Robotica (2008) volume 26, pp. 267–276. © 2007 Cambridge University Press
doi:10.1017/S0263574707003864 Printed in the United Kingdom

Assisted passive snake-like robots: conception and dynamic modeling using Gibbs–Appell method

Gholamreza Vossoughi*, Hodjat Pendar, Zoya Heidari and Saman Mohammadi

Center of Excellence in Design, Robotics and Automation (CEDRA), Mechanical Engineering Department, Sharif University of Technology, Azadi Ave. Tehran 11365-9567, Iran.

(Received in Final Form: August 13, 2007. First published online: October 10, 2007)

SUMMARY

In this paper, we present a novel structure of a snake-like robot. This structure enables passive locomotion in snake-like robots. Dynamic equations are obtained for motion in a horizontal plane, using Gibbs–Appell method. Kinematic model of the robot include numerous nonholonomic constraints, which can be omitted at the beginning by choosing proper coordinates to describe the model in Gibbs–Appell framework. In such a case, dynamic equations will be significantly simplified, resulting in considerable reduction of simulation time. Simulation results show that, by proper selection of initial conditions, joint angles operate in a limit cycle and robot can locomote steadily on a passive trajectory. It can be seen that the passive trajectory is approximately a Serpenoid curve.

KEYWORDS: Snake-like robot; Passive motion; Dynamic modeling; Gibbs–Appell method; Serpenoid curve.

Nomenclature

L_i	ith link
2L	Length of each link
G_i	Center of gravity of ith link
m	Mass of each link
I	Moment of inertia of each link about its center of gravity
θ_i	Orientation of the ith link
ϕ_i	Relative angle between ith and $(i+1)$th links
P_i	The joint between the L_i and L_{i+1}
K	Stiffness of torsional spring
u_i	Relative angle between $(i+1)$th link and A_i link
$\dot{\eta}_i$	Linear velocity of the center of each link
$\dot{\theta}_i$	Rotational velocity of the center of each link
$\dot{\mathbf{q}}_i$	Velocity vector of ith link
$\mathbf{V}_{P_i}\vert_{\hat{e}_{x_{i+1}}}$	Velocity of joint P_i along x-axis of coordinate attached to $(i+1)$th link
${}^{i+1}_{i}A$	Constraint matrix between ith and $(i+1)$th links
A	Constraint matrix
V_{G_i}	Linear velocity of the center of mass
$\boldsymbol{\omega}_i$	Angular velocity of each link
$\mathbf{a}_{G_i}$	Linear acceleration of the center of mass
$\boldsymbol{\alpha}_i$	Angular acceleration of each link
$\mathbf{H}_{G_i}$	Angular momentum of each link about its center of gravity
S	Gibbs–Appell function or energy of acceleration
δW	Total work
$\delta W_{c.}$	Conservative work
$\delta W_{n.c.}$	Nonconservative work
$\mathbf{U}$	Vector of relative angles between each link and the link A_i
μ_1	Viscous coefficients of friction at the wheels
μ_2	Viscous coefficients of friction at the rotary joints between individual links
M_i	Mass matrix of each link
M	Total mass matrix of robot
$\boldsymbol{\lambda}$	Lagrange Multiplier vector
C	Natural orthogonal compliment matrix
ξ	Amplitude of oscillation in serpenoid curve
ζ	Phase difference between successive links in serpenoid curve
γ	Radius of the circular path in serpenoid curve
ω	Speed that the serpenoid curve propagates along the body of robot

* Corresponding author. E-mail: vossough@sharif.edu

1. Introduction

Snakes perform many kinds of movement and can immediately adapt to a new environment by changing the locomotion mechanism and body shape. Unique and interesting gait of snakes enables them to crawl, climb a hill and climb a tree by winding and move on very slippery floors.[1] Because of these features of snakes, it seems useful to make a snake-like robot that can imitate the special functions of snakes.

Snake robots can be used in exploration, in narrow winding environments like pipelines or irregular terrains like ruins and rubbles of collapsed buildings, hazardous environments such as nuclear power plants, inspection and medical interventions. They can also be used for inspection applications in industries.

Hirose[2,3] has long investigated snake robots. He modeled the snake by a wheeled link mechanism with no side slip. As the control inputs, he used sinusoidal functions and tuned the

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

amplitude, frequency, and phase differences to drive the robot to a certain direction. He also described the trace of a snake as what he called a 'Serpenoid curve' which guarantees that the distribution of strain in the muscles become a smooth function. Takanashi *et al*. have developed a 3-dimensional snake robot for the purpose of search and rescue for survivors in collapsed buildings[4] and Worst and Linnemann[5] have developed another 3-dimensional snake robot using tendon-driven mechanism. Mori and Hirose[6] have also built a 3-dimensional robot equipping with larger passive wheels. Prautsch and Mita[7] have derived the equation of motion of a 2-dimensional snake robot with passive wheels using Lagrange method while Ute and Ono[8] have implemented Newton's method for derivation of equation of motion of a 2-dimensional snake robot. Chirikjian[9] has developed a new method to obtain dynamics of snake-like robot using approximation of a continuum hyper redundant manipulator. Jones and Walker[10] have introduced a new method for synthesizing kinematic relationships for a general class of snake-like robots.

One of the main drawbacks of snake-like robots is their low speed and poor power efficiency for surface locomotion.[11] Therefore, it is important to increase the speed and the efficiency of snakelike locomotion. Many animals are superbly designed to exploit passive dynamics and manage to reduce the metabolic cost of running considerably by utilizing the elastic properties of their muscles tendons and bones.[12]

Since in Serpenoid gait, curvature changes sinusoidally over its lengths, it seems to be possible to save energy in springs to attain a sinusoidal behavior and as a result passive motion. Using passive motion reduces the power consumption sometimes up to 90%[13] and thus increasing the efficiency.

The concept of passive dynamic running has been followed in the area of robotics, where hopping machines harness the compliant elements in their knee joint for efficient running. The idea of using the passive motion of a mass-spring system was first proposed by Seifert.[14] In 1990, McGeer[15] developed a mechanism which included a pair of legs, hinged at the hip, which was able to walk downhill only under the influence of gravity (Fig. 1(a),(b)). Raibert and Thompson[16] suggested the addition of a torsional spring to the hip of a one-legged robot, which provides the leg with a passive swing motion (Fig.1(c)).

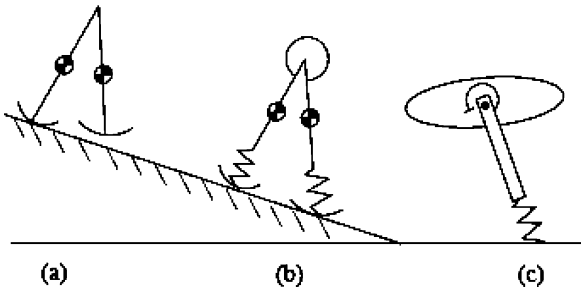


Fig. 1. (a) McGeer's bipedal stable passive dynamic walking machine, (b) McGeer's model for bipedal stable passive dynamic running, and (c) Thompson and Raibert's model for unstable one legged passive dynamic running.

In these robots, a specific initial condition can be found, in which the robot can continue its motion without any active actuator. Actuators are used to recoup the dissipated energy. Finding such initial conditions is sometimes very sophisticated.

