3-D GRASPING DURING SERPENTINE MOTION WITH A SNAKE-LIKE ROBOT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BARIŞ ATAKAN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2005

Approval of the Graduate School of Natural and Applied Science

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Aydan M. Erkmen
Co-Supervisor

Prof. Dr. İsmet Erkmen
Supervisor

Prof. Dr. Kemal Özgören (METU,ME)
Prof. Dr. İsmet Erkmen (METU,EE)
Prof. Dr. Aydan M. Erkmen (METU,EE)
Asst. Prof. Buğra Koku (METU.ME)
Asst. Prof. Afşar Saranlı (METU,EE)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are no original to this work.

Name, Last name: Barış Atakan
Signature:

ABSTRACT

3-D GRASPING DURING SERPENTINE MOTION
WITH A SNAKE-LIKE ROBOT

Atakan, Barış

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. İsmet Erkmen
Co-Supervisor: Prof. Dr. Aydan M. Erkmen

December 2005, 106 pages

In this thesis, we introduce our lasso-type grasping scheme. This 3-D lasso-type grasping scheme, different from the previously performed grasping schemes which are either planar or fixed base, is the novelty of our approach where the snake robot grasps an object with any of its body links which are at close proximity to the object while undergoing its serpentine motion with the remaining links and dragging the grasped object. Since our snake robot has the pitch motion for every link, we can ensure that the links do not run into each other as they wrap around the object. A lasso-type power grasp is then possible for our 15-link snake robot as seen in the simulation results of this thesis.

Furthermore we develop the kinematic and control models for lasso-type grasping and for dragging the grasped object to a desired state. This control model includes an adaptively changing feedback gain which prevents excessively large inputs to corrupt the serpentine locomotion control. According to our lasso-type grasping model, while the snake robot can grasp the object beginning with the any body link at close proximity of the object, the contact points can be controlled to generate the curvilinear grasping curve by using our lasso-type grasping procedure. For dragging the grasped object, we define a scheme which can determine the appropriate desired state to drag the grasped object to a desired position.

The stability of the grasped object is important to resist the disturbance forces as well as the force closure grasping is important to counteract the disturbance force. To analyze the stability of the lasso-type grasping, we introduce a stability model of lasso-type grasping based on contact stiffness matrices that faces the snake to regrasp when gone unstable.

Keywords: Lasso-type grasping, grasping during locomotion, snake robots, grasping in snake robots.

ÖZ

YILANA BENZEYEN BİR ROBOT İLE SÜRÜNGEN HAREKETİ
SIRASINDA ÜÇ BOYUTLU KAVRAMA

Atakan, Barış

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Aydan M. Erkmen

Aralık 2005, 106 sayfa

Bu tezde, kement tipi kavrama yöntemimizi ortaya koyuyoruz. Daha önce yapılan düzeysel yada sabit tabanlı kavrama yöntemlerinden farklı olan üç boyutlu kement tipi kavrama yöntemimiz yılan robotun cisme en yakın halkasının cisme sarılırken kalan halkalarının sürüngen hareketine devam ettiği ve kavranılan cismi sürüklediği bir yöntemdir. Yılan robotumuz alçalma ve yükselme hareketine sahip olduğu için halkaların cismin etrafına sarıldıkça birbirlerine çarpmayacağından emin olabiliriz. Böylece bir kement tipi güçlü kavrama on beş halkalı yılan robotumuz için tezin simülasyon sonuçlarında görüldüğü gibi mümkündür.

Ayrıca kement tipi kavrama için ve kavranılan cismi arzulanan duruma çekmek için hareket ve kontrol modelleri geliştiriyoruz. Bu kontrol modeli yılansı hareket kontrolünü bozan aşırı büyük girdileri engelleyen adapte olarak değişen bir geri besleme kazancı içermektedir. Kement tipi kavrama modelimize göre, yılan robot cisme yakın olan herhangi bir halkasından başlayarak cismi kavrayabilirken, kontak noktaları kement tipi kavrama modelimizi kullanarak lineer kavrama eğrisini oluşturmak için kontrol edilebiliyor. Kavranılan cismi arzu edilen duruma çekmek için uygun durumları belirleyen bir plan tanımlıyoruz.

Kavranılan cismin kararlılığı bozucu kuvvetlere karşı direnebilmesi için önemli. Aynı zamanda kuvvet kapatan kavrama da bozucu kuvvetlere karşı koyabilmek için önemli. Kement tipi kavramanın kararlılığını analiz etmek için kontak sertlik matrisine dayalı kavrama karasızlaştığında yeniden kavrayabilen bir kararlılık modeli tanıtıyoruz.

Anahtar Kelimeler: Kement tipi kavrama, hareket sırasında kavrama, yılan robotlar, yılan robotlarda kavrama.

ACKNOWLEDGEMENTS

I would like to express my honest gratitude to my co-supervisor Prof. Dr. Aydan M. Erkmen and my supervisor Prof. Dr. İsmet Erkmen for their guidance and support either technical or mental.

I would also like to thank my family and my friend Serkan Şahin for their unique support.

To

My Uncle Mustafa

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

SAR	: Search and Rescue
3-D	: Three Dimension
ACM	: Active Cord Mechanism
GMD	: German National Research Center for Information Technology
JPL	: Jet Propulsion Laboratory
DOF	: Degree of Freedom
VGT	: Variable Geometry Truss
VGTM	: Variable Geometry Truss Manipulator
ER	: Electrorheological
KSI	: Kinetic Science Incorporate
PID	: Proportional, Integral and Differential
GALWA	: Grasping and Locomotion Wave

CHAPTER 1

INTRODUCTION

1.1 Serpentine Robots

Hyper-redundant systems are kinematically redundant mechanisms, in which the number of degrees of freedom is very large or infinite. Such systems when serial manipulators identify themselves to snakes or tentacles in nature, are often refered to as serpentine robots in the robotics field. The design of hyper-redundant mechanisms vary as a result of different objectives like locomotion, inspection and manipulation [1] [2] [3] [4]. The advantages of such system can be found in their flexibility in constrained environments and have led caused to the emergence of new challenging fields of robotics. Search and rescue (SAR) Robotics is such new and challenging area, dealing with tasks in extremely hazardous and complex disaster environments [5]. Since natural complexities have to be handled, biologically inspired mobile robots and, in particular serpentine mechanisms have turned out to be widely used, providing effective, immediate and reliable responses to many SAR operations.

Difficulties of manipulating objects in such SAR environments have prompted researchers to explore increasingly sophisticated manipulator designs to improve grasping and object manipulation [3] [4]. Grasping while locomotion has been a critical need in Search and Rescue (SAR) robot missions to enlarge passages around victims, or for any need of passing through during pure navigation or transportation. Hyper-redundant robots analogous in morphology to snakes, tentacles and elephant trunks have found wide usage as SAR robotic devices which need not only to be able to pass through narrow passages but also may need to enlarge a pathway, to bring back a sample, or transport vital support to a victim. Such a device should not have a sharp saliency which is violated when equipped with manipulators which, in this case, would mean more probability of getting stuck among rubbles. Figure (1.1) shows an example of a serpentine system built for navigation and inspection of earthquake debris [6].

Figure (1.1) Search and Rescue Demonstration by a Serpentine Robot [6].

1.2 Lasso-Type Grasping

Snake-robots different from wheeled and legged robots use their entire bodies for locomotion. Since this supplies the great capability of traction and manipulability in complex environments, a new manipulation technique is needed to increase the versatility of such robots in object handling. In the literature, grasping with snake-robot is found to be performed either planarly or with fixed base hyper-redundant mechanism [7] [8] [9]. In the planar examples, grasping has been done using the head link and the consecutive links just in the rear of the head link. This, however, prevents any serpentine motion of the snake robot during grasping. In the fixed base examples, the fixed base arise as a big handicap in taking away the grasped object to any desired new distant location. So, the lasso-type grasping is a novel concept to investigate.

1.3 Objective and Goals

The aim of this thesis with a specific focus a SAR robotics, is a lasso-type grasping by a snake like robot during its serpentine motion within the highly unstructured environments of disaster areas. To meet this objective, we had to extend the snake-like robot grasping to 3-D and expand our novelty in grasping to undertaking it during a serpentine motion.

1.4 Contributions and Methodology

Since previous works in the literature about the snake-like robot locomotion highlight the imitation of natural snake gait generated by the sinusoidal functions, kinematic and control models already exist for serpentine motion and constructing curvilinear wave gaits. So, when we extend the planar snake-robot grasping to 3-D lasso-type grasping, firstly we do the kinematic and control adaptations on previously existing serpentine locomotion models. As the basic novelty of our approach, the snake-robot grasps an object with any of its body link which is at close proximity to the object while undergoing its serpentine motion with the remaining links and dragging the grasped object. Since our snake robot has the pitch motion for every link, we can ensure that the links do not run into each other as they wrap around the object making a lasso-type power grasp. This is possible for a highly redundant snake-like robot which has 15-links in our thesis work when we carry on our simulation.

Furthermore we develop the kinematic and control models for lasso-type grasping and for dragging the grasped object to a desired state. This control model includes an adaptively changing feedback gain which prevents excessively large inputs to corrupt the serpentine locomotion control. According to our lasso-type grasping model, while the snake robot can grasp the object beginning with the any body link at close proximity of the object, the contact points can be controlled to generate the curvilinear grasping curve by using our lasso-type grasping procedure. For dragging the grasped object, we define a scheme which can determine the appropriate desired state to drag the grasped object to a desired position.

The stability of the grasped object is important to resist the disturbance forces as well as the force closure grasping is important to counteract the disturbance force. To analyze the stability of the lasso-type grasping, we introduce a stability model of lasso-type grasping based on contact stiffness matrices that faces the snake to regrasp when gone unstable.

1.5 Outline of the Thesis

In the first chapter, we give the introduction focusing on the motivation behind the thesis work providing the objective and goals together with the contributions of the present work. In the second chapter of thesis, firstly we give the brief overviews to previously designed snake-robots, the natural snake-gaits, and previously used control mechanism for snake-robot control. Secondly previously performed hyper-redundant grasping schemes for the fixed based and for the snake-like robots are overviewed. Thirdly a 3-D model of snake-robot and a stability model of grasped object used in this thesis are given. In the third chapter, we give our contributions. Firstly, we adopt the model given in Chapter 2 for our lasso-type approach. Secondly, we develop the kinematic and control models for lasso-type grasping and for dragging the grasped object to a desired state. Thirdly, we give the stability analysis of object in lasso-type grasping. In fourth chapter, we analyse our approaches given in Chapter 3 with the comparatively manner by showing the graphical and demonstrative examples and we give some performance analysis. Chapter 5 concludes the thesis work by also providing a suggestion for future works.

CHAPTER 2

LITERATURE SURVEY

2.1 Snake-Like Robots

The word “redundant” is used in the context of robotic manipulators to indicate that the number of actuated degrees of freedom exceeds the minimal number required to perform a particular task. For instance, a manipulator required to position and orient an object in space needs six actuated degrees of freedom, and so a manipulator with seven or more actuated degrees of freedom is redundant with respect to this task. Hyper-redundant mechanisms are kinematically redundant manipulators, in which the number of degrees of freedom is very large or infinite. Such system are analogous to serpents or tentacles, and are often refered to as serpentine robots. The advantages of such system can be found in their flexibility in constrained environments.

In recent years, biological systems began to serve as great inspiration for robotic creation. Greater knowledge about biological systems facilitates more accurate and efficient robotic systems. Designs that accurately mimic body structure and movements began to produce robust and controllable systems and snake-like robotresearch and application gained emphasis. History has shown that robotic navigation system have dominantly used the tracked and legged mobility methods. Commonly, robotic vehicles use a wheel and axle propulsion system, but this is often debilitating when traveling on variable terrain. For instance, wheel-based propulsion system must be specifically designed to ascend steps. Successful propulsion systems are often similar to motion methods of animals.

Leg-based mobility is the most common method but it is difficult and inefficient to reproduce mechanically. Snakes can use their entire body for propulsion. The large surface area of the body provides great traction. Snakes remain in close contact with the ground. This produces a low center of gravity which improves stability. This stability is reduced in leg and wheel based system. The body structure of biological snakes can be modeled as a series of independently controlled joints. Each joints has many degrees of freedom. These freedoms allow the snake to raise body sections over obstacles. The snake could elevate onto or above a step or an obstruction. This task would be difficult if not impossible using wheeled propulsion.

As shown in Figure (2.1), the biological snake is an omni-body that articulates at joints formed between vertebras. Ribs extend downward on both sides of each vertebra. Snakes lack appendages simplifying their body structure. Constant contact with the ground also provides excellent traction that the snake manipulates when moving. Forward propulsion is achieved by changing body position and manipulating its contact with the ground. The snake obtains mobility by repeatedly altering its body shape. Motion patterns are formed by repeating certain shapes. This patterns borrowed from legged motion are called “gaits”.

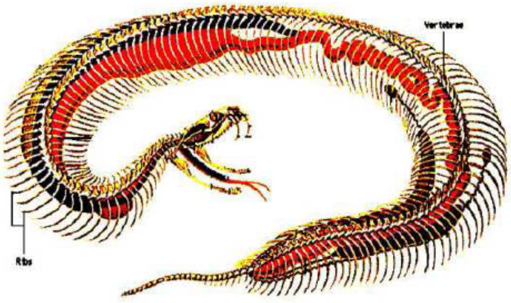

Figure (2.1) Snake Skeletal Structure

Biological snakes utilize several different gaits such as lateral undulation, concertina, and side winding. Controlling these gaits, a snake is able to achieve directional mobility. Biological snake most commonly use the lateral undulation gait which produces propulsion by varying body shape in a horizontal, sinusoidal pattern. This oscillatory motion has both a tangential and a normal component relative to the forward direction. The normal force component is in a direction perpendicular to the forward direction. As undulation changes polarity cancel each other. The tangential forces are in the direction of forward motion and add up to each other in order to propell the snake forward. Lateral undulation Figure(2.2) is not successful on low-friction surfaces and less effective for shorter body lengths and for heavier snakes [10].

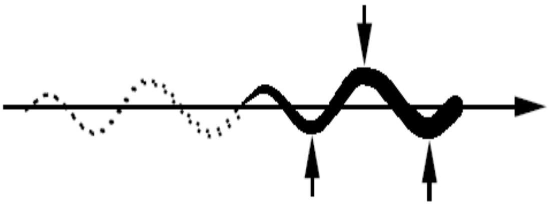

Figure (2.2) Lateral Undulation Motion

The concertina gait uses a progressive body extension pattern where the body folds and unfolds in a posture similar to an accordion. Extending a front section, the snake reaches forward while the back sections remain stationary. These stationary sections provide a fix base for the moving section. The extension remains fixed, as the snake begins to unfold its body and its back section moves forward. In this phase, the front section acts as the fix base, while the back section is in motion. The motion alternates between pushing against a back fix base and pulling against a front fix base. This gait is useful on low-friction surfaces. Figure (2.3) shows the concertina motion of biological snake.

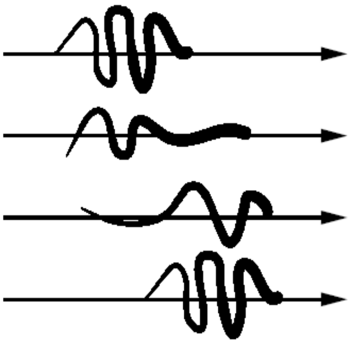

Figure (2.3) Concertina Motion

The side winding gait uses continuous waves of lateral bending to move in a lateral direction. The segments not in contact with the surface are lifted and move to the side and then become the new contact points. Repeating this pattern, the snake moves in a direction to its side. Since sections must be lifted, the snake moves in both horizontal and vertical planes. The side winding gait Figure (2.4) which is very useful on low-friction surfaces requires more complex muscular and skeletal structures to facilitate the two degrees of freedom.

Figure (2.4) Side Winding Motion

Much of the preliminary work about snake-like robot have been done by Shigeo Hirose who has investigated snake robots and natural snakes. Most of his research is focused on the locomotion and gait analysis of such systems, employing an active chord mechanism (ACM). Hirose's active cord mechanism (Figure (2.5)), a chain of serially connected and wheel-based segments, can generate a net forward force by applying the appropriate torques along the length of its body. He models the snake by a link mechanism with no side slip and suggested that the actuating joints with sinusoidal inputs generate typical winding motion of natural snakes [11]. The purpose of the first steps of Hirose's research is as follows: 1-) Design a snake robot which can creep forward and apply the modern control theory to realize a robot with a position control. Check whether the trace can be approximated by the serpenoid function. 2-) Derive some control method which can create an exact serpenoid trace if any, and see what happens with the magnitude and distribution of the control inputs. 3-) Using these results, find the best model of snake robots which use the least energy to move.

