ANALYSIS OF JOINT FAILURES ON THE LATERAL UNDULATION GAIT OF A ROBOTIC SNAKE

Rishad Irani[1], Robert Bauer[2], Lydia North[2], Michael Nicholson[2], David Nolan[2] and Brennan West[2]
[1]*Rolls-Royce Canada Limited Naval Marine, Dartmouth, Nova Scotia, Canada*
[2]*Department of Mechanical Engineering, Dalhousie University, Nova Scotia, Halifax, Canada*
E-mail: robert.bauer@dal.ca

Received March 2014, Accepted January 2015
No. 14-CSME-35, E.I.C. Accession 3696

ABSTRACT

This paper describes the development of a biologically-inspired hyper-redundant wheeled snake robot and a corresponding computer simulator to study the effects that joint failures have on the resulting lateral undulation motion. Experiments and corresponding simulations were carried out to study the robotic snake's lateral undulation gait as power to individual joints was turned off. The results showed that joint failures were most detrimental to the snake's lateral undulation gait when they occurred in the front half of the snake, while joint failures occurring between the midpoint and tail of the robotic snake were found to be less critical and generally resulted in slight lateral drifts as the forward motion progressed. To help compensate for joint failures in the tail-half part of the robot, a bias term was added to the control algorithm. For the conditions tested in this research, the use of a bias term appeared to be effective at reducing the lateral drift.

Keywords: robot snake; joint failures; dynamic modelling; simulation; experimental validation.

ANALYSE DES DÉFAILLANCES D'ARTICULATION SUR LE MOUVEMENT ONDULATOIRE LATÉRAL D'UN ROBOT SERPENT

RÉSUMÉ

Cet article décrit le développement d'un robot serpent sur roue redondant d'inspiration biologique et des simulations numériques correspondantes pour étudier les effets qu'ont les défaillances d'articulation sur le mouvement ondulatoire latéral résultant. Des expériences et simulations ont été faites pour étudier le mouvement ondulatoire du robot serpent lorsque le force motrice sur des articulations individuelles est enlevée. Les résultats montrent que les défaillances d'articulation avait le plus grand impact négatif sur le mouvement ondulatoire latéral du serpent lorsque qu'elles se produisaient dans le moitié avant du serpent ; alors que lorsqu'elles se produisaient dans la moitié antérieure du robot serpent, leur impact était moins critique résultant en une faible dérive latérale pendant le déplacement en avant. Afin de compenser pour les défaillances d'articulation dans la moitié antérieure du robot, un terme de biais a été ajouté à l'algorithme de contrôle. Pour les conditions testées dans cette recherche, l'utilisation du terme de biais semble efficace pour réduire la dérive latérale.

Mots-clés : robot serpent; défaillances d'articulation; modélisation dynamique; simulation; validation expérimentale.

1. INTRODUCTION

The snake-like robotics field is a relatively small sub-group of mobile robotics when compared to other more traditional ground robots such as wheeled or tracked vehicles. The research that is currently being carried out in this robotics field attempts to mimic natural gaits of snakes including side-winding, concertina, slide-pushing, and lateral undulation. Side-winding involves lifting different body segments off of the terrain such that the snake's body is always stationary (rather than sliding) when in contact with the ground. Concertina movement consists of anchoring different body segments in order to pull or push other parts of the body in the direction of motion – similar to an accordion expanding and then contracting. Slide-pushing involves very large side-to-side undulations and is a technique used by snakes on smooth surfaces to incrementally move the body forward. Lateral undulation refers to serpentine motion in which waves propagate through the snake from head to tail – the typical motion which most people associate with a slithering snake. The analyses carried out in this paper focus on this lateral undulation gait.

Hopkins et al. [1] recently published a survey of snake-inspired robot designs, while Transeth et al. [2] and Liljebäck et al. [3] surveyed the available literature on snake modeling and locomotion. The lateral undulation gait studied in the present paper requires anisotropic friction between the robot and the terrain. To achieve this gait, high friction is needed laterally on the robot so that a net forward force can be produced and overcome the lower frictional resistance ahead of the robot. As described by Transeth et al. [2] in their review, small passive wheels can be used to provide the anisotropic friction properties required to achieve lateral undulation locomotion. This anisotropic friction is analogous to the way in which an in-line wheeled skater is able to move forward by "pushing off" in a side-to-side rhythmic motion where the friction is greater in the transverse (lateral) direction and lower in the forward (longitudinal) direction. The snake robot used in the present study also employed passive wheels to achieve the required anisotropic friction.

Characterization of a friction model for snake robots is not a trivial task [4], especially with the anisotropic friction required by wheeled snake robots [5]. Copious amounts of friction research has been performed on the mature field of full-sized wheeled vehicles [6]; however, the same principles applied to full-size wheel models do not necessarily scale down to the smaller wheels used on snake robots [7]. Many researchers have examined and implemented a variety of friction models for their specific robot; however, based on the current literature there appears to be no single friction model that is suitable for all robots and more research needs to be performed to correctly identify and optimize a generalized friction model. To simplify the complex wheel-terrain interaction, most researchers have assumed either a viscous friction model or a Coulomb friction model [8].

Beyond the necessary bio-mimicry control of the individual joint actuators, researchers have also attempted to increase the intelligence of snake-like robots so that they might be able to sense and react to their environment [9]. For example, Transeth et al. [10] have implemented a control algorithm that uses features in the environment as "push-off" points to help propel a wheel-less snake robot. However, in the absences of obstacles the robot has no means of producing the required anisotropic friction and is, therefore, virtually immobile.

Although considerable work has been carried out with respect to the design and implementation of snake-robots, based on the present authors' knowledge, the only prior work on joint failures in snake robots was carried out by Mehta et al. [11]. In their work, Mehta et al. studied the effect of various friction models on the serpentine gait of a five-joint snake robot and implemented the governing equations of motion in Matlab's Simulink software. By comparing simulation results with experiments on a treadmill they determined that a combined viscous and Coulomb friction interaction model yielded promising results. They noted that when joints failed and became rigid near the head or tail link, the net speed increased. They also observed that when a joint failed and became "free" such that it was unable to produce any torque, regardless of where this type of failure occurred along the snake, the net forward speed was compromised.

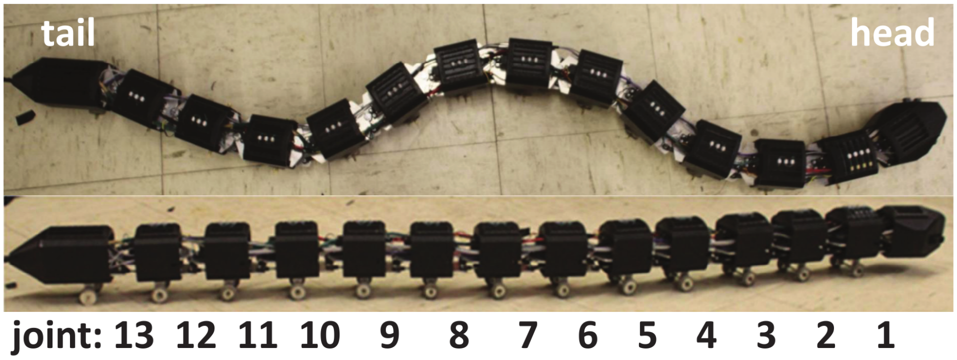

Fig. 1. “Robo-Conda 1” 13-joint arrangement.

