengths are specified by the mechanical setup of our prototype, which is detailed in the next subsection.

B. Mechanical Implementation and Control System

The prototype is shown in Fig. 1 (a). The joints on both plates are on circles with a radius of R. The linkages on a single subchain have the same length L. The robot is actuated by 12 motors, whose angles are defined in Fig. 2 (c). Here, we denote the action as $\boldsymbol{a} = [\theta_1^{(1)},\theta_2^{(1)},\theta_3^{(1)},\theta_1^{(2)},\ldots,\theta_2^{(4)},\theta_3^{(4)}]^\top \in \mathcal{A}$, where, $\theta_1^{(i)}$ represents the motor angle at the top subchain of segment i and $\theta_2^{(i)}$, $\theta_3^{(i)}$ at side subchains, for $i = 1,2,3,4$. The forward of the single-segment mechanism, transforming $\boldsymbol{a}$ to $\{p_d^{(i)},\tau^{(i)},\psi^{(i)}\}$, and inverse kinematics (IK), transforming $\{p_d^{(i)},\tau^{(i)},\psi^{(i)}\}$ back to $\boldsymbol{a}$, have been derived in [12], [13], [22]. To prevent the singular configuration, we limit the actuating joint angles in range $\Theta = [0,\frac{\pi}{2}]$, and thus the action space is $\mathcal{A} = \Theta^{12}$.

Due to the essence of assembling multiple segments, we specify the bias between adjacent segments to align the mechanical design of the prototype and the BLBM of RW-Robot. We illustrate the biases in Fig. 2 (b) and specify them in Tab. I. Besides, we provide some mechanical properties of the prototype in Tab. I.

TABLE I
MECHANICAL CONFIGURATIONS OF OUR PROTOTYPE

R	L	$l(4_i,4_{i+1})$	$l(4_{i-1},4_i)$	Leg length	Mass
25.1mm	33.0mm	11.0mm	7.5mm	45.0mm	340g

The prototype has a cascaded control system. The local controllers on the segments run on embedded microcontrollers. Each of the controllers includes a communication modular and three PD-control modulars. The desired angles sent from the high-level computer are received by the first-segment controller and passed through the remaining controllers.

III. PHASE SYNTHESIS

Crawling gait is a basic gait of caterpillars. The gait possesses a rhythmic pattern of motor trajectories, causing the legs to move alternately to the new footstep position. While the legs periodically move under the dominance of the crawling patterns, their movements in a long-term aspect are supposed to be compliant with the desired spatial directions. Reference trajectories of the motors must be adjusted to fit both gait rhythms and spatial footsteps. Hence, as shown in Fig. 1 (b), we use a rhythmic phase to describe the timing of the gait period for controlling short-term motion and multiple footstep phases to establish the timing of the footstep schedule for controlling long-term behaviors. By representing the patterns of gait and footsteps in the frequency domain, we then design a gait generator to activate the patterns according to these phases.

A. Spatial Gait Generator with Phase Synthesis

Segments have their own rhythms, meaning each segment of the robot has a local rhythmic phase varying asynchronously with the timing of the gait. We use an angle to express a local rhythmic phase. Local filters bridge time and the local rhythmic phases, reconstructing the unique variation of rhythmic phases for the segments. Local rhythmic phases offer two significant advantages: (1) they not only reveal the stage of the gait but also indicate the process of the planned locomotion in a long-term scope, and (2) they isolate the gait patterns from time, allowing the movement to pause and resume at any time. For the first advantage, to further explain, the rhythmic pattern with a period of 2π maintains consistent performance at a particular gait phase, but the resulting motor angles are also altered by a non-periodic feature that aligns with the desired trajectory. The ratio between the local gait phase and 2π indicates the number of footsteps taken by the segment along the intended path. Thus, we define the local rhythmic phases of segment i, i.e., $\phi_r^{(i)}$, as a unit complex number as follows:

$$\phi_r^{(i)}(t) = \int_0^t \delta^{(i)}(t)d\varsigma \tag{4}$$

where $\delta^{(i)}(t)$ represents the varying speed of phase related to the current stage in a gait period, and $i = 1,2,3,4$. For $k \in \mathbb{N}$, $0 \leq t - kT < T$, if $\varsigma^{(i)} \leq t - kT < \varsigma^{(i)} + T^{(i)}$, then $\delta^{(i)}(t) = \dfrac{2\pi}{T^{(i)}}$. Otherwise, $\delta^{(i)}(t) = 0$. The lag $\varsigma^{(i)}$ determines when to activate the motion of the segment, and the durations $T^{(i)}$ and T correspond to motion durations of the segment and the entire body, respectively.

