

Development of Rectilinear Path for a Limbless Robot

Saurabh .A. Ban¹., Sachin .P. Patel²

(¹ Research scholar S.P.C.E Visnagar, ² Asst. Professor Government Engg. Collage Patan.)

Abstract: This paper describes the research carried out on limbless robot and the results of experiments conducted with wheel-less snake inspired robots. This develops a kinematic and dynamic model for rectilinear path. A model using Eulerian framework and coulomb friction yields torque expressions for the joint of the robot. Results are shown for the experiments on trajectories.

Key Words: Limbless locomotion, rectilinear path, Snake inspired, biologically inspired methods.

I. INTRODUCTION

Limbless locomotion has recently motivated the construction of a number of mobile machines taking advantage of the physical phenomena observed particularly with snakes and inchworms. Original works of Hirose [1-3] has inspired many of the following research on snake-like locomotion throughout the world [4]. Most of these locomotors consist of mobile platforms equipped with active or passive wheels. Despite their snake-like mechanical structures, locomotion of machines using active wheels is based on the principles of classical wheeled functioning. Most structures with passive wheels have been designed to exert biologically inspired snake-like locomotion [1-5]. However wheeled snake robots are intended to operate over perfectly smooth substrata, i.e. in artificially structured environments.

In addition wheeled motion does not appear in natural world. Rectilinear locomotion is a form of locomotion that is common to large snakes with well-developed muscles such as boas and pythons [8]. This form of movement is distinct from the other forms of locomotion in that the snake progresses with its body fully aligned with the direction of movement.

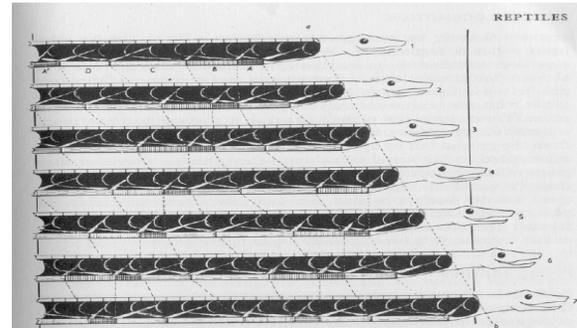


Figure1 . Rectilinear Motion. Progression of snake in a rectilinear path.

Movement is achieved by waves of muscular contraction and expansion passing along the body of the snake. This form of locomotion is best understood by imagining two points located on the ventral (bottom) surface of the snake. With the waves of muscular contraction and expansion, the distance between the two points is oscillating. When the distance between the two points is at a minimum, that segment is at rest. When the distance between the points is either increasing or decreasing, the moving point is moving forward. This is achieved by the frictional characteristics between the snake and the surface, and can be thought of as a “ratcheting” action. The points on the ventral surface move forward in discrete steps. However, the top of the body moves continuously because of the changing geometry of the muscular segments.

The second part describes the kinematic and dynamic model. In the third part, parametric analysis is carried out. The results of experiments are given in the forth part. In the fifth and last part, discussion on the results is carried out.

II. Kinematic and Dynamic models

This section presents a formulation of both the kinematics and dynamics of the model. This is necessary because a dynamics model must be created in order to determine the amount of torque required to actuate the robot. More importantly, the dynamics model can be used to evaluate metrics that determine the performance of the robot. The value in the kinematics model is that it is both necessary to develop the dynamics model, and also can be used to verify the feasibility of the path.

In this Section the path was divided into mechanisms with different topologies. The kinematics and dynamics model is constructed for each mechanism as shown in fig.2. The example of M1 is used to show how the kinematics and dynamics models are developed, and then it is shown how this approach can be expanded and applied to M2. Finally, it is shown how the solutions to the steps and mechanisms can be mapped back to the overall mechanism to achieve one set of global solutions.

A. Path kinematics

Consider the first mechanism in the locomotion path, M1. The links of the robot must move from the flat shape. This motion can be described by three parameters in the joint space: the angular positions of joints 1, 2, and 3. It can also be seen that the feasible joint space is constrained to a region that amounts to a half-space due to the presence of the ground.

The first way to model the sub-problem is to treat it as a free three-link system with three degrees of freedom. In order to ensure that the configuration remains in the feasible joint space, the y-positions of each of the joints and the “tip” of the third link are calculated. If the position of either of the joints is equal to or less than zero, a penalty function is assessed in the fitness function.

