# Parameterized Controller Generation for Multiple Mode Behavior

Chaohui Gong, Matthew J. Travers, Hsien-Tang Kao and Howie Choset
{chaohuig, mtravers, hkao, choset}@cmu.edu

***Abstract*— We derive and demonstrate a new capability for snake robots in which two behaviors—one for locomotion and the other for manipulation—are executed simultaneously on the same robot. This is done in two steps: 1) inverse kinematics via numerical optimization and 2) gait-based locomotion via modal decomposition. The result is an analytical representation of a multiple mode behavior that reduces online execution to simple parameterized control. This representation makes it possible to derive a feedback control law that enables reliable visual servoing using a snake robot while climbing a pole.**

## I. Introduction

Snake robots are versatile, but few research effort have been spent on the problem of making them simultaneously perform more than one task. This work demonstrates the ability of a snake robot to simultaneous perform two different tasks on two different parts of the robot. Specifically, we enable a snake robot to actively align a camera located in its head module while climbing a pole.

A number of researchers have previously studied the problem of online visual tracking. Kelly et al. [1] use inverse kinematics to enable a robot arm with a camera mounted in its hand to visually track objects. This inverse kinematics based approach is used to generate instantaneous joint commands for the robot arm to visually track an object. Murray and Basu [2] presented a visual servoing approach for 2 degree of freedom (DoFs) pan-tilt camera.

In this work, we enable a snake robot to simultaneously perform two different tasks. We accomplish this by partition the robot body into two different parts, one for climbing and the other for aligning the robot head module. The climbing motion is achieved by using the *gait equation* proposed in [3]. And the joint trajectories for the head alignment are computed using numerical optimization. The resultant joint trajectories are in the form of lookup tables. The drawback of representing joint trajectories in this form is that 1) a lookup table needs to be generated for each desired head alignment direction, 2) interpolating the lookup table is inevitable in order to generate smooth controls, and 3) lookup tables cannot be reused for a robot with different number of joints.

To address these problems, we apply a FFT-based modal decomposition method to the lookup tables to find a low dimensional analytical representation of the head alignment behavior. Using this approach, we find that lookup tables can analytically be parameterized in terms of a sinusoidal function. In other words, the joint trajectories for head alignment behavior can be equivalently represented as a sinusoidal function with only two parameters (amplitude and phase). Such a parameterization compactly represents all the joint trajectories that otherwise would be calculated using numerical optimization. These parameterized functions enable the derivation of smooth controllers for continuous head alignment.

Fig. 1: Carnegie Mellon University modular snake robot climbing a tree branch. The robot is consist of 16 identical modules and a specialized head module.

Based on the sinusoidal parameterization, we derive a compact, analytical controller for the head alignment behavior. The coefficients of this controller are associated with physically meaningful values, which in turn allows users more readily control the robot. We enable a snake robot to visually servo a moving object while climbing a pole using this controller.

This paper is organized as follows: First, we introduce the background information which is used throughout the paper. Then, a numerical optimization approach for generating discrete joint angles that align the robot's head module direction is presented. Next, we identify analytical representations of the joint trajectories from the numerical optimization. Finally, based on the analytical equation, we implement a feedback control scheme for online visual tracking using a snake robot climbing a pole.

## II. Background

Our research group has developed the *modular snake robot* [4]. It is a chained of $N+1$ identical modules, which includes a head module. The joints of the modular snake robot alternatively rotate in the dorsal and lateral plane of the robot, see Fig. 2b. Such a configuration enables the robot to form a three-dimensional (3D) shape.

Various ways have been proposed to generate controls for a snake robot. Burdick used a spatial curve, called a *backbone curve* [5], to model the shape of a redundant robot.

The motions of a redundant mechanism then can be modeled as the shape changes of the backbone curve [6], [7]. In order to generate joint commands for a discrete mechanism from the designed backbone curves, a fitting algorithm [8] has to be employed. The high computational cost of the existing fitting algorithms impedes applying the backbone curve approach online.

