Snake Charming: Constrained Kinematics for Unstable Manipulator Bases

Neil Abcouwer, Ellen Cappo, Steven Ford, Howie Choset

Carnegie Mellon University

{nabcouwe, ecappo, stevenf}@andrew.cmu.edu

***Abstract*: Gradient descent is a common optimization algorithm used to find the local minimum of a given function. This algorithm can be used to find solutions to manipulator inverse kinematics problems by finding joint configurations that reduce the distance between an end effector and the desired position. This paper outlines our application of the gradient descent function to achieve kinematics solutions for a hyper-redundant linkage system with the additional constraint of maintaining balance about a non-fixed base of support. We maintain system stability by using gradient descent to minimize both excessive joint velocities and offsets between the centers of mass of the manipulator and base. Applied to a hyper-redundant 16 module snake robot, this approach successfully used 10 modules for inverse kinematic solutions while balancing on a base of only 6 modules.**

***Keywords:* gradient descent, inverse kinematics, non-fixed-base linked manipulator, stability, snake robots, inspection**

I. INTRODUCTION

Fixed base linked manipulators are common robotic systems and subsystems, and solving the inverse kinematics to determine how to maneuver the end-effector of a manipulator chain to a desired position is a familiar and well-studied exercise. The large body of knowledge available for solving fixed base manipulator inverse kinematics provides a valuable starting point for exploring the problem of finding the inverse kinematics for "not-so-fixed" base manipulators, a property exhibited by the Carnegie Mellon University (CMU) Biorobotics Lab's hyper-redundant snake robot during inspection tasks [1].

The CMU snake robot has sixteen linked modules providing sixteen degrees of freedom. The snake excels at maneuvering through small spaces and is therefore used extensively for search and rescue and inspection operations. For situational awareness and for inspection, the end-portion of the snake provides a large base of support while the head and four following links lift off the ground and pan the camera, positioned at the head/end-effector of the snake, through the environment as shown in Figure 1.

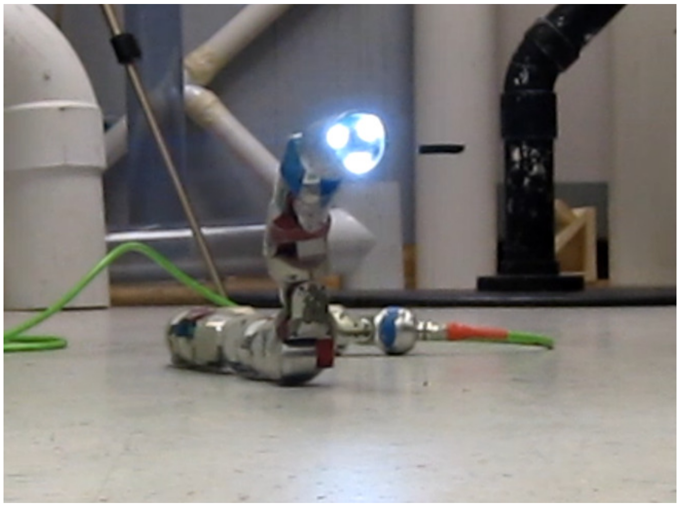

Figure 1. Snake robot performing inspection: tail and base links provide support while the head (containing the camera) and the body of the snake rise into the air to scan the environment.

This inspection method, known as "head-look", is very stable because it uses three quarters of the total number of modules to form a base of support. Because it uses only four modules for motion, however, it is limited and cannot look above or around occluding objects. In order to increase the maneuverability of the camera and the functionality of head-look to inspect past obstructing geometries, we would like to create a new head-look that uses a greater number of modules.

In order to achieve this flexibility, the center of mass of the active upper length, of the snake must remain over the base of support provided by the tail modules. The problem of interest, therefore, is how to maneuver the end effector of the robot to the desired inspection position while balancing the center of mass of the active links over the support base throughout the entire inspection process.

II. APPROACH

Because of the highly redundant and under-constrained nature of a snake robot, a geometric inverse kinematic solution to the problem is costly and infeasible. Numerical inverse kinematic solutions are required.