Mehta et al. concluded that future research is needed to study friction effects and actuator limitations on other hyper-redundant robotic systems; therefore, the present paper builds upon the work of Mehta et al. to comprehensively carry out such studies. To this end, a new 13-joint experimental snake-robot called Robo-Conda 1 (RC1) was developed along with a novel computer simulator of RC1 that uses the combined viscous and Coulomb friction model recommended by Mehta et al. [11]. Rather than implementing the dynamic equations of motion in Simulink, as done by Mehta et al., the simulator developed in the present research uses Matlab’s SimMechanics – a Simulink toolbox that features a three-dimensional multi-body dynamics engine for mechanical systems. In addition, instead of using a treadmill to characterize friction, as done by Mehta et al., the present authors carried out a corresponding friction parameter optimization search routine to identify the coefficients of friction used in the simulator that best match the experimental results. It is important to note that, unlike Mehta et al. [11] where the failed joint was either completely free turning or completely fixed, the present paper studies failed joints that exhibit friction – the coefficients of which are determined using experimental data.

This paper is organized as follows: Section 2 presents the new snake robot RC1 which was designed, constructed and tested in this research, Section 3 describes the development of the corresponding computer simulator, Section 4 compares and discusses experimental and computer-simulation results of various joint power failures along the robot, while Section 5 draws some conclusions.

2. ROBO-CONDA 1 (RC1)

Figure 1 shows a top-view and side-view of RC1 which was designed and fabricated for this research. As shown in Fig. 1, RC1 is comprised of twelve identical body modules with a unique head and tail module giving a total of fourteen modules linked by thirteen joints. As shown in Fig. 1, the joints are numbered from one to thirteen starting from the snake’s head. Except for the head, all modules have passive wheels attached at their mid-sections. The wheels are made from aluminum and are coated with rubber to increase their lateral anisotropic friction characteristics. Each of the fourteen modules are connected through a double four-bar linkage mechanism, as shown in Fig. 2 to achieve relative rotational motion between modules.

Each body module houses a standard-sized HiTec (HS5645MG) pulsed digital servomotor to actuate the linkage mechanism and create up to ±38° of angular motion between modules. Each servomotor uses an internal potentiometer to enable closed-loop control of the motor angular position. The servomotors were modified with additional leads such that the output from their internal potentiometers could be measured using an external data acquisition system and, therefore, the actual angular position of the snake’s joints during an experiment could be recorded. The servomotors receive their angular position commands from an on-board ATmega2560 microprocessor which is located in the snake’s tail module. This processor has a

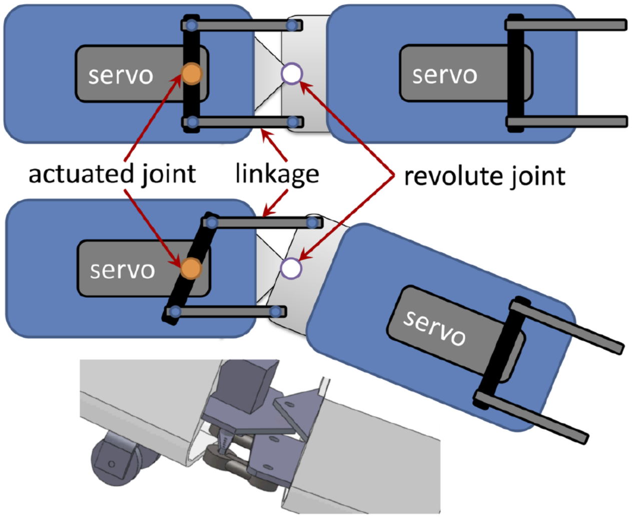

Fig. 2. Double four-bar linkage mechanism at each joint to achive relative rotational motion between modules.

16 MHz clock speed, 256 kB of flash memory, and 4 kB of Electrically Erasable Programmable Read-Only Memory (EEPROM). The microprocessor controller itself is powered by a standard 9V battery which is housed within the tail link. While RC1 was designed with the option to power the servomotors via small battery packs housed in each of the modules, to improve repeatability and power supply consistency, all experiments reported in this paper were externally powered through a tether. This external power supply provided a constant 6V to the robotic snake's servomotors through a single power line at the tail. The main power is then distributed to each servomotor though a wiring harness along the snake's body. A five degree-of-freedom inertial measurement sensor (IDG500/ADXL335) was incorporated into the robotic snake's head to measure the head angular velocity.

The 1.75 m long robot has a mass of 4.3 kg and is capable of carrying up to an additional 5.5 kg of payload before the performance of the robot is compromised. The robot's minimum turning radius is 0.6m, and the maximum forward speed is 750 mm/s when the supply voltage is increased to 7.5V. Based on a literature survey conducted by Hopkins et al. [1], Fig. 3 plots the forward speed of snake-like robots as a function of their length. The results show that, for most snake robots, there appears to be a linear trend between forward speed and length. Figure 3 also shows that, compared to published results, RC1 far exceeds the general trend being 33% faster than the KR II snake robot, and 87.5% faster than ABII and ACMIII robots [1]. The reason RC1 performs so well is likely due to the high-torque (up to 1.19 Nm) servos which enable joint angles to achieve angular speeds of 140 degrees/s. A corresponding robotic snake computer simulator was also developed and validated experimentally using RC1.

3. ROBO-CONDA 1 (RC1) SIMULATOR

To simulate a wheeled robotic snake, Mehta et al. [11] developed a series of two-dimensional (xy plane) governing dynamic equations by adapting the equations of motion presented by Saito et al. [8] to incorporate rolling wheel contact. For an n-link snake robot, where each link's center of mass is described by its position ω and orientation θ, the corresponding equations of motion take the following form [11]:

$$\begin{bmatrix} \mathscr{J} & 0 \\ 0 & MI_n \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\omega} \end{bmatrix} + \begin{bmatrix} \mathscr{C}\dot{\theta}^2 \\ 0 \end{bmatrix} - \begin{bmatrix} \mathscr{L}^T \\ \mathscr{E}^T \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} D^T \\ 0 \end{bmatrix} u \tag{1}$$