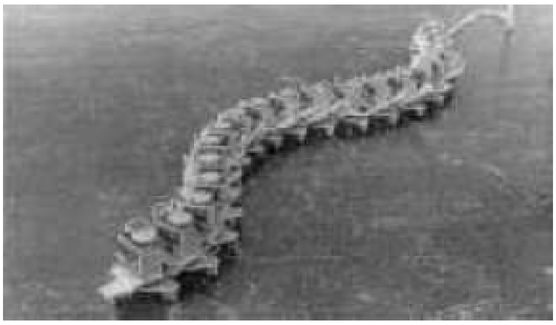

Figure (2.5) Hirose’s ACM III

Another discrete morphology snake-robot, called GMD-Snake (Figure(2.6)), is constructed by Riner Worst and Ralf Linnemann [1]. It uses its body instead of wheels, legs or arms for locomotion as well as handling items. It may be used for inspection and manipulation in areas where motion is restricted. The GMD-Snake is assembled of same uniform sections, each one consisting of two rubber joints. The joints can be bent in arbitrary directions, thus making the robot’s body very flexible. To imitate closely a real snake and to study different ways of movement, appropriate control software which manages the harmonious motion of all the section was given by investigators. Therefore, the individual sections are connected to an external processor via special type of field bus such that the GMD-snake undergoes its locomotion according to predefined commands.

Succeeding the GMD-snake, Bernhard Klaassen, Karl L. Paap present the next generation GMD-Snake2, a robot for inspection task in areas difficult to access by humans [2]. They try to imitate the natural scale-driven propulsion of a snake by wheels around the body and concentrated on a method for motion control. This method allows a very flexible and consistent method of path planning and calculation where smooth curves are desired. Mathematically, it is based on an enhancement of the clothoid curve which is applicable not only to snake-like movement but to any wheeled robot which is able to control the curvature of its path.

Figure (2.6) GMD Robotic Snake

Jet Propulsion Laboratory (JPL) has developed a serpentine robot to test the feasibility of robotic inspection of restricted areas [3]. The robot has 11 degrees of freedom. All joints are direct-drive motor controlled, and all motors are mounted internally (Figure (2.7)). Inspection capability is provided by a fiber-optics borescope transfers images to a camera at the base of the arm and illuminates the inspection site. The goal is to use this robot as an inspection tool to be picked up by one of the large manipulators. The combined macro/micro arm will have 21 degrees of freedom. Algorithms have been developed to guide this arm through small openings in such a way that the rest of the arm automatically follows the tip's path, thus avoiding collisions with the environment. This technology can be used in industry and medical applications.

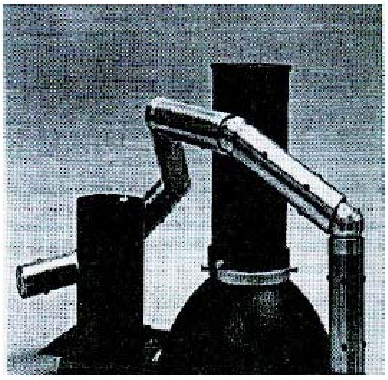

Figure (2.7) The Jet Propulsion Laboratory 12-DOF Serpentine Robot

In 1995, the giant Japanese electronics company, NEC, announced the development of a snake robot which was dubbed 'The Quake Snake' and designed to enter the rubble strewn aftermath of earthquakes and explosions to search for survivors. The device, called Orochi, utilized an active universal joint, a novel form of a Hooke's joint designed by Ikeda and Takanashi [12]. The seven segment device is shown in Figure (2.8) [4]. Control is done manually and the single gait used is akin to a rectilinear or inchworm gait. This class of mechanism has great promise for serpentine robots in real applications.

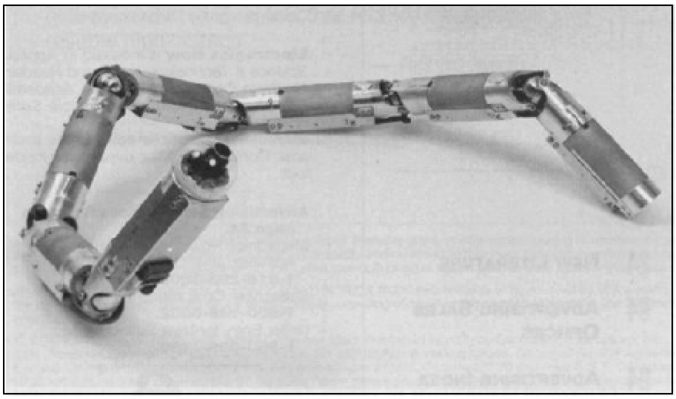

Figure (2.8) The Nec 7-Segment 'Quake Snake'

Research in structural resonance control for large space structures produced the original Variable Geometry Truss (VGT) (Figure (2.9)). First, Chirikjian and Burdick built a 30-DOF VGT composed of 10 identical 3- DOF truss modules [13]. The truss modules contain three prismatic joints, creating planar parallel manipulators. The prismatic joints are operated with direct current servo motors and lead screw drives, and can vary in length from 12 to 18 inches. The method for kinematic analysis which does not rely upon the manipulator Jacobian matrix have been introduced by Chirikjian and Burdick [14]. In their approach, a continuous curve model was used to describe the macroscopic truss geometry and a method to analyze the kinematics and inverse kinematics of planar continuous and discrete morphology of nonextensible (fixed-length) hyper redundant manipulators was presented. These methods are based on an intrinsic parameterization of a 'backbone curve' which captures the macroscopic geometric features of the manipulator. Control of hyper redundant robots using continuous backbone curves has been well studied. This approaches scales very well to large numbers of modules and is of particular interest in hierarchical control approaches with kinematic constraints. Furthermore, Chirikjian and Burdick present a approach, termed "tunneling" to the obstacle avoidance problem which is applied to planar hyper redundant manipulators [15]. Each joint of a snake-robot cannot be simply operated individually because there are too many. These robots require a motion planning algorithm. Motion planning for snake robots is difficult because the robots have many internal degrees of freedom that have to be coordinated to achieve purposeful motion. Chirikjian and Burdick investigated serpentine control with a geometric approach to motion planning [16]. While their analysis was a big step in demonstrating the use of VGTMs, that work has several drawback. First, for spatial manipulators, a curve alone is not sufficient to describe manipulator configuration. Second, unless the curve used to describe the manipulator is parameterized with meaningful physical variables, additional computations are required to specify a desired distribution of actuator displacements. Lastly while special algorithms deal exclusively with VGTMs, it is not clear how they would apply to other types of hyper redundant manipulators.

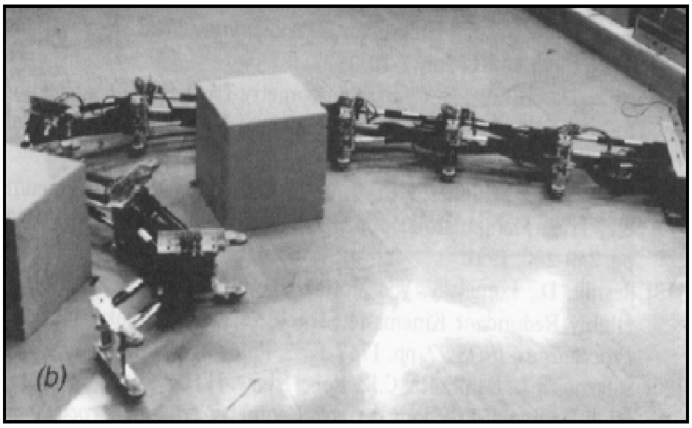

Figure (2.9) Burdick's Snake, a VGT-style hyper-redundant manipulator and locomotor

Munerato presented a flexible mini-robot designed for pipe inspection [17] consisting of twenty a mechanism segments, each with a 30-mm diameter and 50-mm length summing up to a total of 60-DOF to be controlled. Each 3-DOF module is constructed of two platforms linked by three extensible lengths, providing two twist angles of up to 20° amplitude, and extension equal to one-third the total length of the module (Figure (2.10)) enabling an earthworm type displacement. The earthworm displacement was categorized as “extension in a horizontal plane”, with the sole disadvantage being necessary module elongation.

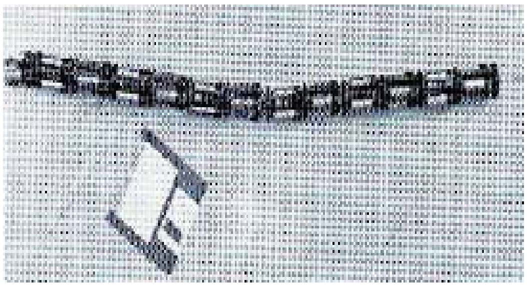

Figure (2.10) The Flexible Mini-Robot designed by Munerato et al

In continuous morphology systems, actuation is distributed throughout the manipulator. The aim of this approach is to create a system comparably efficient to those found in nature, namely muscle tissue. Muscle distributed along the length of a bone has the ability to elongate and contract, allowing for the actuation of limbs through wide ranges of motion. Optimally, a serpentine robot would be actuated with “muscles” just the way a real snake is. Most of the continuous morphology systems are based on the use of pneumatic and hydraulic actuators and chemomechanical materials, or electrorheological (ER) fluids.

Kinetic Science Incorporate (KSI) has built and tested a hybrid electric pneumatic robot called the KSI Tentacle Manipulator [18], utilizing a folding hood actuator and three tendon cables for bending any direction the manipulator in such a tentacle manipulator has six degrees of freedom and can extend more than five times its contracted length (Figure (2.11)). The KSI manipulator is a two-part system: the folding hoods become stiff with increased pressure, while contraction and bending is achieved through electric motor servo winches controlling the tendon cables. This particular design is advantageous, when compliant grasping is required, and is more conducive to manipulate fragile items than rigid mechanism are.

Figure (2.11) The KSI Tentacle Manipulator

Two broad classes of control methods have been used with snake robots [19]. The first class can be described as *trajectory-tracking control* using predefined gait patterns, usually computed as sine waves, that are tracked with a feedback controller (e.g. a PID controller). Typically, the control is open-loop: the set points of the joints are calculated and sent to the motor controllers without any form of feedback (the only feedback present in the system is the one used by the PID controller). Hirose used this type of control. He suggested that actuating the joints with sinusoidal inputs generate typical winding motion of natural snakes [11]. A trace of such winding motion is called a serpenoid curve. He applied some control based on the serpenoid curve. In this method, though singular postures can always be avoided, the gait of the robot is fixed beforehand and exact control of the position is difficult.

As another example of trajectory-tracking control, Ma, Araya and Li in their work developped a snake-like robot that can achieve creeping locomotion only by adjusting the relative angles between adjacent links [20] that allow the robot's shape to form along the Serpenoid curve.

The other class can be described as *online gait generation control.* In this case, gaits are not predefined in advance, but generated online during locomotion.

These approaches that are generally model-based, i.e. they rely on a kinematic or dynamic model of the robot's locomotion in order to design control laws for the gait generation can, therefore, better deal with perturbations and irregular terrains.

As an example of online gait generation control based on the dynamic model, Prautsch and Mita showed a dynamical position control of the snake robot and discussed the effect of constraining trace of the head to a serpenoid curve [21]. They proposed an autonomous locomotion control of the head position based on Lyapunov function method. In this method, winding motion is generating autonomously in real time and exact position control can be achieved. However, when the number of link is large, amplitude of winding motion tends to decrease, namely, tends to have a singular postures. Hence it is difficult to design a controller satisfying keeping good posture and tracking to a desired trajectory.

As another example of online gait generation control based on the dynamic model, Date, Hoshi and Sampei achieve tracking to a desired trajectory, avoiding singular postures without giving any gait beforehand [22]. They utilize a notion of dynamic manipulability to evaluate the locomotability with consideration of the side force wheels discussing an autonomous locomotion control of a snake like robot which consists of multiple links with passive wheels and active joints and also propose a control method of locomotion control based on this manipulability.

As another example of online gait generation control based on kinematic model, Ostrowski and Burdick examine a model for a snake robot [23] based on the ACM-III built by Hirose [24]. By using the intrinsic invariances of the constraints, they realize the three-segment model as a principal kinematic system. The three wheel constraints define a kinematic connection and so fully determine the effect of internal shape changes on net serpentine locomotion. They utilize this fact to control additional segments which are constrained to "follow" the lead of the three-segment kinematic system. By formulating the motion of the system in terms of a connection, they can highlight the important factors which contribute to the serpentine locomotion. Using sinusoidal inputs which are phase-delayed down the length of the snake robot, they are able to simulate various possible gaits, including a serpentine gait and two rotate gaits.

2.2 Snake-Robot Grasping

A multitude of studies have been done about snake like robots on their kinematic models [9] [25] [26], path planning [2], singular configuration avoidance [27] [28], obstacle avoidance [15] [29], and serpentine locomotion control [19-23] [27] [30]. But very few studies exist about snake like robot grasping or lasso-type grasping. Some investigators developed tentacle and elephant trunks based robot manipulators for grasping, in the form of massively redundant manipulation devices being alternative to conventional robot arm/hand combination. They focused on grasping by wrapping the device links around an object in the same manner as an octopus would do. This grasping method is advantageous because these manipulators become an all-in-one arm and such gripping devices are capable of a wide variety of grasp configurations, while still utilizing the mechanics of serial manipulators.

Pettinato and Stephanou proposed a 4-link revolute joint tentacle module [7] (Figure (2.12)) that uses a yaw-pitch-pitch-pitch configuration which allows it to curl around objects. By attaching several of these prototypes together, this curling action is enhanced. 9, 13 and 17 link tentacles can be formed by attaching together two, three and four 4 link modules together respectively and adding a roll to the base. The roll is added to increase the manipulability in the angular z direction. A 21 link tentacle appending five of these modules together with a roll is shown in Figure (2.13)

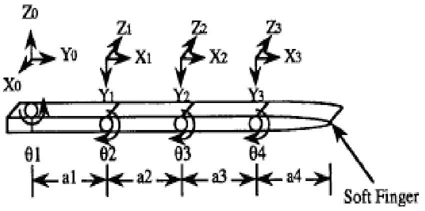

Figure (2.12) Schematic Diagram of a Four Link Tentacle

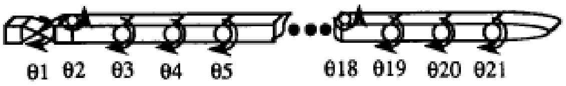

Figure (2.13) Schematic Diagram of a Twenty One Link Tentacle

Pettinato and Stephanou simulated the grasping scheme by the 21-link tentacle arm and gave their tentacle grasping in an algorithmically simple and uniform manner as follows:

1. Position linkage system so that the last link to contact the object is in position

   above it (Figure (2.14), slide 1).
2. Move joint until the link comes in contact with object.
3. Repeat step 2 until all links used for grasping contact object.
4. Adjust torque at each joint until there is no contact slippage.

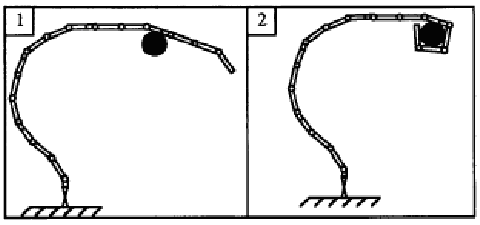

Figure (2.14) Time Lapse Illustration of a Tentacle Grasp

Furthermore, Pettinato and Stephanou applied a power grasping to an object by wrapping its links around it multiple times as shown in Figure (2.15). To ensure that the links do not run into each other as they wrap around the object, they used the yaw motion of each module to offset each encirclement of the object.

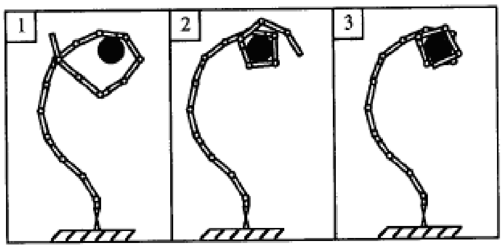

Figure (2.15) Time Lapse Illustration of a Power Grasp Using a Tentacle

Since tentacle and elephant trunks based manipulator is a hyper-redundant mechanism and analogous in morphology to a snake like robot mechanism, the grasping with these redundant manipulators was poineers of the snake-like robot grasping studies. In one of such leading works, Chirikjian and Burdick applied some of the locomotion ideas to the kinematics of grasping on a 30-DOF Variable Geometry Truss, and considered the novel combination of a hyper-redundant grasp with a locomotion wave used to reorient the object [9]. In this work, they considered only the kinematic aspect of grasping and object reorientation with a grasping and locomotion wave (GALWA) method that they propese for manipulating objects consists of the following steps:

- **Shape Initialization:** The hyper-redundant mechanism wraps around the object. The section of the manipulator in contact with the object is termed the *grasp contact segment.*
- **First Phase:** A section of the mechanism which is outside of the grasp contact segment distorts to a wave form. As a result, the object will be displaced by a small amount. This is shown in Figure (2.16-a).
- **Second Phase:** The wave generated in the first phase travels along the robot toward the distal end without changing the position or orientation of the object over which it passes. This phase (shown in Figure (2.16-b)) is similar to traveling wave locomotion where the manipulated object is the terrain. When the wave has traveled to the distal end of the manipulator, the grasp contact segment will be longer (by an amount refered to as *stride length* in locomotion terminology).
- **Third Phase:** The manipulator 'unwraps' part of the grasp contact segment from the object by straightening a small part of the grasp contact segment as shown in Figure (2.16-c). This results in a rotation an displacement of the object.