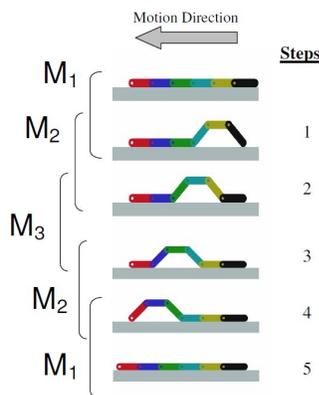


Figure 2 Modified traveling wave path, steps and mechanisms are identified

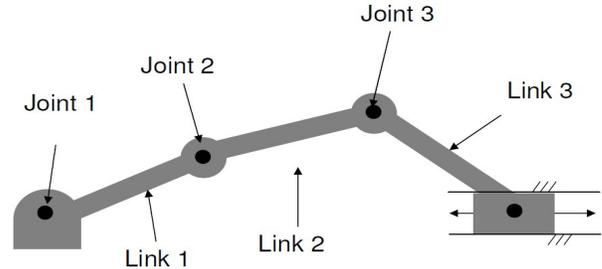


Figure 3. Representation of the first mechanism as a closed-loop, planar, kinematic chain

The second means of modeling the sub-problem is to assume that the “tip” of the third link remains in constant contact with the ground during the motion. In this instance, the problem can be modeled as a kinematic chain with four rotating joints and a slider mechanism that is shown in Figure 3. In this case, the geometric relations involve an equality constraint, as the y-position of the “tip” of link 3 always remains at $y=0$.

In this case, the kinematic configuration of the mechanism is governed by a constraint relationship. The mechanism is converted from one with three-degrees of freedom to one with two degrees of freedom. In this system, the angles of joints 1 and 2 are treated as free variables, and the position of joint 3 is calculated using the following constraint relation:

$$\sin \theta_1 + l_1 \sin (\theta_1 + \theta_2) + l_2 \sin (\theta_1 + \theta_2 + \theta_3) = 0 \dots \dots \dots (i)$$

The parameter f can be introduced to simplify the notation:

$$\Phi_1 = \sum \theta_j \dots \dots \dots (ii)$$

The equation then reduces to:

$$\sin \theta_1 + l_1 \sin \theta_2 + l_2 \sin \theta_3 = 0 \dots \dots \dots (iii)$$

The angle of link 3 can then be calculated by the solving the equation to yield:

$$\theta_3 = \sin^{-1} (-\sin \theta_1 - \sin \theta_2) \dots \dots \dots (iv)$$

The only constraints on the angles of joints 1 and 2 are that they must satisfy the aforementioned criteria

of not resulting in any Cartesian positions below $y=0$, and that:

$$\sin \square_1 + \sin (\square_2) \leq 1 \dots\dots\dots(v)$$

These constraints define the feasible design space of the problem. Like the previous constraint criteria, a penalty is assessed in the fitness formulation if this constraint is not met.

B. Dynamics

The dynamic problem of the path design involves computing which torque values are needed to achieve the desired joint motion. The dynamics have already been developed for the case of the open-loop configuration, as they only involve simple inward-outward propagation. This formulation depends on the manipulator moving in free space. Since there is a frictional force that acts on the “tip” of the third link, a Coulomb friction model represents this force. This means that the force can be calculated using the equation:

$$F_{fr} = -\mu F_R \text{sign} ({}^0v_{3tip}) \dots(vi)$$

The term ${}^0v_{3tip}$ denotes the velocity of the “tip” of the third link, and F_R is the vertical reaction force between the ground and the link. The “sign” denotes the signum function, meaning that the frictional force is always acting in a direction opposite to the velocity of the link. Fig.4 shows the free body diagrams for mechanism M1 in the closed loop configuration. The mechanism consists of three bodies, both enacting forces on each other, and on the external environment. The entire mechanism is contained in a plane, so there are only three degrees of freedom for each body: Translation in the x and y directions, and a rotation about the z-axis. The Newton and Euler equations are applied as follows:

$$\sum F = m \square \text{ and } \sum M = I \square \dots\dots\dots(vii)$$

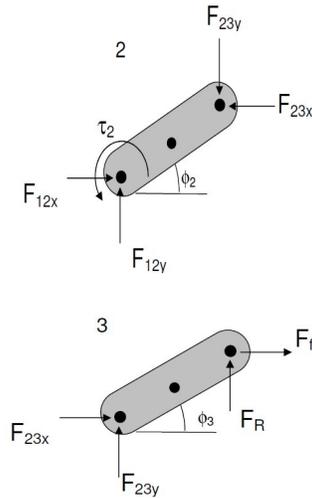
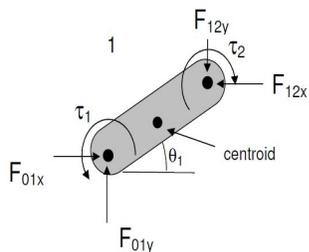


Figure4. Free body diagrams for the links in M1.