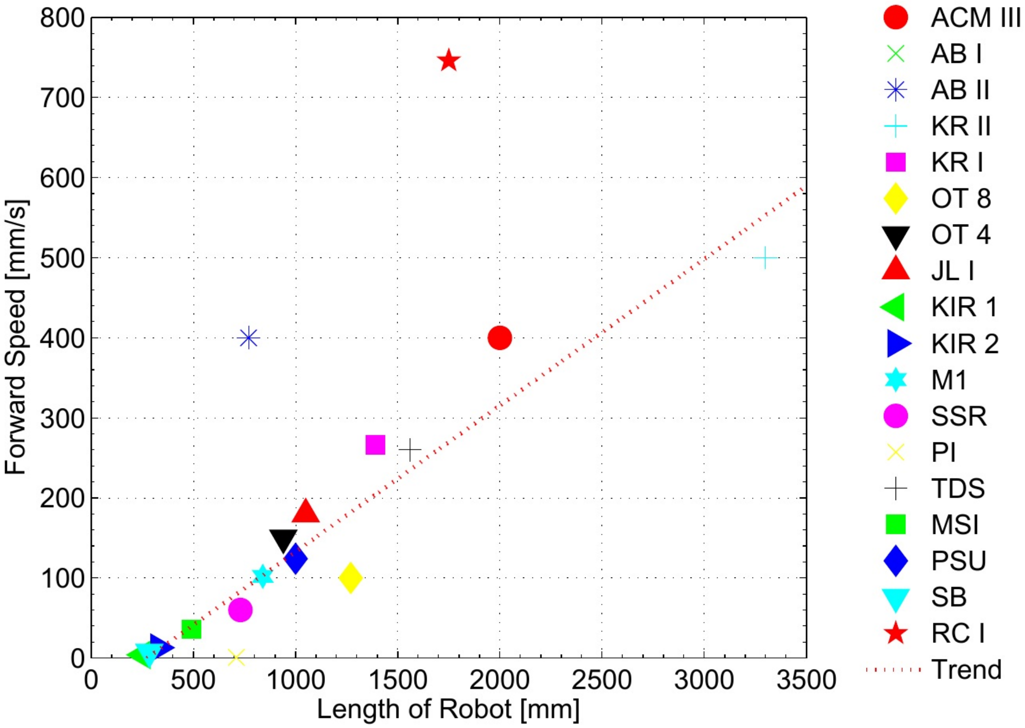

Fig. 3. Summary of forward speed of snake-like robots plotted as a function of their length.

where I_n is the identity matrix of dimension n, $\mathscr{J}$ and MI_n form the generalized mass matrix of the robot, $\mathscr{C}$ is the forcing function (including centrifugal components) due to rotational movement, $\mathscr{L}^T$ and $\mathscr{E}^T$ are transformation matrices which convert the local link coordinates to the global coordinate frame of the robot, f_x and f_y correspond to the friction forces as the snake's wheels interact with the terrain, u represents the torques applied to each joint, and D^T is the transpose of the difference matrix used to relate the reaction forces at each joint where

$$D = \begin{bmatrix} 1 & -1 & & 0 \\ & \ddots & \ddots & \\ 0 & & 1 & -1 \end{bmatrix}_{(n-1)\times n} \tag{2}$$

The present authors took advantage of the high-level programming, graphical user interface, and multi-body dynamics engine offered by the SimMechanics block set in Matlab and Simulink to construct and implement a dynamic simulator that effectively implements the numerical integration of Eq. (1). Moreover, the intuitive nature of Simulink and SimMechanics allows one to easily model joint actuator dynamics with kinematic or kinetic constraints while also modeling the corresponding joint failures – the focus of the current paper using the snake robot RC1.

Each link in the robot was modeled using a SimMechanics body block having a mass m of 2.86×10^{-1} kg and principle moments of inertia I_{xx}, I_{yy} and I_{zz} of 5.362×10^{-4} kg m^2, 2.383×10^{-4} kg m^2 and 2.383×10^{-4} kg m^2, respectively. The experimentally-measured servomotor angular positions (collected from the on-board servomotor potentiometers) were used as inputs to the simulator. These inputs were connected within SimMechanics to kinematic motion joint actuator blocks to ensure that the motion of each simulated joint follows the actual (measured) experimental joint motion.

To mimic lateral undulation motion, the joint angles θ_i of the physical robot RC1 were commanded for each joint i using a phase-shifted sinusoidal signal generated using the following equation:

$$\theta_i = A_i \sin(\omega_i t - \phi_i) \tag{3}$$

where A_i is the amplitude, ω_i is the frequency, and ϕ_i is the phase shift for each joint. In this research, an amplitude of 38°, a frequency of 140 degrees/s and phase shifts ranging from 10° to 90° were used.

Combined Coulomb and viscous friction models having the following form were applied in both the lateral (y-direction) and longitudinal (x-direction) wheel directions (F_{lat} and F_{long}, respectively) to model the interaction between the wheels and the terrain:

$$F_{lat} = \text{sgn}(v_{lat})(c_{lat}|v_{lat}| + \mu_{lat} mg) \tag{4}$$

$$F_{long} = \text{sgn}(v_{long})(c_{long}|v_{long}| + \mu_{long} mg) \tag{5}$$

where m is the mass of a single body module, g is the acceleration due to gravity, v_{lat} and v_{long} represent the components of the wheel velocity in the direction against which friction is applied, c_{lat} and c_{long} correspond to the viscous friction constant coefficients having units of kg/s, while μ_{lat} and μ_{long} represent the Coulomb friction constant coefficients.

To simulate a failed joint where the servomotor was free to rotate uncontrolled, it was necessary to implement joint limits to prevent the failed joint from exceeding the mechanical limits of ±38°. The revolute joints within the simulator that connect the links (when working) are nominally activated using kinematic constraints where the joint angular position, velocity and acceleration are known and commanded. When a joint fails, however, the joint kinematics are not known (when the power is turned off to a servomotor, that servomotor's on-board potentiometer is no longer operational and, therefore, the joint motion cannot be measured) and the joint is free to rotate with joint friction as the only joint torque applied. Since it is currently not possible in SimMechanics to simultaneously configure both kinematic and kinetic constraints for a single joint, Fig. 4 shows the block diagram logic needed within Simulink to enable the option of having a joint work or fail. As depicted in Fig. 4, two revolute joint blocks were used for every joint in RC1 – a primary revolute joint and a secondary revolute joint. The primary revolute joint block represents RC1's joint when it is working (with the joint actuated using kinematic constraints consisting of angular position, velocity and acceleration), while the secondary revolute joint represents RC1's joint when it fails (with the joint actuated using a kinetic constraint consisting of a torque due to joint friction). The primary and secondary revolute joint blocks were then connected via a massless body block within the simulator. To simulate a working joint, the primary revolute joint is kinematically actuated to follow the measured joint motion while, at the same time, the secondary revolute joint is kinetically (torque) regulated using a PD controller to ensure no secondary joint rotation occurs. To simulate a failed joint, the primary revolute joint is kinematically commanded with zeros to ensure no primary joint rotation while, at the same time, the secondary joint experiences a torque due to joint friction. If, however, the secondary joint reaches a joint limit (±38°), a joint limit PD controller is used within the simulator that applies the necessary restoring torque to ensure that the joint remains within the angular limits. It should be noted that, when used to connect co-located joints in the way described, the presence of a massless body block does not create any numerical issues within SimMechanics. For the simulations carried out in this research, the joint friction was assumed to be viscous with a constant viscous friction coefficient c_{joint} having units of kg m^2/s.

