1-1-2010

Toward a Distributed Actuation and Cognition Means for a Miniature Soft Robot

Xiaoting Yang
University of Denver

Toward a Distributed Actuation and Cognition Means for a Miniature Soft Robot

---

Thesis

Presented to

The Faculty of Electrical and Computer Engineering

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Xiaoting Yang

November 2010

Advisor: Prof. Richard M. Voyles

Author: Xiaoting Yang
Title: Toward a Distributed Actuation and Cognition Means for a Miniature Soft Robot
Advisor: Richard M. Voyles
Degree Date: November 2010

Abstract

This thesis presents components of an on-going research project aimed towards developing a miniature soft robot for urban search and rescue (USAR). The three significant contributions of the thesis are verifying the water hammer actuation previous work, developing an estimator of water hammer impulse direction from hose shape, and creating the infrastructure for distributed cognitive networks. There are many technical issues in designing soft robots, in terms of perception, actuation, cognition, power, physical structure and so on. We are focusing on actuation and cognition issues in this thesis. We investigated water hammer actuation as an alternative system which provides a continuously distributed form of actuation results from water hammer effect. It is special because it is a soft actuation method. We generated some comparison experiments and verified the benefits of the water hammer actuation, and also designed our soft robot to be hose-like in order to utilize the water hammer actuator. For the cognition part, we first addressed and verified that the shape of the hose-like robot has impact on impulse direction from the water hammer actuation. And then we implemented an emulated synthetic neural network (ESNN) to analyze the direction of the impulse from the water hammer actuation. Then in order to achieve the long-term goal, we distributed the emulated synthetic neural network onto many embedded system boards to achieve a distributed cognitive network. The distributed nodes in the network are using Bluetooth communication.

In the comparison experiments between the active tether system and passive tether system, we can clearly see the benefits of active tether in momentum transfer and friction reduction. For example, in the drag test, with the water hammer actuation the burden that the tether can pull was increased by about 1.6 times. For the distributed cognitive network, we successfully built an emulated synthetic neural network on distributed embedded system boards. With the shape information as the inputs, the difference on outputs from the ESNN and the experimental results is less than 3%.

Acknowledgements

First of all, I would like to express my appreciation to my adviser Professor Richard Voyles for his invaluable guidance in my two years of graduate studies and research. I also appreciate the support of Professor Matt Rutherford and Professor Mohammad Mahoor for serving as the committee members in my final oral examination. Plus, I appreciate all my teachers that ever taught me and gave me help during these two years.

Secondly, I thank all my lab mates a lot, especially, Robert Nawrocki, Kang Li and Sam Povilus. Robert and I collaborated on the synthetic neuron network and hose shape and directionality analysis part of the research. Robert was instrumental in training the artificial neural networks described in this thesis, using MATLAB. Kang helped me on active tether versus passive tether experiments, and with the Zigbee wedge PCB layout design. Sam helped me on design of experiments for the water hammer system and with the force sensor platform.

Thirdly, I thank my school, University of Denver. I love the campus environment, I love the school culture, and I love everything here. The two years of graduate study has been one of the best memories in my life.

Last but not least, I would like to give my special thanks to my family, their love, concern and encouragement which drives me until the end of my study. Also, thanks so much to my friends and my host family; they make me not lonely here. I enjoy all the happy time with them.

Support for my work was generously provided by the National Science Foundation through grants IIS-0938196, CNS-0923518 and IIP-0719306 and by the NSF Safety, Security and Rescue Research Center.

Table of Contents

Chapter One: Introduction

Motivation

The research of the Collaborative Mechatronics Lab (CML) has focused on the field of search and rescue robotics for a long time. We have contributed to both theory and applications in this field. For example, the CML has developed hardware and software systems for “TerminatorBot”, a small size cylindrical robot that is able to manipulate objects and crawl over difficult terrain.

For my own research area, I am particularly interested in the field of soft robotics. First of all, it is a new and cutting-edge area for search and rescue robotics. Secondly, conformability is an important and helpful feature for search and rescue tasks. In search and rescue activities, robots always face complex and unknown situations, like collapsed buildings, narrow openings and so on [3]. Soft robots are able to deform themselves to fit the environment and this is both necessary and useful for these tasks. For example, the Jambot is highly deformable with its Jamming skin [4]; snake robots [5] promise the ability to reach areas that are difficult for conventional robots to reach. Finally, through polymer materials and fabrication processes, soft robots hold great promise for reducing the cost of deployment, particularly for multi-robot cooperative teams [6] [7].

As with hard robots, all soft robots must contain the basic robotic components of perception, actuation, cognition, power, and physical structure. Among all these components, actuation is one of the hardest parts to achieve for an all-polymer robot; soft actuators of sufficient power density, speed, and range of motion have not yet been made well for soft robots. Therefore, I began my study by examining novel actuation means that did not involve rigid components. The focus of my initial study was a new and promising actuation method called *water hammer actuation*. Water hammer actuation is a new research area for robotics systems. *Water hammer* is a phenomenon that occurs when a flow of water (or other fluid) through a pipe is suddenly stopped [8]. The momentum of the fluid that was in motion applies a force on the parts of the system that are at rest, causing a substantial increase in pressure. Some of the initial study of water hammer actuation was directed by D. P. Perrin and R. Howe at Harvard University [1] [2]. With their students, they developed an "active tether" system, which is a preliminary platform for experimentation. We realized this novel form of actuation, if properly controlled, could provide the basis for a practical, all-polymer robot.

The natural feature of water hammer actuation dictates the morphology to be hose-like, so that the flow is able to move inside. Therefore, our idea is to build a hose-like miniature soft robot. The polymer tubing that forms the soft robot body doubles as both the actuating means and the physical structure of the robot.

With this alternative actuator for the soft robot, we still need to be able to control it so that it is providing the desired directional actuation to the robot. A key contribution of my thesis is the confirmation of the hypothesis that the shape of the tubing conveying the fluid impacts the direction of the applied force at the point of momentum transfer. Stating this hypothesis in other words, "can we affect direction of motion of a hose-like robot body by varying the hose's shape and applying the water hammer effect?" Through the tests and simulations reported in chapter four, we proved the hypothesis above. Given this relationship between the soft robot's morphology and the water hammer propulsion, we must develop a cognitive architecture that can predict the propulsion direction from the perceived morphology information of the hose-like robot, eventually allowing us to control the direction of the propulsion.

To sense and analyze the shape information for the amorphous computational material hose-like robot body, we need to use a distributed sensing, actuation and computation method. Therefore we choose the distributed cognitive network to process the morphology information.

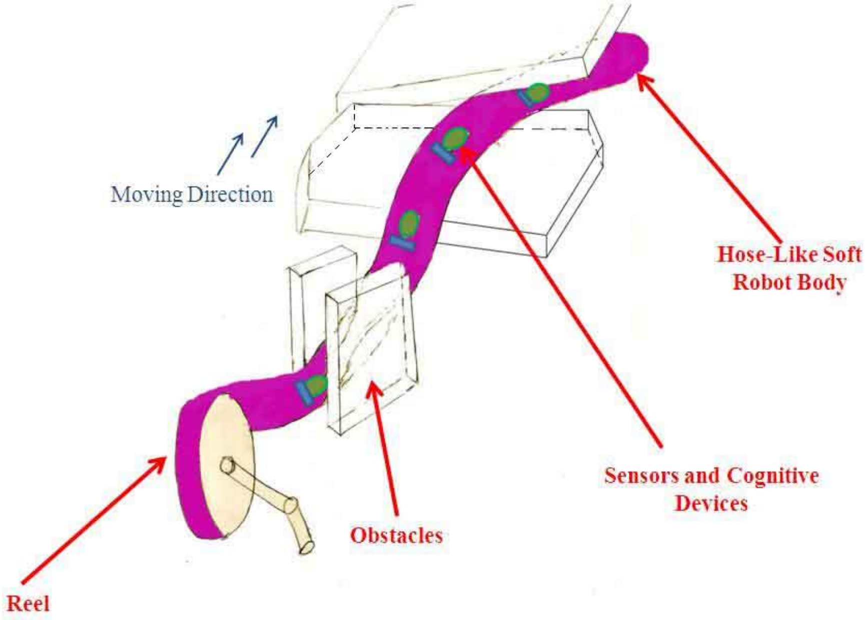


Fig. 1.1. Conceptual view of a hose-like miniature soft robot

Fig1.1 shows the "big picture" of the miniature soft robot for urban search and rescue that conceptually motivates. The whole body of the robot is constructed of soft materials, smooth and flexible. We can describe the information flow inside the soft robot actuation control system as: the sensors, which can be bend sensors or torque sensors made of soft electronic materials, send the sensing information to cognitive devices, where morphology information gets processed; the output of cognitive devices is the directionality information results from the impulse from the water hammer effect. The locomotion controller uses the prediction directionality to control the water hammer actuator, which is the locomotion direction of the soft robot. The information flow from sensors to water hammer actuator can be found in Fig. 1.2.

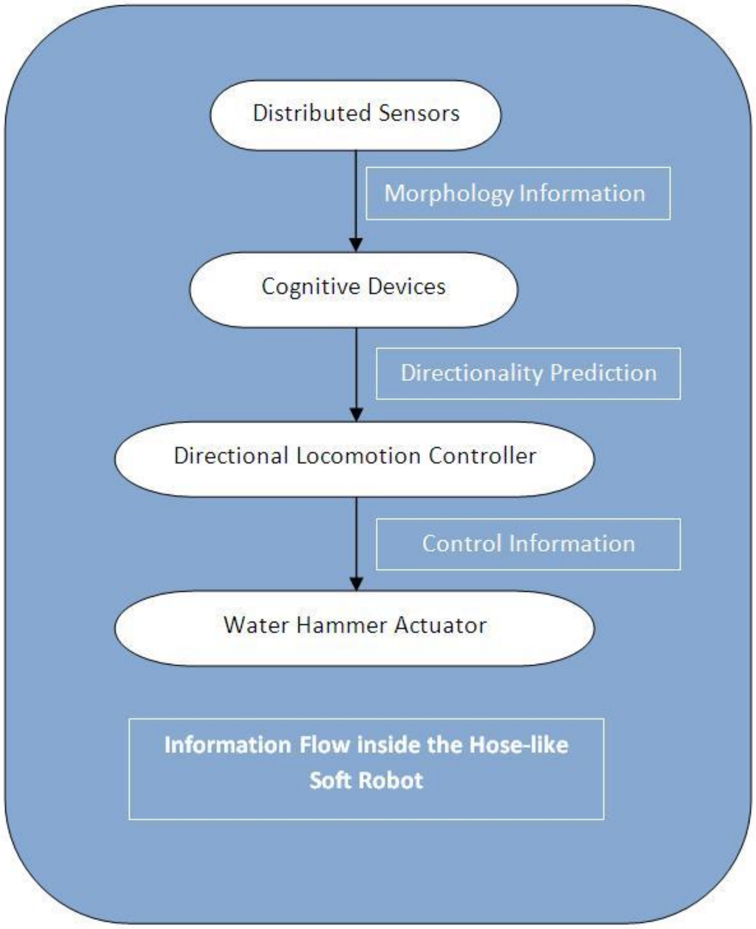

Distributed Sensors
Morphology Information
Cognitive Devices
Directionality Prediction
Directional Locomotion Controller
Control Information
Water Hammer Actuator
Information Flow inside the Hose-like Soft Robot

Fig. 1.2. Information flow inside the hose-like soft robot

Thesis Overview

Given this long-term goal and broad interest within the Collaborative Nechatronics Lab (CML), it is too big for one master thesis. I chose to focus on, as mentioned above, the actuation and cognitive parts. There are other works on perception, power and so on; I am just not going to address them in my thesis. I am not implementing the soft/polymer cognitive network, but setting up an emulation of a prototype cognitive network; I am also not controlling the direction of the water hammer actuation, but simply sensing and analyzing the morphology information for directionality prediction.

The two significant contributions of the thesis are: first verifying the effectiveness of water hammer actuation as active tether; and secondly, determining the directionality based on morphology and creating a distributed cognitive network to predict the directionality.

In order to verify the performance of the water hammer actuator, we first generated three comparison experiments between active tethered system and passive tethered system (Chapter Three). The experiments are: distance test, drag test and sliding friction force test. In this test comparing the active and passive tether, we found that the active water hammer tether can perform better under the test conditions than the passive tether for combating robot stoppage due to the tether, also for the higher dragging capability. These experiments also confirm the results from previous publications [1] and [2].

For the distributed cognitive networks, the hose-like soft robot is analogous to a snake on the ground; it is a finite element model, with each small element pushing in different directions. What’s the effect of all the elements pushing on the frictional surface?

We first addressed and verified that the shape of the hose-like robot has impact on impulse direction from the water hammer actuation. And then we built an artificial neural network to predict the direction of the impulse from the water hammer actuation on an emulated synthetic neural network. Given the laboratorial conditions right now, we couldn’t make sensors and cognitive devices with soft material, so we are using embedded system circuit UM001 as the hardware platform for the cognitive network. We implemented and distributed the artificial neural network onto many embedded system boards using Bluetooth as the communication means in the distributed cognitive network.

Prior Work

Small Size Robot in Search and Rescue

Urban Search and Rescue (USAR) refers to rescue activities in collapsed building or man-made structures after a catastrophic event, such as an earthquake or a bombing [3]. Urban search-and-rescue is considered a "multi-hazard" discipline, as it may be needed for a variety of emergencies or disasters, including earthquakes, hurricanes, typhoons, storms and tornadoes, floods, dam failures, technological accidents, terrorist activities, and hazardous materials releases. Engineers have been researching the field of robotics rescue for decades, especially after the World Trade Center (WTC) disaster. From the WTC disaster human-robot interaction applications, it has been confirmed that small robots have a unique capability to collect useful data. For example, they can aid in search and rescue because their diminutive size enables them to fit into tight spaces, openings, such as those found in rubble and in caves [9]. Field research shows that mobility is one main problem hindering effective use of robots in search and rescue missions [10]. This is why size matters for urban search and rescue tasks.

Soft Robot

Soft robotics is the branch of robotic study that deals with amorphous robotic devices constructed of soft materials. Soft robots can be conventional in morphology, as Trivedi addressed in [11], with articulated limbs and wheels such as the proposed electro rheological balloon animals shown in Fig. 1.3, or they can be unconventional forms, such as Amorphous Computational Materials and Molecule-like robots [12][13].