We chose to solve the inverse kinematics for positioning the head of the snake while maintaining balance by applying gradient descent. Gradient descent is a first order optimization algorithm: given a cost function with multiple input variables, gradient descent iteratively varies each input variable and finds a solution that will minimize the cost.

Gradient descent can be used to solve inverse kinematics by assigning the error between the end effector desired position and end effector current position as the cost and varying the joint positions of the manipulator. The cost function's reduction to zero equates with the end effector's arrival at the desired position.

The gradient descent technique can be made smoother by iterating through multiple desired positions along a trajectory. In order to bring the snake head to an elevated position we choose a trajectory that starts at the current position of the head module, approaches the center of mass of the base of the snake, and then curves up to a position directly above the base center of mass. This path is chosen to guide the active region of the snake through stable regions, either along the ground or over the support region provided by the base.

This trajectory generation requires the position of the head of the snake and the center of mass of the snake as inputs. These inputs are generated by a forward kinematic function. The forward kinematic function takes in each joint position and returns the position of each module in Cartesian space relative to a chosen origin module. This function already existed from prior work with the snake robot.

The center of mass function is a new development for the purpose of this project. The center of mass is estimated by assuming that each module is uniform in mass and that each center of mass is at the center of the module between joint vectors. The center of a mass of a set of modules is estimated as the average of the positions of all the modules in the set.

Forward kinematics, gradient descent, and a trajectory would be enough to move a manipulator with a fixed base. However, the non-fixed robot base adds additional challenges. If our manipulator—the active modules of the robot—moves outside the polygon of support provided by the base, the snake will overbalance. Overbalancing can also occur due to rapid dynamic motions, i.e., if the active modules move too quickly near the boundaries of the stability region.

To keep the manipulator stable, the X and Y distance between the center of mass of the base and the center of mass of the manipulator is penalized. The Z distance is not penalized, as this is the vertical axis (see Figure 2). Rather than use a binary penalty, which would be high if the manipulator center of mass were outside the defined safe region and low otherwise, we chose a proportional function to hedge against disturbances and uncertainty in the actual joint positions. To restrict joint velocities, the new joint positions attempted by gradient descent are compared to the original positions. Large differences correspond to a high velocity to get from one position to another, and differences over a certain threshold are penalized severely.

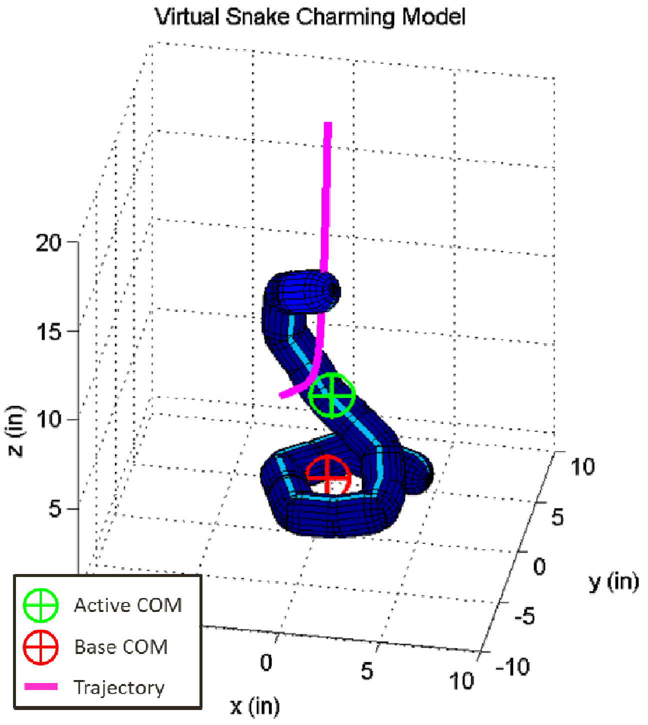

Figure 2: Virtual model of gradient descent execution

As the code iterates through points along the trajectory, the cost function effectively minimizes the end effector error, the center of mass offset, and the joint velocities, allowing the manipulator to stably rise to the desired position (Figure 2).