4. EXPERIMENTAL AND SIMULATION RESULTS

To identify the resulting five friction model coefficients c_{joint}, μ_{lat}, μ_{long}, c_{lat} and c_{long} within the simulator, the following experimental cases were conducted using RC1: no joint failure, joint seven failure, joint

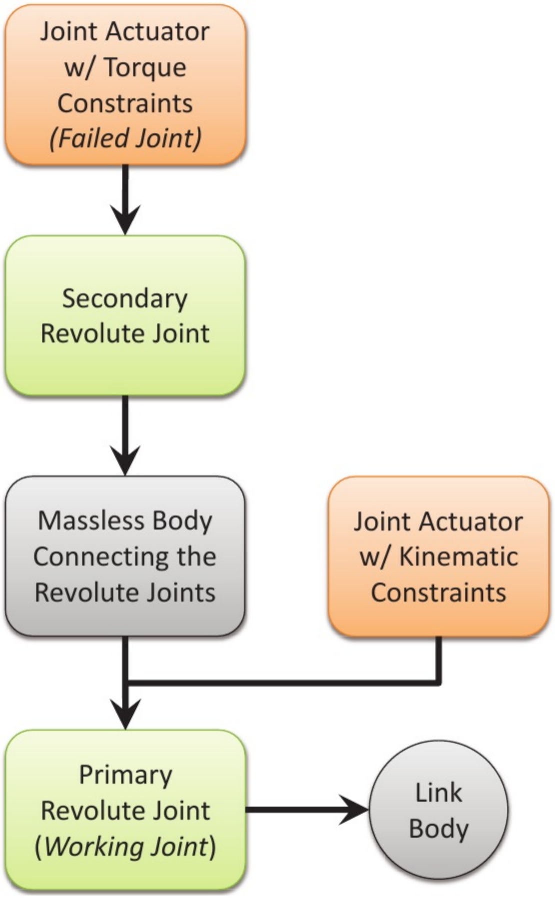
Joint Actuator w/ Torque Constraints *(Failed Joint)*
Secondary Revolute Joint
Massless Body Connecting the Revolute Joints
Joint Actuator w/ Kinematic Constraints
Primary Revolute Joint (*Working Joint*)
Link Body

Fig. 4. Flowchart illustrating the simulator logic for a failed joint.

thirteen failure, and joint seven failure with a bias term added to the commanded joint angles. To improve repeatability and enable better comparisons between experiments, each experiment initialized the sinusoidal angular position of the joints such that the snake started in the same serpenoid wave configuration and orientation as shown in Fig. 5 – with the head of the snake consistently positioned at the same starting location. The small black circles in Fig. 5 represent the initial position of each of the body modules at the beginning of an experiment. Each experiment was carried out five times to establish the level of repeatability of the experiments. The first test case actuated all thirteen joints in order to study the nominal (no joint failure) motion. The snake was driven until the head crossed a line located 3.18m from its starting location, at which point the experiment was terminated and the final snake-head position coordinates (x,y) were recorded. Similar experiments were conducted where individual joints were methodically turned off (corresponding to a failed joint) starting at the head and moving towards the tail one joint at a time.

The following initial values for the friction coefficients were manually found via trial and error by visually comparing the simulator and experimental results: $c_{joint} = 0.045$, $\mu_{lat} = 1.5$, $\mu_{long} = 0.1$, $c_{lat} = 2.0$ and $c_{long} = 0.1$. Using these manually-tuned values as a starting point, an Alternating Variable Search (AVS) algorithm [12] was then implemented to identify the friction model coefficients in the simulator that minimize the differences between experimental and simulated results. To quantitatively compare the experimental and simulated results, the differences between the predicted and experimentally-measured final snake head positions were calculated and this difference was compared against the displacement of the head from its

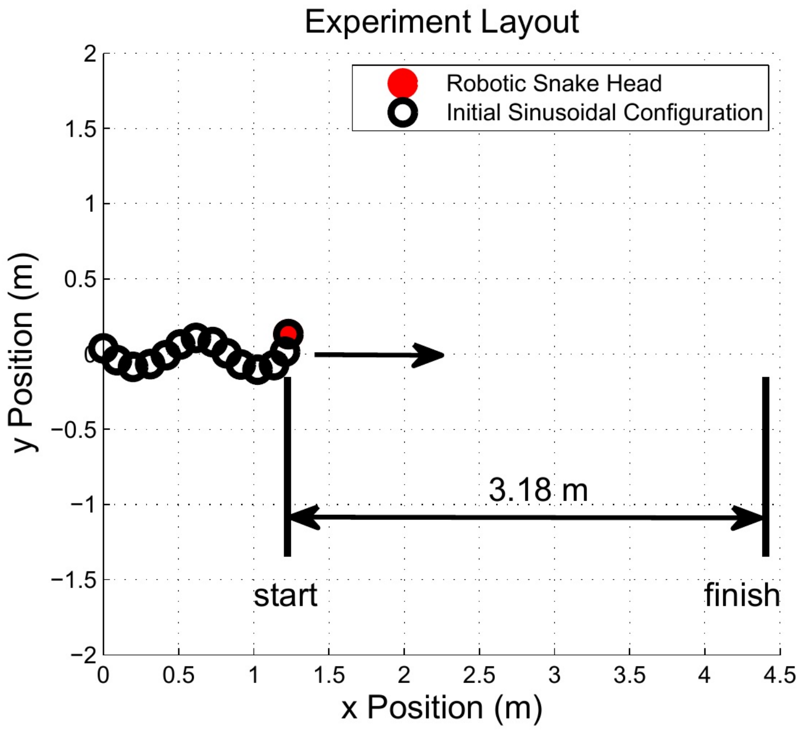

Fig. 5. Experimental layout showing the initial snake configuration and the *x*-*y* coordinate frame of reference from which experimental results were taken.

starting position (total distance traveled) to yield a percentage simulator error as shown in Eq. (6):

$$\text{Simulator Error} = \left(\frac{|\text{predicted head pos} - \text{measured head pos}|}{\text{total distance traveled}}\right)100\% \tag{6}$$

Using the AVS method, one friction model coefficient was varied at a time, starting with c_{joint}, to find the value of this coefficient that minimizes the simulator error. The next iteration would then retain this value of the coefficient and move on to a different friction model coefficient (u_{lat} in this case) to find the value that minimizes the resulting simulator error. As summarized in Table 1 and illustrated in Figs. 6 and 7, this iterative process was repeated ten times (varying one friction parameter at a time) ultimately resulting in the following friction coefficients being identified: $c_{joint} = 0.045$, $\mu_{lat} = 1.5$, $\mu_{long} = 0.12$, $c_{lat} = 3.75$ and $c_{long} = 0.14$. In Table 1, the bold numbers represent the best friction model coefficient found for a particular iteration, while in Figs. 6 and 7 the filled circles correspond to the best friction model coefficient found for a particular iteration. It should be pointed out that in iteration #9 (Fig. 7), two values for c_{lat} competed for minimizing the simulator error: c_{lat}=2.75 and c_{lat}=3.75; however, given that the simulation error was not very sensitive to changes in this parameter (over the 2.75 to 3.75 range), the latter value was selected to be consistent with iteration #4. The final identified friction model coefficients were then kept constant for all remaining simulations presented in this paper.