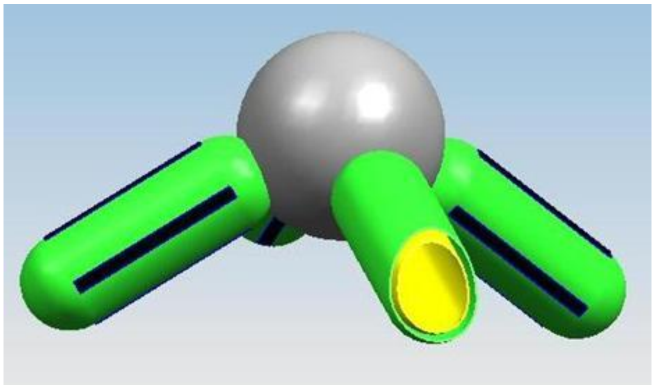


Fig. 1.3. Proposed soft robot with limbs

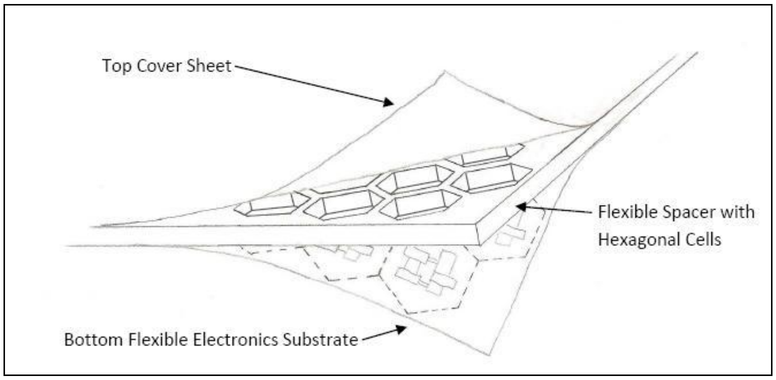

Top Cover Sheet
Flexible Spacer with Hexagonal Cells
Bottom Flexible Electronics Substrate

Fig. 1.4. Electro-rheological (ER) sensor/actuator cells in hexagonal combs made from soft polymers.

Soft robot is commonly thought of as soft, flexible, and compliant. All the sensors, actuators and other devices on the soft robot are all made of soft electronic materials. They have highly conformability; they can go through narrow opening, pass obstacles by their ability of compressing and flexibility, as in Fig. 1.1. These features make soft robot significantly more user friendly than traditional hard robots because humans are more accustomed to interacting with soft, animal-like creatures. Elephant's Trunk [14] robot has the ability of grasping various objects. Hatazaki and Konyo developed an active scope camera [15] which is using ciliary vibration drive mechanism. It can move smoothly while decreasing the sliding friction between the robot body and debris environment by vibrating the thin hair around the robot using the vibrating motor. The JamBot [4] is based on a novel concept, where the body is based on a number of balloons that can be inflated or deflated based upon requirements and environmental constraints.

Distributed Cognitive Networks

The concept of Cognitive Networks has been blooming in the networking research world for a while. Cognitive networks are motivated by complexity of the information. Particularly in wireless networks, there has been a trend towards increasingly complex, heterogeneous, and dynamic environments [16]. Cognitive networks were described in [17] as: "a network with a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals."

As we can see in Fig. 1.1, the morphology of the soft robot is decided by all the points along the hose-like body. In other words, in order to be able to detect the shape of the amorphous computational material, we need to gather the data from many of points along the body. This drives us to design a distributed cognitive network to analyze the data from the distributed points.

Soft Actuator

Soft actuators are the actuators that made of soft material, like electroactive polymers (EAPs). Electroactive polymers are polymers that exhibit a change in size or shape when stimulated by an electric field. One potential application for EAPs is that they can potentially be integrated into micro electro mechanical systems (MEMS) to produce smart actuators. Actuators based on electrochemically-induced volumetric changes in electroactive polymers (EAPs) have been used for artificial muscles and other applications [18]. They have features like high fracture toughness, large actuation strain and inherent vibration damping [19].

Electro-rheological fluid (ER) fluids are suspensions of extremely fine, non-conducting particles in an electrically insulating fluid. The apparent viscosity of these fluids changes reversibly by an order of up to 105 in response to an electric field. Viscosity changes can go from the consistency of a liquid to that of a gel with response times in the order of milliseconds. Cutkosky [20] and Voyles [21] demonstrated tactile sensing and actuation with electro-rheological fluids for robotic applications.

But the EAPs and ER are weak and slow, also they don’t have sufficient range of motion. None of them can provide high power density either.

Water Hammer Actuation

“Water hammer” is a pressure surge or wave when a fluid in motion is forced to stop or change direction suddenly. Water hammer actuation is a continuously distributed form of actuation resulting from an effect commonly known as “water hammer effect”. It is a common phenomenon that happens around lives. For example, water hammer is the phenomenon that causes the gasoline pump hose to jerk as the flow is automatically shut off or household pipes to rattle when a washing machine cycles.

Water hammer actuation helps robot in two aspects, one is the distributed momentum transfer, and the other is the distributed friction reduction

Some of the initial study of water hammer actuation was undertaken by D. P. Perrin in Howe’s group at Harvard University [1] [2]. They demonstrated the feasibility of harnessing this potentially devastating effect towards a useful application. Water hammer actuation is fascinating because there is no rigid part needed. More importantly, because of the forcing pulse in transferring along the whole pipe, the actuation is also distributed along all the hose-like robot body. Which means it can help reduce the friction on all the surfaces of the soft robot, also be able to help get rid of entanglements on any part of the robot body, like the situations in Fig 1.1. These excellent features of the water hammer actuator make it an ideal alternative form of the actuation for our miniature soft robot.

Tether Enhancement

Tethers have been used to assist robotics locomotion for a long time. The need for tethering systems on mobile robots can be seen in applications such as in ground, underwater and aerospace environments. The tether can act as a conduit for any subset of the following: power, data communication between remote controllers and the tethered system, gases or fluids supply [22]. Especially when a small robot descends into a pile of rubble, searches inside a disaster environment and so on, the tether serves as a safety line [23]. While due to the tendency to tangle around the obstacles, the increasing of drags of tethers etc., many tethered robots are stopped before they complete their task by having their tether get caught on an obstacle. Tethers also limit the depth to which the robot can go because a tether is of a finite length. In many cases search and rescue robots work in teams. In this case robots have been known to cross paths and tangle in each other tethers [24]. For these reasons, we consider that it is valuable to maintain a tethered robot but with some improvements on the tether.

Researchers have been trying to improve tether for a long time. For example, E. F. Fukushima, N. Kitamura and S. Hirose [25] developed autonomous tethered mobile robot systems using the ‘hyper-tether’ concept. Its basic function is to actively control the tether’s tension and length. A. Birk and C. Condea [26] set up glass fiber via a cable drum as cable deployment system on mobile robots, which makes the tether lightweight, thin, and very robust.

All these approaches improved the tether performance from different point of views, but none of them solves the problems includes increasing drag and a tendency to catch on obstacles.

Active Tether

In this thesis, I focus on the use of a continuously distributed form of actuation resulting from an effect commonly known as “water hammer”. The concept of “active tether” is first proposed by D. P. Perrin at Harvard University [1]. The active tether system consists of a mobile robot, an on-robot valve, and two pieces of water hoses for water flow. Besides supplying conventional power, active tether will supplement small robots with external driving energy caused by water hammer effect to meet with high power demands especially when robots get hindered. As mentioned before, water hammer occurs when a flow of water (or other fluid) through a pipe is suddenly stopped due to closure of a valve (or other means, as will be discussed); the momentum of the fluid that was in motion applies a force on the parts of the system that are at rest, causing a substantial increase in pressure.

Chapter Two: Water Hammer Theory

“Water hammer” is a pressure surge or wave resulting when a fluid in motion is forced to stop or change direction suddenly. In steady flow there is no change in conditions at a point with time, while in unsteady flow conditions at a point may change with the time. Consider the case of instantaneous stoppage due to the closure of a valve in a horizontal pipe (Fig. 2.1). For purposes of this discussion friction and minor loses can be ignored.

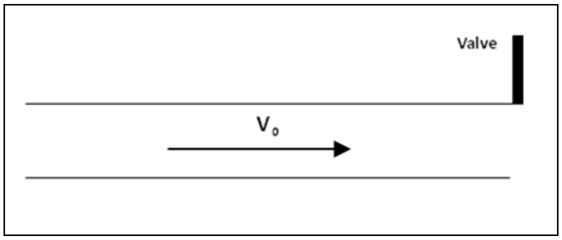


Fig. 2.1. Horizontal pipe with valve

The instant the valve is closed, the fluid immediately adjacent to the valve is brought from velocity V_0 to rest by the impulse of the higher pressure developed at the face of the valve, as in Fig. 2.2. The next layer is brought to a stop by this first layer and so on. Due to this chain of stoppages a pressure wave is created [27].

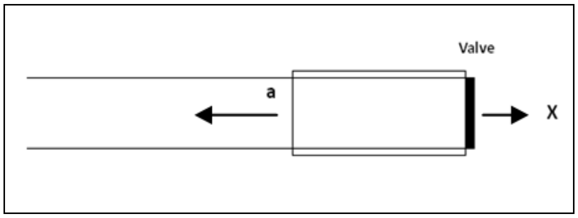


Fig. 2.2. Hydraulic transients after the valve is closed

The Navier-Stokes equations, for constant density and viscosity are:

$$\rho(\frac{\partial u}{\partial t} + u \cdot \nabla u) = -\nabla P + \mu \nabla^2 u + \rho g \qquad \text{(Eqn. 2.1)}$$

where g is the density of the fluid, u is the fluid velocity, P is the pressure, and μ is the viscosity of the fluid. For turbulent flow we can neglect viscosity and for analysis of a water hammer, the changes in pressure due to gravity is much less than the changes in pressure associated with the water hammer ($\nabla P >> \rho g$), leaving:

$$\rho(\frac{\partial u}{\partial t} + u \cdot \nabla u) = -\nabla P \qquad \text{(Eqn. 2.2)}$$

In the situation of a water hammer, the deceleration of the fluid will be much greater than the convection of momentum, that is $\frac{\partial u}{\partial t} >> u \cdot \nabla u$, which leaves:

$$\rho(\frac{\partial u}{\partial t}) = -\nabla P \qquad \text{(Eqn. 2.3)}$$

The gradient of pressure is a change in pressure over some characteristic distance, and the time derivative of velocity is the change in velocity divided by some characteristic time. Rewriting Eqn. 2.3 and solving for ΔP

$$\rho \frac{\Delta u}{t_c} = -\frac{\Delta P}{L_c} \qquad \text{(Eqn. 2.4)}$$

$$\Delta P = -\rho \Delta u \frac{L_c}{t_c} \qquad \text{(Eqn. 2.5)}$$

We can consider the speed of the water hammer wave to be the distance over which the fluid decelerates divided by the time is takes to decelerate the fluid.

Substituting $v_{wave} = \frac{L_c}{t_c}$ gives:

$$\Delta P = -\rho v_{wave} \Delta u \qquad \text{(Eqn. 2.6)}$$

which is exactly the maximum magnitude of a water hammer wave for rapid valve closures considering a small element of fluid.

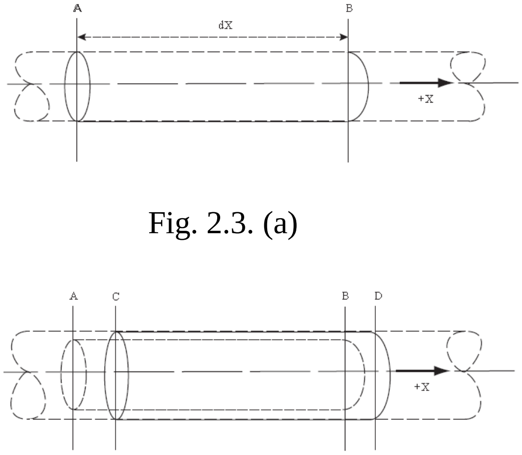


Fig. 2.3. (a)

Fig. 2.3. (b)

Fig. 2.3. Small element of fluid at (a) time t and (b) time $t + \delta t$

To determine the equation for the velocity of the water hammer wave, it is necessary to consider a small element of fluid in the pipe. In Fig. 2.3 a small element of fluid is shown at time t and at time $t + \delta t$. The fluid is assumed to be elastic: between time t and time $t + \delta t$ the element has compressed in length and expanded in cross section and has not necessarily maintained constant volume. By considering the conditions of dynamic equilibrium, continuity, and deformation of the tube, the velocity of the water hammer wave is:

$$v_{wave} = \sqrt{\frac{K}{\rho}}\left(1 + \frac{Dc_1 K}{Ee}\right)^{-\frac{1}{2}} \qquad \text{(Eqn. 2.7)}$$

where K is the bulk modulus of the fluid, D, E, e are the diameter, Young's modulus, and thickness of the tube, and c_1 is a constant determined by the constraints on the deformation of the pipe in a longitudinal direction [28].

The analysis so far has also assumed that the magnitude of the water hammer wave is constant at the maximum value. This assumption is valid for instantaneous valve closures, but actual valve closures involve a finite amount of time. When the water hammer wave reaches the end of the hose, which is maintained at constant pressure, the water hammer wave will be reflected and a negative pressure wave will travel in the opposite direction. For slower valve closures, the change in pressure at a given position is the sum of the initial pressure wave from the valve and the negative pressure wave reflected from the reservoir. For valve closure times less than $\frac{2L}{v_{wave}}$:

$$Length_{\max \Delta P} = (1 - \frac{T}{\frac{2L}{v_{wave}}})\Delta P_{\max} \qquad \text{(Eqn. 2.8)}$$

where T is the valve closure time. For a 30 m tether with $v_{wave} = 1100 m/s$, this corresponds to valve closure times completed in less than 0.05 s. To maximize the effect of the water hammer wave on the tether, we would want to minimize the valve closure time [29].

Chapter Three: Effectiveness of Water Hammer Actuation as Active Tether

"Active tether system" is tethered robotics system that utilizes the water hammer effect as the part of the actuation for the robot [1]. The platform for experiment consists of a mobile robot, an on-robot valve, two pieces of water hoses for water flow and an air bleed on the output water hose for minimizing the recoil force from water hammer effect, as shown in Fig. 3.1. In other to verify the effectiveness of the water hammer actuation, I design three comparison experiments between the active tethered system and the passive tethered system.

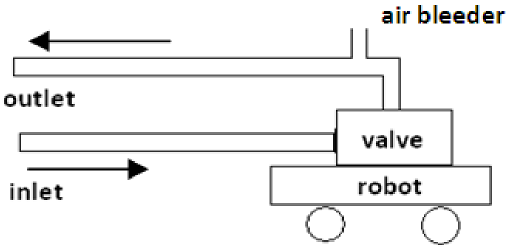


Fig. 3.1. Active tethered system

Experimental Setup

In this experiment, we choose to use a 1/10 scale electric 7.2V battery powered 4WD monster truck (3851-2, Heng Long Plastic Toys Co., Ltd.) as a mobile robot, which is set up with either a normal passive tether or water hammer device to consist of the passive tethered system or active tethered system.

Passive Tethered System

The focus of this research is to compare a passive and active tether. To get a base line test for an example tether we simply attached a cable to the robot that had no purpose but to perform as a passive tether, the robot was still powered by the on board battery. To have a fair baseline for a passive tether system we need to choose a line that could be used in an actual tethered robot, which should be flexible, not easy to be stuck, smooth surface, small diameter and light. Based on this requirement, we choose to use the High-flex Mini Diameter Data Cable (86302CY SL005, Alpha Wire Company), it is 22 AWG cable with diameter of 5.4mm and oil resistant PVC jacket.

Active Tethered System

The active tethered system consists of the remote controlled truck, a fluid control valve, two pieces of water hose, an air bleeder and tanks. In our experiment we use a general purpose solenoid valve (71215, Parker Hannifin Corporation), which is a 2-way, 24V DC, direct operated model. In order to control the valve generating water hammer effect, we programmed UM003 motor control board from CML lab, whose MCU is ATMEGA 128 (Atmel Products -Microcontrollers – AVR), to turn the valve on and off at 8 HZ. For the water hose, we want it be high pressure-resistant and with small diameter. Our choice is the High-Pressure Clear PVC Tubing (52375K12, McMASTER-CARR), whose inside diameter (ID) is 6.35mm and outside diameter (OD) is 11.11mm., the whole experimental platform installation with valve attached on is as shown in Fig. 3.2. In addition, to keep a constant input pressure through the hoses, we used two five gallon tanks and air compressor to provide around pressure 90 psi water.

