Appeared in the proceedings of IROS'95, vol. 3, pp. 250-255, August 1995.

Kaa : An Autonomous Serpentine Robot Utilizes Behavior Control

Rajiv S. Desai, Charles J. Rosenberg, Joseph L. Jones
IS Robotics Inc.
22 McGrath Highway
Somerville, MA 02134

Abstract

Many robot applications require mobility in constrained spaces and a high degree of flexibility. This paper describes the development of an autonomous serpentine robot. This robot is a 12 degree of freedom, untethered, hyper-redundant planar robot, named Kaa. Experiments were conducted in its application to mobility tasks that would be useful in locomotion among parallel pipe structures similar to those found in industrial plants. Kaa utilizes a behavior control system. Unlike classical approaches, that decompose the control system functionally, behavior control approaches decompose the control system by task. This control scheme requires very little computation; Kaa uses two 8-bit, 1 MIP microprocessors with a total of 36 Kbytes of memory. Only simple tactile sensing is employed; the distances between objects in the environment and the robot are neither implicitly modeled nor explicitly sensed by the robot. The behavior control system developed allowed this robot to reliably perform complex autonomous mobility and manipulation tasks with limited computation and sensing resources.

Keywords: Behavior Control, Autonomous Control, Serpentine Robots, Pipe Inspection, Hyper-Redundant Robots.

1: Introduction

There are many applications for robots where a high degree of flexibility is desirable. While a manipulator arm requires only six degrees of freedom to reach any point in a workspace, there is little or no choice with respect to the position each joint occupies in space. This limitation decreases the usefulness of such a device in the presence of many obstacles or in a confined work space. An ideal manipulator consisting of a string of a virtually infinite number of degrees of freedom of zero diameter would avoid this limitation. Many practical hyper-redundant robots have been proposed and constructed to approximate this ideal.[1,2,3,4,5,6] Some hyper-redundant robots have been designed for manipulation tasks. The goal of some designs was to position an end effector in a small or cluttered work space.[1,3] Other authors have worked toward practical whole arm manipulation of objects or studied the kinematics of such a task.[7,4,5,8,6] Mobility in hyper-redundant robots has also been investigated.[7,2,5,9]

In many cases there are practical problems that render serpentine designs of limited use. In some implementations, one end of the robot is stationary, limiting the reach of these robots to a sphere with the radius equal to the fully extended length of the robot. Also, due to the redundant nature of the serpentine design, the inverse kinematics of the robot are non-determinate and solutions to specific problems can be complex. In almost all cases the systems are not autonomous, they carry neither power nor computer control on board.

Our interest in serpentine robots stemmed from the specific application of the external inspection of pipes that commonly occur in industrial plants. A typical pipe structure is shown in figure 1. Such a structure is characterized by pipes arranged with narrow irregular, indeterminate spacing that cannot be readily modeled. Often there are numerous layers of such pipes. In case of one heat exchanger used inside a nuclear power plant, there were 1024, 1” pipes spaced less than 3” apart (center to center). However, there is some regularity to these structures which can be exploited. For example, all the pipes have an essentially circular cross-section and many are arranged parallel to each other in a planar fashion. Such structures are typically inspected manually. Traditional manipulators are not suitable for this inspection task as they suffer from the same limitations as humans with respect to reach.

This problem inspired us to investigate the possibility that a small, autonomous, serpentine robot could navigate such pipe structures and lead us to investigate the capabilities of serpentine robots in general.

For such robots to be practical, they must be highly autonomous and cost effective. Robots cannot be tethered, as tethers can limit the motion of the robot and the mechanical system needed to manage a tether adds to the mass and complexity of the robot.

In most instances, the robots cannot support sustained, high bandwidth communication. This is due both to the power constraints on the robot and fact that the surrounding environment, metallic pipes for example, can cause interference. Off board computation typically does not lend itself to low bandwidth communication; therefore a higher degree of autonomy is advantageous.