The simulation and experimental results for the case study when no joints along the snake failed are presented in Fig. 8. This figure shows a sample of the simulated head trajectory from start to finish as a dashed line. The corresponding experimentally-measured final head position is then superimposed on the figure as a star "*". A line is drawn between the simulated and experimental final head locations to visually show the difference between the predicted and measured head positions. The resulting predicted and measured

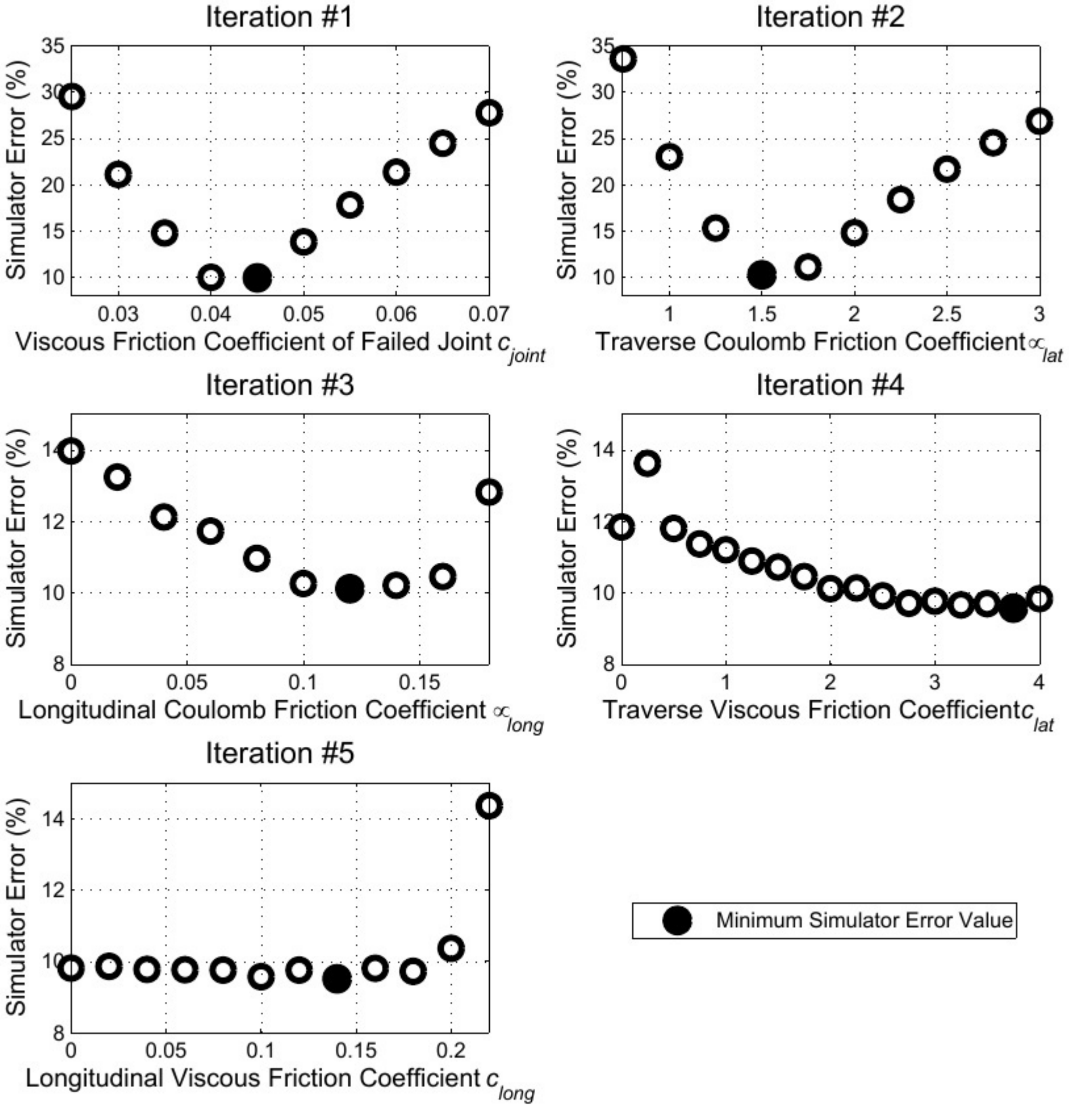

Fig. 6. Friction parameter identification: iterations 1 to 5.

final head positions are also shown in this figure for the remaining four repeatability tests. As can be seen in this figure, there is excellent agreement between the simulation and experimental results with the simulated and experimentally-measured final head positions having a positive component in the y direction between approximately +0.25 and +1m. The first row of Table 2 summarizes the results of the five repeatability tests for this case study and shows that the simulator is able to predict the final snake head position to within, on average, 10.2% when no joints are failed. A sample of the predicted and experimentally-measured head angular velocities are plotted in Fig. 9 as a function of time for this no-failure case study, again showing good agreement between experiment and simulation.

The next series of tests involved turning off the power to joint two (near the head of the snake) to study the effect that such a failure has on the mobility of the robot. Similar to Fig. 8, Fig. 10 plots the initial configu-

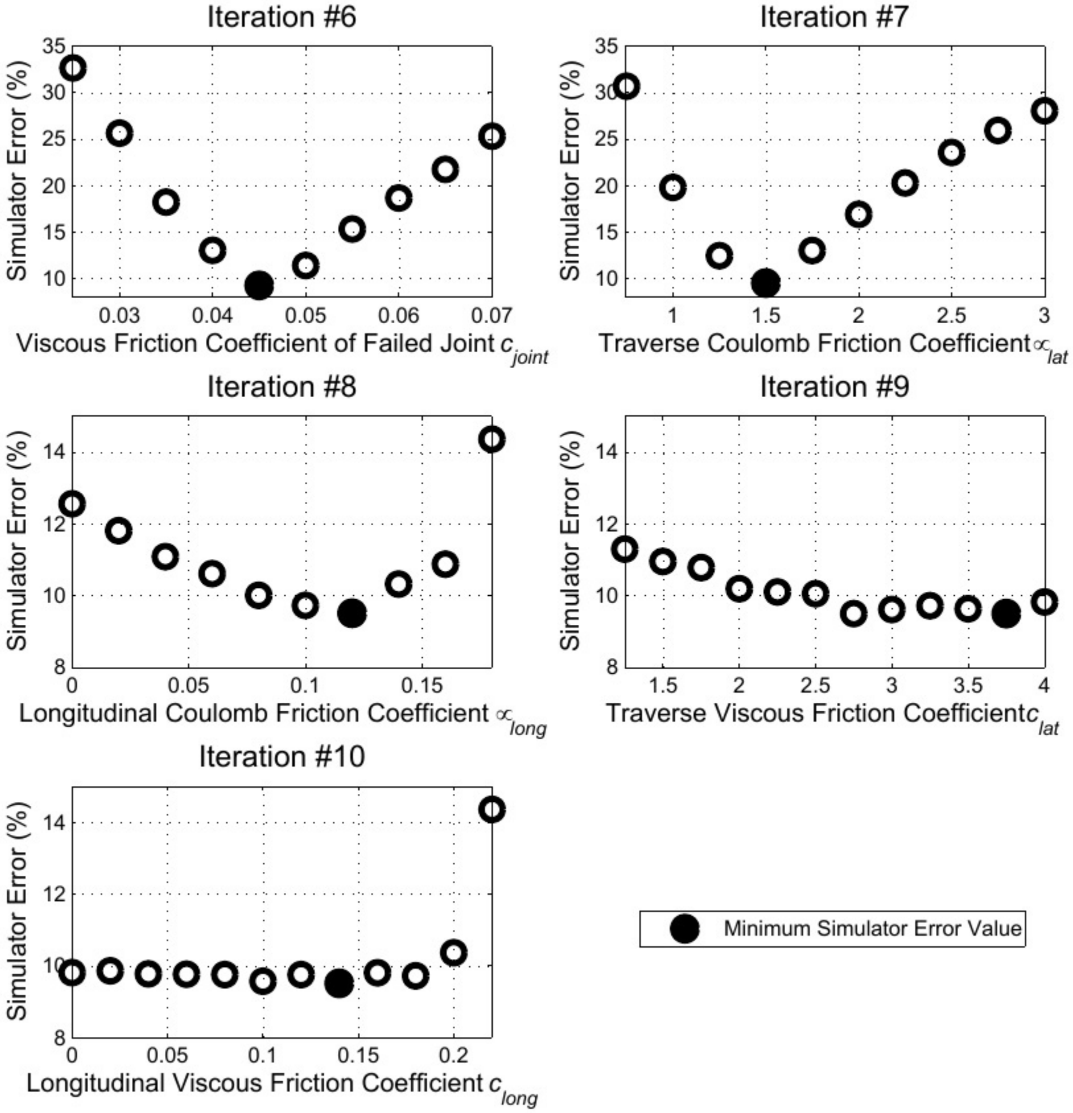

Fig. 7. Friction parameter identification: iterations 6 to 10.