The material of water hose is much heavier than the passive tether per unit length in our experiment, and also, the solenoid valve with the plastic board is around 2kg, these all increase the burden of the mobile robot. This would only be a problem if it was found that the active tether robot could not go as far in our tests.

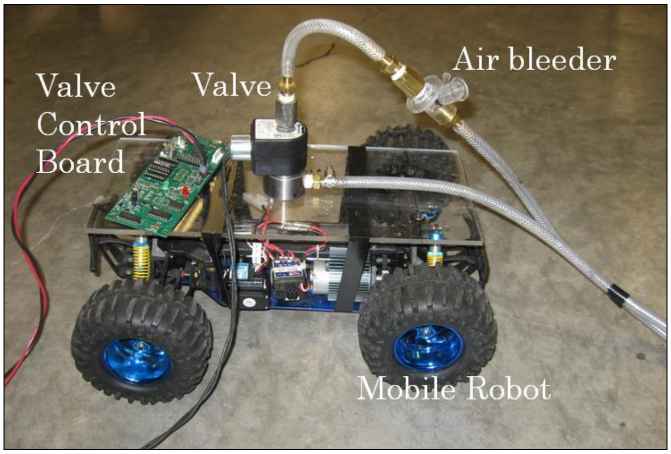


Fig. 3.2. Active tethered system for experiments

Experimental Implementation

There are two types of performance testing for the robotics field: one is testing under laboratory conditions, and the other is field testing. In this chapter, we only operate the experiments under our lab conditions, while we are aiming to make fair and convincing comparable tests between the passive tethered system and active tethered system.

As described in the introduction part, USAR tasks are those robot assisted activities after a disaster in an urban environment. For measuring the performance of robotics search and rescue systems, one approach is to develop standardized or reproducible tests similar to the RoboCup and AAAI rescue robot competitions which rely on the National Institute of Standards and Technology (NIST) test course [30].

In an urban search and rescue mission, a tethered robot can be affected by any unpredictable environmental events, like being held by a nail, getting stuck around a relatively heavy and big obstacle, falling into a hole and so on.

For this chapter, we simulate two scenarios of the possible disaster events in USAR on the tethers: one is getting tangled around some corners, and the other is being pinned under obstacles during the collapse. To simulate the first situation, we design a zigzag path inside a workshop, where the robot needs to pass many fixed table feet corners. With the continuous increase of friction, the tether will eventually become locked. Also we set up some heavy water bottles on the floor, and wrap the tether around the bottles, to see how many bottles it takes to stop the robot pulling its tether; and then measure the sliding friction with different number of bottles. For the second simulation, we choose to quantify the drag capability of the tethers with our experiment device, which is actually testing the drag potential during a collapse. In all three experiments, to develop fair comparisons, we apply the same experimental conditions such as the wheeled robot, floor, etc. to keep friction the same, except for different tethers for the two systems.

Distance Test Comparison

During the robotics search and rescue field application, normally the tethered robot stops due to the tether getting stuck. The distance test is designed to compare tethers' capability in getting rid of tangles.

In the distance test, we generated a specific path (Fig. 3.3) in a workshop in ECE department at the University of Denver. There are four tables that fixed on the floor, each table is 30 inches * 64 inches measured on the table feet, and the distance between tables can be found from Fig. 3.3 also. The tethered robot needs to pass many table feet corners, and the mobile robot will come out from the same position with the same fully charged car battery.

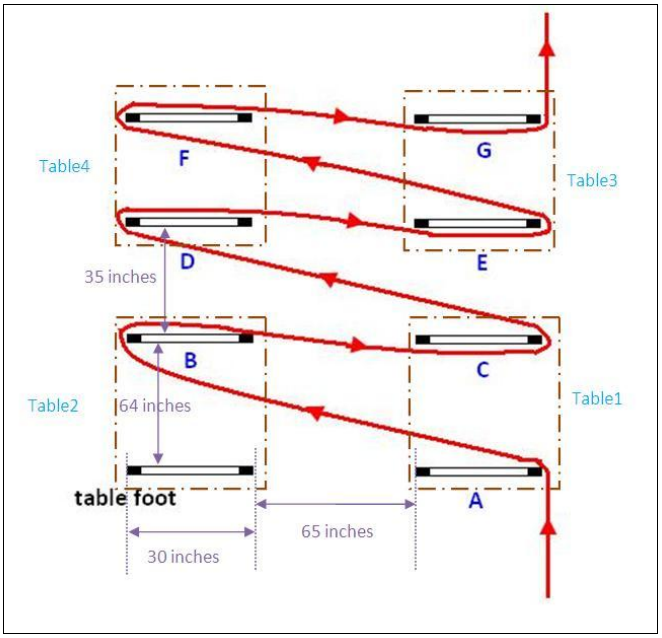


Fig. 3.3. Aero view of robotics moving path for distance test comparison

We execute the experiment ten times with the passive tethered system and ten times with the active tethered system. The results turn out to be that the passive tethered system can normally pass the first three corners (A, B and C), but will stop after corner C, as in figure 3.4 (a). The final result is that the passive tethered robot stops 8.89 m away from start point on average, as shown in figure 3.5(a).

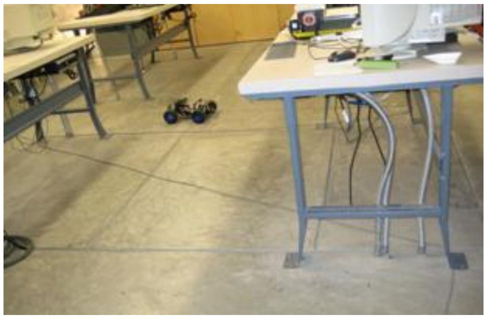


Fig. 3.4. (a)

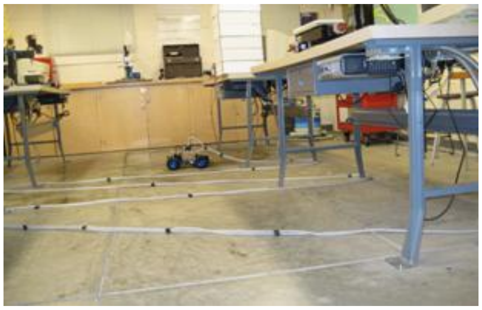
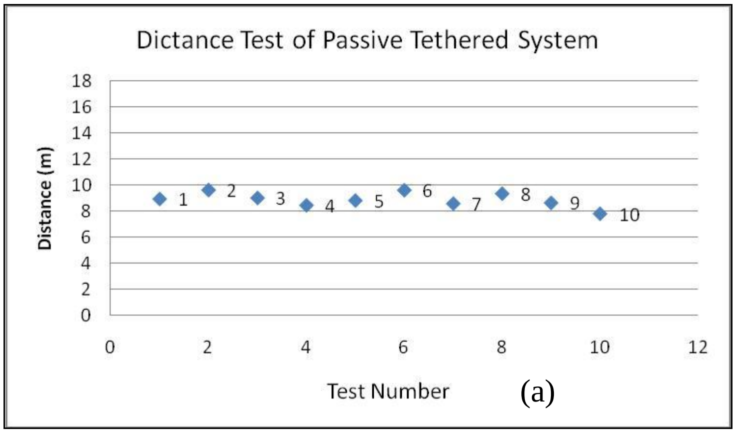


Fig. 3.4. (b)

Fig. 3.3. Distance test comparison result (a: passive tethered robot; b: active tethered robot.)



(a)

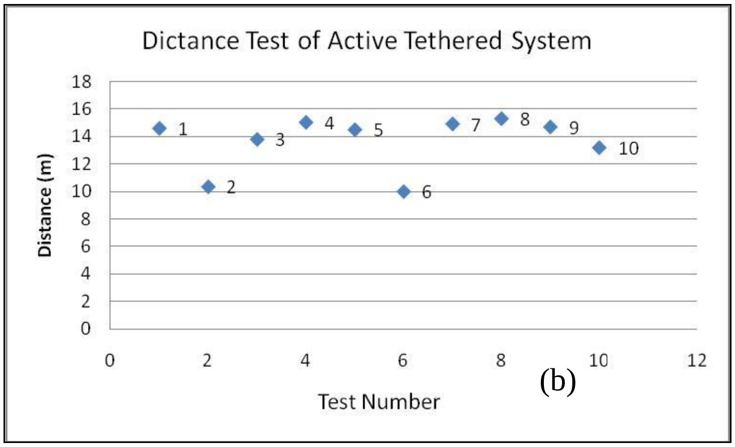


Fig. 3.5. Distance from start point (a. passive tethered system; b. active tethered system)

For the active tethered system testing, the mobile robot passed the fifth corner (corner E) eighty percent of the time, as in figure 3.4 (b). On average the robot stops at a position that is 13.64 meters away from start point, as shown in figure 3.5 (b), runs 2 and 6 of the test are almost 4 meters less than other test results, the reason is that the robot stopped after the fourth corner instead of the fifth. This shows some inconsistency in the active tethered system. If we delete the data from runs 2 and 6, the mean distance is 14.51 m.

From this test, we can see that both passive tethered and active tethered system will stop eventually due to the wrapping around the of table feet by the tether, which is actually due to the high friction results from the sharp corners of the table feet in this path. While the actuation from the water hammer effect, especially the jerks of the pipe, sharply reduces the friction, the active tether system can go on average 1.8 times the number of turns than the normal tethered system. This result proves the water hammer effect to be a powerful way to actuate a tether and reduce friction.

Drag Test Comparison

For USAR missions, the robots may need to be operated inside a collapsed building, descend into a pile of rubble, etc. In those situations the robotics tethers are highly likely to be pinned by bricks, planks or rocks; therefore the drag strength turns out to be an important characteristic of robotics tether equipment.

There are also some drag tests in [1]. In our experiment, we will duplicate a pulling weight test. Perrin [1] tested the robot's ability to pull the additional weight. With the vehicle and the tether in a line, a block of lead and a brass weight weighing a total of 6 kg was placed on top of the tether providing increased drag.

In our experiment, we will see how much additional weight the tethers can pull for the two different systems. We generate the test on the smooth floor in the same workshop at the University of Denver. We put the solid iron bars and weights onto the tethers 15cm away from the mobile robot at the same place above the ground, and the center of gravities of bars and weights are the same. We will measure the maximum weight that the tethered robot can pull. The experiment pictures can be found in Fig. 3.6 and Fig. 3.7 for the two systems.

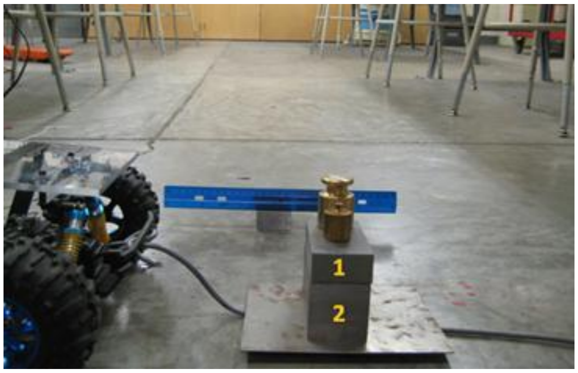


Fig. 3.6. Drag test for passive tether

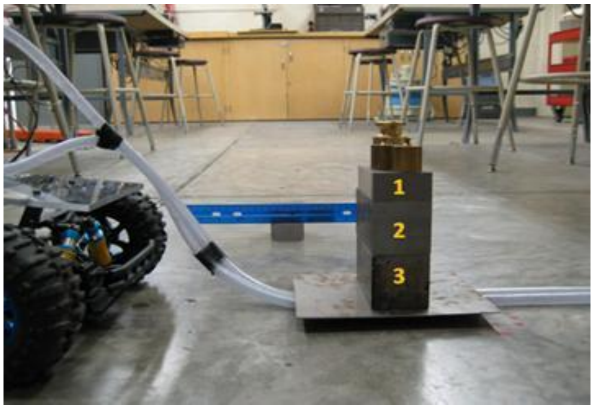


Fig. 3.7. Drag test for active tether

The test result is the passive tether can pull one small iron bar (1.1kg number 1 in Fig. 9), one big iron bar (2.3kg, number 2 in Fig. 3.6) , one 300g weight and one 100g weight in the situation shown in Fig. 3.6, thus the total weight is 3.85 kg (the bottom sheet is 0.05kg).

The test result for active tethered system is the active tether can pull one small iron bar (1.1kg number 1 in Fig. 3.7), two big iron bars (2.3*2kg, number 2 and 3 in Fig. 3.7) , one 300g weight and two 100g weights, thus the total weight is 6.25 kg (the bottom sheet is 0.05kg).

From this test we see that the jerks of the water hammer effect will increase by about 1.6 times the burden that the tether can pull. The pulling force may vary in different situations, such as different friction coefficient, different distance from the mobile car. Nevertheless the results in [1] of increased drag pull by the active tether seem to be accurate.

At the same time, the results from the drag test prove the benefits of active tether on pulling capability.

Sliding Friction Force Test Comparison

The sliding friction force test is designed to compare the tethers' capability to get rid of high friction.

In the sliding friction force test, we wrapped the tether around many round water bottles to test how many bottles it takes to stop the robot pulling its tether. Additional, we measured the force of sliding friction with different numbers of bottles for the two systems by digital scale (Berkly, TEC 100 LB Digital Scale, www.berkly-fishing.com). The water bottles are cylinders with diameters of 25cm, containing 18.9 L (5 gallon) water. The methods of wrapping around different numbers of bottles can be found in Fig. 3.8. The distance between two adjacent bottles is 60cm, and the center of gravities of the round bottles are on the same line. Other conditions are all the same for the two systems. The mobile robot will come out from the same position with the same fully charged car battery.

For the two systems, because of the different material of the tether, the friction coefficients will be different. Therefore with the same number of bottles, the force of sliding friction will also be different. After running the comparison tests, we then measure the forces of sliding friction that are needed with different number of bottles for the two systems. The force is measured by a digital scale, with 0.001 kg precision and 100 lb/ 45.5kg maximal range. The value of sliding friction forces can be found in Table 3.1. The experiment pictures can be found in Fig. 3.9 (a) and Fig. 3.9(b) for the two systems.

