A robot snake to inspect broken buildings

Conference Paper · February 2000

DOI: 10.1109/IROS.2000.895277 · Source: IEEE Xplore

Thomas Christaller
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS

Frank Kirchner
Universität Bremen

Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems

A Robot Snake to Inspect Broken Buildings

K.L.Paap T.Christaller F.Kirchner

Institute for Autonomous intelligent Systems
GMD D-53754 Sankt Augustin
paap@gmd.de

Abstract

Snakes move in rough terrain, cross obstacles and move in narrow space. This inspired us to build a snake-like robot for the inspection of areas that are difficult or dangerous to be accessed by human. Pictures from a camera at the robot's head are sent to a remote screen and can be monitored by a human operator. This human operator can also control the robot's motion but not every single body part. This would be too difficult even for a simple robot snake having only five body sections because of it's many degrees of freedom. So the operator will just give general directives and the robot shall then able to follow autonomously the given directives. This is the semi-autonomous behavior of the robot. But the robot must also be able to act fully autonomous when the contact to the operator is lost.

After a short description of the robot we present a method which makes operator control easy and also allows the robot to act fully autonomous. The autonomous motion in a sewer pipe was already implemented and will be demonstrated in a video.

1. Introduction

Snakes use many different types of motion depending on the environment. They can move like caterpillars by bending and stretching their body which is e.g. useful in trees. When approaching a victim some big snakes move by contracting and stretching like worms which makes the motion scarcely visible. Some snakes moving in hot sands use a sidewinding technique, where each part of the body touches the ground only for a short moment. A good survey of these motions can be found in [1]. But most often snakes use a type of motion which S. Hirose [2] calls serpentine motion. Serpentine motion is characterized by the fact that every single part of the snake's body moves on the same trajectory. One can think of a trajectory which the snake's head defines and the body of the snake just follows this trace.

This seems to be the most natural thing but e.g. caterpillars don't move this way. Even snakes sometimes use other types of motion.

During serpentine movement the main propulsion comes from a source not to be observed at first glance: hundreds of tiny scales on the bottom side of the snake are shifting it's body forward.

If this was the only driving force the snake would better move in a straight line. But as it's motion is undulatory it can stem it's body against the outside of the curves and thus gets additional propulsion.

Serpentine motion has some additional advantages:

- It prevents the snake's body from tilting.
- The path can be defined arbitrarily e.g. to avoid obstacles.
- It is easily controlled as only the motion of the head needs to be defined and the body's motion is then fixed.

This serpentine motion implies that every part of the body moves in the same way at the same spot. The observation of nerve stimulation of snakes showed that stimulation activates only a small part of the body and this stimulation propagates through the snake from head to tail.

This observation leads us to a motion control method for a robot snake which is independent of the number of a snakes body segments:

1. Arbitrarily define the head's motion. This may be done using a joystick if an operator controls the snake.
2. Or the head may follow a given path thereby avoiding obstacles. Store the heads motion parameters such that the motion can be repeated by the body segments.

3. For each section determine its actual position on the trajectory and then execute the appropriate motion as executed by the head.

We built a robot to imitate this snake motion. The next section will give a brief description of our snake robot. After that we will discuss some problems which occur when applying the algorithm above.

2. The Snake Robot

Our first GMD-Snake (for details see [3], [4] or http://www-set.gmd.de/RS/snake) was built with rubber joints to allow flexible bending of the parts. A disadvantage was an uncontrolled torsion effect which occurred when the snake lifted some of its parts (e.g. to climb on a step).

From this experience we decided to construct the next generation in a more rigid way using universal (or cardanic) joints.

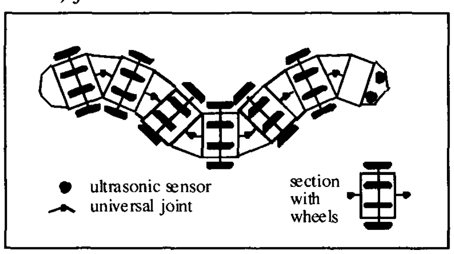


Fig.1: Schematic view of GMD-SNAKE2 with wheels

The new Snake robot (Snake2) is built of five identical body sections which are connected via universal joints. The head carries two ultrasonic sensors for obstacle detection and a video camera with a sender such that the pictures can be monitored on a remote screen. Batteries in the tail allow the robot to operate up to one hour before recharging. A microcontroller in the head takes over central control of the robot. Within each body section a 16Bit microcontroller processes the local sensor information and operates the motors. Within each cylinder there are three 5W-DC motors to control the position of one universal joint by way of ropes. Between these motors all electronic parts have been integrated, including one C167 processor per section, each with a CAN interface for bus communication. We achieved good results with this bus concept and it is used as well in the joint project MAKRO (with FZI Karlsruhe) [5].