For controlling the snake-like robot, we need to derive the dynamic equations of motion. Snake-like robots have a complex dynamics, and due to existence of large number of masses and constraints among them (especially nonholonomic constraints), derivation of dynamic equations becomes difficult.

During the last decades, many researchers have tried to find new algorithms, new numerical integration techniques and better implementation methods in order to speed up the dynamic simulation of complex multibody systems. Among them, the most significant ones who were able to find efficient order n methods to derive the equations of motion were Featherstone,[17] Ginsberg,[18] Kane and Levinson,[19] Rosenthal,[20] Vukobratovi'c,[21] Jerkovsky,[22] Baraff,[23] and Anderson *et al.*[24]

Desoyer and Lugner,[25] were the first to solve the inverse dynamic of manipulators by proposing a recursive development of (n^2) order. He used the Jacobian matrix of the manipulator in order to avoid the explicit development of partial derivatives. Later, a recursive algorithm of $O(n)$ was presented by Rudas and Toth,[26] to solve a similar problem, based on the minimization of the Gibbs function using of Lagrange's multipliers. Recently, Mata *et al.*[27] using the advantages of Gibbs–Appell equations in computing the generalized forces on robotic manipulators, have introduced an efficient $O(n)$ recursive algorithm for the inverse dynamics of robotic manipulators.

In previous researches for deriving snake's equations of motion, using Lagrange's and Kane's method, the computational efficiency of $O(n^3)$ and $O(n^2)$ were obtained.[7,28]

In some methods such as Newton's and Lagrange's, equations of motion appear as n second order differential equations. In some other methods such as Hamilton, Kane and Gibbs–Appell, equations are derived as $2n$ first order differential equations. In the case that system constraints are written separately, using Lagrange's multipliers, a Differential Algebraic Equation (DAE) set can be derived. DAEs are characterized by a differential index.[29] The formulations, which are based on acceleration, have an index of 3, where as the Hamiltonian, Kane and Gibbs–Appell formulations have an index 2.[30] As shown by Brenan *et al.*[31] index 2 DAE's have a better behavior during numerical integration. Due to using Gibbs–Appell method, the derived equations have an index of 2, which leads to decreasing the simulation error.

In this paper, we introduce a new planar structure for snake-like robots in order to obtain passive motion. Because of the sinusoidal change of relative angles between each two links in a Serpenoid curve, adding a torsional spring in the joints can induce passive serpenoid oscillations. We briefly explain kinematic model of the proposed snake-like robot and derive the kinematic equations and constraints. This robot has many nonholonomic constraints. Due to the advantages of Gibbs–Appell method in generating efficient

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

dynamic models for nonholonomic systems, we use this method to derive dynamic equations in horizontal plane. Finally, simulation results will be presented.

2. Model Description

In this paper, we use an n-link serial kinematic chain as a model of snake-like robot (Fig. 2).[7] Similar to Prautsch and Mita,[7] all links are assumed to be structurally identical. We further assume that, the robot moves in a horizontal plane. Locomotion of snake-like robots on slopes is studied by Ma and Tadokoro[32] The ith link denoted by L_i is assumed to have a length 2L and its center of gravity G_i placed at the middle. m and I represent mass and moment of inertia of each link about its center of gravity. Each link has a passive wheel in the middle, which is assumed to have no sideslip. Most snake-like robot's models ignore sideslip at the wheels.[7,11,28] S. Ma[33] has considered sideslip by modeling the coulomb friction between floor and the wheels. So, the velocity of each link is in the direction of that link. $(e_{xi},\ e_{yi})$ denote the coordinates attached to the ith link, θ_i is orientation of the ith link and ϕ_i is relative angle between ith and $(i+1)$th links:

$$\phi_i = \theta_{i+1} - \theta_i \quad i = 1, \ldots, n-1 \tag{1}$$

ϕ_0 and ϕ_n and their derivations are defined as:

$$\phi_0 = 0, \dot{\phi}_0 = 0, \phi_n = 0, \dot{\phi}_0 = 0 \tag{2}$$

These definitions will help us to simplify the final equations. As it can be seen in Fig. 3, we let $\dot{\eta}_i$ to represent the velocity of the center of each link in that link's direction. Note that using wheels in the middle of each link, adds a nonholonomic constraint to the system resulting in a zero lateral velocity of the link.

The joint between the L_i and L_{i+1} is denoted by P_i. A motor and a torsional spring are located in each joint (Fig. 4).

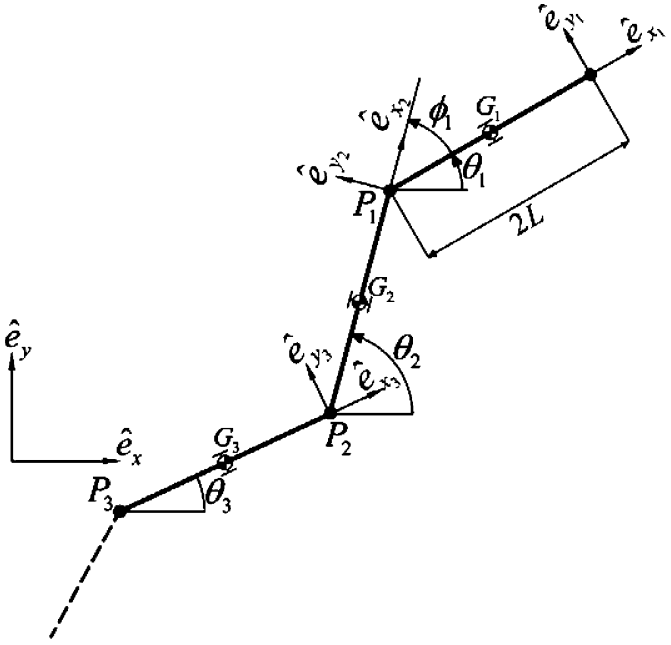


Fig. 2. Kinematic chain of the snake-like robot.

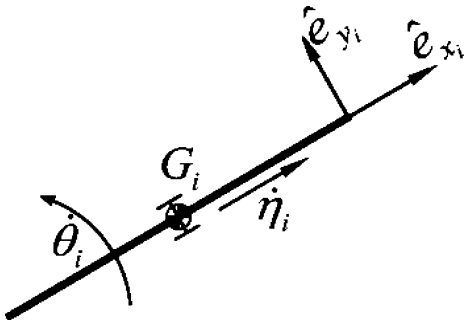


Fig. 3. Linear and angular velocities of L_i.

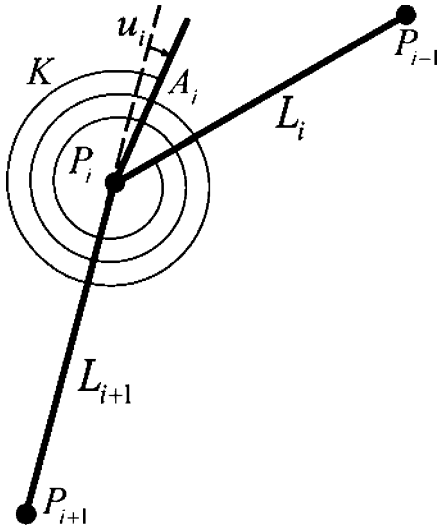


Fig. 4. Joint structure.