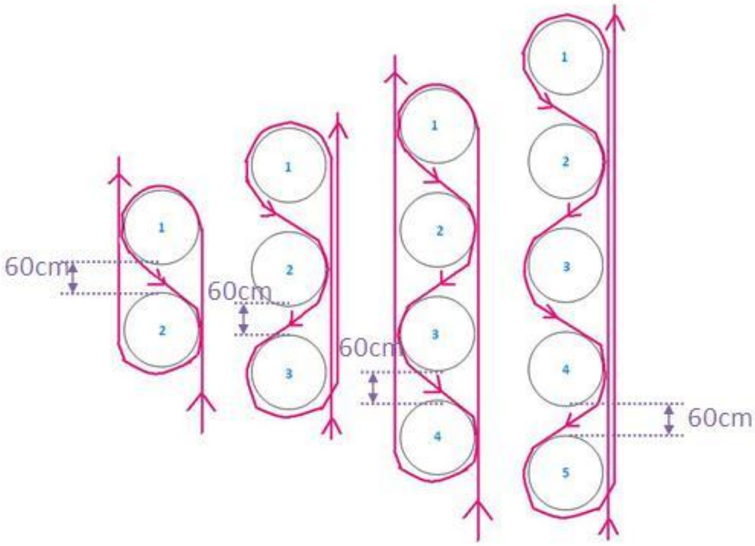


Fig. 3.8. Aero view of tether path for sliding friction force test

Table 3.1. Force of sliding friction from digital scale

	2 bottles	3 bottles	4 bottles
Passive	0.7kg	1.8kg	3.0kg
Active	1.9kg	2.9kg	4.5kg

For the passive tethered system testing, the results turn out to be that the mobile robot is able to pull the tether wrapped around two bottles, but it will stop at the three bottles one, as in figure 3.9 (a). From the data in Table 3.1, we can say that the passive tethered robot can conquer the force of sliding friction between 0.7kg*9.8kg/m^2 and 1.8kg*9.8 kg/m^2, which is about 6.86N to 17.64N.

For the active tethered system testing, the mobile robot is able to pull the tether wrapped around first four bottles, but it will stop at the five bottles path, as in figure

3.9(b). From the data in .1, we can say that the active tethered robot can drag the sliding friction force a little above 4.5kg*9.8kg/m^2, which is 44.1N.

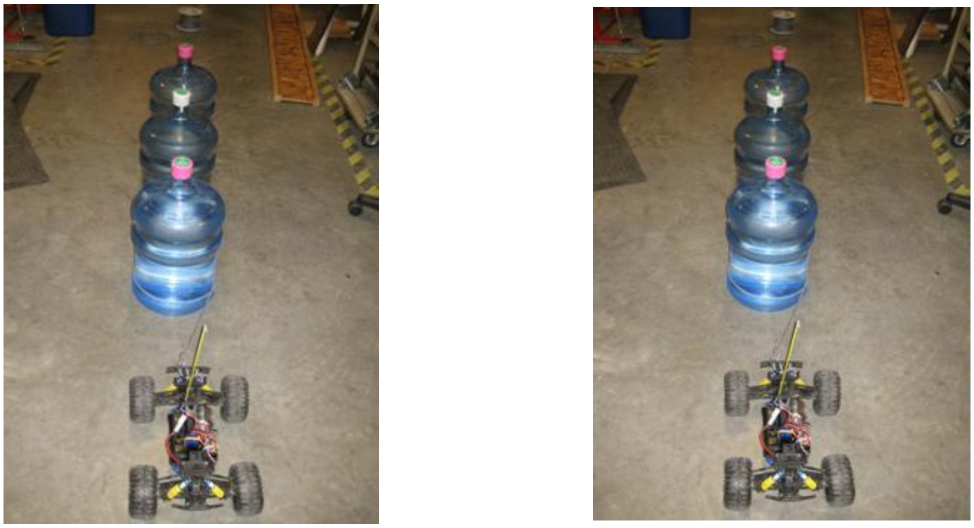


Fig. 3.9. (a)

Fig. 3.9. (b)

Fig. 3.9. Sliding friction force test comparison (a: passive tethered robot; b: active tethered robot.)

From this test, we can see that both passive tethered and active tethered systems will stop eventually due to the high friction. The actuation from the water hammer effect, especially the jerks of the pipe, sharply reduce the friction, the active tether system can go two more bottles than the normal tethered system. This result proves again the benefit of water hammer effect in dealing with entanglement.

We can conclude from the tests above that the active tether provides a new continuously distributed form of actuation for robots; the potential impact of this active tether is significant. It helps and benefits in urban search and rescue missions based on the discussion and experiments in this thesis.

Chapter Four: Estimates of Water Hammer Force Impulse Direction Based on Distributed Morphology

In this chapter, we verified our hypothesis about the impact of tether shape on impulse direction from the water hammer effect. We ran physical experiments to verify the behavior and developed a simplified finite element model to help explain it.

Shapes and Directionality Hypothesis

For a hose-like soft robot, we need to eliminate the wheeled robot from Figure 3.1 so that only the tubing is actuating itself. (Feller et al. discuss ways of eliminating the valve in [2].)

We can simply observe one phenomenon in some tests, which is the shape of the hose impacts the direction of propulsion of the valve, which is consistent with a lumped, finite element model of the fluid in the hose. Therefore, we started to investigate a hypothesis, namely that the shape of the active tether results in directed propulsion. In another word, the hypothesis can be restated as, can we affect direction of motion of a robot placed at the end of a hose, by varying the hose's shape and applying the water hammer effect.

To start the investigation, we first generated a preliminary test. For a preliminary test of this hypothesis, the hose was arranged into two distinctively different shapes denoted as Shape 1 and Shape 2 in Figure 4.1. The direction result can be found in Table 4.1. The valve moved about 4 to 5 cm within 20 second interval.

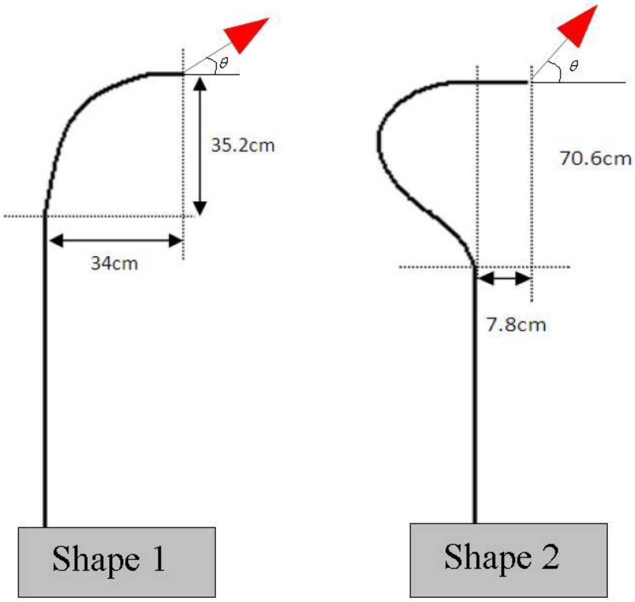


Fig. 4.1. Initial pipe shapes for directed propulsion due to water hammer effect experiment. (Arrows indicate direction but not the magnitude of propulsion.)

Table 4.1. Directional angle values for the two shapes

	Shape 1	Shape 2
1	38.2 ^{o}	65.2 ^{o}
2	35.2 ^{o}	68.0 ^{o}
3	43.2 ^{o}	70.1 ^{o}
4	38.4 ^{o}	64.0 ^{o}
5	39.8 ^{o}	63.6 ^{o}

6	42.4^{o}	62.3^{o}
7	40.5^{o}	66.0^{o}
8	36.0^{o}	71.8^{o}
9	40.5^{o}	66.3^{o}
10	37.1^{o}	64.7^{o}
Mean	39.13^{o}	66.19^{o}

From Figure 4.1 and Table 4.1, we do see two very different directional angles come from the two distinctively different shapes. In other words, we are more confident that there is a strong relationship between the directionality of propulsion due to water hammer and the shape of the hose. This effect could be harnessed either in aiding the steering or perhaps as a sole source of directionality of movement. Therefore next we need to investigate the impact of pipe shapes on directionalities with more accurate experiments.

Experimental Setup for Shapes and Directionality

We attached the valve to a stationary (mounted to a large metal plate) force sensor that would measure the force impacting on the valve through the water hammer, in the X and Y direction. And also, the input point of the pipe is also fixed on the same piece of metal plate, as in Fig. 4.2.

The experiment included two 5-gallon tanks with about 80 psi output, and the pipes were “High-Pressure Clear PVC Tubing,” with inside diameter of 6.35 mm, outside diameter of 11.11 mm, and wall thickness of 2.38 mm. The valve was a general purpose, 2-way, 24V DC solenoid valve with a weight of about 2 kg, operated at 8 Hz. The force sensor was an ATI Industrial Automation, Gamma Model, with sample rate of 2000 Hz and sensing range of 7.5 LBF. The sensor registers force in X, Y, and Z directions, however, for our purposes we disregarded the Z direction as the tube existed in a planar space.

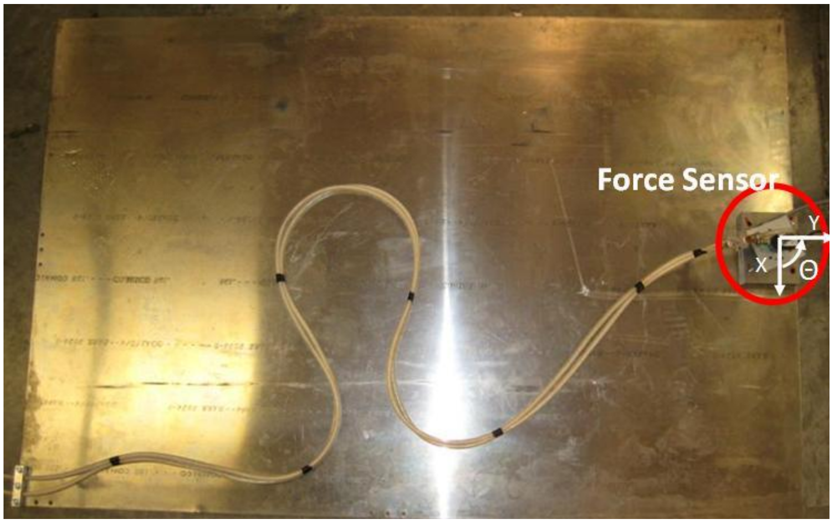


Fig. 4.2. Experimental setup for shapes and directionality tests

Force Sensor Data Calibration in MATLAB and Test Results

As mentioned before, the force sensor is working at sample rate of 2333HZ. On the one hand, the high sample rate is very necessary in order to be able to detect the very fast water hammer effect. On the other hand, with this high frequency, we will get around ten thousand data points just in five seconds. This brings us a problem, how should we deal with the data? How can we get the directional angle when the maximal pulse applies on the force sensor?

To fulfill the requirements above, we think of programming a data analysis software using MATLAB. Because the impulse happens so fast, we cannot get enough samples for the peak value in one impulse. What we do is we set the program to finds the average angle and magnitude above the threshold of the water hammer effect on the force sensor. The basic idea is to plot the data on a polar coordinator, and then set up a threshold on magnitude for isolating the peak (Default is 85% of peak). The final value of angle and magnitude are all the average value of points above threshold. As in Fig. 4.3, the threshold on magnitude is set to be 1.5 LBF, so the points marked as red are the points that used in calculation. The calibration result in Fig. 4.3 matches the shape number 19 in Appendix A.

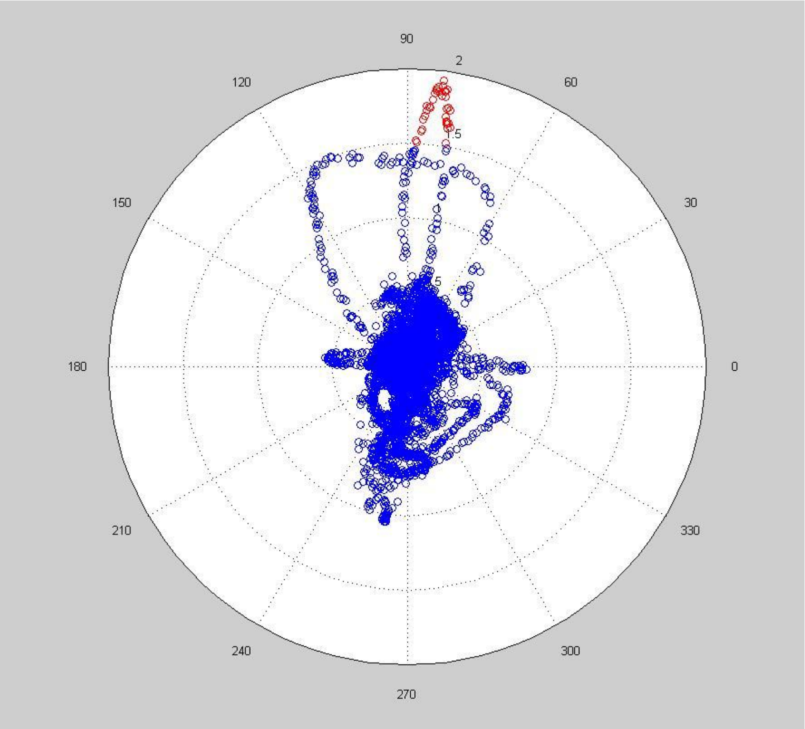


Fig. 4.3. Experimental setup for shapes and directionality

The experiment included measurement of 19 distinct shapes. The 19 shapes can be found in Appendix A. Just using the program introduced above, we get the direction data in table 4.2, which presents the data obtained with 19 different shapes, each with a distinct force vector (the values are given as the angle calculated from the X axis in the counter-clockwise direction).

Table 4.2. Experimental results for the 19 shapes

Shape Number	Directional Angle (^{o})
1	91.931
2	95.0267
3	93.6791
4	98.5166
5	88.3794
6	68.7642
7	66.0154
8	54.1671
9	86.6102
10	98.8555
11	85.0278
12	66.6876
13	87.383

14	88.9315
15	89.424
16	80.7907
17	98.3098
18	106.9707
19	83.2079

Computer Simulation and Comparison of Experimental and Simulation Data

A simplified, naive finite element model was constructed and simulated to model the resultant force vector due to the shape of the hose, acting on the front-mounted object. For the purposes of the simulation, the hose was considered to comprise of a finite number of elements, each in direct contact with adjacent elements. For each of the shapes, the resultant force vector, obtained by extracting point(s) of the greatest magnitude (in XY plane) of all of the impacting forces recorded, was matched with 20 distinct points on the hose that were obtained from pictures taken of the shape before the application of the water hammer (throughout the experiment the shape would slightly change due to the forces generated by the effect). Figure 4.4 demonstrates an example shape used for this experiment. Each finite element had a point placed at its center. These individual points were connected, in the XY-Plane, resulting in an approximate representation of the shape based on the formed angles in a global coordinate frame. Figure 4.5 illustrates the concept.

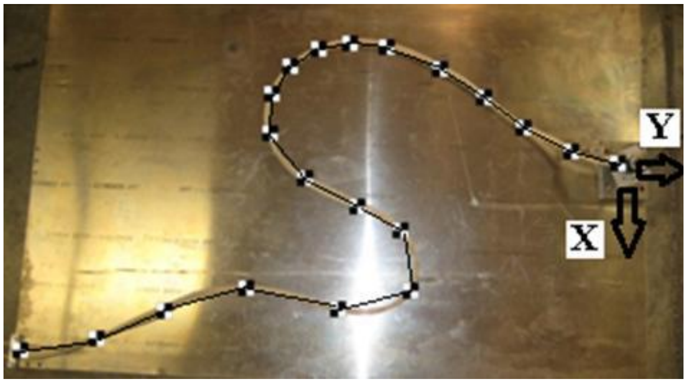


Fig. 4.4. Checker marks used to represent the shape of the hose.