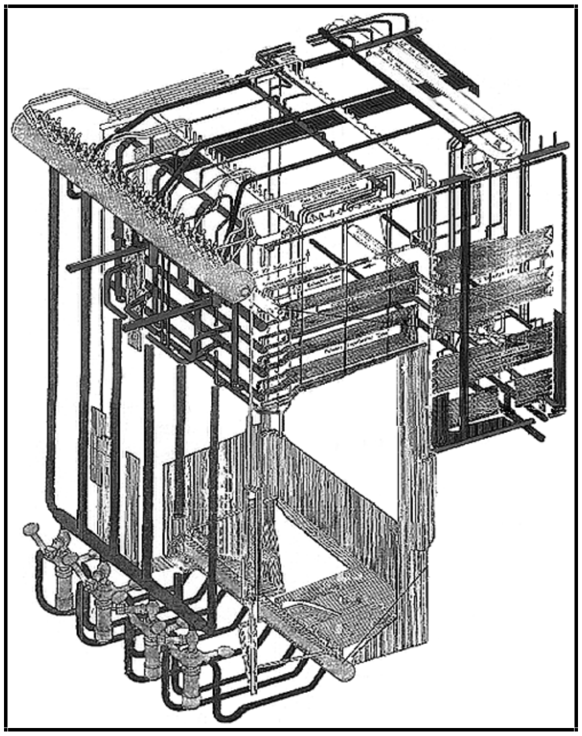

Figure 1: Example of an industrial pipe structure. The general shapes of the pipes (cylindrical), approximate spacing, etc., is known; however, the exact location and shape information is not known.

The behavior control approach allows for the development of highly autonomous robots that require very little computational resources to perform their task. The ability to implement the control system with minimal computational resources helps reduce the overall size and power requirements of the robot. Behavior control is also attractive because of its ability to readily adapt its actions to an unspecified environment. This is extremely important in the case of a serpentine robot where sensing may be sparse at best and uncertainties related to joint positions are cumulative, making it difficult to position specific joints accurately.

2: Kaa - planar mobility system

In 1993 we developed a planar serpentine robot named Kaa, as shown in figure 2. The Kaa robot has 12 degrees of freedom. The robot uses two, 8-bit 1 MIP microprocessors and 36 Kbytes of program memory. The robot does not have any external sensors; however, it can sense the torque in each motor. The robot is completely autonomous - power and computation are on board.

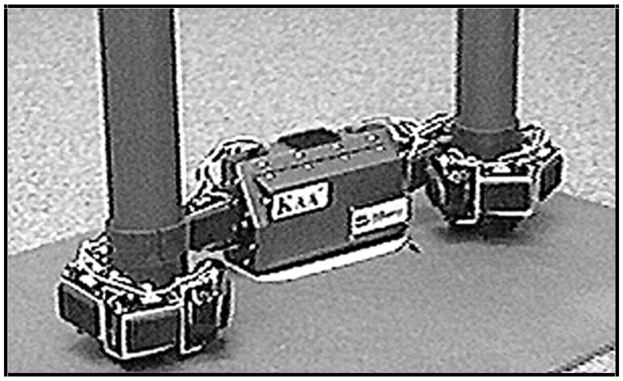

Figure 2: Kaa Robot gripping two pipes.

The mechanical system of Kaa consists of thirteen links and is approximately 30 inches long. A drawing of the mechanical system can be seen in figure 7. The center link is the main body of the robot. This link contains all of the computational hardware and the power source. To each side of the main body are six degree of freedom jointed arms arranged so that their axes of rotation are perpendicular to a common plane. Unlike some other serpentine robot designs that are cable driven [3,4,6], in Kaa the motors are distributed through the links. Each link contains a single degree of freedom implemented by a servo motor and driving electronics which provides position control of the motor. Each motor is capable of exerting a maximum torque of 1.65 Kg-cm. Because of the 2D nature of the task, the robot never has to support the weight of the arm against gravity. Thus links closer to the central body need not generate more torque than those further out. The 10 links closest to the central body are covered with foam rubber on their vertical faces to increase gripping friction. The two links on the ends of the arms are terminated with rounded "finger tips" which are covered with low friction Teflon tape to facilitate tactile exploration of the environment. The links do not have wheels on the bottom for support, as other serpentine robots do, but the arms are offset above ground level so only the central body is in contact with the ground.