Alternatively, a gait equation [9], [10] can be used to generate controls online. The gait equation expresses joint angles as sinusoidal functions,

$$\alpha(n,\tau) = \begin{cases} \beta_{odd} + A\sin(\theta_{odd}) & \text{odd} \\ \beta_{even} + eA\sin(\theta_{even} + \delta) & \text{even} \end{cases} \quad (1)$$
$$\theta_{odd,even} = (\Omega_{odd,even} n + \omega\tau),$$

where $n$, $\tau$, $\beta$, $A$, $\theta$, $e$ and $\delta$ are respectively joint index, time, angle offset, amplitude, phase, aspect ratio and phase shift. Different behaviors can be achieved by properly selecting the gait parameters. For example, when the phase shift $\delta = \frac{\pi}{2}$ and spatial frequency $\Omega_{odd,even} \neq 0$ a rolling helix motion is produced [3]. This gait can be used for climbing poles and trees, see Fig. 1 and Fig. 2a. One limitation of the gait equation is that it does not allow a snake robot to simultaneously perform more than one task without nontrivial modification.

## III. Joint Optimization

The goal of this work is to derive controllers which enable a snake robot to align its head in a static direction while climbing a pole. To simultaneous accomplish these two tasks, we partition the entire snake into two parts, one part for *head alignment* and another for *climbing*, see Fig. 2b. The climbing motion can be readily achieved by using the gait equation presented in the previous section. For the head alignment part, we used numerical optimization to compute joint trajectories which constantly align the head in the desired direction during climbing.

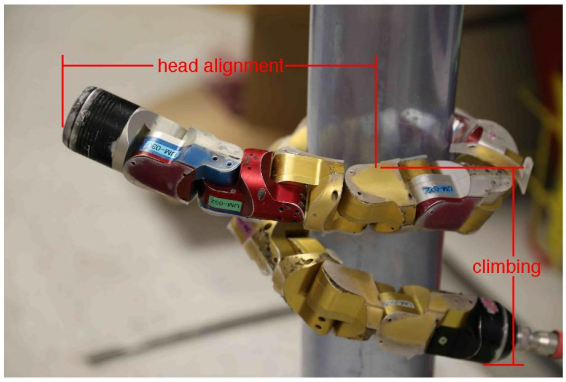
### A. Notation

In this paper, we adopt the notation that the head (has no actuation) is module $N+1$. We will assume that the first $M$ modules of the robot are used to perform pole climbing. And the rest $N-M$ modules are used to perform head alignment. We call the last module of the climbing part (the $M-th$ module) the *base module* of the head alignment. The reason we introduce the base module is that the motion of the climbing part can be decoupled by representing the desired head alignment direction and the kinematics of the head alignment part in the base module frame.

We use $g_i$ to denote the $i\text{-}th$ module reference frame, which originates at the geometric center of the $i\text{-}th$ module. The $\vec{z}$ axis of $g_i$ is aligned with the longitude direction of the robot. Hence, head alignment direction is denoted by $\vec{z}_{N+1}$ (the $\vec{z}$ axis of the head module frame $g_{N+1}$).

In the *world* frame, a desired looking direction at time $\tau$ is denoted by the vector $\vec{z}^w_{d,\tau}$. And the head alignment direction at time $\tau$ is $\vec{z}_{N+1,\tau}$.

(a) In this figure, the robot is commanded to climb a pole by using gait equation.




(b) In this figure, the robot is using the first 4 modules performing head alignment while the rest of the body remaining wrapped around the surface of a pole.

Fig. 2: A snake robot climbs a pole then perform visual surveillance.

### B. Trajectories Generation

The gait equation is used to generate the climbing motion for the climbing part of the snake robot. As time $\tau$ varies, the climbing part of the robot twists around the central axis and generates a net displacement. As being part of the climbing portion of the snake robot, the base module also rotates around its central axis. This amount of rotation is equal the elapsed phase $\omega\tau$ in the gait equation Eqn. 1. Because the head alignment portion is kinematically coupled with the climbing part, the rotation of the base module forces the entire head alignment part to rotate. In order to keep the head aligned in the desired direction, the last $N-M$ modules (head alignment part) have to actuate properly.

Head alignment in a static direction can be achieved by computing a set of joint trajectories which minimize the difference between the head orientation $\vec{z}_{N+1,\tau}$ and the desired head alignment direction $\vec{z}^w_{d,\tau}$ at every time step $\tau$. An optimization based approach can be adopted to generate these trajectories.