Note that the spring, whose stiffness is K, is located between links A_i and the ith link. The motor controls the relative angle u_i between $(i+1)$th link and A_i link. So the angular displacement of the spring is $(\phi_i - u_i)$ By controlling this angle we are able to control the torque between two adjacent links. In addition, this spring helps to attain passive motion of the robot. Friction in gearbox helps us fix the output shaft without consuming energy. This is very important as this friction let the robot to continue its passive motion without power consumption. If we use an ideal motor without any friction, power consumption will be equal to other structures which do not use torsional springs.

3. Kinematics of Robot

The main difference between Lagrange and Gibbs–Appell methods is in definition of coordinates. Lagrange's method is based on generalized coordinates and we are not allowed to use quasi-coordinates, however, in Gibbs–Appell method we may use quasi-coordinates. This type of coordinate selection will often be helpful in systems in which, we have nonholonomic constraints such as the snake-like robot as defined in the last section.

Our system has n links. As each two consecutive links are connected by a revolute joint P_i, there will be $n-1$ revolute joints. Including the associated n nonholonomic constraints,

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

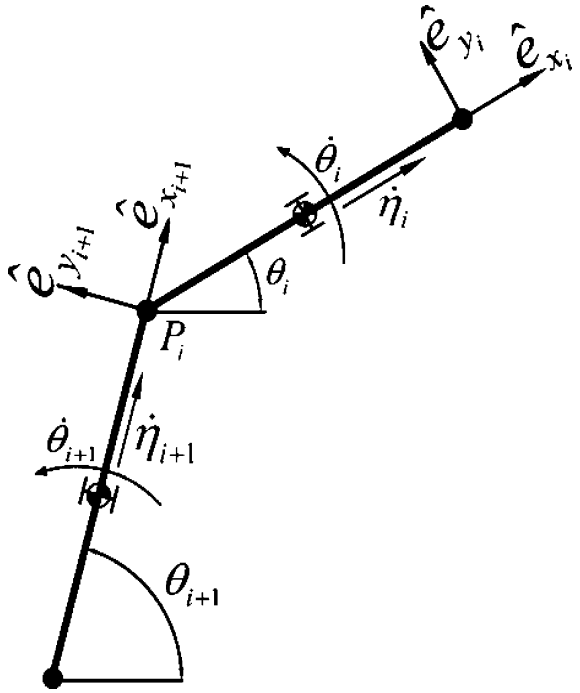


Fig. 5. Kinematic model of snake-like robot.

the system's degree of freedom can be calculated as below:

$$\text{DOF} = 3n - 2(n-1) - n = 2 \tag{3}$$

In this 2-DOF system, two independent general speeds are sufficient to define the speed of the system (Fig. 5). We select these two general speeds as

$$\dot{\mathbf{q}}_1 = \begin{pmatrix} \dot{\eta}_1 \\ \dot{\theta}_1 \end{pmatrix} \tag{4}$$

The other velocities can be defined with respect to these two speeds. We should note that $\dot{\eta}_i$, is a quasi-coordinate whose integral is not a generalized coordinate. By using $\dot{\eta}_i, \dot{\theta}_i$, all nonholonomic constraints are satisfied, since using these generalized velocities imply that the lateral speeds are zero.

$\dot{\eta}_{i+1}$ and $\dot{\theta}_{i+1}$ can be obtained as a function of $\dot{\eta}_i$ and $\dot{\theta}_i$. To obtain this relationship we can express velocity of the ith joint P_i with respect to each of the ith and $(i+1)$th coordinates. These two velocities must be equal. So, we have

$$\begin{aligned} \mathbf{V}_{P_i}|_{\hat{e}_{x_{i+1}}} &= \dot{\eta}_{i+1} = \dot{\eta}_i c\phi_i - L\dot{\theta}_i s\phi_i \\ \mathbf{V}_{P_i}|_{\hat{e}_{y_{i+1}}} &= L\dot{\theta}_{i+1} = -\dot{\eta}_i s\phi_i - L\dot{\theta}_i c\phi_i \end{aligned} \tag{5}$$

Equation (5) can be represented as:

$$\dot{\mathbf{q}}_{i+1} = {}^{i+1}_{i}A\dot{\mathbf{q}}_i \ (i = 1, \ldots, n-1) \tag{6}$$

where

$$\dot{\mathbf{q}}_i = \begin{pmatrix} \dot{\eta}_i \\ \dot{\theta}_i \end{pmatrix}, \quad {}^{i+1}_{i}A = \begin{pmatrix} c\phi_i & -Ls\phi_i \\ \frac{-1}{L}s\phi_i & -c\phi_i \end{pmatrix} \tag{7}$$

Equation (6) is a recursive equation, so, $\dot{\mathbf{q}}_i$ can be written as a function of $\dot{\mathbf{q}}_1$ as

$$\begin{aligned} \dot{\mathbf{q}}_i &= {}^{i}_{i-1}A\dot{\mathbf{q}}_{i-1} = {}^{i}_{i-1}A\,{}^{i-1}_{i-2}A\dot{\mathbf{q}}_{i-2} = \ldots \\ &= \left({}^{i}_{i-1}A\,{}^{i-1}_{i-2}A \ldots {}^{2}_{1}A\right)\dot{\mathbf{q}}_1 = {}^{i}_{1}A\dot{\mathbf{q}}_1 \end{aligned} \tag{8}$$

In Equation (8), ${}^{i}_{1}A$ can be simplified as:

$${}^{i}_{1}A = \begin{pmatrix} c\psi_i & -Ls\psi_i \\ \frac{(-1)^{i-1}}{L}s\psi_i & (-1)^{i-1}c\psi_i \end{pmatrix} \tag{9}$$

where

$$\psi_i = \sum_{\mathrm{k}=0}^{\mathrm{i}-1}(-1)^{\mathrm{k}+1}\phi_k i = 1, 2, \ldots, n \tag{10}$$

Equation (8), contains all kinematic velocity constraints of the robot. All constraints can be gathered in a unique equation as below:

$$A\dot{\mathbf{q}} = 0 \tag{11}$$

where, $\dot{\mathbf{q}}$ is the vector of generalized velocities and can be written as

$$\dot{\mathbf{q}} = \left(\dot{\mathbf{q}}_1^{\mathrm{T}}\dot{\mathbf{q}}_2^T \ldots \dot{\mathbf{q}}_n^T\right)^T \tag{12}$$

and A is the constrained matrix which is defined as

$$A = \begin{pmatrix} -{}^{2}_{1}A| & \\ \vdots\,| & I_{(2n-2)\times(2n-2)} \\ -{}^{n}_{1}A| & \end{pmatrix}_{(2n-2)\times 2n} \tag{13}$$

According to Eqs. (8) and (9), linear velocity of the center of mass and angular velocity of each link, as a function of generalized speeds, can be represented as

$$\begin{cases} \mathbf{V}_{G_i} = \dot{\eta}_i\hat{e}_{x_i} = (\dot{\eta}_1 c\psi_i - L\dot{\theta}_1 s\psi_i)\hat{e}_{x_i} \\ \boldsymbol{\omega}_i = \dot{\theta}_i\hat{e}_k = \left(\frac{(-1)^{i-1}}{L}\dot{\eta}_1 s\psi_i + (-1)^{i-1}\dot{\theta}_1 c\psi_i\right)\hat{e}_k \end{cases} \tag{14}$$