Despite the existence of unknown perturbances and tracking errors, one can manipulate the robot at any time to fix the path biases or perform new actions. To facilitate the expression of the desired path, footstep phases are thus proposed. The footstep phases vary with the local rhythmic phases in spite of a phase difference. Multiple deflection and pitch patterns can add influence to the initial path, each of which possesses a private footstep phase. We use $\phi_{r,0}^{(i)}$ to denote the footstep phase of desired path,

and $\phi_{r,1}^{(i)}, \phi_{r,2}^{(i)}, \phi_{r,3}^{(i)}, \ldots, \phi_{r,k}^{(i)}, \ldots, \phi_{r,M}^{(i)}$ to denote the footstep phases triggered by the manipulation commands at t_k. Thus:

$$\phi_{r,k}^{(i)} = \phi_r^{(i)} - \phi_r^{(i)}(t_k) \tag{5}$$

The gait generator synthesizes the different gait patterns with phases and outputs the reconstructed motor angles in the next several steps. The reference angles are then regarded as the setpoints of the PD controller of the motors. We do not describe the patterns directly in the phase domain. Instead, we utilize the Fast Fourier Transform (FFT) on the model angles to extract the patterns. We will explain how to obtain the reference model angles later. The gait generator reconstructs these patterns by footstep phases back into angles as follows:

$$\hat{\boldsymbol{\theta}}^{(i)}\left(t;\{t_k\}\right) = \theta_e + \sum_{k=0}^{M} \boldsymbol{\theta}_f^{(i)}\left(t;\{t_k\}\right) + \boldsymbol{\theta}_r^{(i)}\left(t;\{t_k\}\right) \tag{6}$$

where θ_e is denoted as the initial motor angle, the summation of $\boldsymbol{\theta}_f^{(i)}$ is denoted as the footstep motor angles indicating the long-term components of the gait, and $\boldsymbol{\theta}_r^{(i)}$ suggests the rhythmic components of the gait. We use $\boldsymbol{\Gamma}^{(i)}$ to denote the value at the corresponding rhythmic or footstep patterns $\boldsymbol{\omega}$. Hence, the motor angles at time t, with manipulation commands at $\{t_k\} := t_0, t_1, t_2, \ldots, t_k, \ldots, t_M$, can be reconstructed via inverse FFT as follows:

$$\boldsymbol{\theta}^{(i)}\left(t;\{t_k\}\right) = \mathrm{Re}\left\{\mathrm{FFT}^{-1}\left[\boldsymbol{\Gamma}^{(\mathrm{i})}\mathrm{e}^{-\mathrm{j}\boldsymbol{\omega}\phi_{\mathrm{r,k}}^{(\mathrm{i})}(\mathrm{t})}\right]_0\right\} \tag{7}$$

where $\boldsymbol{\theta}^{(i)}$ represents the footstep motor angles or the rhythmic motor angles and $\boldsymbol{\Gamma}^{(i)}$ represents the corresponding patterns. $\mathrm{FFT}^{-1}[\cdot]_0$ is denoted as the first element of the inverse FFT, and $\mathrm{Re}\{\cdot\}$ is denoted as the real part of the complex number.

B. Patterns of Rectilinear Gait

Caterpillars lift their legs alternately from the ground while moving the lifting leg to a new footstep by bending and contracting their trunk. We divide the movement of each leg into two stages, i.e., the stance stage and the swing stage. In the stance stage, the leg fixes on the ground, providing friction for the moving trunk. In the swing stage, the leg lifts from the ground, moving in the desired direction, and finally drops down to the new footstep. Exceptionally, the leg in the stance stage can lift and then drop down in place to aid the swinging leg. This is caused by the kinematics limitation of the mechanics of our robot.