For the case where the object to be manipulated is a cylinder, the displacements resulting from the first and third phases cancel, leaving only a net rotation. When the third phase is complete, the cycle repeats starting with the first phase. This repetition results in repeated object rotations, the magnitude of which depend on the size of the wave. The cycle shown in Figure (2.16) can be used to cause counter-clockwise rotations. Alternately, the cycle can be reversed to yield clockwise rotations. For arbitrary objects, net translations can also occur from cycle to cycle.

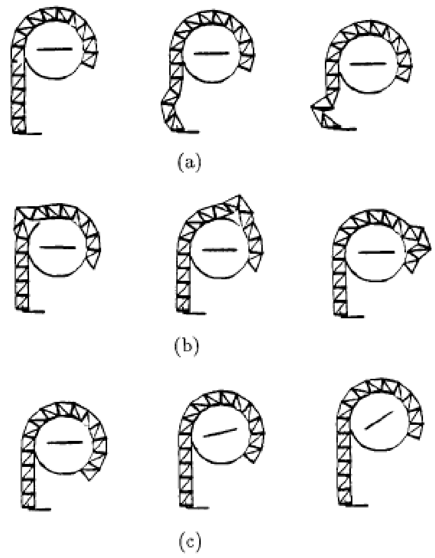

Figure (2.16) Hyper-Redundant Grasping of a Cylinder

IS robotics developed a planar serpentine robot named Kaa [8], shown in Figure (2.17). The Kaa robot has 12 degrees of freedom. The robot does not have any external sensors; however, it can sense the torque in each motor. The robot is completely autonomous, that is, power and computation are on board.

Figure (2.17) IS robotics' Kaa.

One set of behaviours explored using Kaa is mobility among parallel pipes, i.e. the ability to reposition the robot body in a field of parallel vertical pipes (Figure (2.19)). The vertical pipes present a simple circular cross-section to the robot.

This overall behavior is diagrammed in Figure (2.18). The cross-sections of vertical pipes are labeled A, B, and C. The basic behavior steps are as follows:

1) Kaa begins with its left arm wrapped around pipe A, as in Figure (2.18-a).
2) It acquires pipe B by swinging its body in counter-clockwise direction, using the left most joint not involved in the grasp of pipe A, as in Figure (2.18-b,c).
3) The left arm unwraps and folds back parallel with the body, as in Figure (2.18-d,e).
4) Kaa uses its right arm to swing its body clockwise a few degrees, as in Figure (2.18-f).
5) The right arm unfolds on the opposite side of the pipe and ready to grasp a new pipe, as in Figure (2.18-g).
6) This sequence is repeated with the right arm, starting at step 3.

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

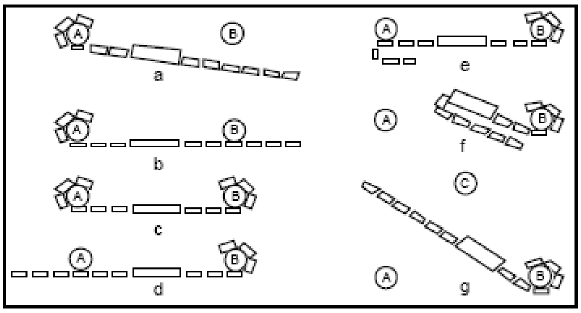

Figure (2.18) A diagram of Kaa's planar mobility sequence steps a-g.

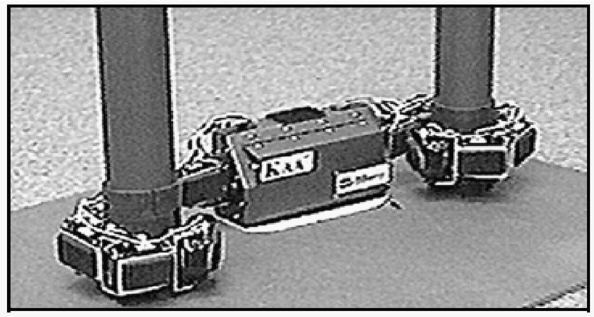

Figure (2.19) Kaa Robot gripping two pipes

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

1) The arm is moved into a straight position; each joint angle is set to zero as shown in Figure (2.20).
2) A wave of activation is initiated at the central link and is passed to the next nearest link, on the side of the robot which grasping is occuring.
3) When a joint is activated, it moves in the direction of the object to be grasped. (This causes all of the arm from this link to the tip to move in that direction.)
4) This motion continues until a position limit is reached or a force is detected (via motor torque for this joint).
5) The joint then goes into a force servoing control mode. It then activates the next joint further out on the arm which repeats the actions outlined in step 3.

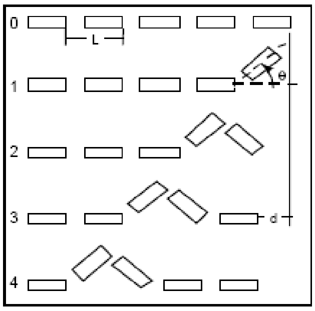

Figure (2.20) A traveling wave moves Kaa on a flat surface

Hirose’s soft gripper is a snake-like mechanism that can flexibly grasp an optionally shaped object [11] [31]. To flexibly grasp an object, it must be possible to create a uniform grasping force on all the gripper surfaces while wrapping around the object. This was accomplished in the action of the soft gripper in the following way.

First, Hirose consideres the bending moment M(s) which is created on the inside part of the member when adding a distributed load to a cantilever beam. This may be reduced in terms of secondary functions as in Figure (2.21). From this, if the gripper is assumed to be a kind of beam which has countless motors, when considering the moment in which there are secondary function changes by these motors facing from the base toward the tip, this moment generates a uniform force along the beam to the exterior part. However, using countless motors cannot be done.

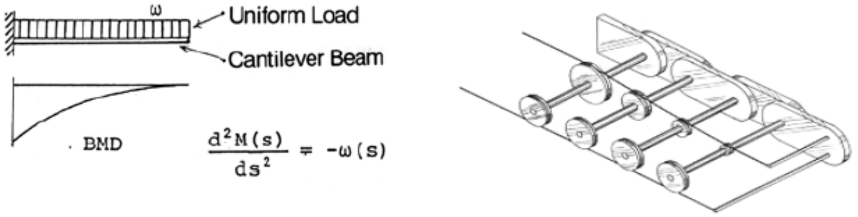

Figure (2.21) Relationship of the uniform load $\omega(s)$ applied to the beam

Thus, Hirose and coworker develop the soft gripper mechanism shown in Figure (2.22), in which the joints and pulleys are all connected to rotate independently, and the wire fixed to the head joint is pulled by coiling around all the pulleys. In this mechanism, when the wire is pulled, a bending moment proportional to the radius of the pulley is generated on each joint. For that reason, when the pulley radius is designed to change in terms of secondary functions and the wire coiling around the pulley is pulled, the moment of distribution described above is obtained, and uniform grasping is possible. The grip release motion is conducted by pulling the series of pulleys of the same radius in the opposite direction.

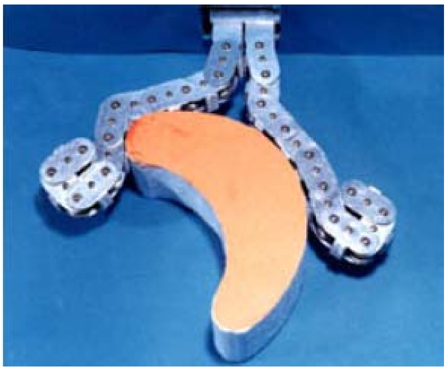

Figure (2.22-a) The flexible gripping action of the Soft Gripper I

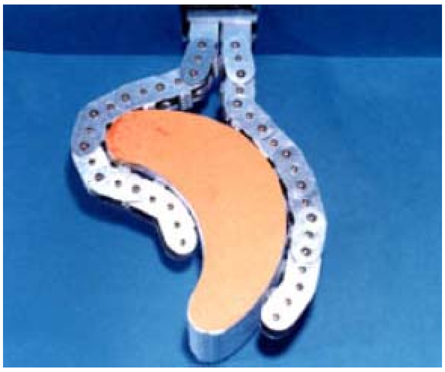

Figure (2.22-b) The flexible gripping action of the Soft Gripper I

Figure (2.22) Hirose’s Soft Gripper I

Figure (2.22) indicates the conditions by which the primary model grasps from the housed state. Figure (2.23) indicates a large-scale model that was developed as a gripper for a robot assisted by a human command to grasp and hold up a human body. Drive systems for vertical grasping and for horizontal grasping are provided. Figure (2.24) is belt driven type three-fingered model.

Figure (2.23) The flexible gripping action by the Soft Gripper II on a human body

Figure (2.24) The belt driven Soft Gripper III

2.3 Grasping Stability

Since the grasping stability is the quantitative ability of a manipulator to compensate for finite disturbances and maintain zero slippage at its points of contact with an object without destroying the object through excessive internal forces, the measure of grasping stability is important to consider the grasping ability of manipulator. There are some way of measuring the stability of grasping. For this purpose, Pettinato and Stephanou used the grasp matrix G which relates the velocities at the contact points to the overall twist being applied to the object [7]. The grasp matrix is given as;

$$[G] = \begin{bmatrix} T_{fi} & \dots & T_{fn} \end{bmatrix} \begin{bmatrix} B_1 & & 0 \\ & B_2 & \\ & & \ddots \\ 0 & & B_3 \end{bmatrix} = T_f B \qquad (2.1)$$

where T_{fi} are the transforms between contact frame and body frame, and B_i is a unit matrix of contact twists. With the grasp matrix representation, the stability measure (μ_s) denotes the ability to resist wrenches.

$$\mu_s = \sqrt{\det\left[GG^T\right]} \tag{2.2}$$

Pollard and Perez approach stability as satisfying equilibrium [32]. Focusing on the use of a three-fingered hand, stability is achieved when the three contact forces are in equilibrium. A “grasp focus” is defined as the intersection of the three contact forces. Similarly, Park and Starr describe a stable grasp as a force closure grasp [33], where force closure has the property of completely constraining an object, compensating for any force through the finger forces at the contacts. Construction of such a grasp is reduced to finding contact locations such that (in the case of a three-fingered hand) the three force directions create force closure. Park and Starr extend their analysis to create a heuristic function that evaluates the grasp quality in terms of the force direction arrangement, the grasp points, the grasp locations with respect to the object centroid, the number of convex vertices, and the number of grasp points located near the end of an edge. Although force closure and stability are often used interchangeably, it should be noted that force closure is not always sufficient for achieving stability. By utilizing the grasp matrix, Hsu, Li and Sastry define stability for a multi-fingered robotic hand as the existence of a choice of joint torque to balance every wrench applied to the body. This approach is devided into two quality measures; one in twist space and the other in wrench space. A three-measure system for evaluating stability is also proposed, consisting of a “minmax” measure, a volumetric measure, and task-oriented measure. All three are numerically computable and exist in the force domain, allowing grasp selection to be treated as an optimization problem [34].

In this thesis, we use the Howard and Kumar’s model for the analysis of grasping stability. This model gives the expressions for the changes in the contact forces as a function of the rigid body relative motion between the contact points and the grasped object in terms of contact stiffness matrices [35]. So, we can analyse the contact forces as well as the grasped object stability. This model given in the following Section (2.3.1) is analyzed for the multifingered robot hand, but it is applicable for the other types of grasping as lasso-type grasping.

2.3.1 Derivation of the Contact Stiffness Matrix

At the point of contact, a second-order model describing the surface of the finger is given by

$$2z_A + K_{u_A} x_A^2 + K_{v_A} y_A^2 = 0 \qquad (2.3)$$

where x_A and y_A are aligned with the principal axes of curvature of the finger. K_{u_A} is the curvature (the signed inverse of the radius of curvature, positive for convex

curves) along the x_A axis, K_{v_A} is the curvature along the y_A axis, and z_A is the outwardly pointing normal as seen in Figure (2.25). x_A, y_A, and z_A define an orthogonal coordinate system, fixed in space, which will be refered to as o_A. The orthogonal curvilinear coordinate system (coordinates u, v) is defined on each surface. As seen in Figure (2.25-c), the coordinates u_A and v_A describe the location of the points on the finger, x_A is tangent to the coordinate curve, $v_A =$constant, passing through the origin, and y_A is tangent to the coordinate curve, $u_A =$constant, passing through the origin. A similar definition is made for the grasped object, using the subscript "B". The angle ψ specifies the orientation about the common normal of one body with respect to the other. In figure (2.25-b), this angle is defined such that a positive (counterclockwise) rotation of x_A about z_A through ψ aligns the axes x_A and x_B. In general, $\psi \neq 0$. That is, the principal axes of curvature of the finger and grasped object are not, in general, aligned.

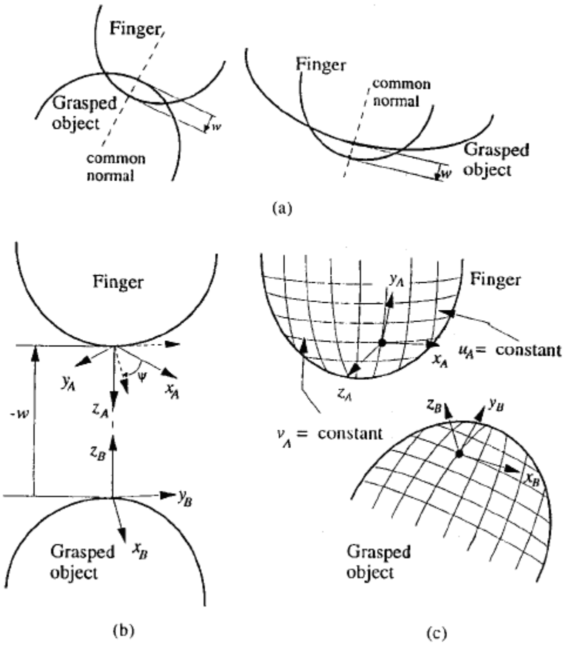

Figure (2.25) Parameterization of two bodies in contact [35]

2.3.1.1 Kinematic of Contacts

A small displacement of the grasped object is given by the twist $\begin{bmatrix}\Delta x_A & \Delta y_A & \Delta z_A & \Delta\theta_{xA} & \Delta\theta_{yA} & \Delta\theta_{zA}\end{bmatrix}^T$ in the fixed frame o_A. The relationship between the change in the location of contact point on the finger and the movement of the grasped object can be written as;

$$\begin{bmatrix}\Delta u_A \\ \Delta v_A\end{bmatrix} = (L_A + \widetilde{L}_B)^{-1}\left(\widetilde{L}_B\begin{bmatrix}\Delta x_A \\ \Delta y_A\end{bmatrix} - \Lambda\begin{bmatrix}\Delta\theta_{xA} \\ \Delta\theta_{yA}\end{bmatrix}\right) \qquad (2.4)$$

$$\Delta\psi = \Delta\theta_{zA} \qquad (2.5)$$

where

$$L_A = \begin{bmatrix}-K_{u_A} & 0 \\ 0 & -K_{v_A}\end{bmatrix}$$

$$\widetilde{L}_B = \begin{bmatrix}\cos\psi & \sin\psi \\ \sin\psi & -\cos\psi\end{bmatrix}\begin{bmatrix}-K_{u_B} & 0 \\ 0 & -K_{v_B}\end{bmatrix}\begin{bmatrix}\cos\psi & \sin\psi \\ \sin\psi & -\cos\psi\end{bmatrix}$$

$$\Lambda = \begin{bmatrix}0 & 1 \\ -1 & 0\end{bmatrix}$$

As seen in Figure (2.25-a), an additional contact coordinate, w have to be defined as the maximum rigid body penetration between the two contacting bodies.

$$\Delta w = -\Delta z \qquad (2.6)$$

2.3.1.2 Compliant Contact

If the contacts are nonconformal and the strains are small, the relationship between the normal force, F_n, and the gross rigid body motion along the line joining the centers of curvature of the contacting bodies (which according to terminology is the penetration w) can be given by

$$F_n \approx f(K_{u_A}, K_{v_A}, K_{u_B}, K_{v_B}, E^*) w^{3/2} \qquad (2.7)$$

Where E^* is a function of the material properties of the contacting bodies. Equation (2.7) can be linearized to give

$$F_n = F_{no} + k_n \Delta w \qquad (2.8)$$

Where F_{no} is the nominal contact force and $\Delta F_n = F_n - F_{no}$ is change in the contact force. The constant of proportionality, k_n, is the linearized spring constant. This equation is valid if the small changes in the generalized coordinates caused by the disturbance twist result in a relative rigid body motion, Δw, such that $\Delta w << w$.