We first define the *head looking deviation* at time $\tau$ as,

$$(\vec{z}_{d,\tau} - \vec{z}_{N+1,\tau})^T W^w_e (\vec{z}_{d,\tau} - \vec{z}_{N+1,\tau}), \quad (2)$$

where $W^w_e$ is an weight matrix represented in the *world* frame.

To simplify the kinematic computation, we transform all the vectors contained in Eqn. 2 into the *base* frame,

$$\vec{z}^b_{d,\tau} = R^T_{b,\tau}\vec{z}^w_d \tag{3}$$

Note the base module rotates around its $\vec{z}_b$ axis at an angular velocity $\omega$ during the climbing motion hence the orientation of the base frame $R^T_{b,\tau}$ changes over time. The head looking direction in the *base* frame at time $\tau$ is represented by the $\vec{z}^b_{N+1,\tau}$ axis of the head frame, which is easily computed by forward kinematics. The weight matrix in Eqn. 2, originally expressed in the *world* frame, is transformed into the base frame as

$$W^b_e = R^T_b W^w_e R_b \tag{4}$$

Eqn. 2 expressed in the *base* frame is,

$$C^b_{e,\tau} = (\vec{z}^b_{N+1,\tau} - \vec{z}^b_{d,\tau})^T W^b_e (\vec{z}^b_{N+1,\tau} - \vec{z}^b_{d,\tau}) \tag{5}$$

The goal of the optimization is to generate joint trajectories which minimize Eqn. 5 at every sampled time step $\tau$. Let $\vec{\alpha}_h(\tau)$ denote the set of joint angles dedicated to head alignment at time $\tau$. The joint trajectories for head alignment then can be represented as the collection of $\vec{\alpha}_h(\tau)$ with $\tau = \{1, ..., T\}$.

Given the joint angles of head alignment portion $\vec{\alpha}_h(\tau)$, the head looking direction $\vec{z}^b_{N+1}$ at time step $\tau$ can be computed using forward kinematics,

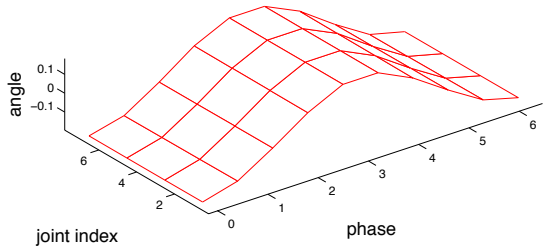
$$\vec{z}^b_{N+1} = \vec{z}^b_{N+1}(\vec{\alpha}_h(\tau)) \tag{6}$$

The total cost in one gait period is defined as,

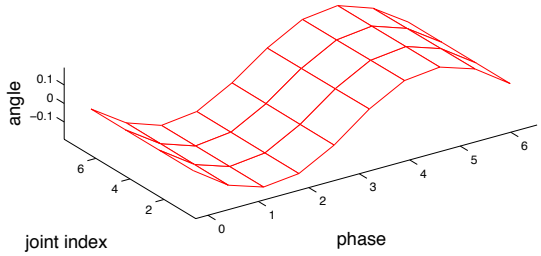
$$C = \sum_{\tau=1}^{T} [C^b_{e,\tau}(\vec{\alpha}_h(\tau)) + \vec{\alpha}^T_h(\tau) W_r \vec{\alpha}_h(\tau)], \tag{7}$$

where $W_r$ is a $(N-M) \times (N-M)$ weight matrix. The second term in the cost function is a regularization term. The regularization term introduces two benefits. First, it keeps the joints away from the joint limits. Second, it distributes the required actuation for head alignment over all the joints and makes the head alignment part of the robot form a smooth shape, reducing the risk of hitting the pole on which the robot is climbing. The two weight matrices $W^b_e$ and $W_r$ trade off between head alignment deviations and regularization. In this work, $W^b_e$ is a $3 \times 3$ diagonal matrix with all the diagonal entries equal to 10 and $W_r$ is an identity matrix.