By differentiating Eq. (14), linear and angular accelerations could be obtained as

$$\begin{cases} \mathbf{a}_{G_i} = [\ddot{\eta}_1 c\psi_i - L\ddot{\theta}_1 s\psi_i - \dot{\psi}_i(\dot{\eta}_1 s\psi_i + L\dot{\theta}_1 c\psi_i)]\hat{e}_{x_i} \\ \qquad + \dot{\theta}_i(\dot{\eta}_1 c\psi_i - L\dot{\theta}_1 s\psi_i)\hat{e}_{y_i} \\ \boldsymbol{\alpha}_i = \left[\frac{(-1)^{i-1}}{L}\ddot{\eta}_1 s\psi_i + (-1)^{i-1}\ddot{\theta}_1 c\psi_i\right. \\ \qquad \left. + (-1)^i\dot{\psi}_i\left(u_2 s\psi_i - \frac{u_1 c\psi_i}{L}\right)\right]\hat{e}_k \end{cases} \tag{15}$$

4. Dynamic Modeling

In Gibbs–Appell method[18] we should obtain the acceleration energy of each link. Acceleration energy for each part is defined as

$$S_i = \frac{1}{2}m\,\mathbf{a}_{G_i}\cdot\mathbf{a}_{G_i} + \frac{1}{2}\boldsymbol{\alpha}_i\cdot\frac{\delta\mathbf{H}_{G_i}}{\delta t} + \boldsymbol{\alpha}_i\cdot(\boldsymbol{\omega}_i\times\mathbf{H}_{G_i}) \tag{16}$$

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

where, $\mathbf{H}_{G_i}$ is the angular momentum of each link about its center of gravity.

Now, we can rewrite Eq. (15) to obtain linear and angular accelerations for each link as follows:

$$\begin{cases} \mathbf{a}_{G_i} = \ddot{\eta}_i \hat{e}_{x_i} + \dot{\eta}_i \dot{\theta}_i \hat{e}_{y_i} \\ \boldsymbol{\alpha}_i = \ddot{\theta}_i \hat{e}_k \end{cases} \tag{17}$$

Angular momentum of each link can be represented as

$$\begin{aligned} \mathbf{H}_{G_i} &= I\dot{\theta}_i \hat{e}_k \\ \frac{\delta \mathbf{H}_{G_i}}{\delta t} &= I\ddot{\theta}_i \hat{e}_k \end{aligned} \tag{18}$$

By substituting Eqs. (14), (15), (17), and (18) into Eq. (16), we have

$$S_i = \frac{1}{2} m \ddot{\eta}_i^2 + \frac{1}{2} I \cdot \ddot{\theta}_i^2 + \text{irrelevant terms} \tag{19}$$

and S can be obtained as

$$S = \sum_{i=1}^{n} S_i = \frac{1}{2} m \sum_{i=1}^{n} \ddot{\eta}_i^2 + \frac{1}{2} I \sum_{i=1}^{n} \ddot{\theta}_i^2 + \text{irrelevant terms} \tag{20}$$

Irrelevant terms, are the terms, which are independent of acceleration. Since, our derivation is based on taking derivatives of S with respect to acceleration, terms independent of acceleration can be omitted.

Total work done by each link is the sum of conservative and nonconservative works:

$$\delta W = \delta W_{c.} + \delta W_{n.c.} \tag{21}$$

It is known that

$$\delta W_{c.} = -\left(\frac{\partial \mathbf{V}}{\partial \mathbf{q}}\right)^{\mathrm{T}} \cdot \delta \mathbf{q} \tag{22}$$

where V is the Potential energy and is defined as:

$$V = \frac{1}{2} k \sum_{i=1}^{n-1} (\phi_i - u_i)^2 \tag{23}$$

and

$$\delta \mathbf{q} = \begin{pmatrix} \delta \mathbf{q}_1 \\ \vdots \\ \delta \mathbf{q}_{\mathrm{n}} \end{pmatrix}; \quad \delta \mathbf{q}_{\mathrm{i}} = \begin{pmatrix} \delta \eta_{\mathrm{i}} \\ \delta \theta_{\mathrm{i}} \end{pmatrix} \tag{24}$$

By substituting Eqs. (23) and (24) into Eq. (22) and taking note of Eq. (1), we have:

$$\begin{aligned} \delta W_{c.} = {} & [k(\phi_1 - u_1)]\delta\theta_1 + [k((\phi_2 - \phi_1) + (u_1 - u_2))]\delta\theta_2 \\ & + \cdots + [k((\phi_{n-1} - \phi_{n-2}) + (u_{n-2} - u_{n-1}))]\delta\theta_{\mathrm{n-1}} \\ & + [k(-\phi_{n-1} + u_{n-1})]\delta\theta_{\mathrm{n}} \end{aligned} \tag{25}$$

Now, Eq. (25) can be written in the form of

$$\delta W_{c.} = (\mathrm{k}\boldsymbol{\Phi} + \mathrm{B}\mathbf{U})^{\mathrm{T}} \delta \mathbf{q} \tag{26}$$

where

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_n \end{pmatrix}; \boldsymbol{\Phi}_i = \begin{pmatrix} 0 \\ \phi_i - \phi_{i-1} \end{pmatrix} \mathrm{i} = 1, \ldots, n \tag{27}$$

$$\mathbf{U} = \begin{pmatrix} \mathrm{u}_1 \\ \mathrm{u}_2 \\ \vdots \\ \mathrm{u}_{\mathrm{n-1}} \end{pmatrix} \tag{28}$$

$$\mathrm{B} = \mathrm{k} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}_{2n \times (n-1)} \tag{29}$$

$\mathbf{U}$ is the vector of relative angles between each link and the link A_i as mentioned in (Fig. 4).

$\delta W_{n.c.}$can be written as

$$\delta W_{n.c.} = \sum_{i=1}^{n} [-\mu_1 \dot{\theta}_i - \mu_2 (\dot{\phi}_{i-1} - \dot{\phi}_i)]\delta\theta_i \tag{30}$$

where, μ_1 and μ_2 represent the viscous coefficients of friction at the wheels and rotary joints between individual links.

So, Eq. (30) can be represented as

$$\delta W_{n.c.} = (-\mu_1 \dot{\boldsymbol{\Theta}} + \mu_2 \dot{\boldsymbol{\Phi}})\delta \mathbf{q} \tag{31}$$

where,

$$\dot{\boldsymbol{\Theta}} = \begin{pmatrix} \begin{pmatrix} 0 \\ \dot{\theta}_1 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} 0 \\ \dot{\theta}_n \end{pmatrix} \end{pmatrix} \tag{32}$$

By substituting Eqs. (26) and (31) into Eq. (21) δW is obtained as

$$\delta W = \boldsymbol{\Gamma}^T \delta \mathbf{q} \tag{33}$$

where,

$$\boldsymbol{\Gamma} = -\mu_1 \dot{\boldsymbol{\Theta}} + \mu_2 \dot{\boldsymbol{\Phi}} + \mathrm{K}\boldsymbol{\Phi} + B\mathbf{U} \tag{34}$$