In the case of M1 there are exactly 9 unique terms and 9 equations. The entire system of equations can be reduced to one matrix equation, which can then be inverted to solve for each unknown term. Similarly, the dynamics of M2 can be developed using the exact same method as the dynamics were developed for M1. The solution for this mechanism involves inverting a 12 X 12 matrix, and the torque values are obtained.

C. Mapping solutions

In other words, for each mechanism, the joints and links have been labeled for that step. The joints in the overall robot are labeled as well. The solutions for each sub-problem must be “mapped” back to the appropriate joints for the entire problem. In order to map the solutions, a simple matrix multiplier scheme is used. The solutions for the step sub problem can be multiplied by a matrix to map them to the global coordinates of the joints using the matrix “ S_n ”, with “ n ” being the number of the step.

$$\square_{global} = S_n \square_{local} \dots\dots(viii)$$

Similarly, the torque solutions can be mapped using the same matrix:

$$\tau_{global} = S_n \tau_{local} \dots\dots(ix)$$

After the solutions for the sub-problem are mapped to their global coordinates, the solutions can be pieced together in the order of the steps to obtain the time-histories of the joint angles and joint torques for the entire step of the path.

III. Trajectory Generation Approach

In view of the fact that complications arise when generating optimal paths, a heuristic based trajectory

generation approach was used to demonstrate how trajectories could be generated using the kinematics and dynamics model. This approach should also be able to lend some insight to what trajectories would be better than others in the trajectory generation problem. This approach can be described as a simple perturbation-based search. The approach begins with an initial set of parameters, and the effort metric is approximated. The parameters are then individually perturbed to see if they result in a decrease in effort, and the results are recorded. A new set of parameter values are then generated based on which perturbations resulted in decreased effort. The search continues until a specified number of iterations is reached. To use this approach, the effort metric must first be approximated. The effort metric can be approximated

$$J(\tau) = \frac{1}{2} \sum_{j=1}^N \sum_{m=1}^{M-1} \left(\frac{\tau_{(m+1)j} + \tau_{mj}}{2} \right)^2 \Delta t \quad \dots (X)$$

This function estimates the average torque over an interval using the endpoints. There are $M-1$ intervals, with M boundary points of the intervals. The value of Δt is calculated by taking the number of intervals and dividing by the total time to complete the step. N is the number of joints in the mechanism. The heuristic-based search approach works by having an algorithm that takes the non-boundary control points of the trajectory and generates the effort required by that trajectory as an output. Fig.5 below shows the algorithm for this approach.

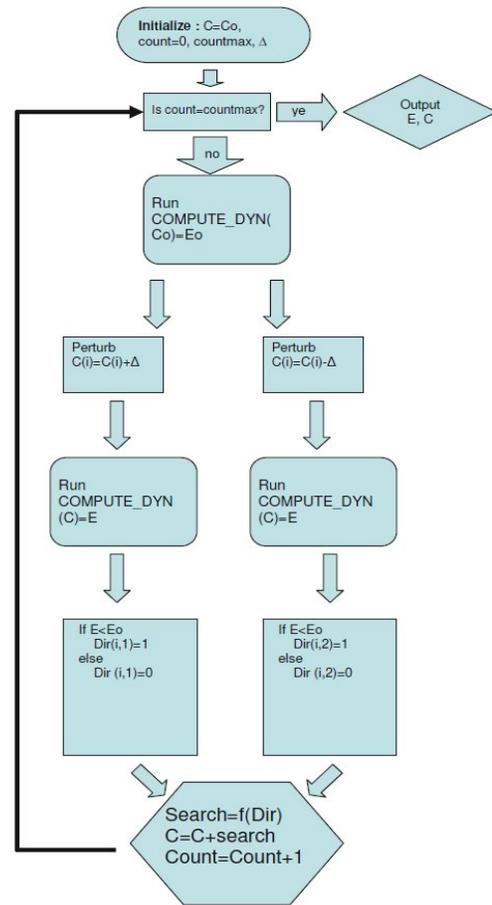


Figure 5. Organization of search algorithm.