The joint trajectories that result from this optimization are sorted into two lookup tables, one for the odd joints and another for the even joints. The values stored in the lookup tables contain the joint angle variations across both joint index and time. The lookup tables can be visualized as *wave surfaces* [8] to show the motion patterns of the resultant joint trajectories. For example, Fig. 3a and Fig. 3b show the wave surfaces (joint trajectories) computed using optimization at a particular head alignment direction.




(a) The joint trajectories for the odd joints in a particular head alignment direction.




(b) The joint trajectories for the even joints in a particular head alignment direction.

Fig. 3: The wave surfaces (joint trajectories) for head alignment in a particular head alignment direction.

## IV. Modal Decomposition

There are several limitations associated with the optimization based approach presented in Sec. III. First, a lookup table needs to be computed offline for each head alignment direction. Interpolation is needed to smoothly vary the head alignment online. Another drawback is that, solutions based on one optimization cannot be generalized to robots that have different numbers of modules. To address these issues, we first apply an FFT-based modal decomposition [11] approach to the resultant joint trajectories to find a low dimensional analytical representation. We then use this representation to derive compact, smooth and analytical controllers which enable the snake robot to actively align its head while climbing a pole.

### A. Principal Modes

To find an analytical representation for head alignment control while climbing, we first need to identify a parametric form. An ideal parameterization should capture all the important variations across all the lookup tables while

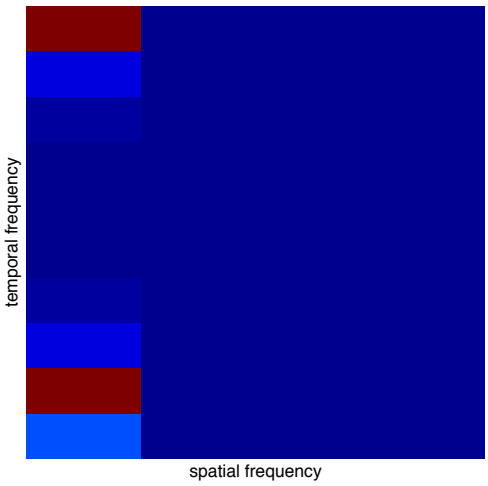



Fig. 4: This figure shows the result of applying 2D FFT to a lookup table resulting from optimization. Block in brown denotes frequency components of high strength. In this figure, there is only one strong component (the second brown block is the complex conjugate).

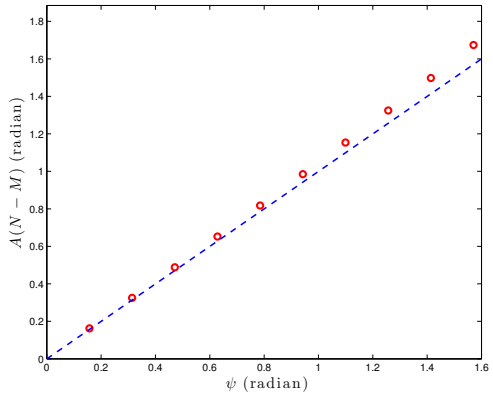
maintaining a simple form. Based on the observation that the head alignment joints move periodically (as shown in Fig. 3a and Fig. 3b), we choose sinusoids as basis functions for the construction of the parametric form. We hence apply a two dimensional fast Fourier Transformation (2D FFT) to identify the dominant frequency components associated in the lookup tables. Fig. 4 shows the result of applying the FFT to one particular lookup table. Blocks at different locations represent Fourier modes at different spatial-temporal frequencies and the block colors represent their strengths. In this figure, only one sinusoidal (two Fourier) component is associated with large amplitude (the two dark regions are the complex conjugate to each other and they together represent one sinusoid), which means the joint trajectories represented by the particular lookup table can be well approximated by a single-spectrum spatial-temporal sinusoidal function. Applying FFT to all the lookup tables leads to similar results. This means that the joint motions for head alignment in any direction can be modeled as a spatial-temporal sinusoidal function and the head orientation is determined by the parameters associated with this sinusoidal function.