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

In Gibbs–Appell method, we have

$$\frac{\partial S}{\partial \ddot{\mathbf{q}}} = \mathbf{\Gamma} + \mathrm{A}^{\mathrm{T}}\boldsymbol{\lambda} \tag{35}$$

where $\boldsymbol{\lambda}$ is the Lagrange Multiplier vector. From Eq. 20, one can conclude as

$$\frac{\partial S}{\partial \ddot{\mathbf{q}}} = M\ddot{\mathbf{q}} \tag{36}$$

where,

$$M = \begin{pmatrix} M_1 \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_n \end{pmatrix} \tag{37}$$

$$M_i = \begin{pmatrix} m & 0 \\ 0 & I \end{pmatrix} \quad (i = 1, \dots, n) \tag{38}$$

Hence, using Eqs. (34), (35), and (36) dynamic equation of the whole robot can be written as

$$M\ddot{q} + \mathbf{N} = B\mathbf{U} + A^T\boldsymbol{\lambda} \tag{39}$$

where,

$$\mathbf{N} = -K\mathbf{\Phi} + \mu_1\dot{\mathbf{\Theta}} - \mu_2\dot{\mathbf{\Phi}} \tag{40}$$

5. Reduced Dynamic Equations

We have a 2-DOF planar system. So, 2 differential equations suffice to represents the dynamic model of the system. We can obtain all the generalized velocities in terms of $\dot{\mathbf{q}}_1$. In the Eq. (39), we have $4n-2$ variables ($2n-2$, $\boldsymbol{\lambda}$s and $2n$, acceleration variables) and $4n-2$ equations ($2n$, differential equations and $2n-2$ constraint equations). These differential equations and algebraic equations build up a Differential Algebraic Equation (DAE). However, the $\boldsymbol{\lambda}$s will be omitted and all of the acceleration variables will be rewritten as a function of $\dot{\mathbf{q}}_1$, using the following method.

We can divide $\mathbf{q}$ into two parts as

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_\mathbf{r} \end{bmatrix} \tag{41}$$

where,

$$\dot{\mathbf{q}}_\mathbf{r} = (\dot{\mathbf{q}}_\mathbf{2}, \dots, \dot{\mathbf{q}}_\mathbf{n})^T \tag{42}$$

We can rew Eq. (11) as

$$\begin{pmatrix} \dot{\mathbf{q}}_\mathbf{2} \\ \vdots \\ \dot{\mathbf{q}}_\mathbf{n} \end{pmatrix} = \begin{pmatrix} {}^2_1A \\ \vdots \\ {}^n_1A \end{pmatrix} \dot{\mathbf{q}}_1 \tag{43}$$

Eq. (43) can be represented as

$$\dot{\mathbf{q}} = C\dot{\mathbf{q}}_1 \tag{44}$$

where,

$$C = \begin{pmatrix} I_{2\times 2} \\ {}^2_1A \\ \vdots \\ {}^n_1A \end{pmatrix} \tag{45}$$

Now, we can use Eq. (44) to obtain $\ddot{\mathbf{q}}$ as a function of $\ddot{\mathbf{q}}_1$ and $\dot{\mathbf{q}}_1$

$$\ddot{\mathbf{q}} = \dot{C}\dot{\mathbf{q}}_1 + C\ddot{\mathbf{q}}_1 \tag{46}$$

where,

$$\dot{C} = \begin{pmatrix} 0_{2\times 2} \\ {}^2_1\dot{A} \\ \vdots \\ {}^n_1\dot{A} \end{pmatrix}, \quad {}^k_1\dot{A} = \dot{\psi}_k \begin{pmatrix} -s\psi_k & -Lc\psi_k \\ \frac{(-1)^{k-1}}{L}c\psi_k & (-1)^k s\psi_k \end{pmatrix} \tag{47}$$

We have

$$AC = \begin{pmatrix} -{}^2_1A| & \\ \vdots| & I_{(2n-2)\times(2n-2)} \\ -{}^n_1A| & \end{pmatrix}_{(2n-2)\times 2n} \times \begin{pmatrix} I_{2\times 2} \\ {}^2_1A \\ \vdots \\ {}^n_1A \end{pmatrix}_{2n\times 2} \tag{48}$$

$$AC = \begin{pmatrix} -{}^2_1A \times I_{2\times 2} + I_{2\times 2} \times {}^2_1A \\ -{}^3_1A \times I_{2\times 2} + I_{2\times 2} \times {}^3_1A \\ \vdots \\ -{}^n_1A \times I_{2\times 2} + I_{2\times 2} \times {}^n_1A \end{pmatrix}_{(2n-2)\times 2} \tag{49}$$

So, it is easily proved that

$$AC = 0 \tag{50}$$

Now, by multiplying Eq. (39) by C^T and substituting from Eqs. (46) and (50), $\boldsymbol{\lambda}$ will be eliminated and we can write the resulting equation as

$$\tilde{M}\ddot{\mathbf{q}}_1 + \tilde{\mathbf{N}} = \tilde{B}\mathbf{U} \tag{51}$$

where,

$$\begin{aligned} \tilde{M} &= C^T M C \\ \tilde{\mathbf{N}} &= C^T(M\dot{C}\dot{\mathbf{q}}_1 + \mathbf{N}) = (C^T M\dot{C})\dot{q}_1 + C^T\mathbf{N} \\ \tilde{B} &= C^T B \end{aligned} \tag{52}$$

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

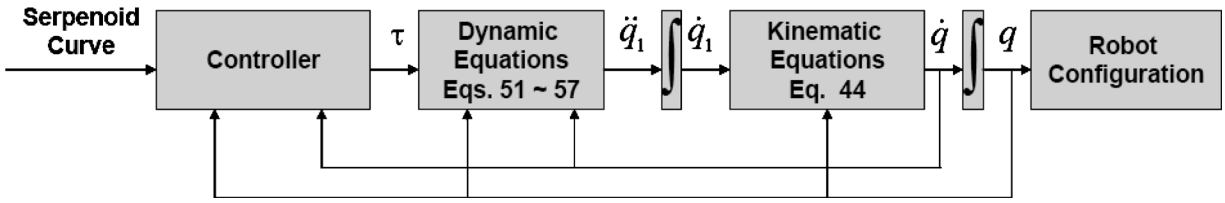


Fig. 6. Simulation algorithm.

By substituting Eqs. (45) and (37) into Eq. (52), we have

$$\tilde{M} = \sum_{i=1}^{n} {}_1^i A^T M_i {}_1^i A \tag{53}$$

$$\tilde{\mathbf{N}} = \sum_{i=1}^{n} \left({}_1^i A^T M_i {}_1^i \dot{A}\right)\dot{\mathbf{q}}_1 + \mathrm{C}^{\mathrm{T}}\mathbf{N} \tag{54}$$

Finally, by substituting Eq. (9) into Eqs. (53) and (54), $\tilde{M}$ and $\tilde{N}$ can be obtained as

$$\tilde{M} = \begin{pmatrix} m\alpha + \frac{I}{L^2}(n-\alpha) & \left(\frac{I}{L} - mL\right)\beta \\ \left(\frac{I}{L} - mL\right)\beta & I\alpha + mL^2(n-\alpha) \end{pmatrix} \tag{55}$$

$$\tilde{\mathbf{N}} = \left[\sum_{k=1}^{n} \dot{\psi}_k \begin{pmatrix} \left(\frac{I}{L^2} - m\right) s\psi_k c\psi_k & -\left(mLc^2\psi_k + \frac{I}{L}s^2\psi_k\right) \\ mLs^2\psi_k + \frac{I}{L}c^2\psi_k & -(I - mL^2)s\psi_k c\psi_k \end{pmatrix}\right] \times \dot{\mathbf{q}}_1 + C^T\mathbf{N} \tag{56}$$

where α and β are defined as follows:

$$\begin{aligned} \alpha &= \sum_{k=1}^{n} c^2\psi_k \\ \beta &= \sum_{k=1}^{n} s\psi_k c\psi_k \end{aligned} \tag{57}$$

As it can be seen, the mass matrix can be simplified to a 2×2 matrix as above.