The program begins with an initial value for the matrix of control points, C_0 , a number specifying the number of iterations the search process should be attempted, and a value specifying the step size for the search. The initial control points C_0 , and obtains the effort value of the path step that is described by the initial control points. After this value is obtained, an approximated gradient of the effort value with respect to the free control points is calculated. This is accomplished by individually perturbing the control points by both Δ and $-\Delta$. After each perturbation, the new effort value is calculated. If the new effort value is lower than the initial effort value, then a 1 is stored in the direction (“Dir” in Fig.5) matrix. If not, a zero is stored. This matrix can then be used to construct an approximated gradient that stores the complete $N \times M$ direction of decreasing effort value. It is important to note that this is a coarse solution because it only considers each parameter (control point) individually,

and it does not store information related to the amount of effort change achieved by each perturbation. Nonetheless, it results in a direction of decreased effort value in most cases. A search direction is then constructed by multiplying this “gradient” by Δ , and the C matrix is updated by $C=C+search$. This process is repeated and the search finds trajectories with lower effort values until it reaches the specified number of iterations.

IV. Results

The heuristic-based search routine was conducted for each of the path steps. Each step was analyzed individually, and effort was minimized for each sub-problem. For the first search run, $\pi/4$ is used as the path angle, 1 second as the path step time, and a friction coefficient of $\mu=0.4$. This would be a typical value of the coefficient of kinetic friction for a material such as plastic on a smooth concrete or metal. In addition, the length of the link is $l=0.14m$ and the mass of the link is $m=0.015kg$ was taken which will be taken when the robot will be physically realized. The moment of inertia is calculated by treating the link as a thin rod, this equation was chosen because in most snake-inspired robot implementations, the links take the form of rods, where the length is significantly greater than the link. The discrete sampling was conducted at a rate of 20 times/interval, and the value of Δ was set at $\alpha/100$. These parameters were generated heuristically after testing the algorithm.

The results are shown here for 50 iterations of the algorithm. This number of iterations was shown to give good convergence. However, there were several cases where the convergence oscillated. In these cases, the control points that produced the lowest effort value were determined to be the appropriate solution. For step 1, both the open and closed-loop models were used to generate a solution. It was determined that the closed-loop solution required significantly less effort than the open loop solution. Thus the closed-loop solution was used. This should be somewhat intuitive, because although friction affects the solution, the joints do not have to support all of the weight of the links in the closed loop solution. Figure 6. Shows the results of the heuristic-based search for path step 1. Each data point represents a solution found during the algorithm. The effort

values are shown to decrease, and then converge to a solution after 27 iterations.

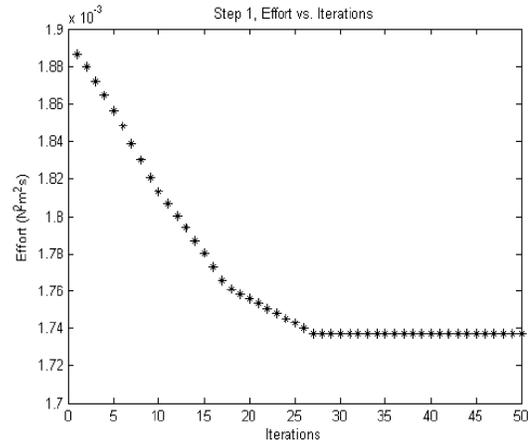


Figure 6. Effort vs. iterations for step 1 search.

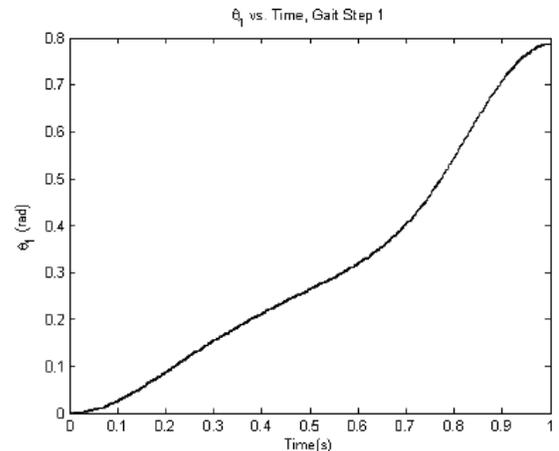


Figure 7. Resultant trajectory function for joint 1.

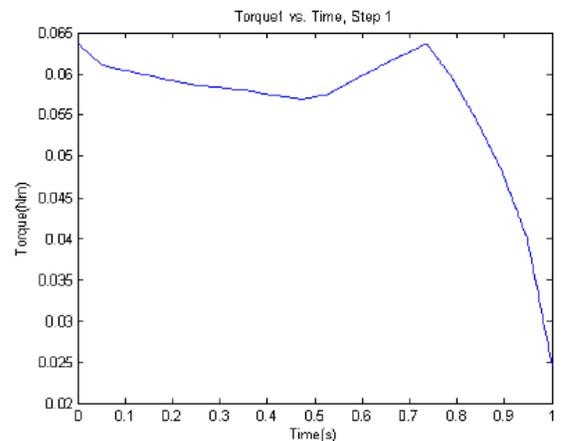


Figure 8. Torque versus time for joint 1.