The legs' trajectories in the entire gait period are the combination of trajectories in both the swing and stance stages. While the trajectories in the swing stage are the integration of the velocity, the trajectories in the stance stage are rather difficult to calculate because of the coupling among legs. We use a gradient-based optimizer with constraints to estimate the reference trajectories. Once we specify the legs' trajectories, we can calculate the trajectories of motor angles according to the inverse kinematics of the Canfield mechanism in [12], [13], [22]. We introduce the detailed generation algorithm in Alg. 1.

To describe the movement of legs in a gait period, we set the coordinate system of the end leg s_0 at the beginning of the gait period as the reference system. We define the velocity of the leg in its swing trajectory as follows:

$$\dot{\boldsymbol{q}}_m(t) = \begin{cases} \begin{bmatrix} \frac{\pi H}{T_s}\sin\frac{2\pi}{T_s}t \\ 0 \\ \frac{\pi L_s}{T_s}\left(1-\cos\frac{2\pi}{T_s}t\right) \end{bmatrix} & 0 \le t \le T_s \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{8}$$

where T_s is the period of the swing stage, we set the swing stages of legs to be equivalent, i.e., $T_s = 0.4\pi$. H represents a parameter adjusting the lifting range of the stepping leg from the ground, and L_s represents the stride length within a gait period. It is worth noting that any reasonable function besides (8) can be used as the velocity function. We set the initial angle θ_e as 60 degrees, the stride length L_s, and the lift range H as $0.5p_d(0)$ (approximately 14.3mm) and 2.5mm, respectively. By estimating the optimized angles and analyzing their spectrum, we discard the circular frequency components larger than 10. The remaining components, as rhythmic patterns, preserve enough information to reconstruct the rectilinear gait.

Algorithm 1 Patterns Generation of Rectilinear Gait.

input: initial motor angle θ_e, stride length L_s, lift range H_s, tick length N
$T \leftarrow 2\pi$, $T_s \leftarrow 0.4\pi$, $\boldsymbol{q}(s_0, 0) \leftarrow [0,0,0]^\top$
$\psi^{(i)} \leftarrow 0$, $\tau^{(i)} \leftarrow 0$ for $i = 0, 1, 2, 3, 4$
compute $p_d^{(i)}(0;\theta_e)$ for $i = 1, 2, 3, 4$ according to the forward kinematics of CM
compute $\boldsymbol{q}\left(s_i, 0\right)$ for $i = 1, 2, 3, \ldots, 16$
for $i = 1, 2, 3, \ldots, N$ **do**
  $t_i \leftarrow \frac{2\pi}{N} i$
  for $n = 0, 4, 8, 12, 16$ **do**
    $\tilde{\boldsymbol{q}}\left(s_n, t_i\right) \leftarrow \boldsymbol{q}\left(s_n, 0\right) + \frac{2\pi}{N}\sum_{k=0}^{i}\dot{\boldsymbol{q}}_m\left(t_k - 2\pi n T_s/4\right)$
  end for
  $\hat{\boldsymbol{p}}_d, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\psi}} \leftarrow \min_{\boldsymbol{p}_d, \boldsymbol{\tau}, \boldsymbol{\psi}} \sum_{n\in\{4,8,12,16\}} \left\|\hat{\boldsymbol{q}}(s_n, t_i) - \tilde{\boldsymbol{q}}(s_n, t_i)\right\|^2$
  compute $\boldsymbol{a}(t_i)$ according to the IK of CM
end for
1-d FFT: $\boldsymbol{A}(\omega) \leftarrow \mathrm{FFT}[\boldsymbol{a}(\mathrm{t})]$
output patterns $\boldsymbol{\Gamma}_r(\hat{\omega}) \leftarrow \boldsymbol{A}(\hat{\omega})$ for $\hat{\omega} \le 10$

C. Patterns of Spatial Moving

With a fixed stride length, we discretize the desired spatial path into footsteps. The number of forecasting footsteps corresponds to the number of periods of rhythmic gait to travel straight. During the locomotion, additional manipulation commands reshape the desired spatial path in the next several steps. The additional paths possess their own patterns and are aligned with the footstep phases. We use the path length parameter s to express the related position of s_0 on

the additional path. At $s = 0$, the robot's tail is positioned at the start of the path. Once $s = 1$, the robot has completed the journey along the path. However, the desired path from the current position of the robot's tail to its head is omitted. Therefore, we formulate s of footstep phase ϕ as follows:

$$s = \frac{L_s}{2\pi L_P}\phi - \frac{L_r}{L_P} \tag{9}$$

where L_P is denoted as the length of the path, and L_r is denoted as the initial length of the robot.