2.3.1.3 Normal Forces

When there is a relative rigid body motion, the rigid body penetration, Δw, changes and the contacts move across the surface of the contacting objects. Thus the normal force changes in magnitude and in direction, and further, its line of action changes. The change in direction about the x_A axis is given by the small angle $-K_{v_A} \Delta v_A$. Thus, in the o_A frame, the change in the normal force is given by

$$\begin{bmatrix} \Delta F_x \\ \Delta F_y \end{bmatrix} = -F_{no} L_A \begin{bmatrix} \Delta u_A \\ \Delta v_A \end{bmatrix} \qquad (2.9)$$

$$\Delta F_z = k_n \Delta w \qquad (2.10)$$

Because the contact point moves through a small displacement given by $[\Delta u_A \quad \Delta v_A \quad -\Delta w]^T$, the resulting moments, expressed in o_A are given by

$$\begin{aligned} \Delta M_x &= \Delta v_A(F_z + \Delta F_z) + \Delta w(F_y + \Delta F_y) \qquad (2.11) \\ &\approx F_{no}\Delta v_A \end{aligned}$$

$$\begin{aligned} \Delta M_y &= -\Delta u_A(F_z + \Delta F_z) - \Delta w(F_x + \Delta F_x) \qquad (2.12) \\ &\approx F_{no}\Delta u_A \end{aligned}$$

2.3.1.4 Tangential Forces

Tangential contact forces arise due to firiction. The tangential force lies in the $x_A - y_A$ plane, and in general has components in both the x_A and y_A directions. These two components are refered as F_{txo} and F_{tyo}, respectively and the tangential force is defined as $F_{to} = [F_{txo} \quad F_{tyo}]^T$. When a relative motion occurs between two objects which remain in contact, the movement is a combination of rolling, twisting, and sliding. Pure rolling and pure twisting about the contact normal do not affect the magnitude of the frictional force, because there is no relative motion between the points in contact. If there is a relative tangential motion at the contact, this motion induces a tangential deformation. The expressions for the change in tangential forces and moments on the grasped object, expressed in o_A, can be written as

$$\begin{bmatrix} \Delta F_x \\ \Delta F_y \end{bmatrix} = -k_t \begin{bmatrix} \Delta x_A \\ \Delta y_A \end{bmatrix} \qquad (2.13)$$

$$\begin{aligned} \Delta M_x &= \Delta y_A(F_z + \Delta F_z) + \Delta w(F_y + \Delta F_y) \qquad (2.14) \\ &\approx F_{tyo}\Delta w \end{aligned}$$

$$\begin{aligned} \Delta M_y &= -\Delta x_A(F_z + \Delta F_z) - \Delta w(F_x + \Delta F_x) \qquad (2.15) \\ &\approx -F_{txo}\Delta w \end{aligned}$$

$$\begin{aligned} \Delta M_z &= \Delta x_A(F_y + \Delta F_y) - \Delta y_A(F_x + \Delta F_x) \qquad (2.16) \\ &\approx F_{tyo}\Delta x_A - F_{txo}\Delta y_A \end{aligned}$$

2.3.1.5 Moments About the Contact Normal

M_o denotes the torsional moment applied to the grasped object at equilibrium. A rotation about the contact normal will produce a change in the moment proportional to the torsional stiffness, k_θ. Because a relative displacement produces a change in the angle of the contact normal, a relative displacement will also cause a change in the moment about the x_A and y_A axes. In Cartesian coordinates, the change in moments on the grasped object can be expressed as

$$\begin{bmatrix} \Delta M_x \\ \Delta M_y \end{bmatrix} = -M_o L_A \begin{bmatrix} \Delta u_A \\ \Delta v_A \end{bmatrix} \tag{2.17}$$

$$\Delta M_z = -k_\theta \Delta \psi \tag{2.18}$$

2.3.1.6 The Contact Stiffness Matrix

At each contact, the above equations can be combined into a single equation, referenced to the local contact frame o_A, given by

$$\Delta F = -\kappa_c \Delta x_A \tag{2.19}$$

where ΔF is the change in the force, Δx_A is the rigid body motion measured in o_A, and κ_c is the 6×6 contact stiffness matrix given the below.

$$\Delta F = \begin{bmatrix} \Delta F_x \\ \Delta F_y \\ \Delta F_z \\ \Delta M_x \\ \Delta M_y \\ \Delta M_z \end{bmatrix}, \quad \Delta x_A = \begin{bmatrix} \Delta x_A \\ \Delta y_A \\ \Delta z_A \\ \Delta \theta_{xA} \\ \Delta \theta_{yA} \\ \Delta \theta_{zA} \end{bmatrix}$$

$$\kappa_c = \begin{bmatrix} F_{no} L_A (L_A + \widetilde{L}_B)^{-1} \widetilde{L}_B + k_t I_{22} & 0_{2\times1} & -F_{no} L_A (L_A + \widetilde{L}_B)^{-1} \Lambda & 0_{2\times1} \\ 0_{1\times2} & k_n & 0_{1\times2} & 0 \\ (M_o L_A - F_{no} \Lambda)(L_A + \widetilde{L}_B)^{-1} \widetilde{L}_B & \Lambda F_{to} & (F_{no} \Lambda - M_o L_A)(L_A + \widetilde{L}_B)^{-1} \Lambda & 0_{2\times1} \\ F_{to}^T \Lambda & 0 & 0_{1\times2} & k_\theta \end{bmatrix}$$

(2.20)

If the contact is frictionless, or a frictional contact, κ_c must be altered as follows:

$$\text{Frictionless contact}: F_{to} = 0 \text{ and } M_o = k_\theta = k_t = 0$$

$$\text{Frictional hard contact:} M_o = k_\theta = 0.$$

2.3.2 Multiple Contact Point Grasp

In the prior section, the equations for the changes in forces and moments at a single contact point in response to an arbitrary motion are formulated. In this section, the individual contributions at each of the multiple contacts into a single expression are combined.

2.3.2.1 Constant External Forces

The constant external forces such as gravitational forces are denoted as g. While these forces are constant, the vector representation of the wrench in a particular coordinate system does depend on the changes in configuration of the system. Consider an external force on the grasped object acting through the center of gravity. The coordinate system is fixed in space, o_{cg}, whose axes are aligned such that the external force acts in the z_{cg} direction. A rigid body motion of the grasped object will not change the force, but will result in changes in the moment of the external force about the origin of o_{cg}. If mg denotes the external force, this result is given by

$$\begin{bmatrix} \Delta F_x \\ \Delta F_y \\ \Delta F_z \\ \Delta M_x \\ \Delta M_y \\ \Delta M_z \end{bmatrix} = -\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -mg & 0 & 0 & 0 & 0 \\ mg & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{cg} \\ \Delta y_{cg} \\ \Delta z_{cg} \\ \Delta \theta_{x_{cg}} \\ \Delta \theta_{y_{cg}} \\ \Delta \theta_{z_{cg}} \end{bmatrix} \quad \text{or}$$

$$\Delta F = -\kappa_{cg} \Delta x_{cg} \tag{2.21}$$

2.3.2.2 Coordinate Transformations of the Stiffness Matrix

When the multiple contacts are modelled, the leading subscript is used to denote the contact. For example, ${}^{i}\kappa_{c}$ would be the stiffness matrix κ_{c} for the i-th contact and ${}^{i}o_{A}$ is the fixed frame at the i-th contact. Note further that Δx_{A} and Δy_{A} refer to the contact frame, but they can be related to other reference frames through appropriate transformations. The fixed frame, O, is used at an arbitrary point. $\left({}^{i}d_{x} \quad {}^{i}d_{y} \quad {}^{i}d_{z}\right)$ is the coordinates of the frame O as seen from the fixed frame for the i-th contact, ${}^{i}o_{A}$, then the change in force at O for any infinitesimal rotation or displacement, due to contact i, is given by

$$\Delta F = -{}^{i}T^{T}\,{}^{i}\kappa_{c}\,{}^{i}T\Delta x_{o} \tag{2.22}$$

where in the spatial case

$${}^{i}T = \begin{bmatrix} {}^{i}R & {}^{i}D \\ 0 & {}^{i}R \end{bmatrix}$$

${}^{i}D$ is a skew-symmetric matrix of i-th contact and ${}^{i}D = \begin{bmatrix} 0 & -{}^{i}d_{z} & {}^{i}d_{y} \\ {}^{i}d_{z} & 0 & -{}^{i}d_{x} \\ -{}^{i}d_{y} & {}^{i}d_{x} & 0 \end{bmatrix}$

$$\begin{bmatrix} {}^{i}x_{A} \\ {}^{i}y_{A} \\ {}^{i}z_{A} \end{bmatrix} = \left[{}^{i}R_{3\times3}\right] \begin{bmatrix} x_{o} \\ y_{o} \\ z_{o} \end{bmatrix}$$

0 is the 3×3 zero matrix, and ${}^{i}R$ is the 3×3 rotation matrix that transforms vectors in frame O to the reference frame ${}^{i}o_{A}$.

2.3.2.3 The Intrinsic Stiffness Matrix

κ_{o} is defined as the intrinsic stiffness matrix, referenced to the world frame O, given by

$$\Delta F = -\kappa_{o}\Delta x_{o} \tag{2.23}$$

where

$$\kappa_o = \sum_{i=1}^{N} {}^iT^T\, {}^i\kappa_c\, {}^iT + T_{cg}^T \kappa_{cg} T_{cg}$$

A grasp will be stable if κ_o is positive definite (whether or not the grasp is force closed).

2.4 Physical Model of a Snake-Like Robot

The choice and arrangement of a hyper-redundant robot's actuators and mechanical structure are important to realize a task. The selection of a particular morphology will obviously depend heavily upon the functional and operational requirements of a particular application. The snake-like robot (Figure (2.26)) used in [27] is selected as a model that we heavily modify for lasso-type grasping in this thesis. Each link of its has the 2-DOF joint equipped with two torque actuators (motor) and two passive wheels on the two side of the links. These wheels are used to decrease the friction between the surface and snake-like robot in tangential directions. Each link is rigid with uniformly distributed mass.

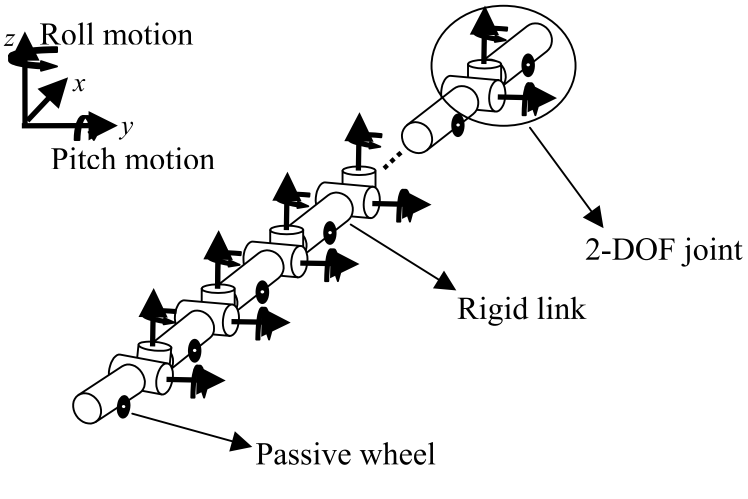

Figure (2.26) Physical Model of Snake-Like Robot Used in [26]

Since the physical morphology of snake-like robot can be changed under particular task achieved by snake-like robot, some physical arrangements must be done to be achieved the particular task. For this thesis, this task is the lasso-type grasping of the snake-like robot. Because of theoretical reasons given in Chapter 3, needed physical arrangements of the model given in Figure (2.26) have to be to achieve the lasso-type grasping as follows.

- The front links involve in the grasping task should be the wheelless links.
- The weight of front links involve in the grasping task should be so light that their motions which construct the grasping curve can not corrupt the stability of the last wheeled links.

Under these arrangements, the snake-like robot that we use in this thesis is seen as in Figure (2.27). The front links involved in grasping task are wheelless links and are less heavy than the last wheeled links. This prevents the corruption of last wheeled links stability during the motion of front wheelless links when grasping the object.

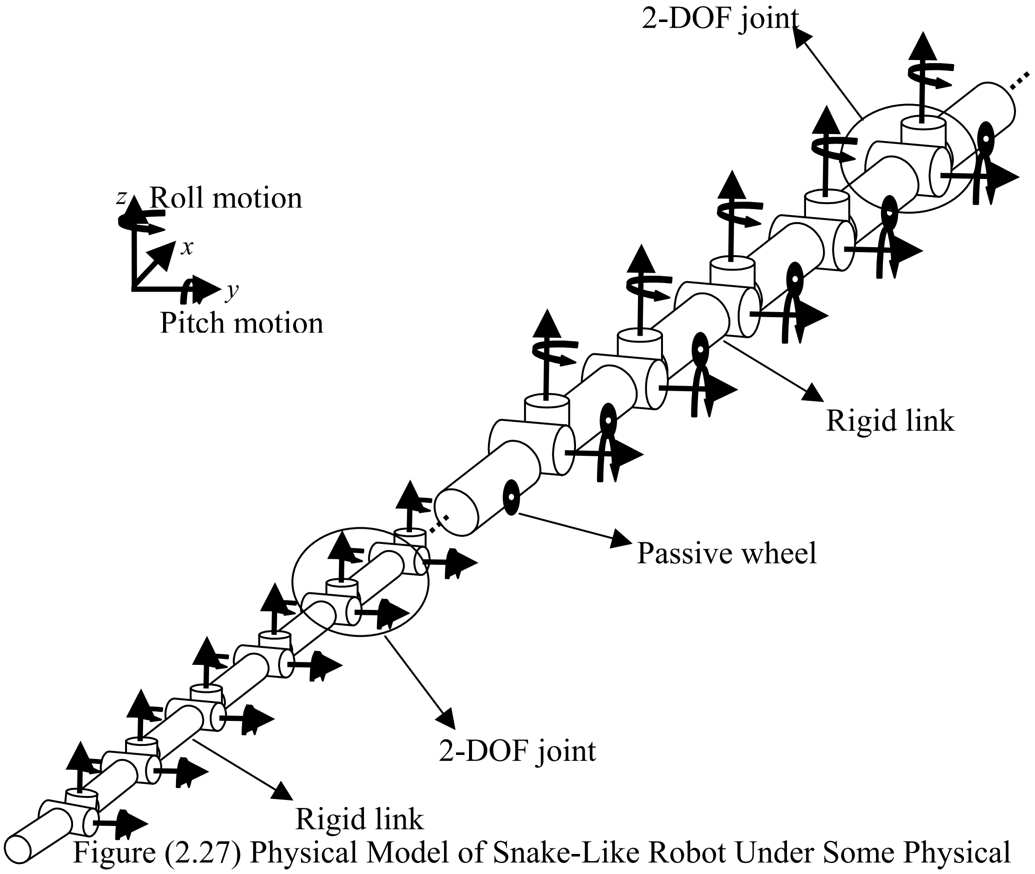

Figure (2.27) Physical Model of Snake-Like Robot Under Some Physical Arrangements

2.4.1 Friction Model of Snake-Like Robot

For this physical model, the key property of snake-like robot in achieving serpentine locomotion is the friction between the links and horizontal surface. Therefore we must model the frictional equation of motion.

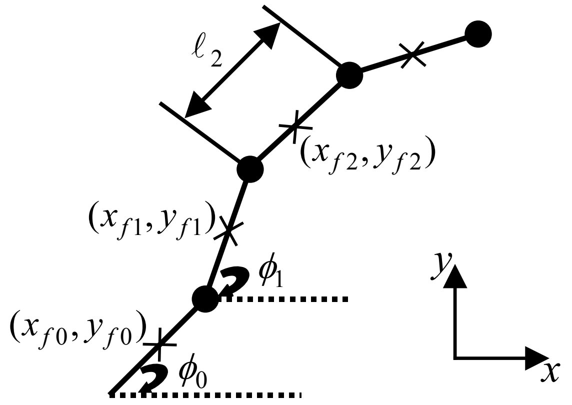

Figure (2.28) The model of the fornt wheelless links

Using the concept given in [36], we can model the friction of the snake-like robot. Let us consider the snake-like robot depicted in Figure (2.28), which consist of n links connected through $n-1$ joints. As given in [36], we give the some notation to model the friction of the snake-like robot as follows.