### B. Analytical Form

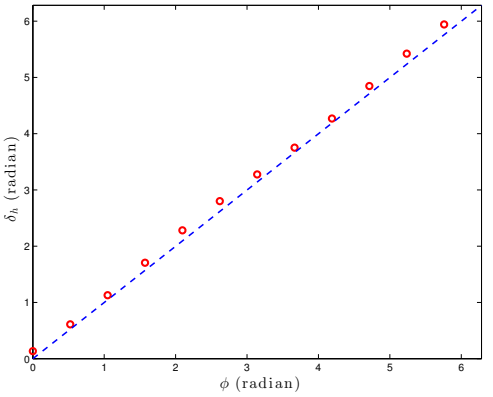
Further analysis on the FFT results shows that for any particular head alignment direction there is a constant phase offset $\frac{\pi}{2}$ between the sinusoidal functions corresponding to the odd and even lookup table. This observation leads to an analytical representation of the joint trajectories for head alignment,

$$\alpha(n,\tau) = \begin{cases} A\sin(\omega\tau) & \text{odd} \\ A\sin(\omega\tau + \frac{\pi}{2}) & \text{even} \end{cases} \tag{8}$$

where $\delta_h$ is a phase offset. This euqation actually corresponds to the parameterization of the head alignment controller which generates joint angle commands to constantly align the head in the desired direction.




(a) The pitch $\psi$ in the *base* coordinates equals to the summation of rolling result from climbing $\omega\tau$ and head alignment $\delta_h$.




(b) The yaw $\phi$ is directly related to the joint angle amplitude $A_h$ of the head alignment part.

Fig. 5: The coefficients in the derived controller are directly related two the pitch and yaw angle in the *base* coordinates.

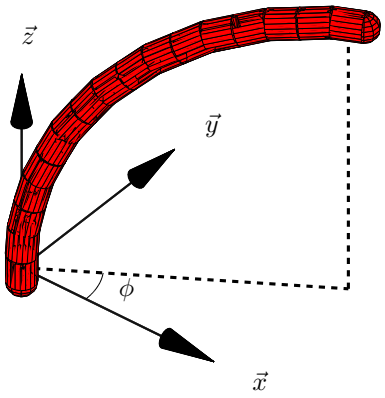
### C. Parameter Identification

Using the FFT based modal decomposition approach, we obtained the analytical form of the controller for head alignment. We further studied the physical meanings of the coefficients contained in Eqn. 8.

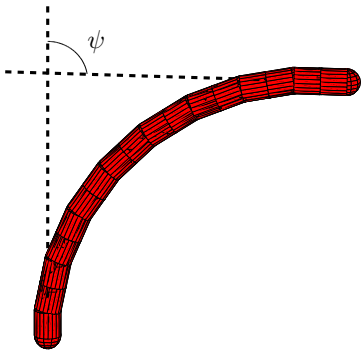
We first introduce a coordinates attached at the center of the base module, see Fig. 6. In this coordinates, a particular head alignment direction is specific by the pitch $\psi$ ($\vec{z}$ direction is 0 pitch) and yaw angle $\phi$ of the head module. These two angles are related to the desired head alignment direction $\vec{z}^{\,b}_{d,\tau}$ as

$$\vec{z}^{\,b}_{d,\tau} = \begin{bmatrix} \sin(\psi)\cos(\phi) \\ \sin(\psi)\sin(\phi) \\ \cos(\psi) \end{bmatrix}. \tag{9}$$

To reveal the physical meanings carried by the coefficients, we sampled $\phi$ and $\psi$ at different values and observed how $A_h$ and $\delta_h$ change correspondingly. As $\phi = 0$ and $\psi$ changes from 0 to $\frac{\pi}{2}$, $\delta_h$ kept constant while $A_h$ changes accordingly, see Fig. 5a. A closer inspection to the results indicate a linear




(a) The yaw $\phi$ in the *base* coordinates equals to the summation of rolling result from climbing $\omega\tau$ and head alignment $\delta_h$.




(b) The pitch $\psi$ is directly related to the joint angle amplitude $A_h$ of the head alignment part.