Once at the end of the trajectory, the top two modules can bend to sweep the head module across the area of interest.

Gradient descent can also be used rotate the head module, through modifying the cost function to include the additional information specifying the head's orientation. A trajectory of vectors can be generated representing desired end effector orientations. The difference between these vectors and the direction of the camera (the z-axis of the head module frame) can be added as another cost to be minimized. This method allows the snake to actively balance while scanning the area.

III. RESULTS

With a 16 link snake, we can successfully and consistently lift ten modules. The six modules on the ground form a stable, square-shaped base while the front ten links, including the head module, rise up into the air. Each module is two inches long and two inches in diameter, meaning a snake in a perfectly straight, vertical configuration could lift its head a maximum of 22 inches off the ground with ten active joints. Our method elevates the head 20 inches off the ground at its peak, where the loss is due to a slight curvature in the snake's body which properly positions the center of mass and helps maintain balance.

From a grounded position, the snake takes approximately 14 seconds to reach its peak. Gradient descent is thus a practical, timely method to employ real-time for head-look—in effect, turning "head-look" into "snake charming." At its height, the robot is balanced, and able to stably scan the area over a full 360° without fear of tipping. We successfully implemented both proposed scanning methods, the simple two module rotation as well as the full-body active balancing technique (Fig. 3).

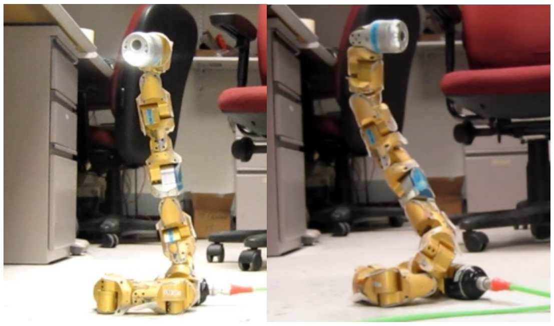

Figure 3: Snake charming, 10 modules lifted vertically with simple 2-module panning (left) and full-body active balancing (right).

Snake charming demonstrates significant improvements over traditional head-look (Table 1). By using 10 modules, we increase the peak achievable head height by 66%, while also actively monitoring and maintaining balance. In some tests we were even able to lift 12 modules up into the air, but the reduced base footprint takes a large toll on stability. For this reason we have settled on 10 modules as an optimal length for both height and balance.

IV. CONCLUSION AND FUTURE WORK

Snake charming is a functional, stable means of elevating a snake robot's head. There are currently no other means of lifting our snake to height of 20", and this solution opens the door to many new applications. From this high vantage point, our scanning range has significantly improved. Short obstacles no longer impair the user's line of sight. In addition, this technique could be used to raise the head into holes or gaps in low ceilings or overhangs.

Given this potential, the Biorobotics Lab would like to include snake charming in our current snake control software as soon as possible. Before then, though, there are improvements we would like to make to the motion. The relative weightings of the costs within our cost function can be tuned to minimize calculation time, and porting our code to a compiled language instead of to Matlab would greatly increase our computation speed. Additional parameters could also be added to the cost function, such as joint torque minimization. In this way, we could ensure that excessive strain is not placed on any one module, and the motion is guaranteed to be achievable in the real world.

Table 1: Results of Traditional Headlook and Snake Charming

Metric	**Traditional Headlook**	**Snake Charming**
Active Modules	4	10
Peak Height	12in	20in
Active Stability?	No	Yes

For operator convenience and field deployment, we would also like to add a user-controlled element to the motion. It would be helpful to autonomously elevate the head, but then give the user control of panning the head and rotating the camera. This would provide useful tele-operated functionality. The active balancing cost function should also be tuned to provide a smoother head rotation movement. The potential for snake charming is great, and its application will improve the snake robot's tele-operation and inspection capabilities.

V. REFERENCES

[1] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, H. Choset, "Design and Architecture of the Unified Snake Robot," *IEEE International Conference on Robotics and Automation*, pp. 4347-4354, 2012.