Fix $x-y$ axes in the inertial frame and consider the i-th link of the snake-like robot. The link is of mass m_i, length ℓ_i, and moment of inertia $J_i(=m_i\ell_i^2/12)$. Let (x_{fi}, y_{fi}) and ϕ_i be the coordinates of the center of gravity and the angle between the link and the $x-$axis, respectively. Denote by x, y, and ϕ the $n-$dimensional vectors whose i-th entries are x_{fi}, y_{fi}, and ϕ_i, respectively. Define

$$A := \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ & & & 1 \end{bmatrix} \in \Re^{(n-1)\times n}, \qquad D := \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & & -1 \\ & & & 1 \end{bmatrix} \in \Re^{(n-1)\times n}$$

$$E := \begin{bmatrix} e & 0 \\ 0 & e \end{bmatrix} \qquad e := \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T \in \Re^n, \qquad m := \sum_{i=1}^{n} m_i$$

$$S_\phi := diag(\sin\phi_1,...,\sin\phi_n), \qquad C_\phi := diag(\cos\phi_1,...,\cos\phi_n)$$

$$J := diag(J_1,...,J_n), \quad M := diag(m_1,...,m_n), \quad L := diag(\ell_1/2,...,\ell_n/2)$$

$$H := LA^T(DM^{-1}D^T)^{-1}AL, \qquad N := M^{-1}D^T(DM^{-1}D^T)^{-1}AL$$

$$\Gamma := J + S_\phi H S_\phi + C_\phi H C_\phi,\ \Lambda := S_\phi H C_\phi - C_\phi H S_\phi,\ \ \eta := \begin{bmatrix} S_\phi N^T & -C_\phi N^T \end{bmatrix}^T$$

$$B := D\Gamma^{-1}D^T,\ \ \kappa := \Gamma^{-1}D^T B^{-1},\ \ \rho := 1/(e^T\Gamma e),\ e_\rho := \rho e$$

In the above, the symbols D and A stand for the “difference” and the “addition” operators, respectively. The vector e is a basis of the kernel of D. It can be shown that the matrices Γ and Λ depend on the relative angles $D\phi$ only, and thus are determined by the shape of the snake robot.

Consider the free-body diagram for the ith link depicted in Figure (2.29), where f_i and τ_i are the force and torque due to the friction between the link and horizontal surface, g_i and g_{i-1} are constraint forces from the adjacent links, and u_i and u_{i-1} are the joint torques from the actuators. Let $\tau \in \Re^n$ be the vector whose ith entry is τ_i, and similarly for $f_x, f_y \in \Re^n$ and $g_x, g_y, u \in \Re^{n-1}$, where (f_{xi}, f_{yi}) are the (x,y) components of the friction force vector f_i. Finally, the total friction force vector f and the whole position vector z can be defined as

$$f := \begin{bmatrix} f_x \\ f_y \end{bmatrix}, \qquad z := \begin{bmatrix} x \\ y \end{bmatrix}$$

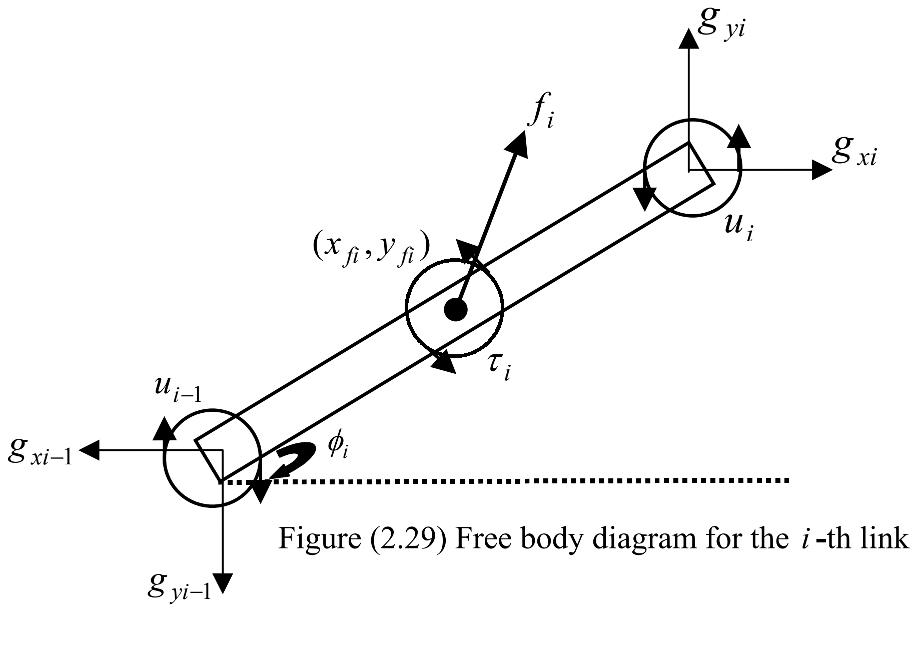

Figure (2.29) Free body diagram for the i-th link

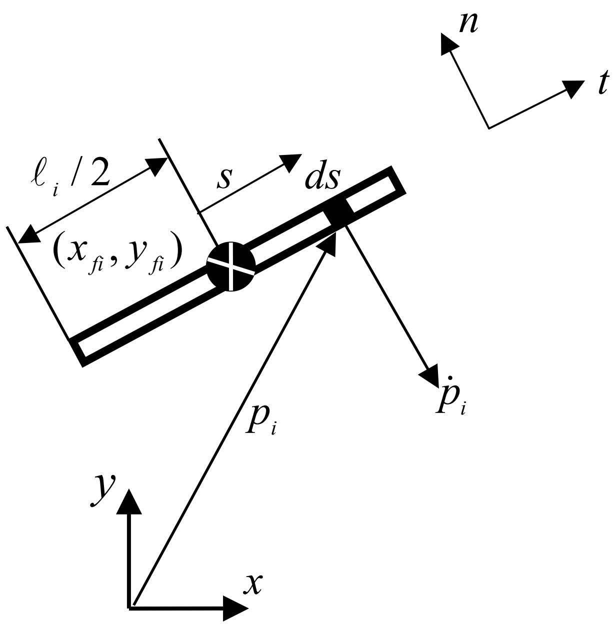

Figure (2.30) Infinitesimal segment on the i th link

2.4.1.1 Simple Friction Model

Consider the i th link of the snake robot depicted in Figure (2.30). The coordinates of the infinitesimal segment ds are given by

$$p_i = \begin{bmatrix} x_{fi} \\ y_{fi} \end{bmatrix} + \begin{bmatrix} \cos\phi_i \\ \sin\phi_i \end{bmatrix} s$$

Taking to derivative with respect to the time, we have the velocity vector in the $x-y$ frame:

$$\dot{p}_i = \begin{bmatrix} \dot{x}_{fi} \\ \dot{y}_{fi} \end{bmatrix} + \begin{bmatrix} -\sin\phi_i \\ \cos\phi_i \end{bmatrix} s\dot{\phi}_i$$

The tangential and normal components of the velocity vector $\dot{p}_i$ are given by

$$\begin{bmatrix} \tilde{v}_{ti} \\ \tilde{v}_{ni} \end{bmatrix} := \begin{bmatrix} \cos\phi_i & \sin\phi_i \\ -\sin\phi_i & \cos\phi_i \end{bmatrix} \dot{p}_i = \begin{bmatrix} v_{ti} \\ v_{ni} \end{bmatrix} + \begin{bmatrix} 0 \\ s\dot{\phi}_i \end{bmatrix} \tag{2.24}$$

where

$$\begin{bmatrix} v_{ti} \\ v_{ni} \end{bmatrix} := \begin{bmatrix} \cos\phi_i & \sin\phi_i \\ -\sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} \dot{x}_{fi} \\ \dot{y}_{fi} \end{bmatrix}. \tag{2.25}$$

Now assume that the friction force is modeled in the body $(t-n)$ frame as

$$\begin{bmatrix} df_{ti} \\ df_{ni} \end{bmatrix} = -\begin{bmatrix} c_{ti} & 0 \\ 0 & c_{ni} \end{bmatrix} \begin{bmatrix} \tilde{v}_{ti} \\ \tilde{v}_{ni} \end{bmatrix} d_{mi}$$

where c_{ti} and c_{ni} are the friction coefficients and dm_i is the mass of the infinitesimal segment (i.e., $dm_i = (m_i ds)/\ell_i$. Transforming the coordinates, the friction force in the inertial frame is

$$\begin{bmatrix} df_{xi} \\ df_{yi} \end{bmatrix} = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} df_{ti} \\ df_{ni} \end{bmatrix}.$$

Integrating over the link, the total friction force is given by

$$\begin{bmatrix} df_{xi} \\ df_{yi} \end{bmatrix} = -m_i \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} c_{ti} & 0 \\ 0 & c_{ni} \end{bmatrix} \begin{bmatrix} \cos\phi_i & \sin\phi_i \\ -\sin\phi_i & \cos\phi_i \end{bmatrix} \begin{bmatrix} \dot{x}_{fi} \\ \dot{y}_{fi} \end{bmatrix}$$

Finally, the total friciton torque around the center of mass of the link is

$$\tau_i = \int s\, df_{ni} = -\frac{m_i \ell_i^2}{12} c_{ni} \dot{\phi}_i = -c_{ni} J_i \dot{\phi}_i$$

where the positive direction is counterclockwise.

We now consider the whole system of connected links. It can be verified that the total friciton force and torque acting on the system can be expressed as

$$\boxed{\begin{aligned} f &= -\Omega_\phi D_f \Omega_\phi^T \dot{z} \\ \tau &= -D_\tau \dot{\phi} \end{aligned}} \qquad (2.26)$$

Where

$$D_f := \begin{bmatrix} C_t M & 0 \\ 0 & C_n M \end{bmatrix}, \qquad D_\tau := C_n J, \qquad \Omega_\phi := \begin{bmatrix} C_\phi & -S_\phi \\ S_\phi & C_\phi \end{bmatrix}$$

$$C_t := diag(c_{t_1}, ..., c_{t_n}), \qquad C_n := diag(c_{n_1}, ..., c_{n_n}).$$

2.4.1.2 Frictional Equations of Motion

Applying the principle to the free-body diagram of the ith link in Figure (2.29) and assembling into the n-link snake-robot, we obtain the equations of translational motion.

$$\boxed{\begin{aligned}&M\ddot{y} = f_y + D^T g_y\\&M\ddot{x} = f_x + D^T g_x\end{aligned}} \tag{2.27}$$

and the equation of rotational motion

$$J\ddot{\phi} = \tau - S_\phi LA^T g_x + C_\phi LA^T g_y + D^T u. \tag{2.28}$$

The equations of motion in terms of the absolute ϕ and the positon of the center of gravity can be given as

$$w := \begin{bmatrix} w_x \\ w_y \end{bmatrix} = \frac{1}{m}\begin{bmatrix} e^T Mx \\ e^T My \end{bmatrix}$$

In view of Figure (2.28), the vectors x, y, and ϕ are constrained by

$$\boxed{\begin{aligned}&Dx + AL\cos\phi = 0\\&Dy + AL\sin\phi = 0.\end{aligned}} \tag{2.29}$$

From the definition of w and Equation (2.29), we have

$$Tx = \begin{bmatrix} -AL\cos\phi \\ w_x \end{bmatrix}, \quad Ty = \begin{bmatrix} -AL\sin\phi \\ w_y \end{bmatrix}, \quad T := \begin{bmatrix} D \\ e^T M/m \end{bmatrix}.$$

Solving these equations for x and y, and taking the time derivatives, the translational velocity can be given by

$$\dot{z} = \eta\dot{\phi} + E\dot{w}$$

where

$$T^{-1} = \left[M^{-1}D^T(DM^{-1}D^T)^{-1}e\right]$$

Next the equations of translational motion Equation (2.27) is decomposed into two parts: expressions for the center of mass acceleration $\ddot{w}$ and the constraint forces g_x and g_y. Multiplying Equation (2.27) with TM^{-1} from the left, we have

$$T\ddot{x} = \begin{bmatrix} D\ddot{x} \\ \ddot{w}_x \end{bmatrix} = \begin{bmatrix} DM^{-1}f_x + DM^{-1}D^T g_x \\ e^T f_x / m \end{bmatrix} \tag{2.30}$$

For the x coordinate and similarly for the y coordinate. Solving for g_x and g_y, we have

$$\boxed{\begin{aligned}&g_x = (DM^{-1}D^T)^{-1}(AL(C_\phi\dot{\phi}^2 + S_\phi\ddot{\phi}) - DM^{-1}f_x)\\&g_y = (DM^{-1}D^T)^{-1}(AL(S_\phi\dot{\phi}^2 + C_\phi\ddot{\phi}) - DM^{-1}f_y)\end{aligned}} \tag{2.31}$$

where we noted from Equation (2.29) that

$$D\ddot{x} = AL(C_\phi \dot{\phi}^2 + S_\phi \ddot{\phi}) \qquad D\ddot{y} = AL(S_\phi \dot{\phi}^2 + C_\phi \ddot{\phi}) .$$

Substituting Equation (2.31) into Equation (2.28), and using the second row of Equation (2.30), we have

$$\Gamma \ddot{\phi} + \Lambda \dot{\phi}^2 = D^T u + \tau + \eta^T f$$

$$m\ddot{w} = E^T f \tag{2.32}$$

The complete set of equations of motion is now derived

$$\begin{bmatrix} \Gamma & 0 \\ 0 & mI \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{w} \end{bmatrix} + \begin{bmatrix} \Lambda \dot{\phi}^2 \\ 0 \end{bmatrix} + \begin{bmatrix} R & S \\ S^T & \varphi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} D^T \\ 0 \end{bmatrix} u \tag{2.33}$$

where

$$\begin{bmatrix} R & S \\ S^T & \varphi \end{bmatrix} := \begin{bmatrix} D_\tau & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \eta^T \\ E^\tau \end{bmatrix} \Omega_\phi D_f \Omega_\phi^T \begin{bmatrix} \eta & E \end{bmatrix}$$

And D_τ and D_f are defined by Equation (2.26).

2.5 Kinematics and Control of Snake-Like Robot

Let us remember again the model of the 3-dimensional snake-like robot (Figure (2.31)) formed by serially connected links that we will adopt in this thesis.

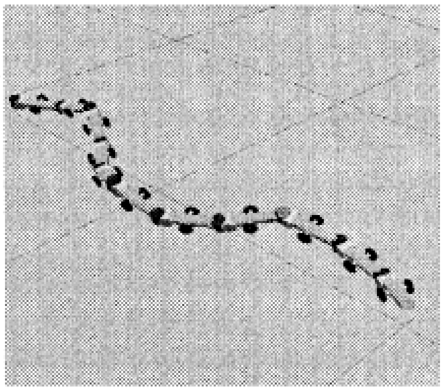

Figure (2.31) Model of a 3-dimensional snake-like robot

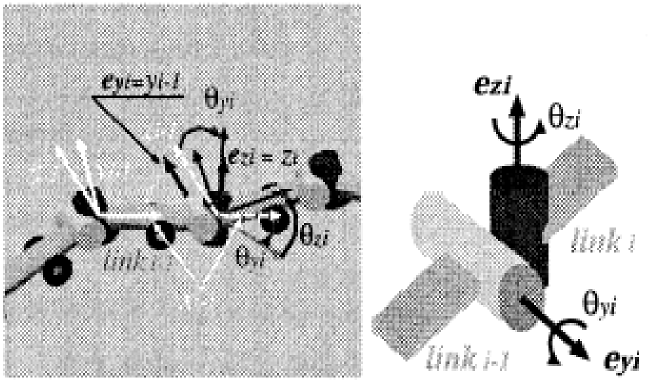

Figure (2.32) Link and Joint Model

2.5.1 Model of a Snake-Like Robot

The link number is n and each joint has two degrees of freedom rotated around the axes $e_{yi} \in \Re^3$ and $e_{zi} \in \Re^3$ with the angles θ_{yi} and θ_{zi}. As shown in Figure (2.32), l_i is the link vector (ℓ is the link length) and the axes e_{zi}, e_{yi} can be described by

$$e_{yi} = \prod_{k=0}^{i-1} E^{j\theta_{yk}} E^{k\theta_{zk}} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \tag{2.34}$$

$$e_{zi} = (\prod_{k=0}^{i-1} E^{j\theta_{yk}} E^{k\theta_{zk}}) E^{j\theta_{yi}} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{2.35}$$

$$l_i = \prod_{k=0}^{i} E^{j\theta_{yk}} E^{k\theta_{zk}} \begin{bmatrix} \ell & 0 & 0 \end{bmatrix}^T \tag{2.36}$$

where θ_{y0} and θ_{z0} are the posture of the head link, $E^{j\theta_y}$ and $E^{k\theta_z}$ are the rotation matrices around the axes y and z, respectively. The position of the wheel mounted onto the link i, $P_{fi} \in \Re^3 (= \begin{bmatrix} x_{fi} & y_{fi} & z_{fi} \end{bmatrix}^T)$, and its velocity $\dot{P}_{fi} \in \Re^3 (= \begin{bmatrix} \dot{x}_{fi} & \dot{y}_{fi} & \dot{z}_{fi} \end{bmatrix}^T)$, are thus derived as

$$P_{fi} = P_0 + \sum_{j=0}^{i-1} l_j + l_{fi} \tag{2.37}$$

$$\dot{P}_{fi} = \dot{P}_0 + \sum_{j=0}^{i-1} (w_j \times l_j) + w_i \times l_{fi} \tag{2.38}$$

$$w_i = \sum_{j=0}^{i} (\dot{\theta}_{yj} e_{yj} + \dot{\theta}_{zj} e_{zj}) \qquad (2.39)$$

where $P_0 (= [x_0 \quad y_0 \quad z_0]^T)$ is the head position of the snake robot, l_{fi} is the vector from the joint to the wheel, and w_i is the rotation velocity vector.