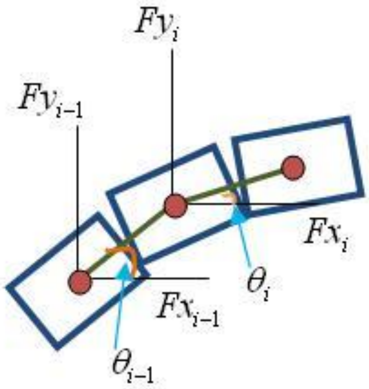


Fig. 4.5. Finite elements used to describe the serpentine shape of the hose, along with their corresponding Fx and Fy components in the world frame.

$$\begin{aligned} F_{x_i} &= f * \cos(\theta_i) + d * Fx_{i-1} \\ F_{y_i} &= f * \sin(\theta_i) + d * Fy_{i-1} \end{aligned} \qquad \text{(Eqn. 4.1)}$$

We are only considering 1st order approximation of the angles. Each subsequent element had its force calculated based on its X and Y components with a scaled force of the previous component added. The scaling factor *d*, left as a variable, was adjusted through various trials, between values of ‘0’ and ‘1’. Increasing the scaling factor corresponded to an increase in the influence of the previous component(s). Equations 4.1 shows the formulae used to calculate the individual force components. *f* is a constant equal to ‘1’.

Through the tests with varying *d* factor, Robert found out that increasing *d* from ‘0’ to ‘0.2’ does not produce a noticeable change in the vector. However, as *d* is increased beyond ‘0.5’, some of the shapes result in greatly varying vectors.

In order to verify the accuracy of our computer simulation we used the shape data obtained during the lab experiment and matched our resultant force vector from the simulation with the resultant force vector obtained from the force sensor. This fitting process involved modifying the parameter d in Equations 4.1 (the influence of individual finite elements on consecutive elements) until the difference was satisfactorily small.

We calculated some different values of *d* factor, and compare the average error compare to the experiment data, for *d*=0.1, the average error is the smallest, which is just under 8% (we also measured the error for values slightly higher and lower than 0.1 but the error was greater in both cases). However, increasing *d* from ‘0’ to ‘0.2’ does not produce a noticeable change in the vector; for *d*=0.5 error increases to about 12%. Also through some comparison tests with different spacing points along the length of the hose, we figured that the spacing the points (equivalent with placing the bend sensors) closest to the front-mounted valve results in the closest approximation of the resultant force. The finding indicates the direction of propulsion is only mildly affected by the overall shape of the hose and the greatest influence is due to the direction or shape of the very end of the hose.

It can be seen that spacing the points (equivalent with placing the bend sensors) closest to the front-mounted valve results in the closest approximation of the resultant.

The conclusion of our experiments is that we have two different sources (force sensor experiment and computer simulation) of information that all lead in the same direction. This, we believe, validates our hose shape and water hammer propulsion directionality relationship statement.

Chapter Five: Distributed Infrastructure for Soft Robot Cognition and Communication

The motivation of this chapter is that although we are able to analyze the directionality with our force sensor and computer simulation method in Chapter Four, it's impossible to test all the possibilities of the shapes. In order to predict the directional result of the water hammer impulse given any arbitrary hose shape, we want to build an intelligent model to generalize the information for different shapes. I chose to apply an Artificial Neuron Network (ANN) to help us predict the direction based on the morphology.

In this chapter, I describe how we first trained an ANN on our impulse/shape dataset using Matlab and the back propagation learning algorithm. Given these weights, which implement an estimator of the impulse direction from the distributed shape of the hose, we ultimately want to construct a distributed synthetic neural network, using polymer electronics, to realize the directional estimation. Since this is beyond the current capabilities of the CML, my goal is to emulate the execution of SNNs on a distributed array of UM003 embedded system boards. The balance of this chapter describes the development of the distributed cognitive network used to implement the impulse direction estimator from distributed morphology information provided by simulated polymer bend sensors along the proposed hose-like soft robot body.

ANN Introduction

An artificial neural network (ANN) is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data [31].

The general form of the output of one neuron is like this:

$$y_i = f_i(\sum_{j=1}^{n} w_{ij} x_j + b_i) \qquad \text{(Eqn. 5.1)}$$

Where $f_i()$ is the activation function, y_i is the output, x_j is the jth input to the node, w_{ij} is the connection weight between nodes i and j , b_i is the bias of the node.

Because the ANN is able to learn the complex relations between inputs and outputs, it is a potential tool to help us model the complex relations between shapes and directionality.

Synthetic Neural Network and Emulated Synthetic Neural Network

In the Collaborative Mechatronics Lab, we have chosen the term “Synthetic Neural Network” to refer to the parallel hardware implementation of an artificial neural network using discrete neurons fabricated with polymer electronics [32] [33]. A number of research groups have tried different ways to implement neural networks in hardware, for example, [34] [35] presented possible designs for realizing neural behavior utilizing transistors.

The goal of my thesis is not to fabricate a real synthetic neural network, but to develop the infrastructure for emulating arbitrary synthetic neural networks on distributed embedded system hardware.

ANN Training and Simulation

Ultimately, the data collected from sensors will be used by our Artificial Neural Network to extrapolate the information about the shape of the hose. Since my thesis does not address the perception component of an all-polymer soft robot, I simulated the behavior of the shape-determining bend sensors using a camera and manual measurements of the tangent to the tubing. Using MATLAB, a neural network was trained with the comprehensive data (or 20 individual points used to describe the shape of the entire hose for each of the 19 shapes) from our water hammer experiment. The network that was successfully trained consisted of three layers and 12 neurons in the hidden layer, and the resulting error was below 0.02%. Of the 19 tests all of the samples are in training set, no separation between training and testing sets was done.

Since the goal of other researchers in the lab is to build actual prototypes of a hose-like polymer robot and its components, there is interest in simplifying the components as much as possible for early-stage prototypes. Therefore, we investigated reducing the number of neurons of the synthetic neural network to simplify the cognition component for future implementation. By sub-sampling sensor data points and examining the performance of the network with various numbers of neurons, we found that four sensors and a single hidden layer of only four neurons produced an adequate estimation of impulse directionality. Consistent with our simplified finite element model with d=0.1, the four sensor points closest to the valve have the greatest effect; we scaled back the network to include an input layer of only four points. By scaling back the input space from 20 to 4, as well as reducing the overall number of hidden neurons from 12 to 4, as in Fig. 5.1., we were able to train this network to the accuracy of just below 0.02%.

Emulated Synthetic Neural Network

In order to be able to build a cognitive network, as mentioned before, we have decided to implement the network in hardware using embedded system boards, which emulate a synthetic neural network. We first used an embedded system circuit UM003, with an ATMega 128 as the MCU. We implemented the network using software coding in embedded C. Because the MCU does not have a floating point coprocessor, we used fixed point calculations. Our network was not capable of a live training but merely calculating the output of the network based on supplied input values and hard-coded connection weights exported from MATLAB. Because of the aforementioned assumption on the number of critical points and our hardware input and output limitations we limited our network to four input neurons, four hidden neurons and one output neuron, as in Fig. 5.1. The activation for the four neurons in the hidden layers is analog hyperbolic tangent sigmoid function:

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \qquad \text{(Eqn. 5.2)}$$

This analog hyperbolic tangent sigmoid function activation function used by MATLAB was implemented as a piecewise linear approximation with 5 segments.

The activation function for the neuron in output layer is just a liner function:

$$f(x) = x \qquad \text{(Eqn. 5.3)}$$

The data used for this training was a rough approximation of the shapes generated for water hammer experiment. The output from embedded system board can be found in Table 5.1. We compared the result with the outputs from MATLAB simulated artificial neuron network, and verified that the output from our PCB-based NN matched that obtained in the software to the accuracy of about 6.26% (Table 5.1).

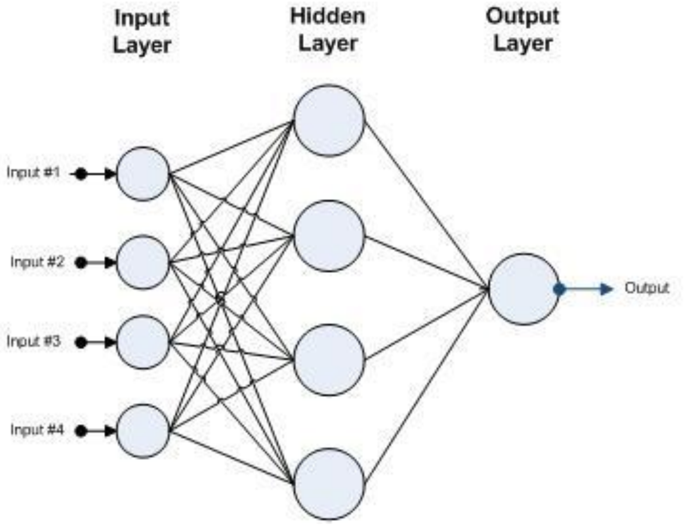


Fig. 5.1. Artificial Neuron Network Architecture

Table 5.1 ANN outputs comparison between emulated synthetic neural network (ESNN) and MATLAB

Shape #	ESNN output	MATLAB output	Error	Shape #	ESNN output	MATLAB output	Error
1	91.77	91.90	0.14%	11	85.70	85.00	0.82%
2	61.99	95.00	34.75%	12	64.82	66.70	2.82%
3	85.70	93.70	8.54%	13	89.13	87.40	1.94%
4	76.99	98.50	21.84%	14	88.82	88.90	0.09%
5	88.91	88.40	0.58%	15	86.36	89.40	3.4%
6	67.62	68.80	1.72%	16	85.70	80.80	6.06%
7	66.26	65.60	1.00%	17	103.94	98.30	5.74%
8	56.90	54.20	4.98%	18	100.24	107.00	7.25%
9	82.46	86.60	4.78%	19	85.70	83.20	3.00%
10	108.27	98.90	9.50%	Average Error			6.26%

Emulated Synthetic Neural Network Model Improvement

We originally planned to get very accurate and very close results from UM003 compare to the MATLAB ANN outputs, because they are using exactly the same ANN model, including the network structure and inputs/weights numbers. But from Table 5.1 we can see, the error is averagely 6.26%, which is much bigger than our first guessing. Here in this section, I would like to analyze the output error between MATLAB model and hardware based model.

First we need to see the differences of the models. For the one on embedded system board, as mentioned before, to enhance the calculation efficiency of the MCU, we use the fixed point calculation. So for the activation functions, instead of using an analog hyperbolic tangent sigmoid function (Fig. 5.2(a)), we use a piecewise linear approximation function (Fig. 5.2(b)). The piecewise linear sigmoid function consist of five segments. The X and Y axis are amplified 100 times, for the sake of fixed point calculation. The X-axis is divided into $[-\infty,-300]$, $[-300,-100]$, $[-100,+100]$, $[+100,+300]$ and $[+100,+\infty]$. From Fig. 5.2 we can see that, except the scaling issue (which is taken care in the code), the linear approximation function has obvious inaccuracies on the corners. If the point happens to be appearing at the corner areas of the function, it has more inaccuracy.

There is another source can cause the output error, which exists in the data processing. We are using the fixed point computation on our MCU, so we have to magnify the activation functions, input data, weights and biases. As just mentioned, the activation function is scaled 100 times. For the input values, we multiply by 10 on the original values and round the float to the nearest integer. The numbers can be found in Table 5.2.

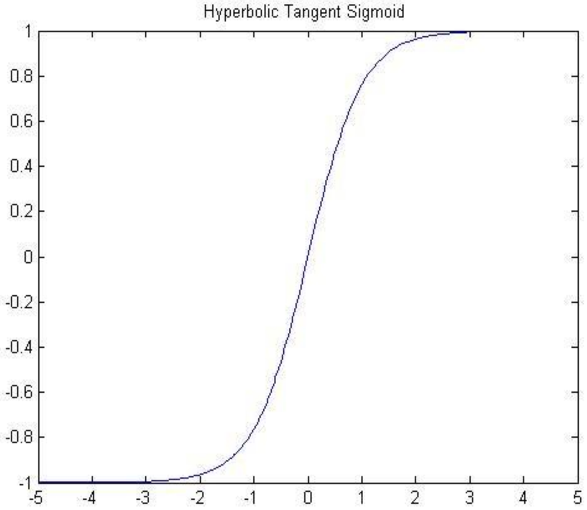


Fig. 5.2 (a). Analog hyperbolic tangent sigmoid function

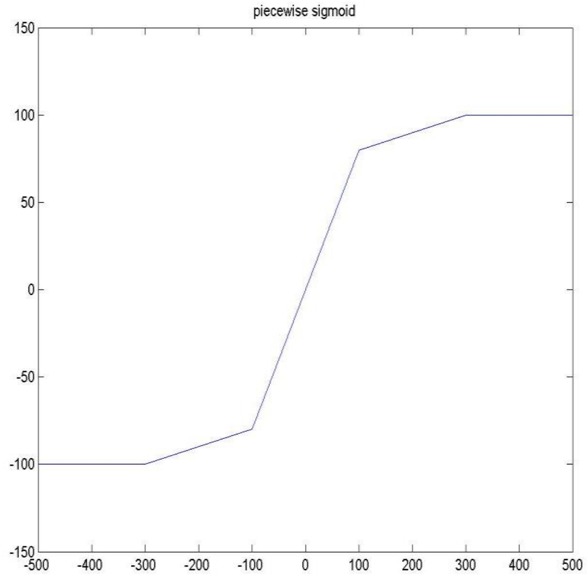


Fig. 5.2 (b). Piecewise linear sigmoid function

For the connection weight and bias values for the four neurons in hidden layer, we 10 times the connection weights and 100 times the bias, this makes the inputs to the activation functions magnified 100 times (as in Table 5.3). Because the activation function on ESNN is scaled 100 times on both X and Y axis, the outputs from the activation functions is still magnified 100 times.

Next, for the connection weight and bias values for the neuron in output layer, since the inputs to the neuron are all 100 times already, the connection weights are just the integer values by rounding the float, but the bias is 100 times magnified (as in Table 5.3). Therefore, the final output from the neuron in output layer is 100 times magnified. We just manually divide 100 on those output values. This is how the results for the embedded system board come from.