6. Serpenoid Curve

Assuming that snakes develop an efficient way of creeping, Hirose[3] by looking back to the nature have proposed the Serpenoid curves as a gait for snake-like robot's locomotion.

In order to generalize the continuous Serpenoid curve, for a snake-like robot, this curve should be divided into n parts. By assuming n links for the robot, the relative angle between two adjacent links, $\phi_i(t)$, can be obtained as

$$\phi_i(t) = \xi\, Sin(\omega t + (i-1)\zeta) + \gamma_i \quad (\mathrm{i} = 1, \ldots, \mathrm{n}-1) \tag{58}$$

in which, ξ is the amplitude of oscillation, ζ is the phase difference, γ determines the radius of the circular path and ω is the speed that the Serpenoid curve propagates along the body. ω and γ, respectively, specify the velocity and direction of the snake's motion. If $\gamma = 0$, then robots path will be a straight line. Otherwise, robot turns right or left through its movement.

It can be shown that the Serpenoid curve, best resembles snake's motion, among the curves which are defined for this purpose.[3] In this efficient gait, as it can be seen in Eq. (58), angles between two links vary sinusoidally. Therefore, it seems that passive motion generated by using torsional springs between each two consecutive links could result in a serpenoid gait.

7. Simulation Study

Simulation algorithm block diagram is shown in Fig. 6. As it can be seen, $\ddot{q}_1$ can be obtained from reduced dynamics equations (Eqs. (51)–(57)). By integrating this value, $\dot{\boldsymbol{q}}_1$ will be obtained. Then, by using constraint equations all velocities ($\dot{\boldsymbol{q}}$) can be computed. By integrating $\dot{\boldsymbol{q}}$, robot position is specified and robot configuration is obtained in each step. Matlab's Simulink toolbox is used for simulation.

$\boldsymbol{U}$ can be defined as a PD control law as below:

$$\boldsymbol{U} = k_p(\phi_{id} - \phi_i) + k_v(\dot{\phi}_{id} - \dot{\phi}_i) \tag{59}$$

where, k_p and k_v are constants and ϕ_{id} and $\dot{\phi}_{id}$ are desired angle and angular velocities of joints, which are obtained from Serpenoid curve as

$$\phi_{id} = \xi\, \sin(\omega t + i\beta) + \gamma_i \tag{60}$$

$$\dot{\phi}_{id} = \xi\omega \cos(\omega t + i\beta) \tag{61}$$

During a passive motion and without a controller, k_p and k_v vanish. Robot parameters in the simulation are listed in Table I. As it can be seen from Eq. (58) as the controller gains vanish, the motor's output shafts are fixed and robot's motion becomes passive.

First, we assume a four link snake-like robot with the initial conditions in Table II. Initial conditions for joint

Table I. Robot parameters.

m (kg)	I (Kg·m^2)	L (m)	μ_1	μ_2	K (m/rad)	k_p	k_v
1.15	0. 003614	0.185	0	0	0.0955	0	0

Table II. Initial conditions for a four link snake-like robot.

θ_1	$x(0)$	$y(0)$	ϕ_i	$\dot{\eta}_1(m/s)$	$\dot{\theta}_1(rad/s)$
0	0	0	$\frac{\pi}{12}\sin(i\frac{\pi}{12})$ $(i = 1, 2, 3)$	.4	1.5

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

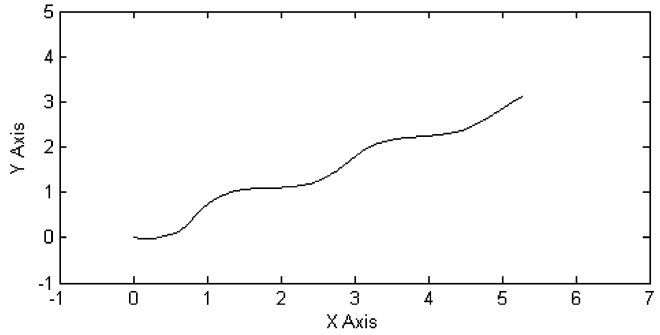


Fig. 7. Head path in $x - y$ plane for a 4-link robot.

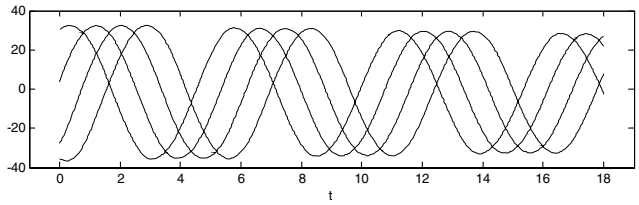


Fig. 8. ϕ_i for passive motion of a 4-link robot.

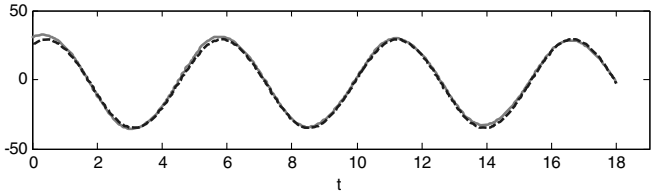


Fig. 9. Comparison between a Serpenoid curve and ϕ_1 of passive motion for 4-link robot (dashed curve is Serpenoid curve).

angles are calculated, using Serpenoid curve. Linear and angular velocities are chosen to result in a Serpenoid passive motion. Here, we use random search method to find these initial conditions, but in the future works we shall present an analytical method to find these initial conditions.

Figures 7 and 8 show head path in the x-y plane and the joint angles, respectively. As it can be seen, joint angles operate in sustained oscillation. In other words, robot remains in a limit cycle and continue this motion, indefinitely.

Now, we may ask whether this oscillating motion follows a Serpenoid curve. In Fig. 9, we compare ϕ_1, which is obtained from the passive motion with a sinusoidal wave, indicating a near sinusoidal curve. Since, joint angles in passive motion follow sinusoidal behavior with appropriate phase shift, one may conclude that passive motion of the robot (with appropriate initial conditions) is a Serpenoid gait.

Next, simulation is repeated for a 12-Link snake-like robot with the initial conditions in Table III. Figures 10 and 11 show

Table III. Initial conditions for a 12-link snake-like robot.

θ_1	$x(0)$	$y(0)$	ϕ_i	$\dot{\eta}_1(m/s)$	$\dot{\theta}_1$ (rad/s)
0	0	0	$\frac{\pi}{12}\sin(i\frac{\pi}{12})(i = 1, 2, \ldots, 11)$	0.42	0.5

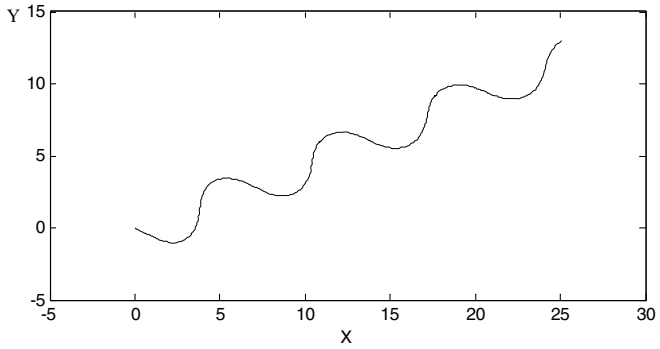


Fig. 10. Head path in $x - y$ plane for a 12-link robot.