If the non-holonomic constraint is assumed, the normal direction slip does not happen at the contact point of the wheel with the ground, the following constraint must be established.

$$h_i^T \dot{P}_{fi} = \dot{x}_{fi} \sin\phi_i - \dot{y}_{fi} \cos\phi_i = 0 \qquad (2.40)$$

where $h_i = [\sin\phi_i \quad -\cos\phi_i \quad 0]^T$, and ϕ_i is the absolute angle of the link i mapped onto the XY plane, as shown in Figure (2.32).

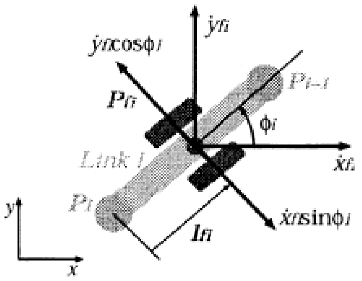

Figure (2.33) Non-holonomic constraint of the link contacted with the ground

From Equation (2.38) and (2.40), we can have the non-holonomic constraint equation of the robot. It is given by

$$h_i^T (\dot{P}_0 + \sum_{j=0}^{i-1} (w_j \times l_j) + w_i \times l_{fi}) = 0$$

$$h_i^T \left\{ \dot{P}_0 - (\sum_{k=0}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{y0} e_{y0} + \dot{\theta}_{z0} e_{z0}) \right\} = h_i^T \left\{ \sum_{j=1}^{i} \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{yj} e_{yj} + \dot{\theta}_{zj} e_{zj}) \right\} \right\}$$

(2.41)

2.5.2 Control of Snake-Like Robot

The head of the snake-like robot must be possibly controlled to a given position or along a designed path in addition of its creeping locomotion, while the robot performs 3-dimensional motion. For the purpose of future analysis of the 3-dimensional motion of the robot, a control loop for possible control of the head of the robot, its wheel-heights and the oscillating joint used for avoiding the singularity is built.

2.5.2.1 Control of head position and orientation

We know that if the m wheels contact with the ground and so, m active non-holonomic constraints are established, we have the system Equation (2.41) with the input vector of the joint velocities $u \in \Re^{2(n-1)} (= \begin{bmatrix} \dot{\theta}_{y1} & \dots & \dot{\theta}_{y(n-1)} & \dot{\theta}_{z1} & \dots & \dot{\theta}_{z(n-1)} \end{bmatrix}^T)$, and the output vector of head position and orientation $w_0 \in \Re^4 (= \begin{bmatrix} x_0 & y_0 & \theta_{y0} & \theta_{z0} \end{bmatrix}^T)$. And therefore, from the non-holonomic constraint and derivative of the wheel position, the head position and orientation can be well controlled.

$$\mathrm{A}_0 \dot{w}_0 = \mathrm{B}_0 u \tag{2.42}$$

where

$$\mathrm{A}_0 = \{a_{0ij}\} \in \Re^{m \times 4}$$

$$\mathrm{B}_0 = \begin{bmatrix} \mathrm{B}_{0y} & | & \mathrm{B}_{0z} \end{bmatrix} \in \Re^{m \times 2(n-1)}$$

$$\mathrm{B}_{0y} = \{b_{0yij}\} \in \Re^{m \times (n-1)}$$

$$\mathrm{B}_{0z} = \{b_{0zij}\} \in \Re^{m \times (n-1)}$$

Therein,

$$a_{0ij} = \begin{cases} \sin\phi_i & j=1 \\ -\cos\phi_i & j=2 \\ -h_i^T \left\{ (\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{y0} \right\} & j=3 \\ -h_i^T \left\{ (\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{z0} \right\} & j=4 \end{cases}$$

$$b_{0yij} = \begin{cases} h_i^T \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{yj} \right\} & j \le i \\ 0 & j > i \end{cases}$$

$$b_{0zij} = \begin{cases} h_i^T \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{zj} \right\} & j \le i \\ 0 & j > i \end{cases}.$$

Since m is smaller than $2(n-1)$ for the case $n > 2$, the system (2.42) is always controllable.

2.5.2.2 Control of link rise

Link rise in link i is the difference between the z-component of head link (z_0) and the z-component of i-th link (z_{fi}). To control the link rise, we can derive the vertical velocity of position of the wheel mounted onto the links and pick the *z*-components from Equation (2.38) by

$$h_z^T \dot{P}_{fi} = \dot{z}_{fi} \quad (i = 1 \sim n-1) \qquad (2.43)$$

where $h_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, the relation between the z-component of the velocities and the rotation velocities of joints is given by

$$h_z^T (\dot{P}_0 + \sum_{j=0}^{i-1} (w_j \times l_j) + w_i \times l_{fi}) = 0 \qquad (2.44)$$

$$\dot{z}_{fi} - \dot{z}_0 + h_z^T \left\{ (\sum_{k=0}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{y0} e_{y0} + \dot{\theta}_{z0} e_{z0}) \right\} = -h_z{}^T \left\{ \sum_{j=1}^{i} \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{yj} e_{yj} + \dot{\theta}_{zj} e_{zj}) \right\} \right\}$$

The system equation for controlling the link rise is described by

$$\mathrm{A}_z \dot{w}_0 + \dot{z} = \mathrm{B}_z u \qquad (2.45)$$

where $\dot{z} = \begin{bmatrix} \dot{z}_{f1} - \dot{z}_0 & \dots & \dot{z}_{f(n-1)} - \dot{z}_0 \end{bmatrix}^T \in \Re^{n-1}$ and

$$\mathrm{A}_z = \{a_{zyij}\} \in \Re^{(n-1)\times 4}$$

$$\mathrm{B}_z = \begin{bmatrix} \mathrm{B}_{zy} & | & \mathrm{B}_{zz} \end{bmatrix} \in \Re^{(n-1)\times 2(n-1)}$$

$$\mathrm{B}_{zy} = \{b_{zyij}\} \in \Re^{(n-1)\times(n-1)}$$

$$\mathrm{B}_{zz} = \{b_{zzij}\} \in \Re^{(n-1)\times(n-1)}$$

Therein,

$$a_{zyij} = \begin{cases} 0 & j = 1,2 \\ -h_z^T\left\{(\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{y0}\right\} & j = 3 \\ -h_z^T\left\{(\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{z0}\right\} & j = 4 \end{cases}$$

$$b_{zyij} = \begin{cases} -h_z^T\left\{(\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{yj}\right\} & j \le i \\ 0 & j > i \end{cases}$$

$$b_{zzij} = \begin{cases} -h_z^T\left\{(\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{zj}\right\} & j \le i \\ 0 & j > i \end{cases}$$

2.5.2.3 Avoiding singular configuration

In order to avoid the singular configuration of the robot, the relative angle ψ_i of the vector that is the mapped one of the link $l_i(=\begin{bmatrix} l_{xi} & l_{yi} & l_{zi}\end{bmatrix}^T, i = 0 \sim n-1)$ onto the *XY* plane can be controlled. The equation to show the relation of $\dot{\psi}_i$ and the rotation velocities of links can be derived, an given by

$$\dot{\psi}_i = \dot{\phi}_i - \dot{\phi}_{i-1} \tag{2.46}$$

$$\dot{\phi}_i = \frac{\varsigma_{yi} l_{xi} - l_{yi}\varsigma_{xi}}{l_{xi}^2 + l_{yi}^2}, \qquad \varsigma_i = w_j \times l_i, (=\begin{bmatrix}\varsigma_{xi} & \varsigma_{yi} & \varsigma_{zi}\end{bmatrix}^T)$$

$$\varsigma_i = (\dot{\theta}_{y0} e_{y0} + \dot{\theta}_{z0} e_{z0}) \times l_i + \sum_{j=1}^{i} (\dot{\theta}_{yj} e_{yj} + \dot{\theta}_{zj} e_{zj}) \times l_i$$

$$\varsigma_i = \begin{bmatrix}\varsigma_{xi}^0 & \varsigma_{yi}^0 & \varsigma_{zi}^0\end{bmatrix}^T + \begin{bmatrix}\varsigma_{xi}^j & \varsigma_{yi}^j & \varsigma_{zi}^j\end{bmatrix}^T \tag{2.47}$$

$$\dot{\psi}_i - \frac{(\varsigma^0_{yi} l_{xi} - l_{yi}\varsigma^0_{xi})}{l^2_{xi} + l^2_{yi}} + \frac{(\varsigma^0_{yi} l_{x(i-1)} - l_{yi}\varsigma^0_{x(i-1)})}{l^2_{x(i-1)} + l^2_{y(i-1)}} = \frac{(\varsigma^j_{yi} l_{xi} - l_{yi}\varsigma^j_{xi})}{l^2_{xi} + l^2_{yi}} - \frac{(\varsigma^j_{yi} l_{x(i-1)} - l_{yi}\varsigma^j_{x(i-1)})}{l^2_{x(i-1)} + l^2_{y(i-1)}} \tag{2.48}$$

The control system equation of the robot shape on the XY plane is expressed by

$$\mathrm{A}_\psi \dot{w}_0 + \dot{\psi}_i = \mathrm{B}_\psi u \tag{2.49}$$

where

$$\mathrm{A}_\psi = \{a_{\psi ij}\} \in \Re^{1\times 4}$$

$$\mathrm{B}_\psi = \left[\mathrm{B}_{\psi y} \quad | \quad \mathrm{B}_{\psi z}\right] \in \Re^{1\times 2(n-1)}$$

$$\mathrm{B}_{\psi y} = \{b_{\psi yij}\} \in \Re^{1\times(n-1)}$$

$$\mathrm{B}_{\psi z} = \{b_{\psi zij}\} \in \Re^{1\times(n-1)}$$

Therein,

$$a_{\psi ij} = \begin{cases} 0 & j = 1,2 \\ \dfrac{{}^y\xi_y l_{xi} - l_{yi}\,{}^y\xi_{xi0}}{l^2_{xi} + l^2_{yi}} + \dfrac{{}^y\xi_{y(i-1)0} l_{x(i-1)} - l_{y(i-1)}\,{}^y\xi_{x(i-1)0}}{l^2_{x(i-1)} + l^2_{y(i-1)}} & j = 3 \\ \dfrac{{}^z\xi_y l_{xi} - l_{yi}\,{}^z\xi_{xi0}}{l^2_{xi} + l^2_{yi}} + \dfrac{{}^z\xi_{y(i-1)0} l_{x(i-1)} - l_{y(i-1)}\,{}^z\xi_{x(i-1)0}}{l^2_{x(i-1)} + l^2_{y(i-1)}} & j = 4 \end{cases}$$

$$b_{\psi yij} = \begin{cases} \dfrac{{}^y\xi_{yij} l_{xi} - l_{yi}\,{}^y\xi_{xij}}{l^2_{xi} + l^2_{yi}} - \dfrac{({}^y\xi_{y(i-1)j} l_{x(i-1)} - l_{y(i-1)}\,{}^y\xi_{x(i-1)j})}{l^2_{x(i-1)} + l^2_{y(i-1)}} & j \le i \\ 0 & j > i \end{cases}$$

$$b_{\psi zij} = \begin{cases} \dfrac{{}^z\xi_{yij} l_{xi} - l_{yi}\,{}^z\xi_{xij}}{l^2_{xi} + l^2_{yi}} - \dfrac{({}^z\xi_{y(i-1)j} l_{x(i-1)} - l_{y(i-1)}\,{}^z\xi_{x(i-1)j})}{l^2_{x(i-1)} + l^2_{y(i-1)}} & j \le i \\ 0 & j > i \end{cases}$$

and ${}^{yij}\xi = e_{yj} \times l_i = \left[{}^y\xi_{xij} \quad {}^y\xi_{yij} \quad {}^y\xi_{zij}\right]^T, {}^{zij}\xi = e_{zj} \times l_i = \left[{}^z\xi_{xij} \quad {}^z\xi_{yij} \quad {}^z\xi_{zij}\right]^T$.

The system equations for controlling the head position and orientation, the link rises, and the robot shape on the XY plane, can thus be derived from all of the system equations as follows;

$$\mathrm{A}\dot{w} = \mathrm{B}u \tag{2.50}$$

where

$$\dot{w} = \begin{bmatrix} \dot{w}_0 & \dot{z} & \dot{\psi}_i \end{bmatrix} \in \Re^{n+4},$$

$$\dot{w} = \begin{bmatrix} \dot{x}_0 & \dot{y}_0 & \dot{\theta}_{y0} & \dot{\theta}_{z0} & \dot{z}_{f1} - \dot{z}_0 & \dots & \dot{z}_{f(n-1)} - \dot{z}_0 & \dot{\psi}_i \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_0 & \mathrm{O}_{m\times(n-1)} & \mathrm{O}_{m\times 1} \\ \mathrm{A}_z & \mathrm{I}_{(n-1)\times(n-1)} & \mathrm{O}_{(n-1)\times 1} \\ \mathrm{A}_\psi & \mathrm{O}_{1\times(n-1)} & 1 \end{bmatrix} \in \Re^{(m+n)\times(n+4)} \qquad \mathrm{B} = \begin{bmatrix} \mathrm{B}_0 \\ \mathrm{B}_z \\ \mathrm{B}_\psi \end{bmatrix} \in \Re^{(m+n)\times 2(n-1)}$$

and I is the unit matrix, O is the zero matrix or vector.

2.5.2.4 Controller design

If $m \le n-2$, the matrix B in Equation (2.50) has row full rank except at singular configuration. The system (2.50) is thus controllable and has redundant degrees of freedom while $m < n-2$. The control input u to control the head position and orientation, the link rises, and the robot shape mapped on the XY plane, w, can thus be derived from Equation (2.50), and given by

$$u = \mathrm{B}^{+}\mathrm{A}\{\dot{w}_d + \mathrm{K}(w_d - w)\} \tag{2.51}$$

where B^{+} is the pseudo- inverse of the matrix B derived by $\mathrm{B} = \mathrm{B}^T(\mathrm{B}\mathrm{B}^T)^{-1}$, K is the gain matrix given by $\mathrm{K} = diag\{K_i\}$, and w_d & $\dot{w}_d$ are the commanded input. Figure (2.36) shows the block diagram of the controller.

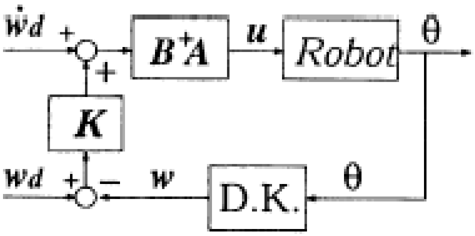

Figure (2.34) Block diagram of the controller

In computer simulation, while we have the control input u, the link rises and the robot shape mapped onto the XY plane can be derived from the joint angles obtained through integrating the joint velocities u. The head position and orientation w_0, however, can be only derived by integrating $\dot{w}_0 = \begin{bmatrix} \dot{x}_0 & \dot{y}_0 & \dot{\theta}_{y0} & \dot{\theta}_{z0} \end{bmatrix}^T$, that is the inverse solution of the system (2.52), given by

$$\dot{w}_0 = \mathrm{A}_0^+ \mathrm{B}_0 u \qquad (2.52)$$

Where $\mathrm{A}_0^+ = (\mathrm{A}_0^T \mathrm{A}_0)^{-1} \mathrm{A}_0^T$.

In order to have solution of the head position and orientation (or make the system observable), the matix A (or the matrix A_0) must be column full rank. That is,

$$m + n \geq n + 4 \quad \Rightarrow \quad m \geq 4 \qquad (2.53)$$

To make system controllable and observable, the number of wheels, that contact with the ground must controlled for satisfying $4 \leq m \leq n-2$.

This model given in Section (2.5) has to be modified in order to provide

1.) Enabling 3-D snake-like robot grasping starting from the any links nearest to the object.

2.) Showing that it is possible to drag the grasp object to a desired state during serpentine motion of snake-like robot.

for our lasso-type snake-like robot grasping. Our approach that builds and uses this modified model is developpped in Chapter 3.

CHAPTER 3

OUR LASSO-TYPE GRASPING APPROACH

3.1 Snake Kinematic Model Adaptation

When the snake grasps objects, it raises some parts of its body together with its head, and wraps around the object while undergoing serpentine motion. If we maintain the non-holonomic constraint which prevents the side-slip of the wheels active for all of the links, the curvilinear snake body motion which is necessary for the lasso-type wrapping becomes difficult if not impossible. So, some changes should be made to the kinematic and control models introduced in Chapter 2. Links under non-holonomic constraints move under the effect of absolute joint angles. However, during grasping they will have to move into the direction of side slip of the wheels violating the non-holonomic constraints. Such motion during grasping are illustrated in Figure (3.1).