Table 5.2 ANN input values for and MATLAB model and emulated synthetic neural network (ESNN)

	Shape 1		Shape 2		Shape 3		Shape 4	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	0.79	8	3.58	36	3.81	38	3.27	33
Input 2	-5.48	-55	-12.48	-125	-7.74	-77	-16.79	-168
Input 3	-10.96	-110	-35.14	-351	28.27	283	-19.48	-195
Input 4	-8.98	-90	-44.79	-448	48.01	480	-15.47	-155
	Shape 5		Shape 6		Shape 7		Shape 8	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	10.41	104	17.35	174	12.43	124	12.63	126
Input 2	-8.06	-81	12.35	124	17.05	171	4.62	46
Input 3	-28.73	-287	-15.43	-154	5.82	58	7.98	80
Input 4	-14.25	-143	-45.79	-458	-20.53	-205	-18.28	-183
	Shape 9		Shape 10		Shape 11		Shape 12	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	11.22	112	-13.45	-135	9.73	97	21.62	216
Input 2	15.86	159	-34.67	-347	38.13	381	31.21	312
Input 3	34.62	346	-43.45	-435	38.56	386	-1.59	-16
Input 4	5.56	56	23.48	235	33.40	334	-27.08	-271
	Shape 13		Shape 14		Shape 15		Shape 16	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	-4.64	-46	-0.62	-6	3.95	40	10.95	110
Input 2	-0.29	-3	-0.15	-2	-5.63	-56	14.69	147
Input 3	2.72	27	-0.25	-3	0.35	4	8.57	86
Input 4	5.98	60	-0.34	-3	1.33	13	20.03	200
	Shape 17		Shape 18		Shape 19			
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN		
Input 1	-14.74	-147	-3.58	-36	0.00	0		
Input 2	-5.81	-58	3.58	36	3.09	31		
Input 3	-14.66	-147	-16.50	-165	9.63	96		
Input 4	-16.91	-169	-19.03	-190	6.20	62		

Table 5.3 ANN connection weights and bias values for and MATLAB model and emulated synthetic neural network (ESNN)

Connection Weights – Hidden Layer								
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
1st Neuron	0.187985423	2	0.0583138	1	0.009799	0	0.1233049	1
2nd Neuron	-2.22737366	-22	0.7812735	8	-1.620899	-16	1.0037673	10
3rd Neuron	-0.24935495	-2	0.4630343	5	-0.308402	-3	0.2588001	3
4th Neuron	0.525324214	5	-0.3030798	-3	-0.301955	-3	0.4187965	4
Bias - Hidden Layer								
	MATLAB	ESNN						
1st Neuron	2.084942174	208						
2nd Neuron	0.787153812	79						
3rd Neuron	7.862609462	786						
4th Neuron	9.447020522	945						
Connection Weights – Output Layer								
MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	
-11.3839028	-11	0.5063467	1	24.40795	24	36.312271	36	
Bias - Output Layer								
MATLAB	ESNN							
37.70492943	3770							

Now we can clearly see, during the processing of rounding the floats, we lose the accuracy of the calculation again. Given the two reasons of the inaccuracy, we can understand some of the error outputs, but some of them are still too big to be accepted, for example, the error for shape 2 is 34.75%. Therefore, we decided to improve the model on embedded system board by increasing the scales on numbers, especially on the weight values (Because most of the weights are smaller than 1). For the new model, in the hidden layer, we scale the inputs for 100 times, the weights 1000 times, bias 100,000 times, and also scale the activation function 1000 times compare to the one in Fig. 5.2 (b);

in the output layer, we scale the bias for 1,000,000 times and weights for 10 times. So the new numbers for the embedded system board based ANN can be found in Table 5.4 and Table 5.5.

Table 5.4 ANN input values for and MATLAB model and emulated synthetic neural network (ESNN)

	Shape 1		Shape 2		Shape 3		Shape 4	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	0.79	79	3.58	358	3.81	381	3.27	327
Input 2	-5.48	-548	-12.48	-1248	-7.74	-774	-16.79	-1679
Input 3	-10.96	-1096	-35.14	-3514	28.27	2827	-19.48	-1948
Input 4	-8.98	-898	-44.79	-4479	48.01	4801	-15.47	-1547
	Shape 5		Shape 6		Shape 7		Shape 8	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	10.41	1041	17.35	1735	12.43	1243	12.63	1263
Input 2	-8.06	-806	12.35	1235	17.05	1705	4.62	462
Input 3	-28.73	-2873	-15.43	-1543	5.82	582	7.98	798
Input 4	-14.25	-1425	-45.79	-4579	-20.53	-2053	-18.28	-1828
	Shape 9		Shape 10		Shape 11		Shape 12	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	11.22	1122	-13.45	-1345	9.73	973	973	973
Input 2	15.86	1586	-34.67	-3467	38.13	3813	38.13	3813
Input 3	34.62	3462	-43.45	-4345	38.56	3856	38.56	3856
Input 4	5.56	556	23.48	2348	33.40	3340	33.40	3340
	Shape 13		Shape 14		Shape 15		Shape 16	
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN
Input 1	-4.64	-464	-0.62	-62	3.95	395	10.95	1095
Input 2	-0.29	-29	-0.15	-15	-5.63	-563	14.69	1469
Input 3	2.72	272	-0.25	-25	0.35	35	8.57	857
Input 4	5.98	598	-0.34	-34	1.33	133	20.03	2003
	Shape 17		Shape 18		Shape 19			
	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN		
Input 1	-14.74	-1474	-3.58	-358	0.00	0		

Input 2	-5.81	-581	3.58	358	3.09	309	
Input 3	-14.66	-1466	-16.50	-1650	9.63	963	
Input 4	-16.91	-1691	-19.03	-1903	6.20	620	

Table 5.5 ANN connection weights and bias values for and MATLAB model and emulated synthetic neural network (ESNN)

Connection Weights – Hidden Layer								
	MATLAB	ESNN	MATLAB	MCU	MATLAB	ESNN	MATLAB	ESNN
1st Neuron	0.187985423	188	0.0583138	58	0.009799	10	0.1233049	123
2nd Neuron	-2.22737366	-2227	0.7812735	781	-1.620899	-1621	1.0037673	1004
3rd Neuron	-0.24935495	-249	0.4630343	463	-0.308402	-308	0.2588001	259
4th Neuron	0.525324214	525	-0.3030798	-303	-0.301955	-302	0.4187965	419
Bias - Hidden Layer								
	MATLAB	ESNN						
1st Neuron	2.084942174	208494						
2nd Neuron	0.787153812	78715						
3rd Neuron	7.862609462	786261						
4th Neuron	9.447020522	944702						
Connection Weights – Output Layer								
MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	MATLAB	ESNN	
-11.3839028	-114	0.5063467	5	24.40795	244	36.312271	363	
Bias - Output Layer								
MATLAB	ESNN							
37.70492943	37704929							

With this new model, we are able to get results that compare favorably with the MATLAB outputs as shown in Table 5.6. We can clearly see that with the changing on scaling of numbers, the output error between the emulated synthetic neural network and MATLAB is below 3%. So we can say that the emulated synthetic neural network model is improved a lot. With this more accurate ESNN, we compared the results from the

ESNN and the water hammer experiments directly in Table 5.7. The average error is still lower than 3%, which is a pretty good result.

Table 5.6 ANN outputs comparison between emulated synthetic neural network (ESNN) and MATLAB

Shape #	ESNN output	MATLAB output	Error	Shape #	ESNN output	MATLAB output	Error
1	92.507	91.90	0.66%	11	86.505	85.00	1.77%
2	93.015	95.00	2.08%	12	63.986	66.70	4.07%
3	86.505	93.70	7.67%	13	88.692	87.40	1.48%
4	97.068	98.50	1.45%	14	88.697	88.90	0.23%
5	89.176	88.40	0.87%	15	86.883	89.40	2.82%
6	65.368	68.80	4.99%	16	86.505	80.80	7.06%
7	63.247	65.60	3.59%	17	97.556	98.30	0.76%
8	56.058	54.20	3.43%	18	106.987	107.00	0.01%
9	84.350	86.60	2.60%	19	86.505	83.20	3.97%
10	98.914	98.90	0.01%	Average Error			2.61%

Table 5.7 Comparison between emulated synthetic neural network (ESNN) output and experiment results

Shape #	ESNN output	Experiment result	Error	Shape #	ESNN output	Experiment result	Error
1	92.507	91.931	0.63%	11	86.505	85.0278	1.74%
2	93.015	95.0267	2.12%	12	63.986	66.6876	4.05%
3	86.505	93.6791	7.66%	13	88.692	87.383	1.50%
4	97.068	98.5166	1.47%	14	88.697	88.9315	0.26%
5	89.176	88.3794	0.90%	15	86.883	89.424	2.84%
6	65.368	68.7642	4.94%	16	86.505	80.7907	7.07%
7	63.247	66.0154	4.19%	17	97.556	98.3098	0.77%
8	56.058	54.1671	3.49%	18	106.987	106.9707	0.02%
9	84.350	86.6102	2.61%	19	86.505	83.2079	3.96%
10	98.914	98.8555	0.06%	Average Error			2.65%

Error Analysis

From the results of the new ESNN model, we can clearly see the average error compared to experimental data is reduced. But defining an error metric is difficult in such a case. For example, adding 360^{o} to every value is also an accurate result, but the percent error would appear much lower (artificially lower) in this case. In fact, the results are not scattered from 0 to 360, but are really deviations from straight ahead. Why not compute error with respect to zero degrees (straight ahead), rather than 90 degrees?

What I really want to show is not an absolute error metric, but that my neural network estimator works well. Can I prove that our ESNN is one of the most accurate models to estimate the directionality? Since many of the values from the experiments are around 90^{o}, is the ESNN really better than simply guessing that no matter what the shape is, the valve will always go straight ahead (always 90^{o})? And how about only getting the direction that the last point on the hose is pointing, ignoring the other parts of the hose. Will this get a better estimation compared to ESNN?

With these concerns in mind, I decided to not only compare our ESNN only with the experimental results, but also compare it to other estimating models. First, we generated another estimation matrix, which is only measuring the direction of one point that closest to the valve (about 3cm away from the valve on the hose). This is equivalent to setting the scaling factor *d* to 0 in Eqn.4.1. (This zeros out the contribution from all nodes, i-1.) The results can be found in Table 5.8, under the column heading "*d*=0". The second estimation model compared is that of always guessing the direction is straight, which equals to 90 degree. This is similar to Roy Godzdanker's prior work in [36] that assumed that the water hammer propulsion is always pointing straight ahead. The results from this estimation model can also be found in Table 5.8.

From Table 5.8, we can easily see that our new ESNN model has the smallest average errors compare to the experiment results. In addition, we also calculated the standard deviation compare to the experiment results as Eqn. 5.1.

$$\sigma = \sqrt{\frac{1}{N}\sum_{1}^{N}(Est_i - Exp_i)^2} \qquad \text{(Eqn. 5.1)}$$

Where Est is the result from estimation model, and Exp is the experiment result. In this way, we can also see how much the results from estimation model is off from the experiments. The standard deviation values can be found in Table 5.8. A modified standard deviation calculation was performed that computes the deviation of each model from the experimental baseline (replacing the mean value subtracted from each estimate with the experimental value). This also confirms that our ESNN is the one that is closest to the experiment results.

Table 5.8 Average error and standard deviation comparison between different estimation models

Shape #	Experiment result	ESNN output	*d*=0	Always Straight Propulsion
1	91.931	92.507	89	90
2	95.0267	93.015	86	90
3	93.6791	86.505	86	90
4	98.5166	97.068	87	90
5	88.3794	89.176	80	90
6	68.7642	65.368	73	90
7	66.0154	63.247	78	90
8	54.1671	56.058	77	90
9	86.6102	84.350	79	90
10	98.8555	98.914	103	90

11	85.0278	86.505	80	90
12	66.6876	63.986	68	90
13	87.383	88.692	86	90
14	88.9315	88.697	91	90
15	89.424	86.883	86	90
16	80.7907	86.505	79	90
17	98.3098	97.556	105	90
18	106.9707	106.987	94	90
19	83.2079	86.505	90	90
Average Error Compare to Experiment Result		2.65%	8.76%	13.56%
Standard Deviation Compare to Experiment Result		2.793447	8.636357	13.73113

Finally, we examined how often the estimate is in the correct quadrant, which means is it on the left side of the Y axis (>90^{o}), on the right side of Y axis (<90^{o}) or straight (=90^{o}). This is important because we want the estimation model to give us predictions that have the correct sign, compared to the actual directions. Therefore, we check the signs of the estimation results, and classify them into three groups, "correct sign" means the estimation result and experiment result are pointing to the same side of Y axis, "incorrect sign" refers to the estimation results that are not on the same side of Y axis compare to experiment results. Thirdly, the "no sign" means the estimations that are straight ahead. The results are in Table 5.9, we can see that with the ESNN model, the estimation gets more correct signs than the other two models.

Table 5.9 Sign comparison between different estimation models

	Number of Correct Sign	Number of Incorrect Sign	Number of No Sign(Straight)
ESNN	18	1	0
d=0	13	5	1
Always Straight Propulsion	0	0	19

In summary, we compared the average error, standard deviation and sign of different estimation models, and all the evidence points to the same conclusion, which is our ESNN model is the best one in predicting the directionality based on the shape information.

Distributed Cognitive Networks

The ANN is getting the directionality information of the water hammer propulsion using the shape. It is actually a cognitive network for our soft robot. As we know, the hose-like robot is consists of amorphous computational material, which needs distributed sensing and cognition in order to gather all the information along the body. Therefore, we need to distribute our ANN onto distributed embedded system boards, which is a distributed cognitive network.

Talking about the distributed cognitive networks, there is one thing we should first consider: the communications between the distributed nodes in the network. Do we choose wired or wireless communication? Since the hose-like robot body can be very long, (though the diameter of the hose is small), if we do the wired communication, we need a lot of extra wires along the body. But if we use the wireless communications, we get much simpler design; the networking can be achieved by the wireless communication and malt hopping. Therefore, we decided to use the wireless communication networks.

Instead of using UM003 as above, we use UM001 embedded system circuit as the node in distributed cognitive networks. UM001 (Fig. 5.3) is actually very similar to UM003, they are using the same MCU, but it has a Bluetooth chip LMX9830 on it, which provides the potential of wireless communications.

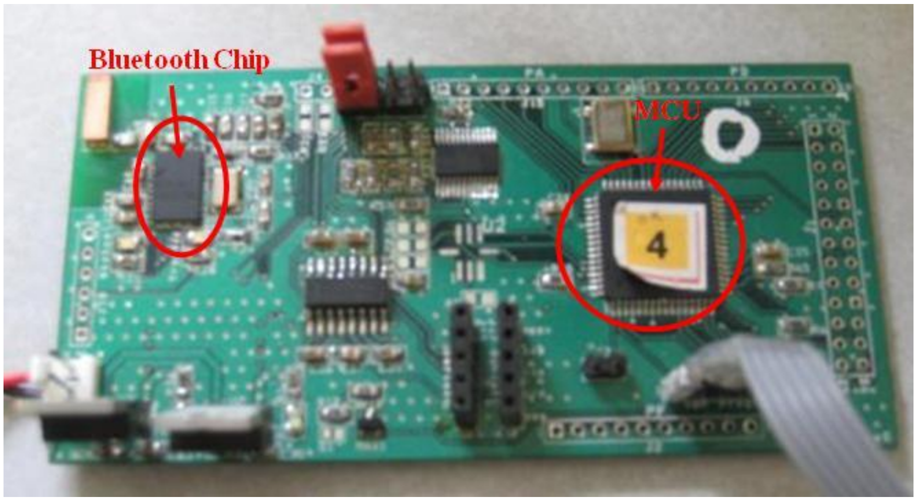


Figure 5.3. UM001 board

Because UM001 and UM003 both have the same MCU, the software calculation of ANN doesn't need to be change a lot. The thing we need to change is the topology of the ANN. Instead of running all the neurons on a single board, we are running them distributed. For example, we can run the inputs 1, 3 and the third neuron in hidden layer on board one, input 2, the first and fourth neurons in hidden layer on board two, and input 4, the second neuron in hidden layer and the neuron in output layer on board three. In this way, we distributed our network onto three embedded system boards. One thing to emphasis, in this thesis, I am not discussing the topology optimization of the distributed cognitive networks. The work will be done by other members in my research group in future.