We modulate the path by a pair of orthogonal footstep models, representing the degrees of deflection and pitch. We introduce a deflection path rotating the robot's orientation around the x axis, which can be expressed by the position as follows:

$$\boldsymbol{q}_d(s) = \begin{cases} [0, 0, \rho\Psi s]^\top, & s \le 0 \\ [0, -\rho(1-\cos\Psi s), \rho\sin\Psi s]^\top, & 0 < s \le 1 \\ \begin{bmatrix} 0 \\ \rho(\cos\Psi - 1) - \rho\Psi(s-1)\sin\Psi \\ \rho\sin\Psi + \rho\Psi(s-1)\cos\Psi \end{bmatrix}, & s > 1 \end{cases} \tag{10}$$

where Ψ is the yaw angle after the deflection, ρ is the turning radius. In this model, $L_P = \rho\Psi$. We note that when turning in the opposite direction, the yaw angle Ψ and turning radius ρ do not change. Instead, the opposition deflection path is formulated by multiplying -1 by the y-axis element of (10). Similarly, a pitch model can be expressed by switching the x-axis and y-axis of the deflection model in (10), i.e., $\boldsymbol{q}_p(s) = [q_{d,y}(s), q_{d,x}(s), q_{d,z}(s)]^\top$. It is worth noting that the deflection and pitch model can be substituted for any other proper functions. When we specify a model path, the orientation at point s_0, in terms of $\boldsymbol{R}_P(s)$, is also specified in the meantime. Thus, we can estimate the optimized motor angles at the current footstep phase $\phi_{r,k}^{(i)}$ as the footstep motor angles $\boldsymbol{\theta}_f^{(i)}$ of the corresponding manipulation command. In detail, we illustrate the process of generating the spatial-moving patterns in Alg. 2.

Algorithm 2 Patterns Generation of Spatial Moving

input: stride length L_s, tick length N, model path $\boldsymbol{q}_P(s)$, initial positions $\boldsymbol{q}(s_i, 0)$ for $i = 0, 1, 2, \ldots, 16$
compute $\boldsymbol{p}_d$ at the initial position
$L_r \leftarrow \sum_{i=1}^{16} \|\boldsymbol{q}(s_i, 0) - \boldsymbol{q}(s_{i-1}, 0)\|$
$L_P \leftarrow \int_0^1 \boldsymbol{q}_P(s)ds$
$N_{step} \leftarrow \dfrac{L_r + L_P}{L_s}$
for $i = 1, 2, 3, \ldots, N$ **do**
  $s \leftarrow \dfrac{L_P + L_r}{NL_P} i - \dfrac{L_r}{L_P}$
  compute $\boldsymbol{R}_P(s)$, $\boldsymbol{q}_P(s)$
  for $j = 1, 2, 3, \ldots, 16$ **do**
    $s_j \leftarrow s + \sum_{k=1}^{j} \|\boldsymbol{q}(s_k, 0) - \boldsymbol{q}(s_{k-1}, 0)\|$
    compute $\boldsymbol{q}_P(s_j)$
    $\tilde{\boldsymbol{q}}(s_j) \leftarrow \boldsymbol{R}_P^\top(s) \cdot [\boldsymbol{q}_P(s_j) - \boldsymbol{q}_P(s)]$
  end for
  $\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\psi}} \leftarrow \min_{\boldsymbol{\tau}, \boldsymbol{\psi}} \sum_{k=1}^{16} \|\hat{\boldsymbol{q}}(s_k) - \tilde{\boldsymbol{q}}(s_k)\|^2$
  compute $\boldsymbol{a}(s; \boldsymbol{p}_d, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\psi}})$ according to the IK of CM
end for
1-d FFT: $\boldsymbol{A}(\omega) \leftarrow \mathrm{FFT}[\boldsymbol{a}(\phi)]$
output patterns $\boldsymbol{\Gamma}_f(\hat{\omega}) \leftarrow \boldsymbol{A}(\hat{\omega})$ for $\hat{\omega} \le \dfrac{10}{N_{step}}$

IV. Experiment and Results

A. Rectilinear Gait and Spatial Moving Patterns

Before deploying the phase synthesis, we generate the rhythmic and footstep patterns for the robot's rectilinear and spatial moving. We set the initial motor angles as 60 degrees. For rectilinear gait, we remark that the stride length is 14.3mm, and the lift range is 2.5mm. For spatial moving, we set the yaw angle to 30 degrees and the turning radius to 330mm. The footstep number of spatial moving N_{step} is 32.