The computer system in Kaa consists of two networked 1 MIP, 8 bit microprocessors. One processor controls the servo motors using less than 2 Kbytes of code. This processor also provides Kaa's sole means of sensing its environment. Motor current is utilized to estimate the motor torque at each joint. The second processor acts as the system controller and has 34 Kbytes of program space.

3: Computing architecture

The control program for the Kaa robot utilizes the behavior control paradigm.[10, 13] Such a control program is highly reactive, the overall task is decomposed into subtasks each of which links sensing to actuation. The behavior control program for the Kaa robot is programmed in the behavior language developed and implemented by Brooks.[11] The compiler for this language runs under Common Lisp and can be re-targeted for a number of different microprocessors. This language is specifically designed to facilitate the generation of behavior control systems for robots. The language allows for the implementation of many small parallel processes, often referred to as augmented finite state machines, which link sensing to action. This allows for task decomposition rather than the more traditional functional decomposition. The processes communicate by passing messages to one another via software "wires". All of the processes are run in a real time multi-tasking kernel on the system processor. Kaa's control program uses no explicit models of obstacles or their location in the environment.

The overall design methodology for all of the programs in Kaa was to control each joint with a set of associated processes, a module. Modules have minimal connections to other program modules. Connections between modules tend to occur between modules that control joints which are physically adjacent on the robot. The specific set of parallel processes which controlled each joint of the robot was functionally identical. Processes controlling each joint differed only in their connections to other sets of processes, associated with physically adjacent joints. The joints closest to the central link communicate to the processes which co-ordinate the robot's overall action. This design lends itself to a hardware implementation of the control system where each joint is controlled by a local computation block. Such a design is advantageous in the control of micromachines.[12]

4: Behaviors

4.1: Pipe mobility

One set of behaviors explored using Kaa was mobility among parallel pipes, i.e. the ability to reposition the robot body in a field of parallel vertical pipes. The vertical pipes present a simple circular cross-section to the robot. This overall behavior is diagrammed in figure 3. The basic behavior steps are as follows:

1) Kaa begins with its left arm wrapped around pipe A, as in figure 3a.
2) It acquires pipe B by swinging its body in a counter-clockwise direction, using the left most joint not involved in the grasp of pipe A, as in figure 3b, 3c.
3) The left arm unwraps and folds back parallel with the body, as in figure 3d, 3e.
4) Kaa uses its right arm to swing its body clockwise a few degrees, as in figure 3f.
5) The right arm unfolds on the opposite side of the pipe and ready to grasp a new pipe, as in figure 3g.
6) This sequence is repeated with the right arm, starting at step 3.

Grasping, folding, and unfolding are three important sub-behaviors of this sequence. Grasping is the ability to acquire and grip an object. Folding and unfolding are a means of moving the arm through a tight space.

4.2: Organization

The behaviors in Kaa are organized by task. There is one set of behaviors for grasping, one set for folding, and another for unfolding. Each side of the robot ("A", the left side and "B", the right side) has a distinct group of behaviors that control it. Also, a task specific process is associated with each joint of the arm. Figure 4 gives an overall view of this system. Boxes in the figure represent processes or groups of processes. Lines represent bi-directional message passing paths. Thin lines represent connections to hardware interface processes. The task sequencer controls the overall pipe mobility behavior, it can initiate a specific behavior. There is a top level behavior for each task such as Grasp-A, Fold-B, etc. This behavior initiates the action by activating the appropriate behaviors for the task associated with each joint and monitors the task's progress, passing a "task complete" message back to the task sequencer. In figure 4 the lowest level processes, j-xn where x=A,B and n=0,1,2,3,4,5, provide the interface between the task processes and the motor hardware.