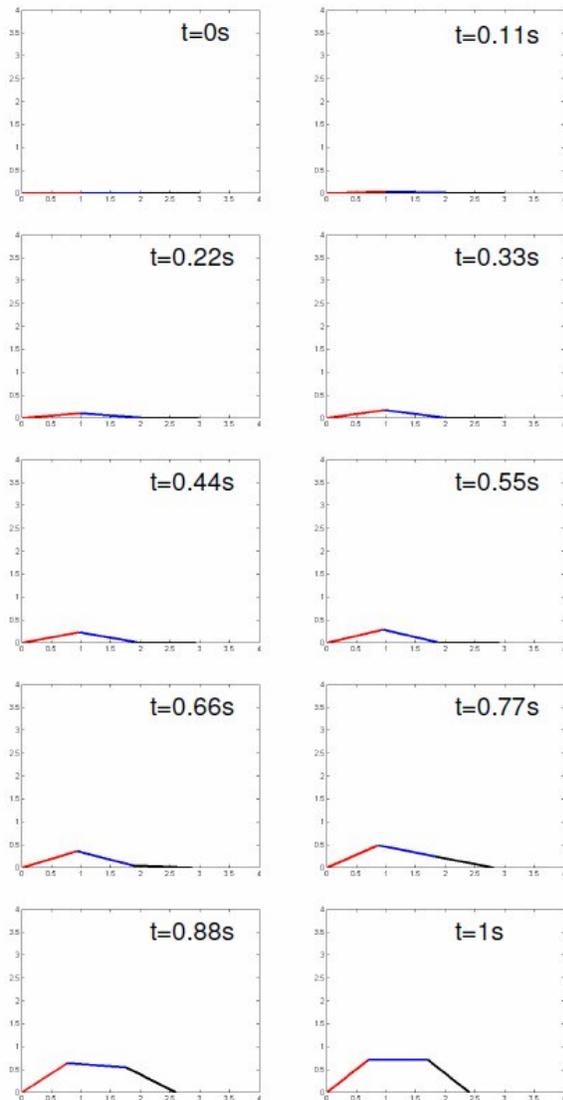


Figure9. Snapshots of path step 1, shown at 0.1s intervals.

Similarly readings for the successive steps were taken and the results were analyzed.

IV. Conclusion

To summarize, in this section a means of computing the kinematics and dynamics of a snake-inspired robot moving with a vertical, rectilinear gait has been developed. This allows for direct computation of joint torques during this motion. This model was developed by breaking the motion up into a series of

steps with their respective topological mechanisms, and analyzing each individually. The work in this chapter allows computation of all of the relevant dynamic information (especially joint torques) at any instant in time for a particular class of gait, and generates joint trajectories for these gaits.

REFERENCES

- [1] S. Hirose, *Biologically inspired robots / Snake-like locomotors and manipulators*, Oxford University Press, 1993.
- [2] M. Mori and S. Hirose, "Development of active cord mechanism ACMR3 with agile 3D mobility," *Proc. IEEE/RSJ IROS'01 Maui, HI*, vol. 3, pp. 1552-1557, 2001.
- [3] S. Hirose and F. Matsuno, "Development of Snake Robots for Rescue Operation) Design of the Shape and Its Control," *Journal of Japan Society of Mechanical Engineers*, vol. 106, no. 1019, pp. 769-773, 2003.
- [4] G.S. Chirikjian, *Theory and Applications of Hyper-Redundant Robotic Manipulators*, Ph.D. Thesis, CalTech, Pasadena, CA, 1992.
- [5] J.P. Ostrowski, *Geometric Perspectives on the Mechanics and the Control of Undulatory Robotic Locomotion*, Ph.D. Thesis, CalTech, Pasadena, CA, 1995.
- [6] K. Dowling. *Limless locomotion: learning to crawl with a snake robot*. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1997.
- [7] G.S. Chirikjian and J.W. Burdick. The Kinematics of Hyper-Redundant Robot Locomotion. *IEEE Transactions on Robotics and Automation*, 11(6): 781-793, 1995.
- [8] H.W. Lissman. Rectilinear motion in a snake (*Boa Occidentalis*). *J. Exp. Biol.*, 26: 368-379, 1950.