Fig. 6: Head alignment direction in the *base* coordinates.

relation between $\psi$ and $A_h$,

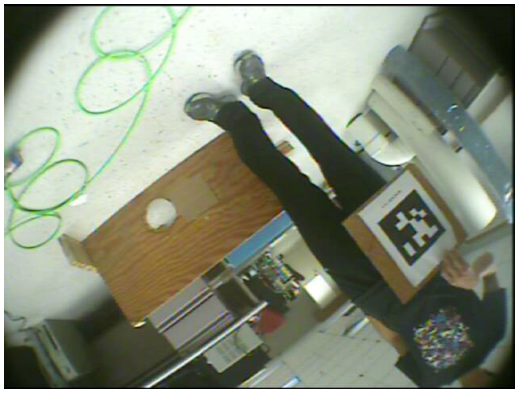
$$\psi = A_h(N - M), \tag{10}$$

where $(N-M)$ is the number of joints in the head alignment part of the robot. Though least-square fit can used, we here choose a linear model for finding a minimal parameterization. Intuitively, Eqn. 10 can be understood as $\psi$ being evenly distributed over all the modules dedicated to the head alignment. When $\psi = \frac{\pi}{2}$ and $\phi$ changed from 0 to $2\pi$, $A_h$ kept constant while $\delta_h$ changed from 0 to $2\pi$. The results in Fig. 5b show that $\phi = \delta_h$.

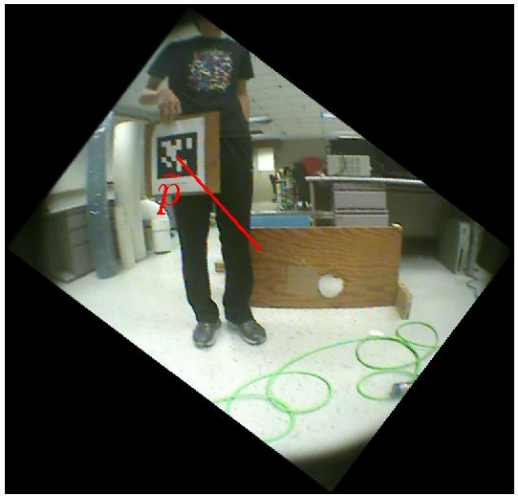
The physical meanings of the coefficients $A_h$ and $\delta_h$ lead to an analytical controller for head alignment,

$$\alpha(n,\tau) = \begin{cases} \frac{\psi}{N-M}\sin(\omega\tau + \phi) & \text{odd} \\ \frac{\psi}{N-M}\sin(\omega\tau + \frac{\pi}{2} + \phi) & \text{even} \end{cases} \tag{11}$$

This controller is compact, smooth and have physically intuitive parameterizations, which in turn allow users more readily control the robot. In addition, this controller now can be applied to a snake robot with different number of modules. The joint inputs generated using this controller can be understood as the feedforward control for the snake robot to maintain a static head alignment.



(a) This figure shows a raw image received from the head camera when the robot is visually servoing and climbing.




(b) The raw image is unrotated to align with the *world* frame.

Fig. 7: To retrieve intuitive measurements, the raw image is unrotated into the *world* frame.

Interestingly, this controller is identical to a pre-existing gait, referred to as "rolling arc." This observation shows that head alignment while pole climbing is actually composing a rolling arc and rolling helix motion, which further implies the possibility of generating new families of behaviors for a snake robot by composing existing gaits.

## V. Visual Servoing

The results in Sec. IV make it possible to derive a feedback control law for online visual tracking. In this section, we implement this controller and demonstrate its efficacy on the CMU modular snake robot. Specifically, we let the snake robot climb a pole while visually tracking an Apriltag [12].

### A. Image Rotation

While the robot is climbing a pole, the head module is forced to spin around its $\vec{z}_{N+1}$ axis. As a result, the world, as viewed through the camera mounted in the head module, is constantly rolling, Fig. 7a. To retrieve intuitive visual data, the images need to be aligned with the *world* frame. The

Fig. 8: This figure shows the robot visually track an Apriltag being manually moved while climbing a pole.

amount of rotation can be explicitly computed as,

$$\theta_h = \omega\tau + \delta_h \tag{12}$$

### B. April Tag Detection

We evaluate the performance of the visual tracking capability by commanding the robot to visually track an Apriltag which is manually moved, Fig. 8. The detection algorithm returns the coordinate of the center of the Apriltag in the image,

$$\vec{p} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{13}$$

The image is then unrotated to retrieve the Apriltag position in the *world* frame, Fig. 7b,

$$\tilde{p} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = R_I(\theta_h)\vec{p} \tag{14}$$