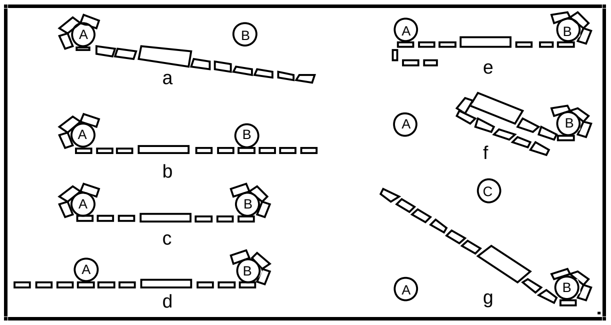

Figure 3: A diagram of Kaa's planar mobility sequence, top view, steps a-g. The cross-sections of vertical pipes are labeled A, B, and C.

4.3: Grasping

Grasping is a key capability of hyper-redundant robots. Other researchers have investigated grasping with hyper-redundant robots.[7,8]. Pettinato and Stephanou in [8] propose a grasping scheme similar to that ours.

In Kaa, a simple set of behaviors implements a robust grasping action. Objects of various sizes and shapes and at various positions along the arm can be grasped. Each joint is controlled by a set of processes. Each set of processes is identical and is connected to its neighbors. To grasp an object, a wave of activation is passed from the central link to the joint closest to it on one side. This wave of activation spreads joint by joint to the arm tip. The resulting action is that the arm swings out and when it encounters an object, it wraps around it, very much like the action of an elephant's trunk. The specific steps in the grasping behavior are as follows:

1) The arm is moved into a straight position; each joint angle is set to zero as defined in figure 6.
2) A wave of activation is initiated at the central link and is passed to the next nearest link, on the side of the robot which grasping is occurring.
3) When a joint is activated it moves in the direction of the object to be grasped. (This causes all of the arm from this link to the tip to move in that direction.)
4) This motion continues until a position limit is reached or a force is detected (via motor torque for this joint).
5) The joint then goes into a force servoing control mode. It then activates the next joint further out on the arm which repeats the actions outlined in step 3.

After the grip is established, the joints are actively servoed to maintain a particular torque. This is useful when the initial contact point of a joint with an object slips because of the motion of the robot or the object. Maintaining a particular gripping force ensures that the robot's grip on the object is not lost. Compliance was also found to be useful in gripping deformable objects, like the soft insulation on the outside of many pipes.

Kaa achieves a reliable, functional grip using only local control and the relatively sparse sensing provided by motor torques at each joint.

Figure 5 details the processes which comprise the gripping task. The top level process Grasp-A or Grasp-B, for the right and left arms, takes in a start message and outputs a done message to the task sequencer. To initiate a grasp, the Grasp-X process outputs a signal to the link closest the body to initiate a grasp. Each link has an associated process, grasp-xn, where x is A or B specifying the robot arm and n specifies the link number. Each link grasp process communicates with the joint actuator via a joint process. The joint process provides an interface to the motor hardware and is shared by all of the processes running on the robot. The interface to the joint process consists of inputs which choose an absolute servo position for the joint or a force servoing mode with high and low force bounds on the servo dead zone. The joint process passes force and joint position back to the link grasp process. The grasp-xn process also has an excite input and an excite output. The input triggers the grasping behavior for that link and when it is complete, passes on the activation to grasp-x(n+1), where x is A,B and n is 0,1,2,3,4.

4.4: Folding and unfolding

One of the major advantages of hyper-redundant robots is their ability to operate in cluttered and highly constrained spaces. One way to deploy a hyper-redundant robot in a tight space (say a tunnel or the narrow space between two pipes) is to have it fold in on itself and then unfurl its arm into the space. A fold/unfold maneuver can be seen in steps d, e, f, and g of figure 3.