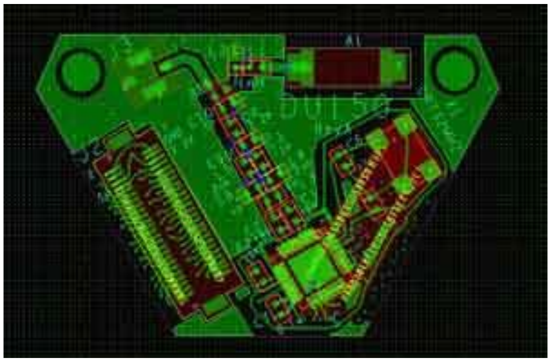
Upgraded Cognitive Network Node

The platform of our sensor and cognitive devices are designed for a single purpose, which is sensing the morphology of the hose-like robot, and analyze the directionality information of the water hammer propulsion. However, this is only focusing the application of the actuation; we want our soft robot can perform some other tasks, especially in a different environment. For example, sense the environment with camera and process the video information in the cognitive network; or sense the temperature and humanity in the environment, and process the data in the cognitive network. Because it’s impossible for a robot to carry all kinds of sensors and operate them at the same time in a complex urban search and rescue environment, it challenges the battery, the hardware complicity of the robot. Therefore we think of assigning and configuring potential usable sensors before the robots deployment and then decide specific ones we need to use after deployment. In this case, a fixed-architecture sensing and cognition device cannot fulfill all their requirements. Our research group has actually already developed a concept of “RecoNode” [37] [38]. The idea is to design a FPGA based reconfigurable platform for sensing and cognation.

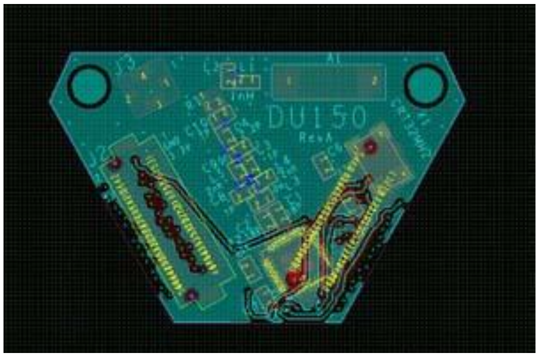
I am particularly in charge of the wireless communication module for this RecoNode platform. We choose to use Zigbee protocol as the wireless communication protocol for RecoNode. ZigBee is a global standard for wireless communication, which provides a short-range cost effective networking capability. ZigBee technology is a low data rate, low power consumption; low cost wireless networking protocol targeted towards automation and remote control applications [39]. We are using one of the most popular ZigBee chip CC2520 (Texas Instruments) for our hardware platform, it is a single-chip 2.4 GHz IEEE 802.15.4 compliant RF transceiver, it also provides extensive hardware support for packet handling, data buffering, burst transmissions, data encryption, data authentication, clear channel assessment, link quality indication and packet timing information.

The Zigbee wedge is named DU150 in our research lab, its schematics can be found in Appendix B. The printable circuit board (PCB) layout can be found in Fig. 5.4. It is a four layer RF circuit.

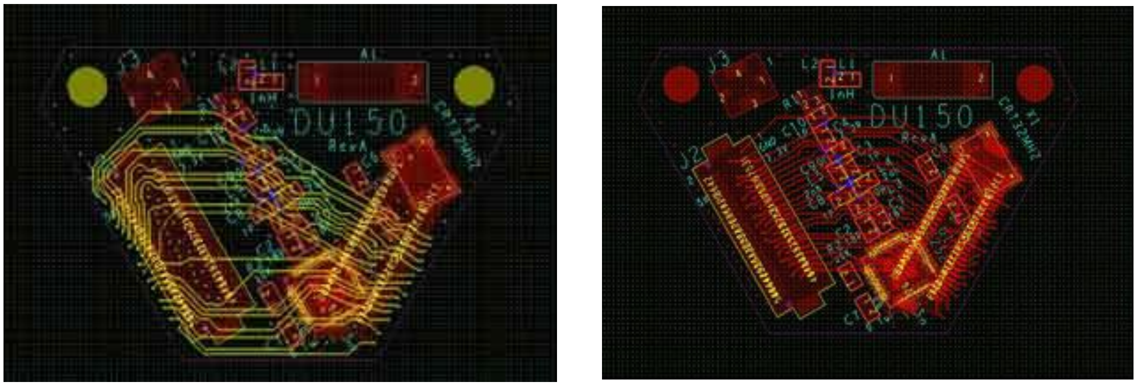
DU 150 as introduced before is the wireless communication interface for the RecoNode platform. Once our lab finished the FPGA based platform hardware design and software operating system, we can transplant our ANN software from MCU to FPGAs and run the distributed cognitive network on this upgraded node in the future.



Layer 1(Top Layer)



Layer 2



Layer 3

Layer 4(Bottom Layer)

Fig. 5.4. DU150 PCB layout

Chapter Six: Summary and Future Work

Summary

This thesis presents components for an on-going research project at Collaborative Mechatronics Lab. The long-term goal is to design a miniature hose-like soft robot with distributed soft sensors and cognitive devices along its body, using soft actuation.

The contributions of this thesis to the big research project are, 1) we found out an alternative form of soft actuator--water hammer actuation, and verified the benefits of water hammer actuator by some laboratorial tests. 2) We figured out that there is a relationship between the hose shape and the direction of water hammer propulsion at the end of the hose (Chapter Four). 3) With the study in Chapter Four, we implemented an emulated synthetic neural network on an embedded system circuit UM003, as a cognitive device to predict the directionality information based on the morphology information of the hose. 4) We distributed the ESNN onto many embedded system boards UM001, which is using the Bluetooth communication. This means we achieved a distributed cognitive network. 5) As mentioned in Chapter Five, we made the RF circuit DU150 for a RecoNode platform, which is a FPGA based configurable sensing and cognition platform. It is actually an upgrade platform compared to the embedded system circuits UM001/003 we are using right now.

Future Work

Topology Optimization for Distributed Cognitive Networks

As discussed in Chapter Five, we were able to distribute the ANN onto many embedded system boards to achieve a distributed cognitive network, but we didn’t do the research on the topology optimization. While topology is very important for networks, it affects the computer efficiency and communication robustness a lot. Therefore, in the future, our lab will work on the topology optimization for the distributed cognitive network.

Software Infrastructure for RecoNode Wireless Communication

We developed the DU150 hardware circuit as an upgraded version of the embedded system board UM001. Next step, we have to transplant the C code running on the MCU of UM001 onto the FPGA on RecoNode, so that the distributed cognitive network can be running on the RecoNode platform.

Control System Design for Controlling the Water Hammer Actuation

We are able to predict the water hammer propulsion based on the morphology information right now, but in order to achieve our long term goal, we also need to design a control system (both hardware and software) which is used to steer and control the water hammer actuation. The control system should also be made of soft material. It can be set up on the end of the hose-like robot, or be distributed along the whole body, based on different requirements and situations.

References

[1] D. P. Perrin, A. Kwon and R. D. Howe., “A Novel Actuated Tether Design for Rescue Robots Using Hydraulic Transients”, Proc. 2004 IEEE ICRA, pp.3482-3487, 2004.

[2] R.L. Feller, D.P. Perrin and RD Howe, “Validation and Explanation of Water hammer-Based Locomotion”, M Cambridge - Robotics and Automation, ICRA 2006.

[3] R. R. Murphy, “Human-robot interaction in rescue robotics”, IEEE Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 34, no. 2, pp. 138-153, 2004.

[4] E. Steltz, A. Mozeika, N. Rodenberg, E. Brown, and H. Jaeger, “Jsel: Jamming skin enabled locomotion,” in Proceedings of the IEEE/RSJ Conf. on Intelligent Robots and Systems, 2009, pp. 5672–5677.

[5] A. A. Transeth, K. Y. Pettersen, P. Liljeback, “A survey on snake robot modeling and locomotion,” Robotica, Vol 27, Pg 999-1015, Dec 2009.

[6] Smith A.F.G, Newborough M. “Low-cost polymer electrolysers and electrolyser implementation scenarios for carbon abatement”. Report to the Carbon Trust and ITM Power, 2004.

[7] R. Vyas, A. Rida, S. Bhattacharya, M.M. Tentzeris, “Liquid Crystal Polymer (LCP): The Ultimate Solution for Low-Cost RF Flexible Electronics and Antennas,” IEEE Antennas and Propagation International Symposium (2007) 1729-1732.

[8] http://en.wikipedia.org/wiki/Water_hammer

[9] Morrey, J.M., Lambrecht, B., Horchler, A.D., Ritzmann, R.E., Quinn, R.D., 2003. Highly Mobile and Robust Small Quadruped Robots, IEEE International Conference On Intelligent Robots and Systems (IROS’03), Las Vegas, Nevada.

[10] J. Casper, “Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at theWorld Trade Center,” Ph.D. dissertation, M.S. thesis, Dept. Comput. Sci. Eng., Univ. South, Florida, Tampa, 2002.

[11] Trivedi D, Lotfi A, Rahn CD. 2007. Geometrically exact dynamic models for soft robotic manipulators. Paper presented at: Proceedings of the 2007 IEEE International Conference on Intelligent Robots and Systems, San Diego, CA: IEEE.

[12] D. Duff, M. Yim, K. Roufas,“Evolution of PolyBot: A Modular Reconfigurable Robot”, Proc. of the Harmonic Drive Intl. Symposium, Nagano, Japan, Nov. 2001, and Proc. of COE/Super-Mechano-Systems Workshop, Tokyo, Japan, Nov. 2001.

[13] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, “Modular Reconfigurable Robots in Space Applications”, Autonomous Robot Journal, special issue for Robots in Space, Springer Verlag, 2003.

[14] M. W. Hannan, I. D. Walker, “Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots,” Journal of Robotic Systems, vol 20, Issue 2, Pg 45-63, Feb 2003.

[15] K. Hatazaki, M. Konyo, S. Tadokoro, “Active Scope Camera for Urban Search and Rescue”, Proc. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2007), pp.2596-2602, 2007

[16] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B.MacKenzie “Cognitive Networks: Adaptation and Learning to Achieve End-to-end Performance Objectives,” IEEE Communications Magazine, vol. 44, pp. 51–57, December 2006.

[17] R.W. Thomas, L.A. DaSilva, A.B. MacKenzie, “Cognitive networks”, in: Proc. IEEE DySPAN 2005, November 2005, pp. 352–360.

[18] A. S. Lee, J. Ly, S. F. Peteu, A. A. G. Requicha, M. E. Thompson and C. Zhou, "Electroactive polymer actuation at the nanoscale", *Proc. IEEE Int'l Conf. on Nanotechnology*, Cincinnati, OH, July 17-20, 2006.

[19] Madden, J. D., P. G. Madden, et al. (2001). Characterization of polypyrrole actuators: modeling and performance. Proceedings of the SPIE 8th Annual Symposium on Smart Structures and Materials: Electroactive Polymer Actuators and Devices, San Diego, CA, USA, SPIE Press.

[20] Kenaley, G.L. and M.R. Cutkosky, “Electrorheological Fluid-Based Robotic Fingers with Tactile Sensing,” in Proceedings of the 1989 IEEE International Conference on Robotics and Automation, v. 1, May, 1989, pp. 132-136.

[21] Voyles RM, Fedder Jr G, Khosla PK. Design of a modular tactile sensor and actuator based on an electrorheological gel. In: Proceedings of IEEE International Conference on Robotics and Automation; 1996. p. 13–7.

[22] M. Krishna and J.Bares , “Tethering System Design for Dante II”, IEEE Conference on Robotics and Automation, April 1997, April, 1997, pp. 1100-1105.

[23] R. R. Murphy. “Rescue robotics for homeland security”. Com. of the ACM, Special Issue on Homeland Security, 27(3), 2004.

[24] Hert, S., Lumelsky, V., “Computational Geometry Issues in the Tethered Robot Problem,” Technical Report RL-94002, February 1994.

[25] E. F. Fukushima, N. Kitamura and S. Hirose (2001) “Development of Tethered Autonomous Mobile Robot Systems for Field Works”, Advanced Robotics, 15-4, 481–496.

[26]A. Birk and C. Condea, “Mobile Robot Communication without the Drawbacks of Wireless Networking,” RoboCup 2005: Robot WorldCup IX, I. Noda, A. Jacoff, A. Bredenfeld, Y. Takahashi(Eds.) Lecture Notes in Artificial Intelligence (LNAI) 4020, pp. 585-592, Springer, 2006.

[27] L. W. Mays and Y.-K. Tung. Hydrosystems Engineering and Management. McGraw-Hill, 1988.

[28] Victor L. Streeter, E. Benjamin Wylie, “Hydraulic Transients”, New York: McGraw-Hill [1967]

[29] J. Parmakian. “Waterhammer Analysis”. Dover Publications, New York, NY, 1963.

[30] A. Jacoff, E. Messina, and J. Evans, “A standard test course for urban search and rescue robots,” in Proc. 2000 PerMIS Workshop, 2001, pp. 253–259.

[31] http://en.wikipedia.org/wiki/Artificial_neural_network

[32] Pugh, G. Allen, 1989, Synthetic neural networks for process control. Computers & Industrial Engineering, 17, 24-26.

[33] I. Jouny, F. D. Garber, and S. C. Ahalt, “Classification of radar targets using synthetic neural networks,” IEEE Trans. Aerosp. Electron. Syst., vol. 29, pp. 336–344, Apr. 1993.

[34] H. B. Hammouda, M. Mhiri, Z. Gafsi, K. Besbes, “Neural-Based Models of Semiconductor Devices for SPICE Simulator,” American Jour. Of Appl Sci. 5 (4): 385-391, 2008.

[35] R. Genov, G. Cauwenberghs, “Dynamic MOS Sigmoid Array Folding Analog-to-Digital Conversion,” IEEE Trans. On Circuits and Systems-I: Reg Papers, Vol 51, NO. 1, Jan 2004.

[36] R. Godzdanker and R.M. Voyles, " Steering Control of an Active Tether Through Mass Matrix Control," in IEEE Workshop on Safety, Security and Rescue Robots, Sendai, Japan, Oct., 2008, pp. 128-133.

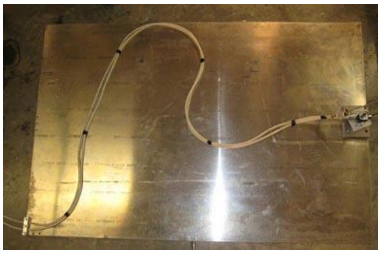
[37] W. Zhao, B. H. Kim, A. C. Larson, and R. M. Voyles, “FPGA implementation of closed-loop control system for small-scale robot,” in Proceedings of the 2005 International Conference on Advanced Robotics, July 2005, pp.70-77.

[38] R. M. Voyles, S. Povilus, R. Mangharam and K. Li, "RecoNode: A Reconfigurable Node for Heterogeneous Multi-Robot Search and Rescue", accepted by 2010 IEEE International Workshop on Safety, Security, and Rescue Robotics

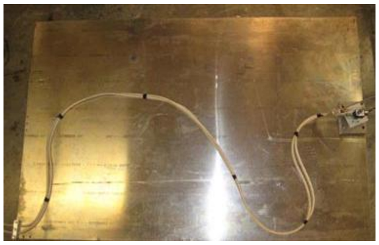
[39] J. S. Lee, Y. W. Su, and C. C. Shen. A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wifi. In IEEE IECON, Nov. 2007.