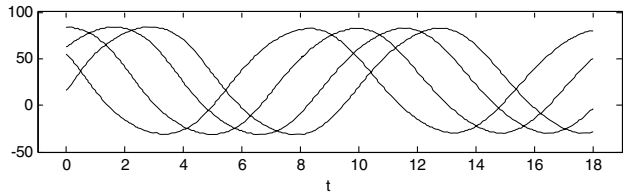


Fig. 11. ϕ_1, ϕ_5, ϕ_9, and ϕ_{11} of passive motion for a 12-link robot.

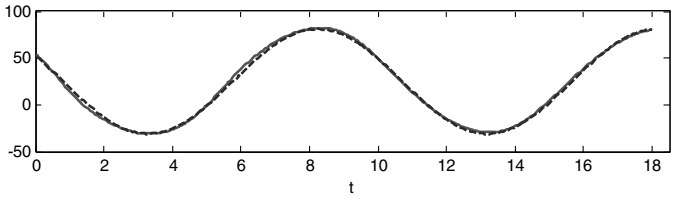


Fig. 12. Comparison between a Serpenoid curve and ϕ_1 of passive motion for 12-link robot (dashed curve is serpenoid curve).

the head path in $x - y$ plane and joint angles, respectively. Figure 12 compares a sinusoidal wave withϕ_1, the first joint angle. Figure 13 shows error between θ of an ideal serpenoid gait and θ_1 of a 12-link robot. Serpenoid gait is a periodic motion in which joint angles are vary sinusoidally. From Fig. 12 and 13, we may conclude that the joint angles, in snake-like robot, are varying periodically, and converges to a stable periodic orbit.

It is interesting to note that if, for example, the same initial joint angles are used for a 12-link robot, where

$$\dot{\eta}_1 = 1\text{m/s},\ \dot{\theta}_1 = 0.8\,\text{rad/s} \tag{62}$$

It can be seen that the head path does not follow a Serpenoid curve (Fig. 14).

8. The Comparative Discussions

1. Number of Operations
   In Table IV the total number of mathematical operations required in each step for an n-link robot using Gibbs–Appell, Kane and Lagrange methods[25] are compared.
   The exact values for the number of operations in a 4, 12, and 100 link robot can be seen in Table V.
2. Simulation Time
   Another important issue is comparing simulation time in Lagrange and Gibbs–Appell methods. In Table VI, the

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

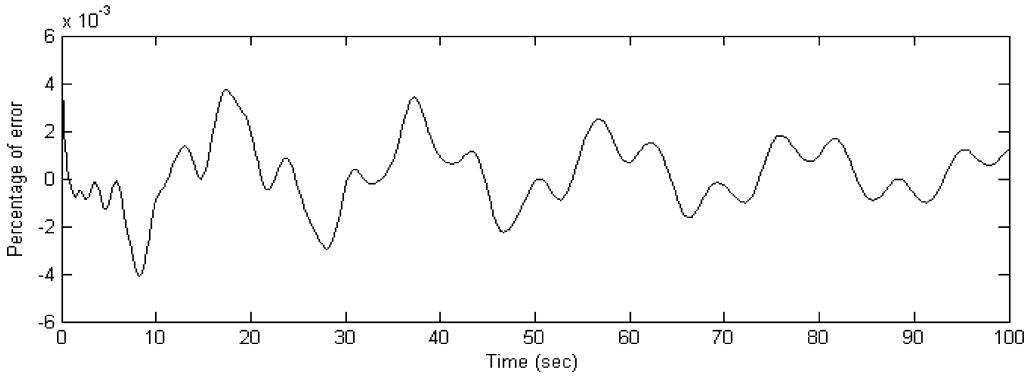


Fig. 13. Difference (error) between a serpenoid curve and ϕ_1 of passive motion for 12-link robot.

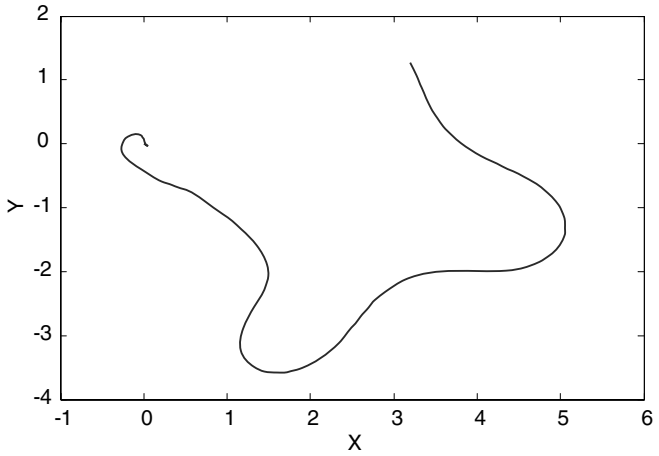


Fig. 14. Head path for a random initial condition.

Table IV. Comparison of total number of mathematical operations for an n-link Snake-Like robot.

Method	Appell Gibbs–	Kane[25]	Lagrange[25]
$+/-$	$30n+3$	$15/2n^2+97/2n-21$	$3/2n^3+11n^2+31/2n-2$
$\times/\div$	$49n+28$	$3n^2+24n-27$	$3n^3+31/2n^2+39/2n-11$
sin/cos	$11n-9$	------*	------*

*Number of these operations is not mentioned.[25]

required time for a 4, 12, and 100 link robot in a 200 second simulation, is compared for Gibbs–Appell and Lagrange method.[28] Table VI shows that Gibbs–Appell method is by far a more efficient algorithm for simulating snake-like robots.

Table VI. Required time for 200 s Simulation on a 2.4GHz Pentium IV.

No. of links Method	4	12	100
Gibbs–Appell	10 s	59 s	4150 s
Lagrange	6 min	15 min	*

*Because simulation of a 100-link robot using Lagrange method is extremely time-consuming, we neglect to measure the required time.

9. Conclusion

We have introduced a new structure for snake-like robots. Using spring in series with angular displacement actuator, passive motion can be obtained. As it can be seen in the "Simulation Study," initial conditions can be found, from which if the robot is released, it will have a steady motion without any actuation. In such a case, joint angles remain in a limit cycle and approximately vary sinusoidally, resulting in a Serpenoid motion. If the robot deviates from passive motion trajectory due to friction or external disturbance, actuators can return it to the passive trajectory. Locomotion of the robot on the passive trajectories reduces power consumption especially during high speed maneuvers.

In this paper, we have used Gibbs–Appell method to derive the dynamic equations, for an n-link snake-like robot. By simplifying the equations, the order of computations for simulation is reduced to $O(n)$, which is outstanding when compared to other works which are at best of $O(n^2)$. Using these equations, it will be possible to simulate snake-like robots with large number of links, with minimal computational effort. In the assumed model of the robot, there are numerous nonholonomic constraints. Methods such as Gibbs–Appell and Kane which are based on generalized velocities or generalized speeds deal with systems with nonholonomic constraints much more efficiently. In these methods, we may use quasi-coordinates. By using quasi-coordinates, nonholonomic constraints may be omitted at the

Table V. Exact values of the number of operations for a four, twelve and hundred-link robot.