The generated patterns are shown in Fig 3. The rhythmic patterns of each motor possess ten circular frequency components ranging from 0 rad/s to 10 rad/s. The footstep patterns vary from 0 rad/s to 0.32 rad/s, which is significantly smaller than the components of rhythmic patterns. The smaller pattern components indicate that spatial moving is a long-term locomotion compared to rhythmic locomotion. We note that the values of pattern components are complex numbers whose phase is not demonstrated in Fig. 3. The mean reconstruction error of the rhythmic motor angles according to the rhythmic patterns is 0.007 degrees, while the standard error is 0.018 degrees. The reconstruction error is deemed acceptable as it falls within the controller's level of precision. The mean construction error of four-type footstep motor angles is 4.452 degrees, and the stand error is 5.125 degrees, which is acceptable for long-term moving.

B. Steering on A Plane

We assess the robot's movement performance while being controlled by a human manipulator on a plane. The human manipulator applies the turn-left or turn-right commands to control the prototype crawling through two obstacles, as shown in Fig. 4 (a). The reconstructed motor angles are sent to the prototype from the high-level computer at intervals of 75ms. We set the actual gait period to 750ms. The locomotion contains 55 footsteps, taking 80.5s. We note that due to the hesitation of the human manipulator, the locomotion takes longer than expected. Within the first seven gait cycles, turn-left commands are sent repeatedly. From the 16th to 20th gait cycles, five turn-right commands are sent. As a result, the prototype follows an S-shaped path through the obstacles. The continuous-sent turning commands conduct a larger steering angle with a smaller turning radius, as demonstrated in Fig. 4 (b). Our experiment shows that the turning radius is limited to at least 80mm due to mechanical constraints.

C. Climbing A Slope

We evaluate the movement performance of spatial tasks. The prototype is manipulated to climb a 45-degree slope, as

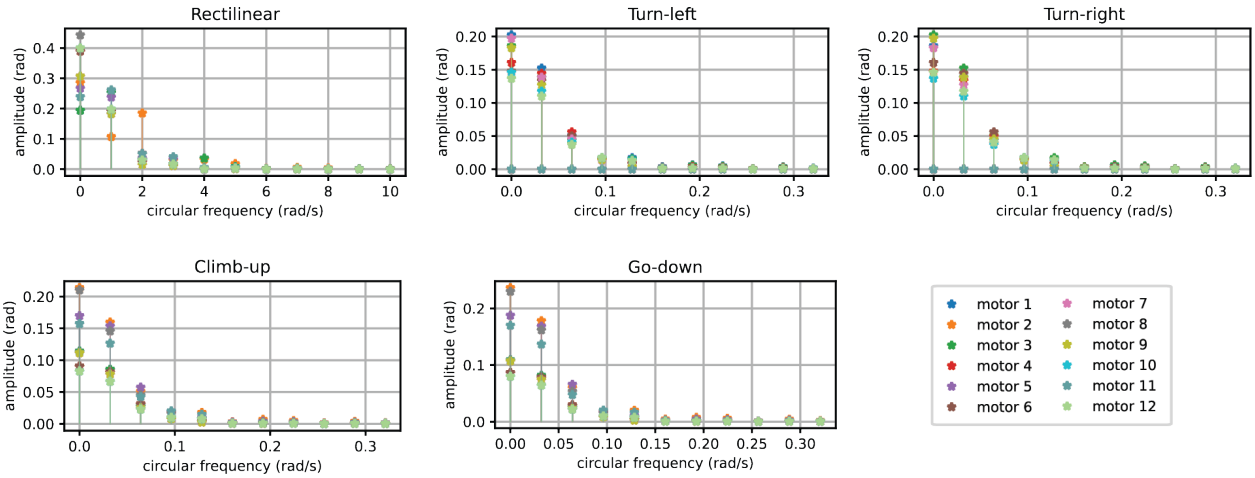

Fig. 3. The amplitudes of rhythmic patterns and footstep patterns. The rhythmic patterns are displayed in the rectilinear figure. The footstep patterns of four manipulation command types, i.e., *turn-left*, *turn-right*, *climb-up*, and *go-down*, are displayed in the remaining figures, respectively.