Often snake-like robots are designed for studying control problems with high number of degrees-of-freedom. (e.g. [6] or [7]). Our SNAKE2 is intended for practical tasks (inspection of areas hardly to reach by men, e.g. narrow sewerage pipes or destroyed buildings) [8].

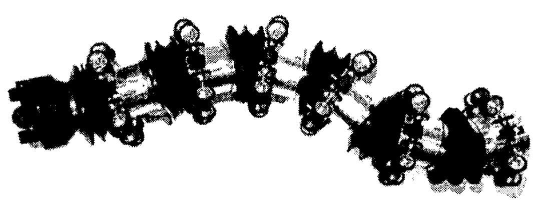
We have studied the numerous types of movement performed by real snakes [1] and found one essential technique which is usually not implemented in robot snakes: the forward forces generated by hundreds of active scales under the snake's body. These forces allow a normal land snake to slide within it's curved track.

To realize a technically equivalent movement we implemented a ring of wheels around each section of our snake. So e.g. the snake can fall on its back and move upside down. Each ring of wheels is driven (via special joints) by one additional DC motor per section such that every section can control its own forward force.

By the active joints we can obviously adjust the curvature of the snake's body. This is only limited by the maximal angle (here 45°) and by the discretization of the path induced by the straight rigid parts between the joints.

The diameter of the snake (including wheels) is 18cm while the length of each section is 13.5cm. Hence, with 10 sections plus head we end up with a total length of about 1.5m and 15kg weight.

So, in contrast to several existing snake robots, here the length of one section is clearly smaller than its diameter which is important for a flexible behavior.



For test runs we can drive our snake with external power through a wire, but for the autonomous version we use an extra section at the tail, filled with batteries and without any motors. So this last joint and ring of wheels is not driven actively.

The following picture (fig. 2) shows four sections of our snake. The joints are covered with rubber bellows to protect them from dust and humidity.

Fig. 2: Head, 5 sections and battery of Snake2

For more snake robots see the collection [9]. There exist several snake-like robots in research institutes around the world. Two big problems for those of them which are intended as inspection systems (not as fixed manipulators) are weight and autonomy.

3. Motion Control

In the introduction we described the snake's motion as to follow a given path. Let us define the path as constructed from the points where the wheels touch the ground. Then only special types of trajectories will result for serpentine motion, namely trajectories constructed from clothoids as shown in [10]. This is natural as the robot snake will move such that the wheel's axes are always nearly perpendicular to the tangent of the trajectory. This reduces friction and will result in minimum energy consumption. Fig. 3 shows that energy consumption raises fast if a wheel not just rolls but is also pushed or pulled in a direction which deviates by an angle delta from it's rolling direction.

Now let us define a model to describe the snake's motion:

We number the sections from 0 (head) to k (tail).

All section are of equal length L. The joint's position is given by two angles a and b and the speed of the wheels is v. So the action of one joint is to change the angle's speed da/dt, db/dt and v, where t is time. These actions are executed only at discrete times given by t_0, t_1, t_2,.... The head chooses the actions arbitrarily e.g. such that no obstacle lies ahead and stores the actions. The body sections can then execute the action after a certain delay. But how should this delay be calculated or estimated? For the wheel's speed of course this delay should be zero, as the whole body moves with same velocity. E.g. section i has to execute the actions which the head executed before it moved a distance of $i*L$. But in the meantime the speed may have changed. So the actions cannot be just repeated at the same speed the head used but must be modulated with speed. That is, if e.g. speed slows down also the bending of the joints must slow down proportionally.