Method	Gibbs–Appell			Kane[25]			Lagrange[25]		
Operator	$+/-$	$\times/\div$	sin/cos	$+/-$	$\times/\div$	sin/cos	$+/-$	$\times/\div$	sin/cos
$n=4$	123	219	35	293	117	–	332	507	–
$n=12$	363	611	255	1641	693	–	4360	7639	–
$n=100$	3003	4923	1091	79829	32373	–	1611548	3156939	–

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press

beginning and therefore, dynamic equations' complexity and simulation time is significantly reduced. It is obvious that one may obtain the same dynamic equations using other methods. However, we should emphasize different modeling methods and also selection of generalized or quasi coordinates directly affect the simplicity of equations.

Some of the issues that required further investigations are:

1. Validation of the simulation results on the real robot.
2. Designing an optimal controller to generate and follow passive trajectories at different speeds and paths.
3. Finding initial conditions for achieving passive trajectories' and controlling the robot to follow these trajectories under more realistic and un-ideal environmental conditions.

References

1. J. Gray, *Animal Locomotion* (Norton, New York, 1993) pp. 166–193.
2. S. Hirose, Biomechanical Engineering (Kougyou Tyousa Kai, Tokyo, Japan, 1987).
3. S. Hirose, Biologically Inspired Robots: Snake-like Locomotor and Manipulator (Oxford University Press, New York, 1993).
4. O. Takanashi, K. Aoki and S. Yashima, "A Gait Control for the Hyper-Redundant Robot O-RO-CHI," *Proceedings of the ROBOMECÕ96*, (Ube, Japan, 1996) pp. 78–80.
5. R. Worst and R. Linnemann, "Construction and Operation of a Snake-Like robot," *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*, (Los Alamitos, CA, IEEE Press, 1996) pp. 164–169.
6. M. Mori and S. Hirose, "Development of active cord mechanism ACM-R3 with agile 3D mobility," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Outrigger Wailea Resort, Maui, Hawaii, USA, IEEE Industrial Electronics Society, 2001) pp. 1552–1557.
7. P. Prautsch and Mita, "Control and Analysis of the Gait of Snake Robots", *Proceedings of IEEE International Conference on Control Applications*, (Hawaii, 1999) pp. 502–507.
8. J. Ute and K. Ono, "Fast and Efficient Locomotion of a Snake Robot Based on Self-Excitation Principle," *IEEE Proceedings of AMC*, (Maribor, Slovenia, 2002) pp. 532–539.
9. Chirikjian, G. S. "Hyper-Redundant Manipulators Dynamics—A Continuum Approximation," *Adv. Robot.*, **9**(3), 217–243 (1995).
10. B. A. Jones and I. D. Walker, "Kinematics for multi-section continuum robots," *IEEE Trans Robot*. **22**(1), 43–55 (2006).
11. M. Saito, M. Fukaya and T. Iwasaki, "Serpentine Locomotion with Robotic Snakes," *IEEE Contr. Syst. Mag.* **22**(1), 64–81 (2002).
12. T. A. McMahon, "The role of compliance in mammalian running gaits," *J. Exp. Biology* **115**(11), 263–282 (1985).
13. M. Ahmadi, Stable Control of a One-Legged Robot Exploiting Passive Dynamics *Ph.D. Thesis* Mechanical Engineering Department, McGill University, 1998).
14. H. S. Seifert, "The lunar pogo stick," *J. Spacecraft*, **4**(7), 914–943 (1967).
15. T. McGeer, "Passive dynamic walking" *Int. J. Robot Res.* **9**(2), 62–82 (1990).
16. M. H. Raibert and C. M. Thompson, "Passive Dynamic Running," **In:** Experimental Robotics I (V. Hayward and O. Khatib, eds.) (Springer-Verlag NY, 1989) pp. 74–83.
17. R. Featherstone, *Robot Dynamics Algorithms* (Kluwer Academic Publishers, Nor-Well, MA, 1987).
18. J. H. Ginsberg, *Advanced Engineering Dynamics*, (New York, Cambridge University Press, 2nd ed. 1998).
19. T. R. Kane and D. A. Levinson, *Dynamics: Theory and Applications* (New York, McGraw-Hill 1985).
20. D. E. "Rosenthal, An Order n Formulation for Robotic Systems," *J. Astronaut. Sci.* **38**(4) 511–529 (1990).
21. M. K. Vukobratovi'c, V. F. Filaretov and A. I. Korzun, "A Unified Approach to Mathematical Modeling of Robotic Manipulator Dynamics," Robotica **12** (5), 411–420 (1994).
22. W. Jerkovsky, "The Structure of Multibody Dynamics Equations," *J. Guid. Cont.*, **1**(3) 173–182 (1978).
23. D. Baraff, "Linear-time dynamics using Lagrange multipliers," *Proc. Annu. Conf. Comp. Graph. Interact. Techn.* **30** 137–146 (ACM Press, New York, 1996).
24. K. S. Anderson and J. H. Critchley, "Improved 'Order-N' performance algorithm for the simulation of constrained multi-rigid-body dynamic systems," *Multibody Syst. Dyn.* **9**(2), pp. 185–212 (Springer 2003).
25. K. Desoyer and P. Lugner, "Recursive formulation for the analytical or numerical application of the Gibbs–Appell method to the dynamics of robots," *Robotica* **7** (4) 343–347 (1989).
26. I. Rudas and A. Toth, "Efficient recursive algorithm for inverse dynamics," Mechatronics **3**(2), 205–214 (1993).
27. V. Mata, S. Provenzano, J. I. Cuadrado and F. Valero, "An O(n) Algorithm for Solving the Inverse Dynamic Problem in Robots by Using the Gibbs–Appell Formulation," *Proceedings of* 10^{th} *World Congress on Theory of Machines and Mechanisms*, Oulu, Finland (XXX, 1999) pp. 1208–1215.
28. H. Tavakoli Nia, H. Nejat Pishkenari and A. Meghdari, "A Recursive Approach for Analysis of Snake Robots Using Kane's Equations," *Proceedings of IDETC/CIE* (Long Beach, CA, 2005).
29. J. Naudet, D. Lefeber and Z. Terze, "General Formulation of an Efficient Recursive Algorithm Based On Canonical Momenta for Forward Dynamics of Open-loop Multibody Systems," *IDMEC/IST*, Lisbon, Portugal (Jul. 1–4, 2003).
30. J. Garc'ıa de Jal'on and E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge (Springer-Verlag), 1994.
31. K. E. Brenan, S. L. Campbell and L. R. Petzold, "The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations," (Elsevier Science Publishing New York, 1989).
32. S. Ma and N. Tadokoro, "Analysis of Creeping Locomotion of a Snake-like Robot on a Slope," *J. Autonom. Robots* **20** (1), 15–23 (2006).
33. S. Ma, "Analysis of creeping locomotion of a snake-like robot," *Advan. Robot.* **15** (2), 205–224 (2001).

https://doi.org/10.1017/S0263574707003864 Published online by Cambridge University Press