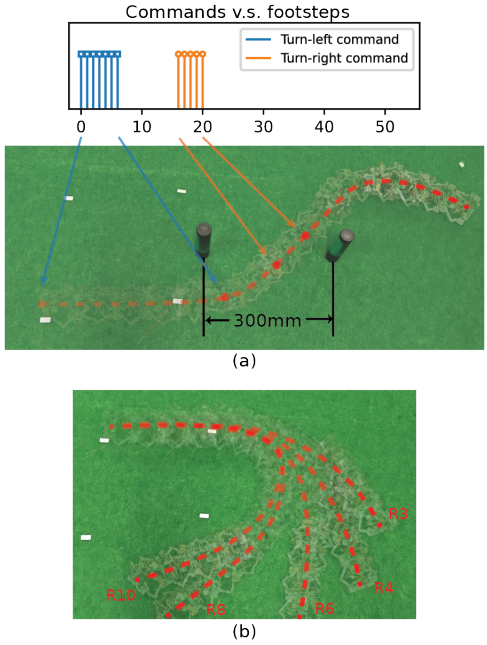

Fig. 4. Plane steering of the prototype. (a) The command list of the locomotion from the first footstep to the 55th footstep and the entire steering trajectory through two obstacles. The arrows point out the positions when the commands are to be applied. (b) The steering trajectories when turn-right commands are repeatedly applied 3 (R3), 4 (R4), 6 (R6), 8 (R8), and 10 (R10) times, respectively.

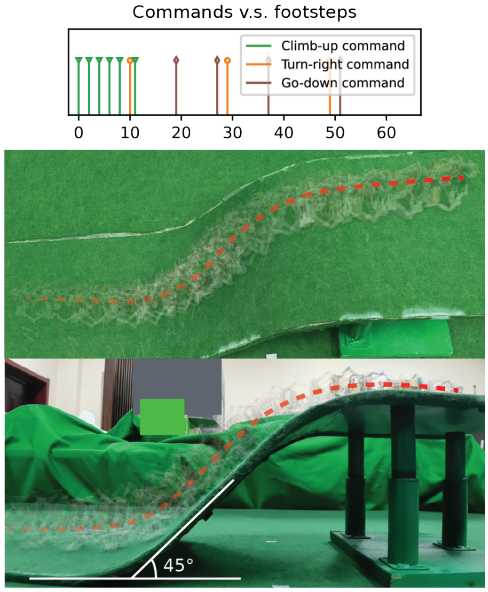

Fig. 5. The command list and locomotion trajectory when climbing a slope. The maximum angle of the slope is up to 45 degrees.

displayed in Fig. 5. The locomotion takes 66 footsteps in 149.4s. Although the path of the locomotion is biased to the left on the slope due to the unexpected angle-tracking error, the manipulator succeeds in fixing the direction by applying turn-right commands. The trajectory shows that our control scheme synthesizes the steering and pitch motor skills and achieves spatial locomotion.

V. Conclusion

In this work, we propose a Phase Synthesis scheme for spatial locomotion control of retractable worm robots. The scheme combines the periodic short-term locomotion skill, i.e., rectilinear gait, and multiple long-term spatial movements by extracting their patterns in the frequency domain and aligning the locomotion by their phases. Experimental results show the RW-Robot prototype crawling through obstacles and climbing on a 45-degree slope with on-slope steering, indicating that the PS scheme allows the RW-Robot to perform spatial movement under human control input. By applying repeated commands, the robot yields larger scales of movement beyond what the model parameters assigned. The adaptation allows the robot to handle potential challenges on complex terrains. In the future, we plan to use imitation learning techniques to generate patterns of rich motor skills with adjustable amplitudes that can address complex terrains at high speeds.

REFERENCES

[1] P. Vartholomeos, P. Marantos, G. Karras, E. Menendez, M. Rodriguez, S. Martinez, and C. Balaguer, "Modeling, gait sequence design, and control architecture of BADGER underground robot," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1160–1167, 2021.