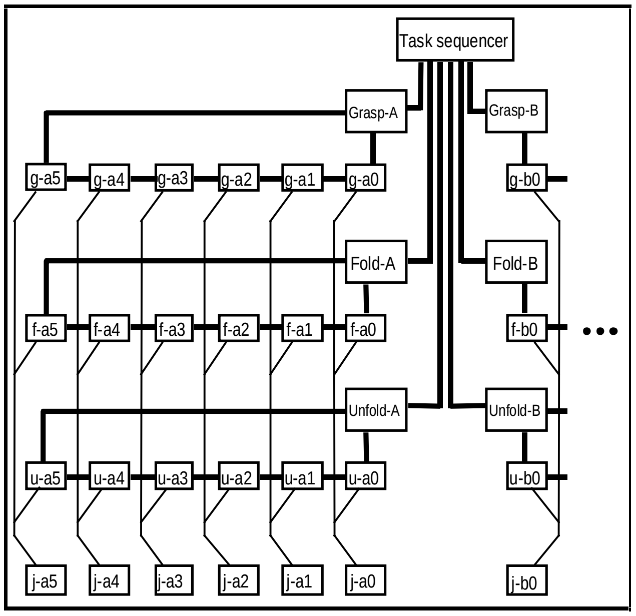

Figure 4: A diagram of the overall behavioral structure of Kaa. Lines represent bi-directional software "wires" along which simple messages are passed.

Again, each link is controlled by a local set of processes, each of which is connected only to processes controlling physically adjacent links. The folding behavior is similar to the grasping behavior in that a wave of activation spreads from one link to another. However, in the case of folding, two activation waves are spread from the link furthest from the robots central body, inward. The second wave is delayed by approximately one link compared to the first wave. The first wave causes the links to curl towards the body one by one. The second wave causes the links to re-straighten one by one. Throughout this process the robot uses the motor torque measured on the last link of the arm as tactile sensor. This allows it to keep the tip of the arm in contact with the body and occupy a minimum of space. The folding behavior consists of the following specific steps:

1) One wave of activation begins at the central link and is passed to the outermost link, and from there it is passed inwards. This is the fold wave.
2) A second wave of activation starts at the central link and is passed to the outermost link and is passed inwards. This is the straighten wave. When a joint is activated by the fold wave it moves in the direction of the fold.
3) The folding motion continues until a position limit is reached and then activation is passed on to the next link inward.
4) When a joint is activated by the straighten wave, it waits until the tip of the arm senses a force. When this occurs the joint is straightened and the straighten wave is passed on to the next link inward.
5) When both waves have propagated through the arm, the third joint from the body is placed in a force controlled servo loop. This presses the arm up against the body, minimizing the space occupied by the robot.

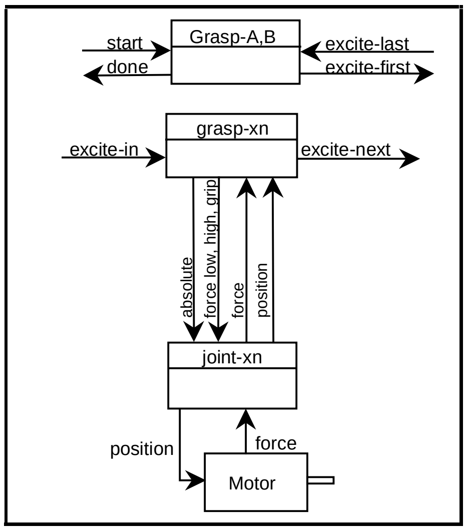

Figure 5: Diagram of the grasp behavior. The top half of the figure is the top level grasping behavior and the bottom half are the behaviors associated with each link.

Unfolding is accomplished by a similar set of steps. It is important to note that the robot contains no explicit model of its body. It uses its motor torque sensors to fold itself into a minimum configuration. The behavior of the system is much like that of a blind man feeling his way by tapping along a path. The resulting folding behavior is quite robust. Kaa was able to fold its arm from either an initial straight or gripping configuration.

By appropriately combining these behaviors, Kaa can move among pipes that are parallel and spaced within the reach of its arms. The actual spacing and sizes of the pipes need not be known. The structure of the environment has been effectively exploited. Kaa's control program is approximately 32 kilobytes long, no attempt was made to optimize it. Kaa performs no explicit forward or reverse kinematics computations.