ration of the links as small circles and a sample of the resulting simulated head trajectory as a dashed line. For this case study, the robot did not move forward and, instead, immediately began to continuously turn in a counterclockwise direction as viewed from above following a circular-like trajectory. The experiment continued until the head of the snake had made two complete revolutions. To capture the salient characteristics of this failure mode, the number of times the snake completed a circular trajectory (two in this case) and the mean diameter of the resulting circular motion (0.56 m in this case) were measured. This mean diameter is depicted by the large circle superimposed on Fig. 10. Using the experimentally-measured joint angles (except joint two for which the power was turned off in this test) the simulator was used to try to predict this circular motion. The simulated head trajectory in Fig. 10 shows the start and predicted finish locations of the head as well as the continuous counterclockwise turning of the snake during the experiment. As can

Table 1. Friction model parameter identification: iterations 1 to 10.

Param.	Initial	Iteration Number									
	Value	1	2	3	4	5	6	7	8	9	10
c_{joint}	0.045	**0.045**	0.045	0.045	0.045	0.045	**0.045**	0.045	0.045	0.045	0.045
u_{lat}	1.5	1.5	**1.5**	1.5	1.5	1.5	1.5	**1.5**	1.5	1.5	1.5
u_{long}	0.1	0.1	0.1	**0.12**	0.12	0.12	0.12	0.12	**0.12**	0.12	0.12
c_{lat}	2	2	2	2	**3.75**	3.75	3.75	3.75	3.75	**3.75**	3.75
c_{long}	0.1	0.1	0.1	0.1	0.1	**0.14**	0.14	0.14	0.14	0.14	**0.14**

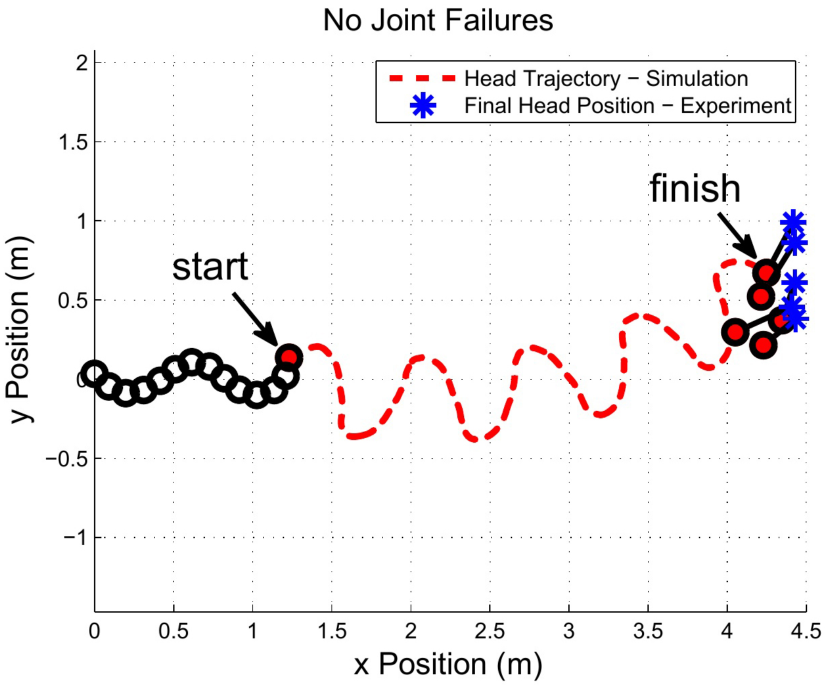

Fig. 8. Case study: no joint failures.

be seen in this figure, the simulator is able to accurately capture the resulting circular motion as it correctly predicts the snake-head's trajectory making two complete revolutions. It is clear that a failure of joint two is a critical failure that inhibits the robot's ability to move forward and is effectively captured by the simulator.

Similar circular motions were observed when the power was turned off (individually) to joints three, four, five and six – with the circle diameters gradually increasing as the joint failures progressed from the head towards the snake's center. Only when a failure occurred at joint seven (the middle joint along the robot), did forward motion actually occur without the snake turning back on itself. Figure 11 presents the experimental and simulated results for this joint seven failure case and includes all five repeatability tests where, to capture the relevant characteristics of this failure mode, the final head position was recorded and compared. Referring to Fig. 11, although the snake has a net forward motion, instead of a final head position with a slightly positive y-component (as with the no joint failure case), the joint seven failure resulted in a head position with a significant negative y-component between approximately -1 and -2m. Table 2 summarizes the results for this case study on line 2 and it can be seen from the table that the predicted and experimental results agree well with an average error of only 5.8%.

As the individual joint-failure experiments continued towards the back of the robotic snake, the snake continued to move forward with either a slightly negative or slightly positive drift in the y-direction. For

Table 2. Summary of simulator error for the no joint failure case as well as cases where failure occurred at joint 7, 13 and 7 with bias added.

	Repeatability Experiment					
Failure Joint	1	2	3	4	5	Avg
None	10.8%	7.7%	12.2%	8.2%	12.2%	10.2%
J7	5.5%	3.0%	16.3%	2.0%	2.3%	5.8%
J13	21.9%	5.7%	18.2%	12.3%	3.6%	12.3%
J7+bias	7.3%	2.1%	22.4%	11.2%	5.2%	9.6%

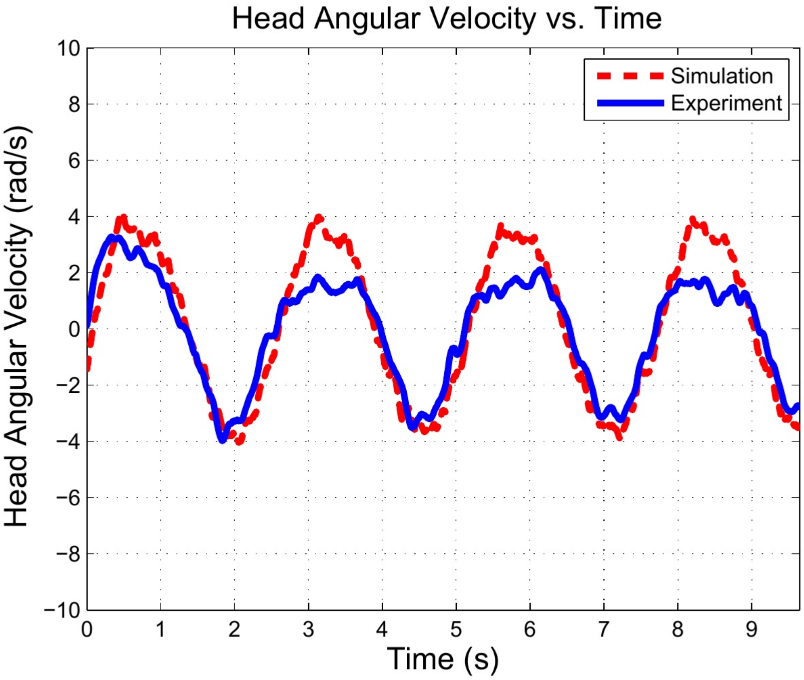

Fig. 9. Snake head angular velocity vs. time for the case of no joint failures.