Appendix A

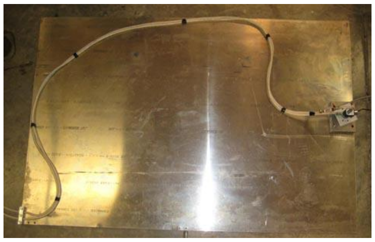
The 19 shapes from experiments that are used in computer simulation:



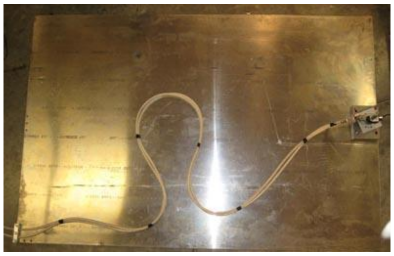
Shape 1



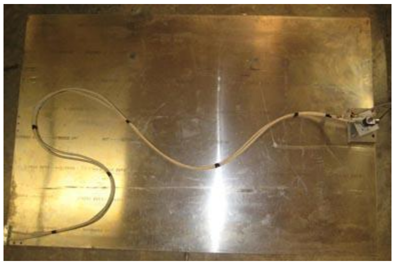
Shape 2



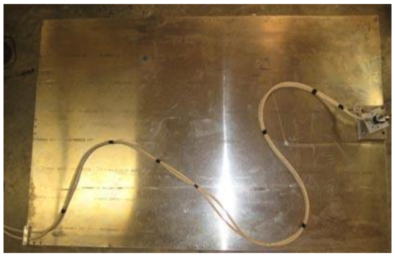
Shape 3



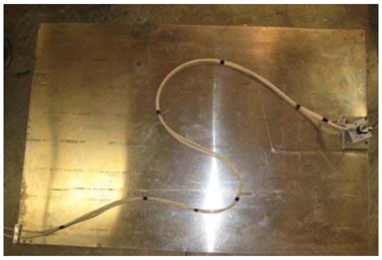
Shape 4



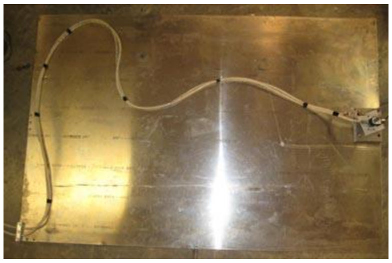
Shape 5



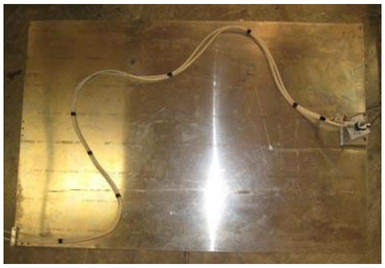
Shape 6



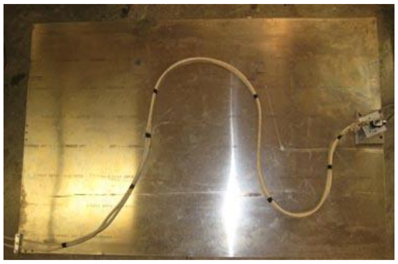
Shape 7



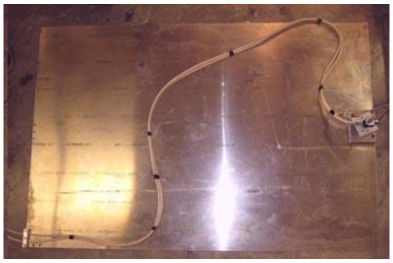
Shape 8



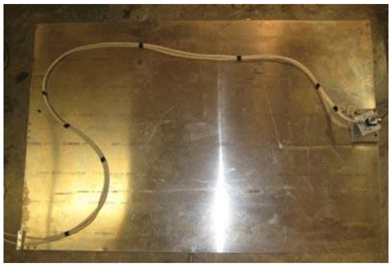
Shape 9



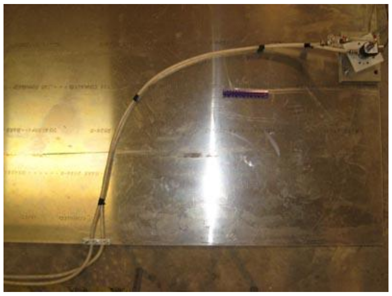
Shape 10



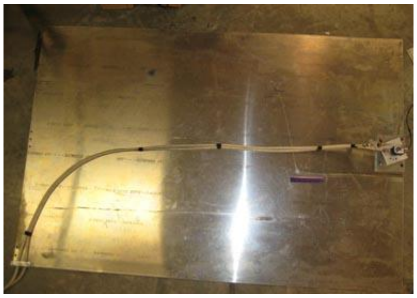
Shape 11



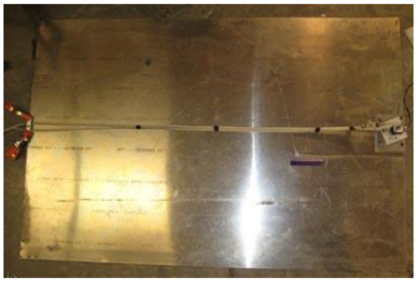
Shape 12



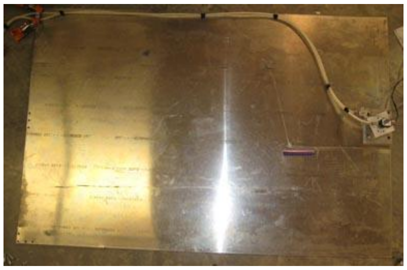
Shape 13



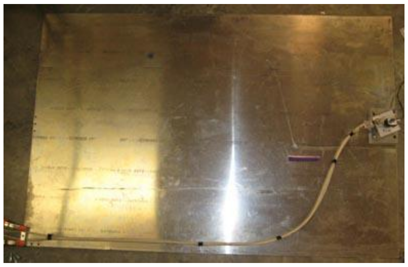
Shape 14



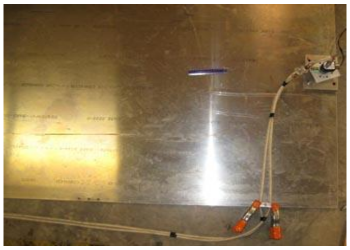
Shape 15



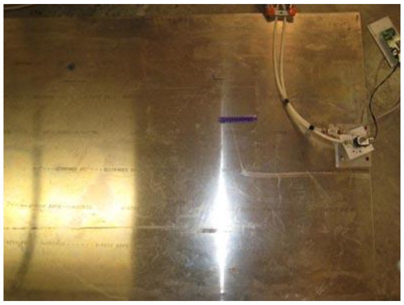
Shape 16



Shape 17

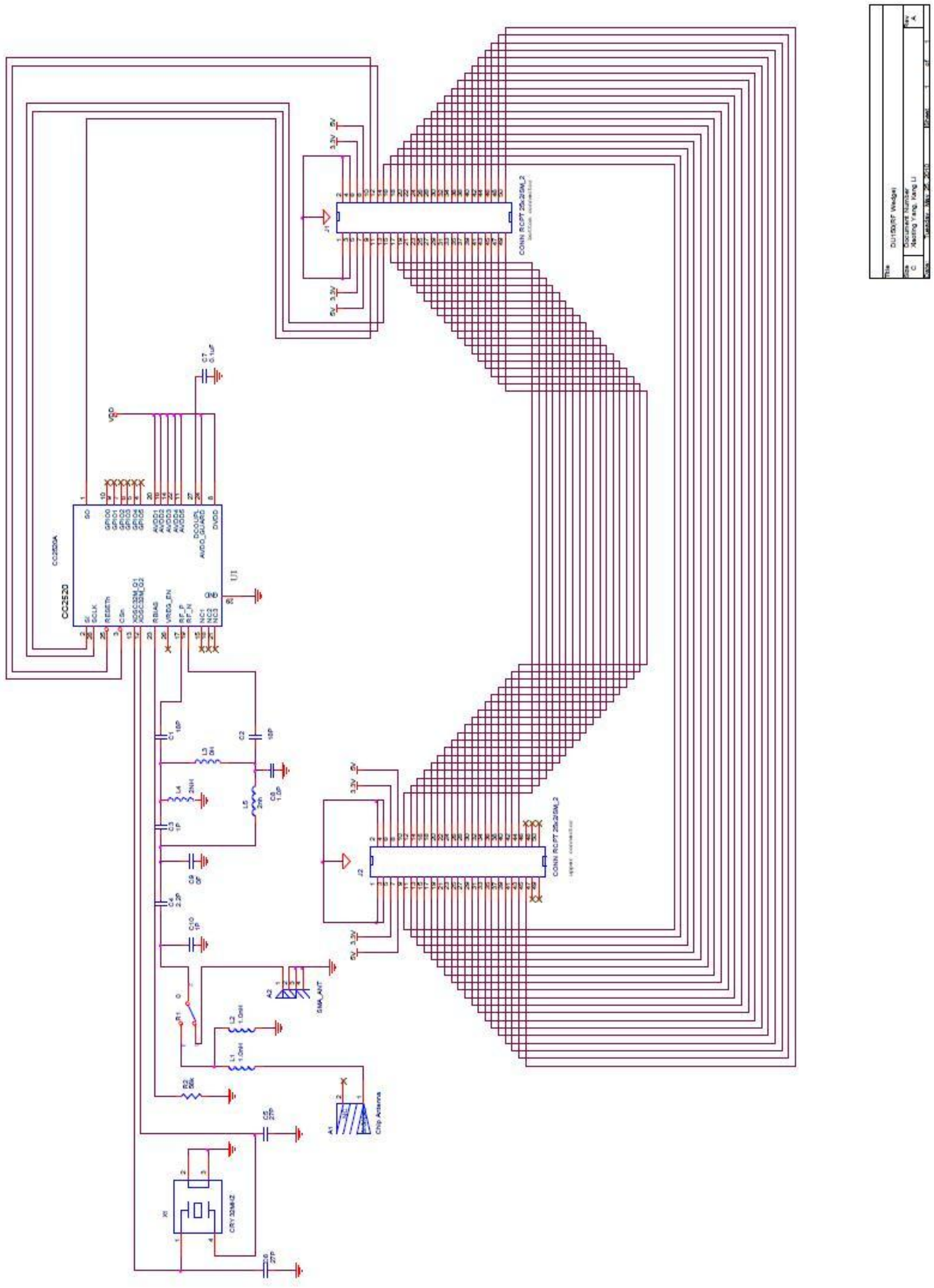


Shape 18



Shape 19

Appendix B



DU 150 schematics

Appendix C

In addition to the laboratory comparison experiments between active tether and passive tether in Chapter Three, we also did a field test at Disaster City in Texas. In the field test, we used a bigger and more powerful robot, Inuktun robot. It is a miniature inspection system designed to access confined spaces and challenging terrain in a variety of applications, as shown in Fig. C.1. We set up the active tether system on this Inuktun robot and used it for the field test; because the Inuktun robot has a control tether for itself, we tied the two pieces of water hoses and the data cable together with the pipe sleeve as the active tethered system. For the passive tethered system, we use its own flexible control cable as the passive tether.

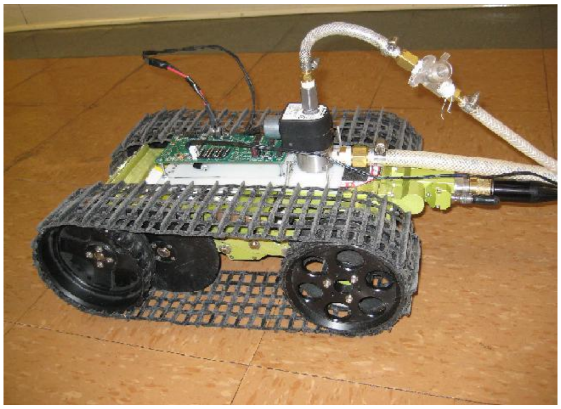


Fig. C.1. Inuktun robot with active tether

The Disaster City test is to run the robot on a slope on the roof of the "House of Pancakes". The slope is covered with tough sand and small rocks; in other words, it has a high friction surface, and the slope is about 25^o and 12 meters long, as in Fig. C.2. We want to see how far the robot is able to climb with the two tether systems.

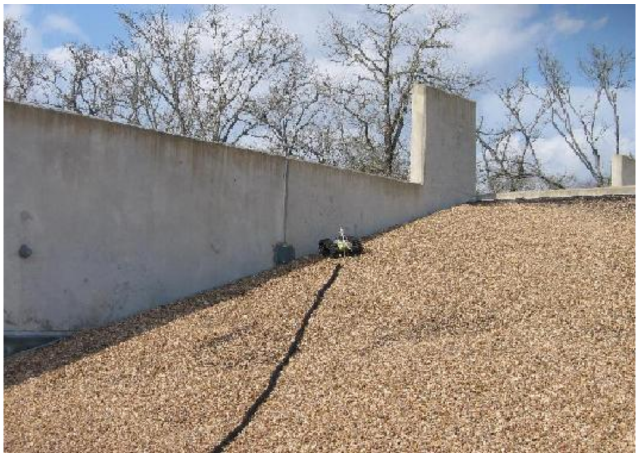


Fig. C.2. Inuktun robot with active tether on the slope

We ran the experiments three times for each system. For the passive tether system, the Inuktun robot was able to get to the top of the slope for three times; but for the active tether system, the robot could only get to the top of the slope once, and the other two times, they all stopped at the points around 85% to 90% of the whole length.

The active tether was less effective than the passive tether actually in this case. The reason we think this happened is, first of all, the active tether is not the optimal tether; it has its own drawbacks, and it cannot perform better than the passive tether under all situations. For example, the active tether system always needs another system to provide high pressure water flow in, like air compressor, pump and so on. Secondly, the water hose is much heavier than normal data cable. This may cause problems especially in the situations like the above; the tether didn't get stuck with obstacles, but the robot stopped due to both the high friction and the extra gravity from the tether. As in Fig. C.3, we know on a slope, the tether gravity G_1 added to the robot equals to the tether gravity multiply by $\sin\theta$. Although the jerks on the pipe caused by water hammer effect could help reduce the friction on the tether, the extra gravity was still applied to the robot. The unit weight of the Inukton cable is about 0.02kg/m, but the two pieces of water hose with sleeve and water inside is 1.2kg/m; it dramatically increased the burden of the robot over long distance. For the passive tether system, on the top of the slopes, G_1 is 0.99N; for the active tether system, G_1 is 59.64N on the top of the slope. This is the main reason for the bad performance of the active tether in this case.

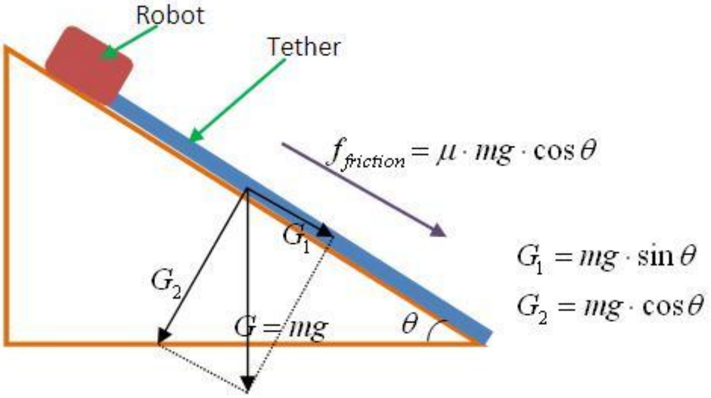

Robot
Tether
$f_{friction} = \mu \cdot mg \cdot \cos\theta$
G_1
G_2
$G = mg$
θ
$G_1 = mg \cdot \sin\theta$
$G_2 = mg \cdot \cos\theta$

Fig. C.3. Burden of tethered robot on slope

This field test also told us that we need to do a lot of design improvement on the active tether, for example find lighter material for the tubing; find alternative forms of lower density fluid instead of water and so on.