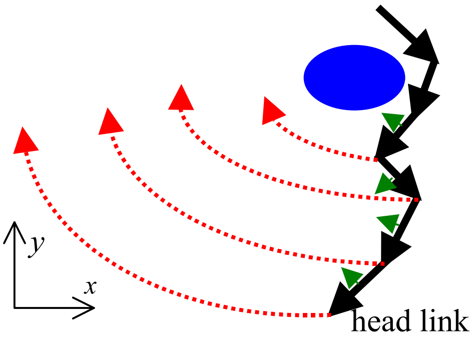

Figure (3.1) Grasping trajectories of the links

In the Figure (3.1), the black arrows indicate the direction of wheels with respect to non-holonomic constraint, the dotted arrows indicate the needed links trajectories for the curvilinear motion during grasping and the green (gray in “black and white” picture) arrows indicate the wheel direction with respect to the curved motion. Therefore, we need to modify the kinematic equation of the model as:

$$h_i^T \dot{P}_{fi} = \dot{x}_{fi} \sin(\phi_i) - \dot{y}_{fi} \cos(\phi_i) = 0$$

$$\frac{\dot{y}_{fi}}{\dot{x}_{fi}} = \tan(\phi_i) \qquad (3.1)$$

i. If $0 \le \phi_i < \pi/2$ and $\pi \le \phi_i < 3\pi/2$, $\tan(\phi_i) \ge 0$ and x_{fi} and y_{fi} increase or decrease during locomotion.

ii. If $\pi/2 \le \phi_i < \pi$ and $3\pi/2 \le \phi_i < 2\pi$, $\tan(\phi_i) \le 0$ therefore, while x_{fi} increases, y_{fi} decreases or vice versa during locomotion.

In Figure (3.1), for the head link if $0 \le \phi_i < \pi/2$, x_{f0} and y_{f0} decrease or increase with respect to the non-holonomic constraint. But from Figure (3.1), to begin grasping the object, the head link must increase in y_{f0} and decrease in x_{f0}. So, the non-holonomic constraint of head link should be removed when gasping is initiated. Since link 1 and 2 are similar to head link, the non-holonomic constraint of these links should also be removed.

For link 3 and 4, if $\pi/2 \le \phi_{3,4} < \pi$, $x_{f3,4}$ increases while $y_{f3,4}$ decreases with respect to non-holonomic constraint. But from Figure (3.1), to begin grasping the object, $x_{f3,4}$ must decrease while $y_{f3,4}$ must increase. Thus, here also the non-holonomic constraint of link 3,4 should be removed during gasping. Since removing the non-holonomic constraints of the links engaged in grasping the object affects the controllability of the system, the minimum number of links whose non-holonomic constraints are active should be determined to obtain a controllable system. This number (m) will be determined in Chapter 3.2. Thus we first have to find the links involved in the grasping and remove their non-holonomic constraints during their wrapping motion, beginning from the head.

For our computer simulation, 9 links from the head link to link 8 are determined as the links that can be involved in grasping and their non-holonomic constraints can be momentarily removed when the snake robot uses some of these links as the wrapping effector. Other links from the 9-th link to link 14 are determined as the links whose non-holonomic constraints are always active, and are equipped with wheels.

Moreover, the singular configuration avoidance for the system expressed in Equation (2.50) is achieved by means of an oscillating joint. Since the oscillating joint results from the difference of absolute angles of two consecutive links, we should investigate the effect of oscillating joints for the lasso-type grasping. Firstly, to investigate the effect of oscillating joints Equation (2.49) can be changed as follows:

$$\dot{\psi}_i = \begin{bmatrix} -\mathrm{A}_\psi & \mathrm{B}_\psi \end{bmatrix} \begin{bmatrix} \dot{w}_0 \\ u \end{bmatrix} \tag{3.2}$$

$$\dot{\psi}_i = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_i \\ \dot{\theta}_{i-1} \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} \mathrm{A}_\psi & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{w}_0 \\ \dot{\theta}_i \\ \dot{\theta}_{i-1} \end{bmatrix} = \mathrm{B}_\psi u \tag{3.4}$$

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_i \\ \dot{\theta}_{i-1} \end{bmatrix} = \begin{bmatrix} -\mathrm{A}_\psi & \mathrm{B}_\psi \end{bmatrix} \begin{bmatrix} w_0 \\ u \end{bmatrix} \tag{3.5}$$

The system expressed in Equation (3.2) is an observable system, while systems (3.4) and (3.5) are not observable. So, for these systems, $\dot{\phi}_i$ and $\dot{\phi}_{i-1}$ can not be uniquely determined. However the links which have the absolute angles $\dot{\phi}_{i-1}$ and $\dot{\phi}_i$ can effect the absolute angles and the motion direction of the other links whose non-holonomic constraints are removed. In this case, undetermined $\dot{\phi}_i$ and $\dot{\phi}_{i-1}$ corrupt the homogeneity of the grasping curve. As shown in Figure (3.3), while links at the end of the snake robot oscillate to avoid singular configuration, the grasping curve is corrupted by oscillating links and these links prevent the grasping of snake robot. This problem is not encountered in Figure (3.4) the case without oscillating joint. So far we can conclude that although the joint angle that oscillates to avoid the singular configuration can be well controlled by the above system, the absolute angles of consecutive links can not be controlled by the system (3.5). Because of the above result, singular configuration avoidance with the oscillating joint is not appropriate for grasping.

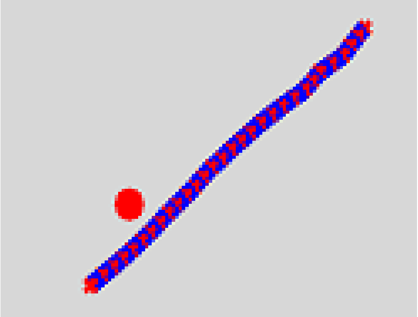

Figure (3.2) Initial configuration of snake robot

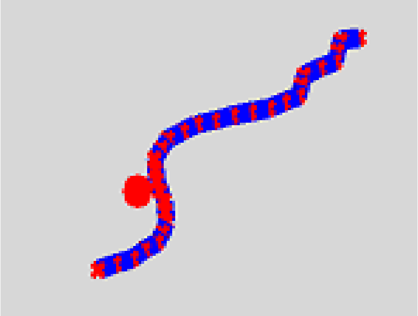

Figure (3.3) Grasping of snake robot of snake robot
with oscillating joint

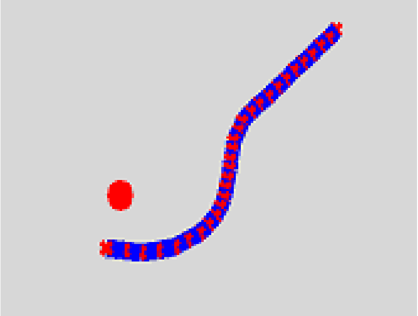

Figure (3.4) Grasping of snake robot without oscillating joint

Therefore, for the lasso-type grasping we have to further modify system (2.50) as:

$$\mathrm{A}'\dot{w}' = \mathrm{B}'u \qquad (3.6)$$

where

$$\dot{w}' = \begin{bmatrix}\dot{w}_0 & \dot{z}\end{bmatrix} \in \Re^{n+3}$$

$$u = \begin{bmatrix}\dot{\theta}_{y1} & \dots & \dot{\theta}_{y(n-1)} & \dot{\theta}_{z1} & \dots & \dot{\theta}_{z(n-1)}\end{bmatrix}^T \in \Re^{2(n-1)}$$

$$\mathrm{A}' = \begin{bmatrix}\mathrm{A}_0 & \mathrm{O}_{m\times(n-1)} \\ \mathrm{A}_z & \mathrm{I}_{(n-1)\times(n-1)}\end{bmatrix} \in \Re^{(m+n-1)\times(n+3)}$$

$$\mathrm{B}' = \begin{bmatrix}\mathrm{B}_0 \\ \mathrm{B}_z\end{bmatrix} \in \Re^{(m+n-1)\times 2(n-1)}$$

Now we are ready to adjust our control to the modified model we have expressed in Equation (3.6).

3.2 Control Strategy Changes under the Modified Model

In order to make the system (3.6) controllable and observable, if $m \le n-1$, the matrix B' in system (3.6) has full row rank except at singular configuration, and the system is controllable; and if $m+n-1 \ge n+3$, that is $m \ge 4$, the matrix A' in system (3.6) has full column rank, and the system is observable. If for system (3.6), the criterions of the controllablity and observablity are combined, we obtain the condition:

$$4 \le m \le n-1 \qquad (3.7)$$

For $m < n-1$, the system (3.6) has redundant degrees of freedom. A manipulator is said to have redundancy if it has more degrees of freedom than are necessary to perform a given task. If a given task is complex, adopting the divide and conquer concept, this task given to a manipulator can be broken down into several subtasks with priority order. Each subtask is then performed by using the degrees of freedom that remain after all the subtasks with higher priority have been implemented. So, our control approach for our snake-like robot uses also the subtask method by regarding a task to be done by our snake-like robot as the subtask with first priority and regarding the reason for using the redundancy as the subtask with second priority. We give the second subtask in the form of a cost function for describing the subtask. In some appropriate postures, the snake-robots can locomote using the constraining force of wheels against sideslip arising from actuating joints. On the contrary, there exist singular postures in which the snake-robots can not move in some direction. A singular posture arises when the snake's body is in a shape such as a straight line or a parabolic curve. In these postures the snake is helpless and can not move out of this shape. Essentially, the singularity represents an instability of the snake's motion. Thus, it is necessary to design a control scheme that will maintain stability and avoid these singular postures. Hence the locomotion ability of the robot largely depends on its posture and it is important to keep some suitable posture to control the locomotion. So, singular configuration avoidance is again an important problem for our robotic hyper redundant snake locomotion/lasso-type grasp system. In system (3.6), singular configuration avoidance has not been performed yet. Consequently, since for $m < n-1$, the system (3.6) has redundant degrees of freedom, the singular configuration avoidance here, can be the second subtask for system (3.6), thus we expanded the second subtask with a second input as

$$\mathrm{A}'\dot{w}' = \mathrm{B}'u \qquad u = u_1 + u_2 \qquad (3.8)$$

In order to make system (3.6) redundancy controllable, using concepts in [28], we must have $2(n-1) > n+3$, $2(n-1) > m+n-1$, full column rankness of matrix A', full row rankness of matrix B', existance of an input u_1 which accomplishes the main objective of the convergence of the vector w' to the desired state w'_d $(w' \to w'_d, \dot{w}' \to \dot{w}'_d)$ and an input $u = u_1 + u_2$ which accomplishes the increase (or decrease) of a cost function V, which is related to the second subtask and which does not disturb the main objective. If the criteria of redundancy controllable are integrated in system (3.6), the condition making system (3.6) redundancy controllable can be written as

$$4 \leq m < n-1 \tag{3.9}$$

In order to have a solution for system (3.6), we can write from [28]:

$$u = \mathrm{B}'^{+}\mathrm{A}'\{w_d' + \mathrm{K}(w_d' - w')\} + (\mathrm{I}_k - \mathrm{B}'^{+}\mathrm{B}')\alpha\eta \tag{3.10}$$

$$\eta = \nabla_\theta V(q) = \begin{bmatrix} \partial V/\partial\theta_{y1} & \dots & \partial V/\partial\theta_{y(n-1)} & \partial V/\partial\theta_{z1} & \dots & \partial V/\partial\theta_{z(n-1)} \end{bmatrix}^T$$

Where η is the gradient of the cost function V with respect to the vector θ related to the input vector u, B'^{+} is a pseudo-inverse matrix of B', and $\alpha \geq 0$, $\mathrm{K} > 0$. The first term of the right side of Equation (3.10) is the control input term to accomplish the main objective of the convergence of the state vector w' to the desired value w_d'. The second term accomplishes the increase of the cost function V.

3.3 Grasping Strategy

With these model changes of Sections 3.1 and 3.2, our snake robot can now perform serpentine motion with some of its links by using head position-orientation, changing the link rises, while grasping with the remaining links. But this can corrupt the homogeneity of the snake shape and prevent the controllability of the snake robot shape. So, we need to also generate a grasping strategy as

1) Orient the head link so that the shape of snake robot turn into the enwrapping curve on the object. Figure (3.5) shows the initial configuration and Figure (3.6) shows the curve formed from this initial configuration.

2) Position and orient the head link to approach the links, nearest to the object.This is shown in Figure (3.7).

3) Raise the links used for grasping and wrap the object from the head position to the grasping links. This 3-D enwrapment of the cylindrical object is shown in Figure (3.8).

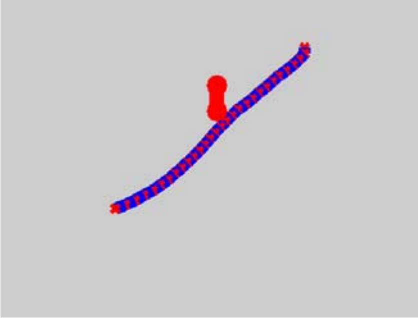

Figure (3.5) Initial configuration of snake robot

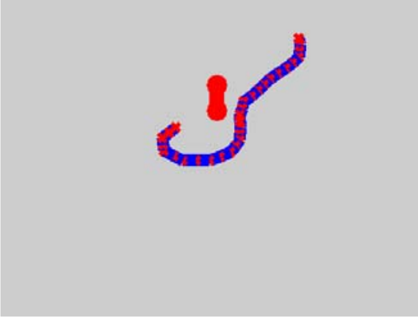

Figure (3.6) Formed curve

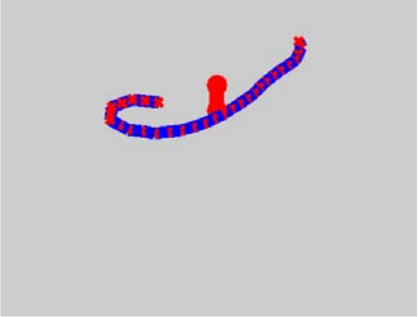

Figure (3.7) Approaching to the object

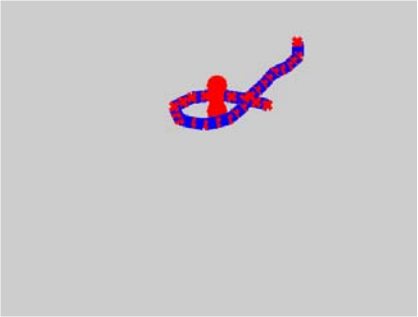

Figure (3.8) Wrapping the object

Figure (3.9) shows the lasso-type grasping of more complex prismatic object with our grasping strategy. The above steps of our grasping strategy are hereby followed.

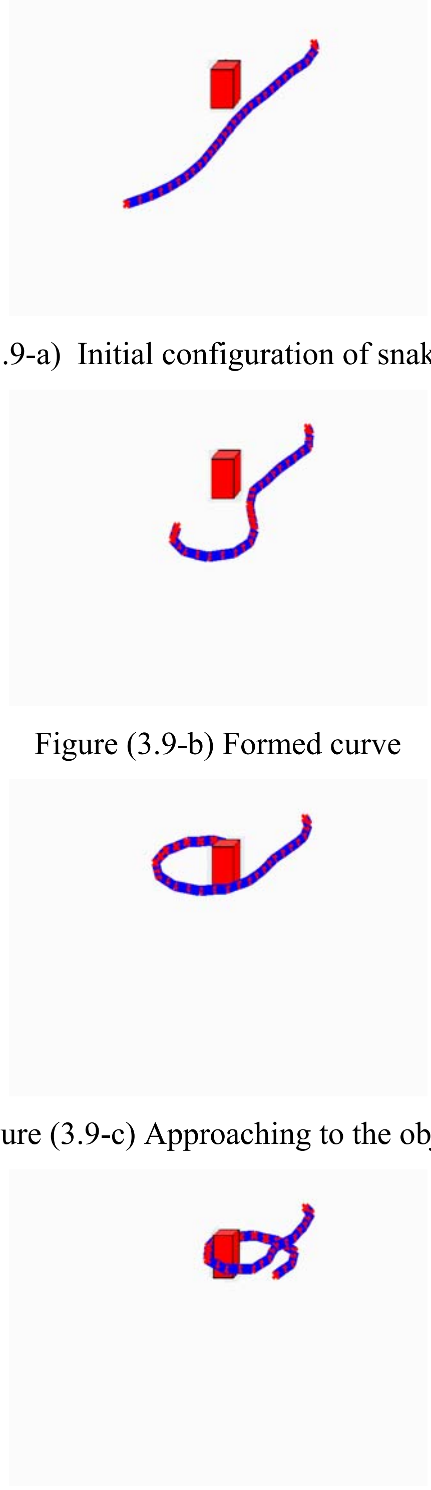

Figure (3.9-a) Initial configuration of snake-like robot

Figure (3.9-b) Formed curve

Figure (3.9-c) Approaching to the object

Figure (3.9-d) Wrapping the object

Figure (3.9) Lasso-type grasping of a prismatic object.

Since our snake robot has the pitch motion for every link, we can ensure that the links do not run into each other as they wrap around the object. Multiple enwrapments for a lasso-type power grasp is then possible for our 15-link snake robot. This can be seen in Figure (3.10). Figure (3.10-a) shows the enwrapment of the snake-like robot. Figure (3.10-b,c) shows the snake sliding its body around the object in a forward direction while enwrapping it in a lasso-type. Figure (3.10-d) demonstrates the power grasp of the snake robot by a lasso-type enwrapment. In figure (3.10-e), a second enwrapment of the object is executed to secure further the power grasp.