4.5: Wave locomotion

In the absence of pipes to grab, Kaa can move along the ground via wave motion. To use this locomotion scheme the axis of rotation of the servo motors is made parallel to the plane of the ground.

A number of gaits have been proposed for serpentine robots utilizing solely body motion, without the use of explicit legs or wheels. Perhaps the most unusual is to mimic the sidewinding motion of snakes.[9] The same researchers, in another paper suggest two types of wave motion.[14] One gait utilizes a stationary wave of varying amplitude. This is much like the gait of an inch worm. Another wave gait utilizes a traveling wave of constant amplitude. This gait is much like that of a caterpillar. This is the gait that was investigated here.

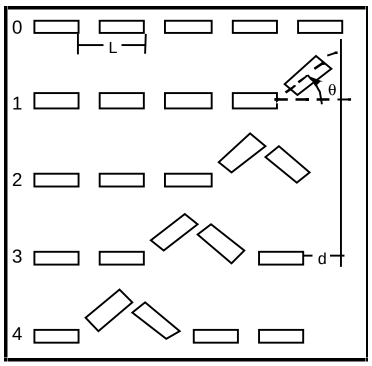

Figure 6: A traveling wave moves Kaa on a flat surface.

Kaa moves as a triangular wave propagates along the robot's length. Figure 6 shows Kaa's five rightmost links. Each time a wave traverses its length (assuming no slippage), the robot moves a distance, d, given by $d = 2 L (1 - \sin \theta)$, where L is the length of a link. Every joint is associated with a separately instantiated wave behavior. When triggered, a joint wave behavior executes the four steps listed below. Each step takes an equal amount of time. Each process is wired to the process associated with the adjacent link. Thus, behaviors execute identical operations with a 90 degree phase difference.

The process associated with joint n does the following:

1) Move joint n to angle +q.
2) Move joint n to angle -2θ while triggering the behavior controlling joint n+1.
3) Move joint n to angle $+\theta$.
4) Return joint n to the zero angle.

When the wave reaches the central body of the robot, the joint angles are modified so that the amplitude of the wave remains constant as it passes the central body.

Joint coordination is necessary for coherent wave motion. Such coordination is hampered, however, because Kaa's hardware does not support velocity control of joint motions. In order to guarantee that all the joints involved in a wave motion reach their destinations approximately simultaneously, each step of the above process is broken into some number of substeps. By an appropriate choice of parameters, e.g. the number of substeps and substep duration, joint motions are roughly synchronized.

5: Conclusions and future work

The Kaa robot demonstrates a functional, completely autonomous serpentine robot, carrying both power and computation on board. Grasping and confined space mobility behaviors were developed to demonstrate the minimum competence necessary for locomotion among a collection of parallel pipes. The same physical mechanism, without modification, has also demonstrated locomotion along a plane using a traveling wave constant amplitude gait. All of this was implemented utilizing behavior control combined with simple tactile sensing without explicit models of the environment. The practicality of using this specific system for mobility in a pipe environment is unproven; however, we believe the lessons learned will be useful in a practical system in this domain.

In the future we would like to extend Kaa's mobility and sensing capabilities. To extend its mobility, a new degree of freedom perpendicular to the other degrees of freedom would be added to each arm. This would allow motion out of plane and the ability to climb up and down a pair of parallel vertical pipes. This would also provide a simple means of steering the traveling wave gait. On the sensing side we would like to add infrared proximity sensors. These would be added, minimally, to the arm tips and would allow the system to find and scan for a free path between pipes as well as avoiding obstacles while implementing the traveling wave gait.