example, Fig. 12 shows the effect of a failure in joint thirteen where, again, the initial configuration of the links is shown and a sample simulated head trajectory is depicted as a dashed line from the start to the finish of the test. The corresponding experimentally-measured head positions are also superimposed on the figure for all of the five repeatability tests. Table 2 summarizes the results for this case study on line 3 and shows that the simulator is able to predict the final snake head position with an average difference between simulation-predicted and experimentally-measured final head positions of 12.3%. Figures 10, 11 and 12 demonstrate that, while joint failures near the head of the snake inhibit forward motion, joint failures between the middle and tail of the snake caused the snake to drift in the *y*-direction. Such drifts in head position could potentially be compensated for by modifying the commanded joint angle in Eq. (3) with the addition of a bias parameter B_i as follows:

$$\theta_i = A_i \sin(\omega_i t - \phi_i) + B_i \tag{7}$$

Figure 13 shows a sample of the corresponding experimental and simulated results when a constant bias B_i of 3.0 was added to all of the joint angles. As seen in this figure, the added bias effectively enables the robot to move forward without the relatively large drift in the negative *y*-direction originally observed in Fig. 11.

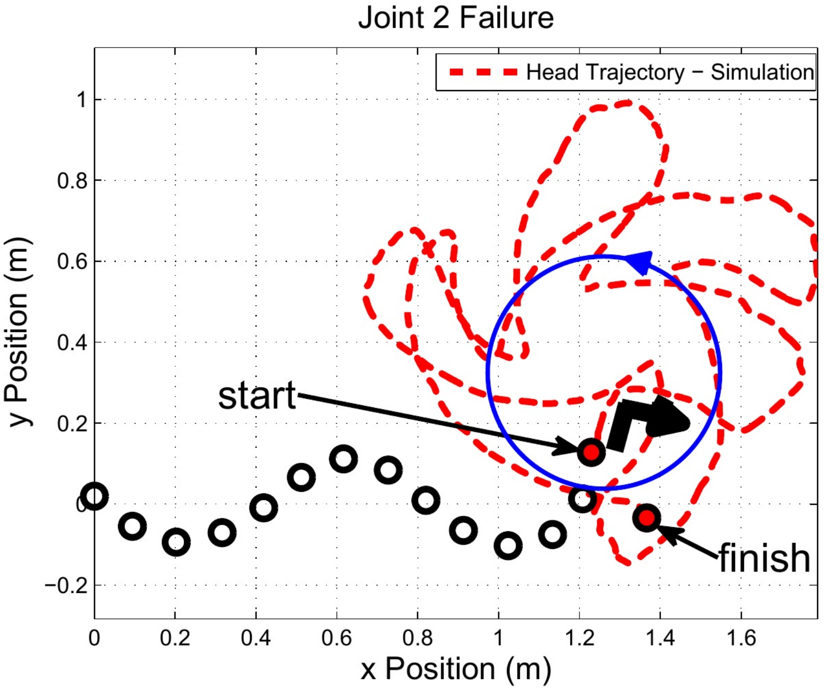

Fig. 10. Case study: joint two failure

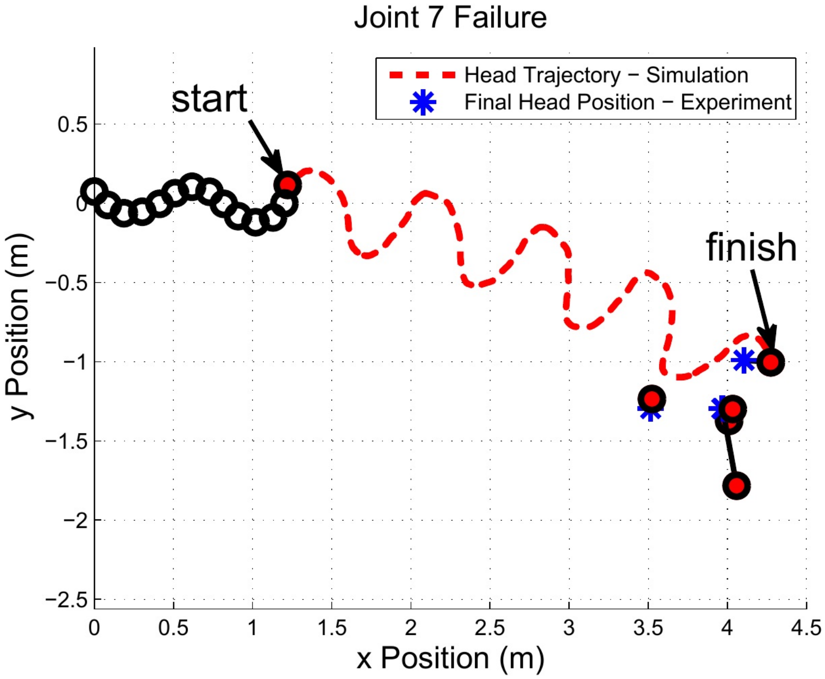

Fig. 11. Case study: joint seven failure.

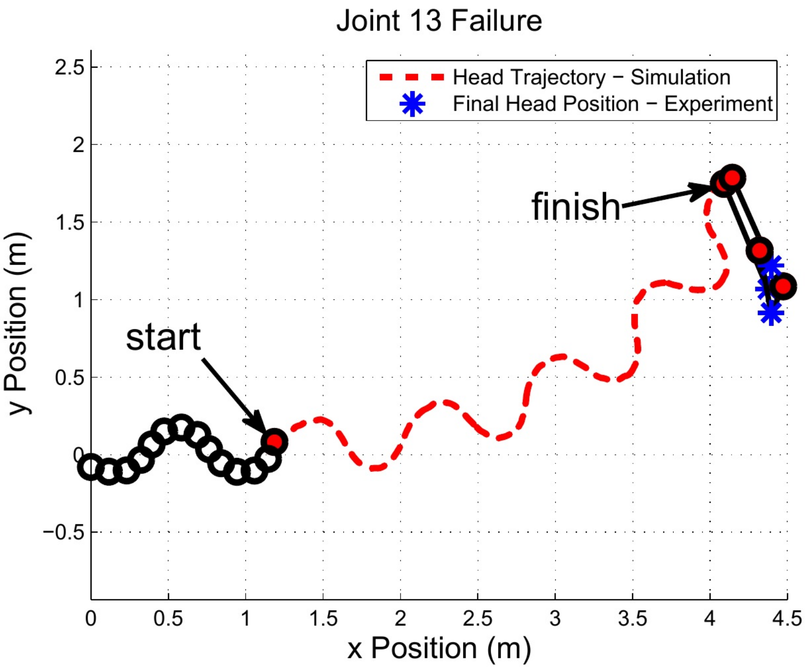

Fig. 12. Case study: joint thirteen failure.