### C. Feedback Control

It is possible to derive a simple feedback control law for visual tracking in terms of the pitch and yaw angle in the *base* coordinates,

$$\begin{aligned} \Delta\phi &= K_\phi \tilde{x} \\ \Delta\psi &= K_\psi \tilde{y} \end{aligned} \tag{15}$$

According to Eqn. 10, we established the following linearized control law,

$$\begin{aligned} \Delta\delta_h &= K_\delta \tilde{x} \\ \Delta A_h &= K_A \tilde{y} \end{aligned}$$

where $K_\delta$ and $K_A$ are proportional gains. In the experiment, we use 12 modules for climbing and 4 modules for head alignment. The controller gains are set to $K_\delta = 0.1$ and $K_A = -0.1$. Fig. 8 shows the robot performing visual tracking while climbing a pole at the speed of 0.05 m/s. The robot is able to visually track the Apriltag moving at a speed about 0.2 m/s at a distance of 3 meters. (with a maximum angle deviation less than 10 degrees). The limiting factor for the performance of the visual tracking is the frequency of the Apriltag detection. Because of the heavy communication between the robot and an off-board computer and the limited computing power, the Apriltag was detected at a frequency varying from 1 Hz to 5 Hz in the experiment.

## VI. Conclusions and Future Work

In this paper demonstrates the capability of a snake robot to simultaneously perform two different tasks using two kinematically coupled portions of its body. The development of this capability was originally based on joint angle optimization. To address the drawbacks of the optimization approach, we apply an FFT based modal decomposition technique to derive an analytical feedforward gait controller. The resultant analytical representation compactly parameterizes head alignment motion with only two parameters. As a result, it facilitates the derivation of a control law for online visual tracking.

Currently, the visual servoing is purely reactive, and thus the feedback control scheme suffers from the problem that once the target goes out of view of the camera, track loss occurs instantly. A trajectory interpolation or a state estimation technique will be employed to address this problem in future work. Additionally, the presented work showed that it is possible to design new gaits by using the existing gaits as the building block. We plan to investigate into this problem in the future work.

## References


[1] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *Mechatronics, IEEE/ASME Transactions on*, vol. 5, no. 1, pp. 39–48, Mar 2000.

[2] D. Murray and A. Basu, "Motion tracking with an active camera," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 5, pp. 449–459, May 1994.

[3] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and scripted gaits for modular snake robots," *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, 2009.

[4] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, "Design of a Modular Snake Robot," in *Proceedigns of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA USA, October 2007, pp. 2609–2614.

[5] J. Burdick, J. Radford, and G. Chirikjian, "A "Sidewinding" Locomotion Gait for Hyper-redundant Robots," *Robotics and Automation*, vol. 3, pp. 101–106, 1993.

[6] G. Chirikjian and J. Burdick, "A Modal Approach to Hyper-redundant Manipulator Kinematics," in *IEEE Transactions on Robotics and Autmation*, vol. 10, 1994, pp. 343–354.

[7] G. Chirikjian, "Theory and Applications of Hyper-Redundant Robotic Manipulators," Ph.D. dissertation, California Institute of Technology, 1992.

[8] R. L. Hatton and H. Choset, "Generating gaits for snake robots by annealed chain fitting and keyframe wave extraction," in *Proceedigns of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO USA, October 2009.

[9] K. Lipkin, I. Brown, A. Peck, H. Choset, J. Rembisz, P. Gianfortoni, and A. Naaktgeboren, "Differentiable and Piecewise Differentiable Gaits for Snake Robots," in *Proceedings of IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct 29 - Nov 2 2007, pp. 1864–1869.

[10] S. Yu, S. Ma, B. Li, and Y. Wang, "Analysis of Helical Gait of a Snake-like Robot," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2008.

[11] C. Gong, M. J. Travers, X. Fu, and H. Choset, "Extended gait equation for sidewinding," *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.

[12] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 3400–3407.