6: Acknowledgments

The development of the Kaa robot was funded by Mitsubishi Heavy Industries. The associated behavior control program development was funded by the MITI Micromachine Center as the "Research and Development of Micromachine Technology" supported by NEDO (New Energy and Industrial Technology Development Organization). Thanks go to all the people that brought Kaa to life. Colin Angle and Helen Greiner completed the final mechanical design of the system and fabricated the first prototype. Rodney Brooks developed the initial behavior control frame work for Kaa. Tri Freed revived Kaa in its second, more robust, implementation. Rudyard Kipling worked on nomenclature.

7: References

[1] Taylor W. K., Lavie D., Esat I. I., "A curvilinear snake arm robot with gripper-axis fibre optic image processor feedback", *Int J. Robotica*, vol. 1, pp. 33-39, 1983.

[2] Hirose, S., Morishima, A., "Design and Control of a Mobile Robot with an Articulated Body", *Intl. J. of Robotics Research*, vol. 9, no. 2, pp. 99-114, 1990.

[3] Ma, S., Yoshinada, H., Hirose, S., "CT ARM-I: Coupled Tendon-driven Manipulator Model I", *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, pp. 2094-2100, 1992.

[4] Greiner, H., Passive and Active Grasping with a Prehensile End Effector, MIT Master's Thesis, 1990.

[5] Chirikjian, G. S., Burdick, J. W., "Design and Experiments with a 30 DOF Robot", *Proceedings of the International Conference on Robotics and Automation*, vol. 3, pp. 113-119, 1993.

[6] Salisbury, K., Eberman, B. Levin, M., Townsend, W., "The Design and Control of an Experimental Whole-Arm Manipulator", *The Fifth International Symposium on Robotics Research*, pp. 233-241, 1989.

[7] Chirikjian, G., Burdick, J., "Kinematics of a Hyper-Redundant Robot Locomotion with Applications to Grasping", *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, pp. 720-725, 1991.

[8] Pettinato, J. S., Stephanou, H. E., "Manipulability and Stability of a Tentacle Based Robot Manipulator", *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 458-463, 1989.

[9] Burdick, J. W., Radford, J., Chiriljian, G. S., "A 'Sidewinding' Locomotion Gait for Hyper-Redundant Robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 101-106, 1993.

[10] Brooks R. A., "A robust layered control system for a mobile robot", *IEEE Trans. Robotics and Automation*, vol. 2, no. 1, 1986.

[11] Brooks, R. A., "The Behavior Language User's Guide", *Memo 1227*, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, MA, 1990.

[12] Brooks, R. A., "Behavior-Based Control of Autonomous Micro-Robots", *Micro Machine*, vol. 6, no. 1, April 1993

[13] Gat, E., "ALPHA: A Language for Programming Reactive Robotic Control Systems," IEEE Conf. on Robotics and Automation, 1991.

[14] Chirikjian, G. Burdick, J., "Kinematics of a Hyper-Redundant robot Locomotion with Application to Grasping," IEEE Conf. on Robotics and Automation, 1991.

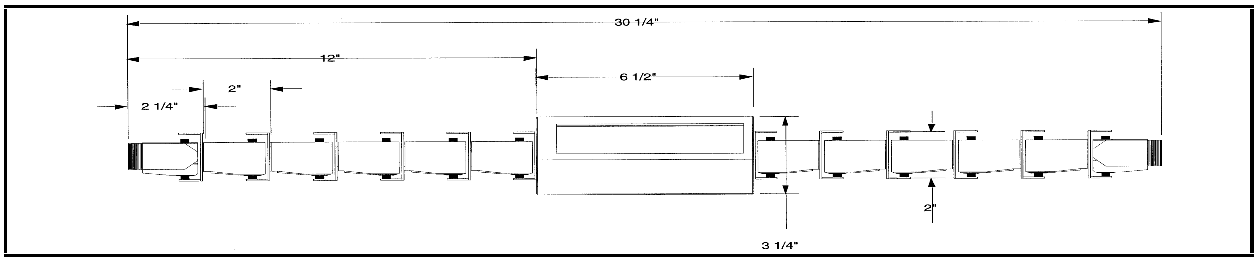

Figure 7: Details of the Kaa mechanical system, a front view, with dimensions in inches.