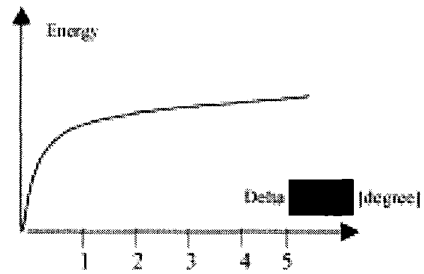


Fig 3: Energy consumption as function of deviation

These requirements can easily be fulfilled. If t_j is the last time we choose the next time t_{j+1} such that the robot has moved a path of length L/N since t_j. We can then implement the motion control in a way as if we had a large shift register distributed on the sections. The shift registers are concatenated and each shift register stores N actions. The procedure works as follows:

The head calculates a new action and the time t_{j+1} when this action should be finished. This time will be sent to all sections as they all have to finish their actions at the same time. The new action is stored to the shift register and the oldest action stored is sent to the next section. All subsequent sections then do the following: First send the oldest action to the next section. Then execute the just received action within the given time. As mentioned before this method is very close to biological snakes.

If the speed is known and assumed to be constant until the next time t_{j+1} it can be calculated using:

$$t_{j+1} = t_j + L / (v \cdot N)$$

In some cases speed can be directly measured by the ultrasonic sensors. This is true when the snake robot just approaches an obstacle. Otherwise speed must be estimated.

4. Inspection of Damaged Buildings

Let us assume it is necessary to inspect a broken building where it is too dangerous for men to go in. There may be people inside who need help or it may be necessary to find out whether there are urgent actions to prevent more damage.

We are convinced that the snake robot could be very useful in this case.

An operator who controls the snake robot would send the robot into the building. During inspection the TV camera at the robot's head sends pictures. A laptop may be used to observe these pictures. The laptop is also helpful for controlling the robot.

For an easy control a joystick–like equipment is used to define the actions of the first joint. This fixes the direction of the head's motion. The speed v of the robot may be controlled by a button. As was pointed out in the previous section, this is all to control the motion of the whole snake robot.

So the operator needs only few equipment and will become accustomed to that soon.

For some reasons the radio contact between the robot and the controlling laptop may be lost. The robot must then act fully autonomous until the radio contact is established again.

The simplest (but in some cases not easy) autonomous action will be to turn back to it's path into the building and then follow this path in backward direction.

It would be much more complicated if the robot would have to continue inspection on it's own. For this purpose the robot should be able to process or at least store pictures which then could be processed later.

5. Status and Future Work

Within our studies on sewer robotics [KIR97] we applied the above methods to our snake robot when it was creeping through a sewer pipe with bifurcations. At the bifurcation points the robot was able to measure it's speed exactly using ultrasonic sensors in the head.

Using the motion control and speed estimation the robot had to follow a sequence of commands as e.g. 'turn right at next bifurcation' etc. The robot was always able to move as expected.

References

[1] R. Bauchot, (Editor), 'Schlangen', Naturbuch Verlag, Augsburg 1994, ISBN 3-89440-075-7.

[2] S. Hirose, 'Biological inspired Robots', Oxford University Press, 1993, ISBN 0 19 856261 6

[3] K.L. Paap, M. Dehlwisch, B. Klaassen, "GMD-Snake: A Semi-Autonomous Snake-like Robot", In: Distributed Autonomous Robotic Systems 2, Springer-Verlag, Tokyo, 1996

[4] [WOR96] Rainer Worst, Ralf Linnemann, "Construction and Operation of a Snake-like Robot", In: Proceedings IEEE International Joint Symposia on Intelligence and Systems, IEEE Computer Society Press, Los Alamitos, CA, 1996

[5] Project MAKRO see http://www.gmd.de

[6] Chirikjian, G.S. and J.W.Burdick. 'Design implementation, and experiments with a thirty-degree of-freedom `hyper-redundant' robot.' ISRAM 1992, November 1992

[7] R.S. Desai, C. J. Rosenberg, J.L.Jones, 'Kaa: An Autonomous Serpentine Robot Utilizes Behaviour Control', IROS '95, vol. 3, pp. 250-255, 1995

[8] M. A. Lewis, D. Zehnpfennig, 'R7: A Snake-Like Robot for 3-D Inspection', IROS 1994, Munich

[9] http://borneo.gmd.de/~worst/snake-collection.html