[2] J. Whitman, N. Zevallos, M. Travers, and H. Choset, "Snake Robot Urban Search After the 2017 Mexico City Earthquake," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 8 2018, pp. 1–6.

[3] X. Guo, W. Zhu, and Y. Fang, "Guided Motion Planning for Snake-like Robots Based on Geometry Mechanics and HJB Equation," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7120–7130, 2019.

[4] T. Liu, T. Yang, W. Xu, G. Mylonas, and B. Liang, "Efficient Inverse Kinematics and Planning of a Hybrid Active and Passive Cable-Driven Segmented Manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2021.

[5] Z. Hu, H. Yuan, W. Xu, T. Yang, and B. Liang, "Equivalent kinematics and pose-configuration planning of segmented hyper-redundant space manipulators," *Acta Astronautica*, vol. 185, no. February, pp. 102–116, 2021.

[6] T. Yang, J. Huang, W. Xu, K. Shao, and B. Liang, "Development of a cable-driven redundant space manipulator with large bending angle by combining quaternion joints and segmented coupled linkages mechanism," *Chinese Journal of Aeronautics*, 3 2023.

[7] L. I. van Griethuijsen and B. A. Trimmer, "Locomotion in caterpillars," *Biological Reviews*, vol. 89, no. 3, pp. 656–670, 2014.

[8] S. J. Newman and B. C. Jayne, "Crawling without wiggling: Muscular mechanisms and kinematics of rectilinear locomotion in boa constrictors," *Journal of Experimental Biology*, vol. 221, no. 4, 2018.

[9] W. Zhao, J. Wang, and Y. Fei, "A Multigait Continuous Flexible Snake Robot for Locomotion in Complex Terrain," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3751–3761, 10 2022.

[10] D. Ramesh, Q. Fu, and C. Li, "SenSnake: A snake robot with contact force sensing for studying locomotion in complex 3-D terrain," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2068–2075.

[11] Z. Bi, Q. Zhou, and H. Fang, "A worm-snake-inspired metameric robot for multi-modal locomotion: Design, modeling, and unified gait control," *International Journal of Mechanical Sciences*, vol. 254, no. March, p. 108436, 9 2023.

[12] M. K. Hayat, J. Yang, and A. K. T. Shaharyar, "Locomotion patterns of a vermicular robot," in *2016 2nd International Conference on Robotics and Artificial Intelligence, ICRAI 2016*, 2016, pp. 152–157.

[13] M. K. Hayat, J. Yang, X. Jiang, and Y. He, "Implementation and Kinematics Calculation of a Modified Canfield Mechanism," *Mechanical Science and Technology for Aerospace Engineering*, vol. 38, no. 6, pp. 828–832, 2019.

[14] 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.

[15] R. Liu and Y. a. Yao, "A novel serial–parallel hybrid worm-like robot with multi-mode undulatory locomotion," *Mechanism and Machine Theory*, vol. 137, pp. 404–431, 2019. [Online]. Available: https://doi.org/10.1016/j.mechmachtheory.2019.03.033

[16] X. Zhan, H. Fang, J. Xu, and K. W. Wang, "Planar locomotion of earthworm-like metameric robots," *International Journal of Robotics Research*, vol. 38, no. 14, pp. 1751–1774, 2019.

[17] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[18] G. Bellegarda and A. Ijspeert, "CPG-RL: Learning Central Pattern Generators for Quadruped Locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 547–12 554, 10 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9932888/

[19] J. Chen, G. Li, J. Zhang, and J. Yu, "Caterpillar-Like Climbing Method Incorporating a Dual-Mode Optimal Controller," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1492–1503, 2015.

[20] X. Zhan, J. Xu, and H. Fang, "In-plane gait planning for earthworm-like metameric robots using genetic algorithm," *Bioinspiration and Biomimetics*, vol. 15, no. 5, 2020.

[21] S. Canfield, "Developing Capture Mechanisms and High-Fidelity Dynamic Models for the MXER Tether System," NASA, Tech. Rep. September, 2007.

[22] C. Bueno, K. Collins, A. Hylton, and R. Short, "A Geometric Approach to the Kinematics of the Canfield Joint," 2021. [Online]. Available: http://arxiv.org/abs/2105.05955