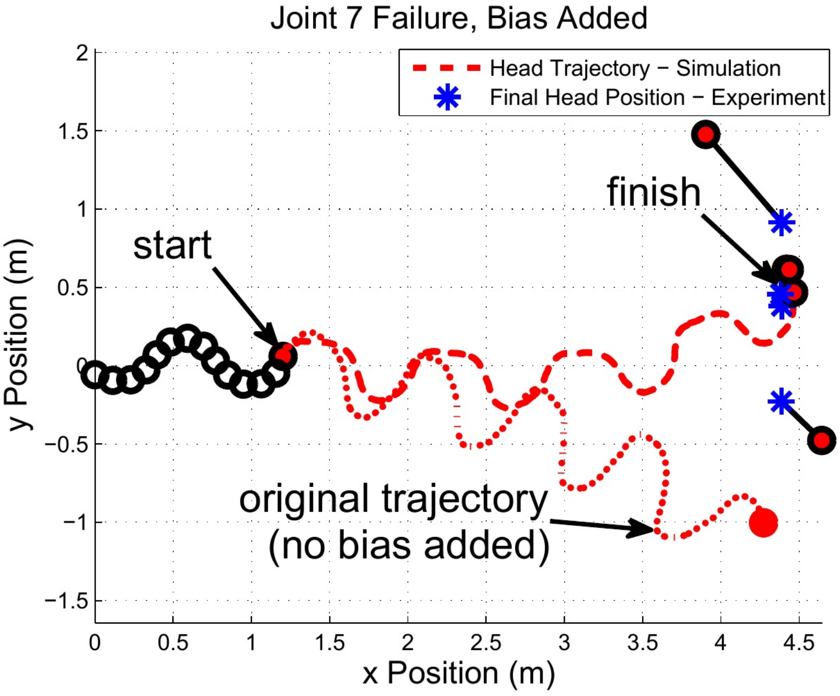

Fig. 13. Case study: joint seven failure with bias added.

The value for B_i of 3.0 was selected because it produced the smallest deviation from the desired straight-line trajectory. Increasing this value beyond 3.0 simply caused the snake to turn further away from its original no-bias-added trajectory, while values less than 3.0 caused the snake to turn the other direction and more closely follow its original no-bias-added trajectory. The last line of Table 2 shows that the simulator's predictions for this case agree with the experimental data to, on average, 9.6%.

The experimental and simulated joint failure results suggest that, for the conditions used in this research, joint failures from the front of the robotic snake to the midpoint are significantly more critical than joint failures occurring in the back half of the snake. As shown in Figs. 11 and 13, a possible solution for the latter case is to add a bias term to compensate for any lateral drift caused by the joint failure. This important observation suggests that failures near the front of the snake could also be compensated for by driving the snake backwards (such that the failure near the head becomes a failure near the tail) and adding a bias term to correct for any drift.

5. CONCLUSIONS

The fourteen-segment wheeled robotic snake described in this paper (referred to as RC1) uses thirteen commercially-available digital servomotors to actuate the links with a double four-bar linkage. A corresponding computer simulator was developed with the SimMechanics toolbox that captures the salient characteristics of RC1 as it effectively predicts the lateral undulation motions observed in the experiments with typical head-position errors averaging to approximately 10%. The Alternating Variable Search (AVS) algorithm in conjunction with the computer simulator was effectively used to solve for the various friction parameters of the system. The computer simulator also shows that using a massless body block with a primary and secondary revolute joint in SimMechanics is a viable method of simulating joint failures within SimMechanics. Power failures applied to individual joints along the robot's body showed that joint failures within the front half of the snake severely compromise the robot's gait, while power failures in the rear half of the snake were able to be effectively compensated for by incorporating a bias into the joint trajectory equations. This important result suggests that, if a joint fails near the head of the snake, one could simply drive the snake in reverse such that the failed joint would effectively be near the tail and be compensated for by adding the necessary bias to the joint motion. Developing a method to select this bias term is an area of future research and could involve periodically updating the bias term as the snake attempts to follow its desired trajectory to help keep it on course. Furthermore, while the current work and scope did not include the use of an overhead vision system to further verify the ground truth motions, this overhead data collection could be investigated and used by future researchers.

ACKNOWLEDGMENTS

The authors would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC), Shell Canada, and the Department of Mechanical Engineering at Dalhousie University for their financial support of this research. The authors would also like to thank Riddhi Sharma for her help with the experiments.

REFERENCES

1. Hopkins, J., Spranklin, B. and Gupta, S., "A survey of snake-inspired robot designs", *Bioinspiration & Biomimetics*, Vol. 4, p. 021001, 2009.
2. Transeth, A., Pettersen, K. and Liljebäck, P., "A survey on snake robot modeling and locomotion", *Robotica*, Vol. 27, No. 07, pp. 999–1015, 2009.
3. Liljebäck, P., Pettersen, K., Stavdahl, O. and Gravdahl, J., "A review on modelling, implementation, and control of snake robots", *Robotics and Autonomous Systems*, Vol. 60, No. 1, pp. 29–40, 2012.
4. Olsson, H., Åström, K., Canudas de Wit, C., Gäfvert, M. and Lischinsky, P., "Friction models and friction compensation", *European Journal of Control*, Vol. 4, pp. 176–195, 1998.
5. Kalani, H. and Akbarzadeh, A., "Effect of friction models on snake robot performance", *International Journal of Modeling and Optimization*, Vol. 1, No. 2, pp. 129–133, 2011.

6. Sextro, W., *Dynamical Contact Problems with Friction: Models, Methods, Experiments and Applications*, Vol. 3, Springer Verlag, 2007.
7. Bekker, M., *Introduction to Terrain-Vehicle Systems*, University of Michigan Press, Ann Arbor, 1969.
8. Saito, M., Fukaya, M. and Iwasaki, T., "Serpentine locomotion with robotic snakes", *Control Systems, IEEE*, Vol. 22, No. 1, pp. 64–81, 2002.
9. Shammas, E., Wolf, A. and Choset, H., "Three degrees-of-freedom joint for spatial hyper-redundant robots", *Mechanism and Machine Theory*, Vol. 41, No. 2, pp. 170–190, 2006.
10. Transeth, A., Leine, R., Glocker, C., Pettersen, K. and Liljeback, P., "Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments", *IEEE Transactions on Robotics*, Vol. 24, No. 1, pp. 88–104, 2008.
11. Mehta, V., Brennan, S. and Gandhi, F., "Experimentally verified optimal serpentine gait and hyperredundancy of a rigid-link snake robot", *IEEE Transactions on Robotics*, Vol. 24, No. 2, pp. 348–360, 2008.
12. Antia, H., *Numerical Methods for Scientists and Engineers*, Birkhauser Verlag, 2002.
13. Hatton, R. and Choset, H., "Generating gaits for snake robots: Annealed chain fitting and keyframe wave extraction", *Autonomous Robots*, Vol. 28, No. 3, pp. 271–281, 2010.