Figure (3.10-a) Grasping of the object

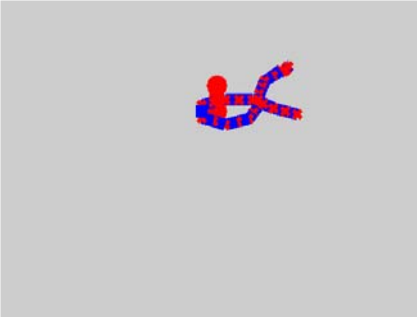

Figure (3.10-b) Snake body sliding on the enwrapped object

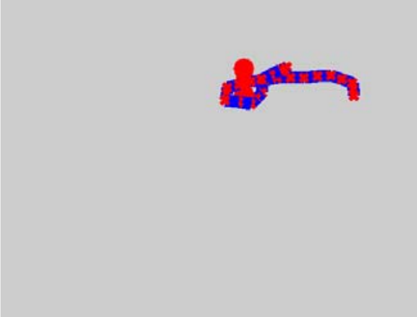

Figure (3.10-c) End of snake body sliding

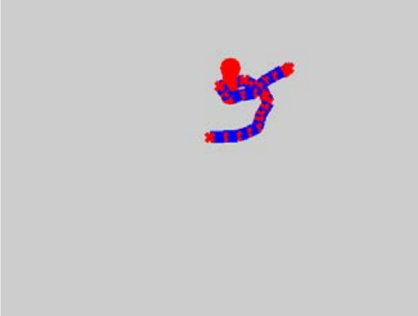

Figure (3.10-d) Getting in position for beginning second enwrapment

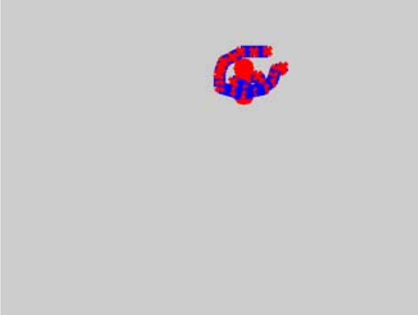

Figure (3.10-e) Power grasp of the snake robot
with double enwrapment of the object

Figure (3.10) Lasso-type grasping with power grasp and serpentine motion

3.4 Model of Lasso-Type Snake Robot Grasping and Dragging

Now we have to further adapt the system given in Equation (3.6) to a lasso-type grasping where contacts change position along links as the grasp tightens yielding also the dragging control. Grasping with wheeled and legged train of vehicles or multi-fingered robot hands can be achieved by controlling the contact point changes on the grasped object. Similarly, for our lasso type grasping using the body links of the snake-robot, the link contact point changes must be determined and controlled to grasp and drag the object to a desired positon during serpentine motion of the snake robot. Moreover, according to the kinematic model of grasping in Equation (3.6), the head link motion of the snake robot and thus the entire snake robot motion are generated by the last wheeled links which have non-holonomic constraint. Therefore, the contact points must be generated as a result of the these last wheeled links having non-holonomic constraints so that the grasped object can be moved as the result of entire snake robot motion. For this purpose, the relative position of the contact points $x_{fi} - x_0$ and $y_{fi} - y_0$ should be controlled with respect to the head link motion and thus with respect to the entire snake-robot motion. Here x_{fi} and y_{fi} are the respective x and y-components of ith contact point, x_0 and y_0 are the respective x and y-components of the head position. For simplicity, we assume that the middle point of the contact links can undergo point contact. For any contact on links, the system equations can be derived for controling the $x_{fi} - x_0$ and $y_{fi} - y_0$ of contact points as follows.

$$h_x^T \dot{P}_{fi} = \dot{x}_{fi} \tag{3.11}$$

$$h_x^T (\dot{P}_0 + \sum_{j=0}^{i-1} (w_j \times l_j) + w_i \times l_{fi}) = 0$$

$$\dot{x}_{fi} - \dot{x}_0 + h_x^T \left\{ (\sum_{k=0}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{y0} e_{y0} + \dot{\theta}_{z0} e_{z0}) \right\} = -h_x{}^T \left\{ \sum_{j=1}^{i} \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{yj} e_{yj} + \dot{\theta}_{zj} e_{zj}) \right\} \right\}$$

$$h_y^T \dot{P}_{fi} = \dot{y}_{fi} \tag{3.12}$$

$$h_y^T (\dot{P}_0 + \sum_{j=0}^{i-1} (w_j \times l_j) + w_i \times l_{fi}) = 0$$

$$\dot{y}_{fi} - \dot{y}_0 + h_y^T \left\{ (\sum_{k=0}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{y0} e_{y0} + \dot{\theta}_{z0} e_{z0}) \right\} = -h_y{}^T \left\{ \sum_{j=1}^{i} \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times (\dot{\theta}_{yj} e_{yj} + \dot{\theta}_{zj} e_{zj}) \right\} \right\}$$

where $h_x = [1 \quad 0 \quad 0]^T$, $h_y = [0 \quad 1 \quad 0]^T$. So, the system equations for controlling the x_{fi} and y_{fi} of contact points can be described by

$$\mathrm{A}_x \dot{w}_0 + \dot{x} = \mathrm{B}_x u \tag{3.13}$$

where $\dot{x} = [\dot{x}_{fi} - \dot{x}_0 \quad \dot{x}_{f(i-1)} - \dot{x}_0 \quad \dot{x}_{f(i-2)} - \dot{x}_0 ......\dot{x}_{f(i-r)} - \dot{x}_0]^T \in \Re^{r\times 1}$ and

$$\mathrm{A}_x = \{a_{xyij}\} \in \Re^{r\times 4}$$

$$\mathrm{B}_x = [\mathrm{B}_{xy} \quad | \quad \mathrm{B}_{xz}] \in \Re^{r\times 2(n-1)}$$

$$\mathrm{B}_{xy} = \{b_{xyij}\} \in \Re^{r\times (n-1)}$$

$$\mathrm{B}_{xz} = \{b_{xzij}\} \in \Re^{r\times(n-1)}$$

Therein,

$$a_{xyij} = \begin{cases} 0 & j = 1,2 \\ -h_x^T\left\{(\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{y0}\right\} & j = 3 \\ -h_x^T\left\{(\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{z0}\right\} & j = 4 \end{cases}$$

$$b_{xyij} = \begin{cases} -h_x^T\left\{(\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{yj}\right\} & j \le i \\ 0 & j > i \end{cases}$$

$$b_{xzij} = \begin{cases} -h_x^T\left\{(\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{zj}\right\} & j \le i \\ 0 & j > i \end{cases}$$

$$\mathrm{A}_y \dot{w}_0 + \dot{y} = \mathrm{B}_y u \tag{3.14}$$

where $\dot{y} = \left[\dot{y}_{fi} - \dot{y}_0 \quad \dot{y}_{f(i-1)} - \dot{y}_0 \quad \dot{y}_{f(i-2)} - \dot{y}_0 ......\dot{y}_{f(i-r)} - \dot{y}_0\right]^T \in \Re^{r\times1}$ and

$$\mathrm{A}_y = \{a_{yyij}\} \in \Re^{r\times4}$$

$$\mathrm{B}_y = \left[\mathrm{B}_{yy} \quad | \quad \mathrm{B}_{yz}\right] \in \Re^{r\times2(n-1)}$$

$$\mathrm{B}_{yy} = \{b_{yyij}\} \in \Re^{r\times(n-1)}$$

$$\mathrm{B}_{yz} = \{b_{yzij}\} \in \Re^{r\times(n-1)}$$

Therein,

$$a_{yyij} = \begin{cases} 0 & j = 1,2 \\ -h_y^T\left\{(\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{y0}\right\} & j = 3 \\ -h_y^T\left\{(\sum_{k=0}^{i-1} l_k + l_{fi}) \times e_{z0}\right\} & j = 4 \end{cases}$$

$$b_{yyij} = \begin{cases} -h_y^T \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{yj} \right\} & j \le i \\ 0 & j > i \end{cases}$$

$$b_{yzij} = \begin{cases} -h_y^T \left\{ (\sum_{k=j}^{i-1} l_k + l_{fi}) \times e_{zj} \right\} & j \le i \\ 0 & j > i \end{cases}$$

The system model given in Equation (3.6) can be further modified by adding the matrices $\mathrm{A}_x, \mathrm{B}_x, \mathrm{A}_y, \mathrm{B}_y$ to matrices A', B' given in Equation (3.6) and by adding the vectors x, y to state vector w' given in Equation (3.6). So, the further modified model can be written as:

$$\mathrm{A}''\dot{w}'' = \mathrm{B}''u \qquad (3.15)$$

$$\dot{w}'' = \begin{bmatrix} \dot{w} & \dot{z} & \dot{x} & \dot{y} \end{bmatrix}^T \in \Re^{n+2r+3}$$

$$u = \begin{bmatrix} \dot{\theta}_{y1} & \dots & \dot{\theta}_{y(n-1)} & \dot{\theta}_{z1} & \dots & \dot{\theta}_{z(n-1)} \end{bmatrix}^T \in \Re^{2(n-1)}$$

$$\mathrm{A}'' = \begin{bmatrix} \mathrm{A}_0 & \mathrm{O}_{m \times n-1} & \mathrm{O}_{m \times r} & \mathrm{O}_{m \times r} \\ \mathrm{A}_z & \mathrm{I}_{n-1 \times n-1} & \mathrm{O}_{n-1 \times r} & \mathrm{O}_{n-1 \times r} \\ \mathrm{A}_x & \mathrm{O}_{r \times n-1} & \mathrm{I}_{r \times r} & \mathrm{O}_{r \times r} \\ \mathrm{A}_y & \mathrm{O}_{r \times n-1} & \mathrm{O}_{r \times r} & \mathrm{I}_{r \times r} \end{bmatrix} \in \Re^{m+n+2r-1 \times n+2r+3}$$

$$\mathrm{B}'' = \begin{bmatrix} \mathrm{B}_0 \\ \mathrm{B}_z \\ \mathrm{B}_x \\ \mathrm{B}_y \end{bmatrix} \in \Re^{(m+n+2r-1) \times 2(n-1)}$$

where I is the unit matrix and O is the zero matrix or vector. If $m + 2r \le n - 1$, the system given in Equation (3.15) is the controllable and has full row rank except at singular configuration, and this system has redundant degrees of freedom for which inequality $m + 2r < n - 1$. Furthermore since it is necessary to find inverse solution of this system for controling the snake robot motion, the system given in Equation (3.15) must be invertiable as:

$$\dot{w}'' = \mathrm{A}''^{+}\mathrm{B}''u \qquad (3.16)$$

where $\mathrm{A}''^{+} = (\mathrm{A}''^{T}\mathrm{A}'')^{-1}\mathrm{A}''^{T}$, If $m+n+2r-1 \geq n+2r+3 \Rightarrow m \geq 4$, the matrix A'' has full column rank and the Equation (3.15) has an inverse solution.

3.5 Lasso-Type Grasping: Constructing the Grasping Curve

The model derived in Equation (3.15) enables us to control the contact points as the function of entire snake-like robot motion which can be determined by the position and orientation of the head link. Therefore, the appropriate desired states w''_d which construct the grasping curve should be defined to control the generation of grasping curve through $u = \mathrm{B}''^{+}\mathrm{A}''\{\dot{w}''_d + \mathrm{K}(w''_d - w'')\}$ just as Equation (2.51), by the new model matrices A'' and B'' of Equation(3.15).

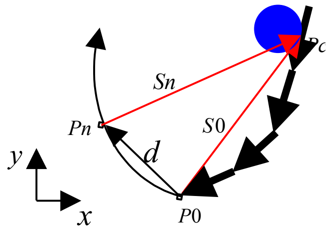

Figure (3.11) Two state of snake-robot during lasso-type grasping

We give the procedure for determining the desired state w''_d which constructs the enwrapment grasping curve. Figure (3.11) shows the two state of snake-robot during lasso-type grasping, where $P0 = [x_0 \quad y_0 \quad z_0]$ is present position of the head link, $Pn = [x_0 + a \quad y_0 + b \quad z_0]$ is the any next position of head link, (the head is not rising) $Pc = [x_{fi} \quad y_{fi} \quad z_{fi}]$ is the first contact point of snake robot, $d = [a \quad b \quad 0]$ is the difference vector between the present position of head link ($P0$) and the next position of head link (Pn), $S0 = [x_{fi} - x_0 \quad y_{fi} - y_0 \quad z_{fi} - z_0]$ is the difference vector between Pc and $P0$. $P0$ and $S0$ are determined from the present state of snake-robot, found from Equation (3.15) as:

$$w'' = \lfloor x_0 \quad y_0 \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad z_{fi} - z_0 \quad \ldots \quad z_{f(n-1)} - z_0 \quad x_{fi} - x_0 \quad y_{fi} - y_0 \rfloor$$

or

$$w'' = \lfloor P0(1) \quad P0(2) \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad S0(3) \quad \ldots \quad z_{f(n-1)} - z_0 \quad S0(1) \quad S0(2) \rfloor$$

To determine the desired state w''_d which construct the grasping curve, we should find Sn which is equal to ($S0 - d$) or ($Pc - Pn$). So, the desired state which construct the grasping curve is as follows.

$$w''_d = \lfloor x_0 + a \quad y_0 + b \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad z_{fi} - z_0 \quad \ldots \quad z_{f(n-1)} - z_0 \quad x_{fi} - x_0 - a \quad y_{fi} - y_0 - b \rfloor$$

or

$$w''_d = \lfloor Pn(1) \quad Pn(2) \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad Sn(3) \quad \ldots \quad z_{f(n-1)} - z_0 \quad Sn(1) \quad Sn(2) \rfloor$$

While the snake-robot tracks this desired state, it constructs the grasping curve, until the second contact. After the second contact and for the other contacts from 3 to r, the desired states which construct the grasping curve can be determined similarly as above. But the limiting number of contact points ($m + 2r < n - 1$) found in Equation (3.15) should not be exceeded to control the system. Figure (3.12) shows a demonstrative example of our lasso-type grasping scheme where contact by contact the desired state w''_d is found as just described and control system of Equation (3.15) runs to enwrap the object in a control manner. Figure (3.12-a) to Figure (3.12-e) show the lasso-type enwrapment periods with one contact point which is on the seventh link. Figure (3.12-f) to Figure (3.12-h) show lasso-type enwrapment with two contact points, first contact point being on the seventh link and the second on the sixth link. After the second contact point, the grasping curve is constructed for two contact point by means of our lasso-type grasping procedure given above. Figure (3.12-i) shows the lasso type grasping with three contact points. Third contact point is on the fifth link. With this contact point, the lasso-type grasping is concluded.

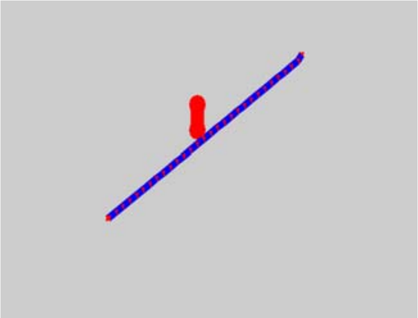

Figure (3.12-a) Initial state of snake-robot

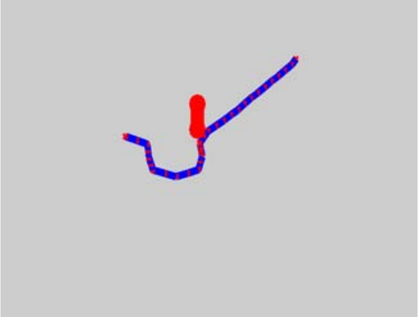

Figure (3.12-b) Snake robot stars the grasping with one contact point

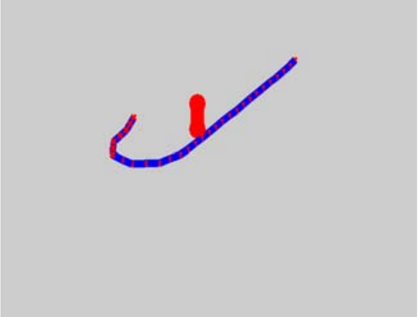

Figure (3.12-c) Snake robot continues the grasping with one contact point

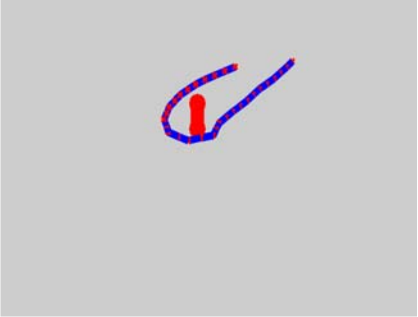

Figure (3.12-d) Snake robot continues the grasping with one contact point

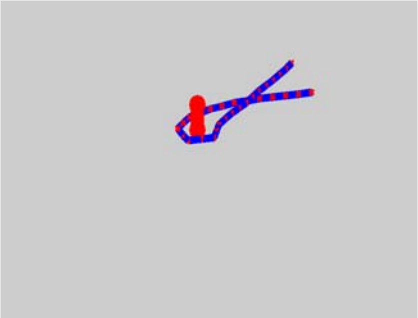

Figure (3.12-e) Snake robot continues the grasping with one contact point

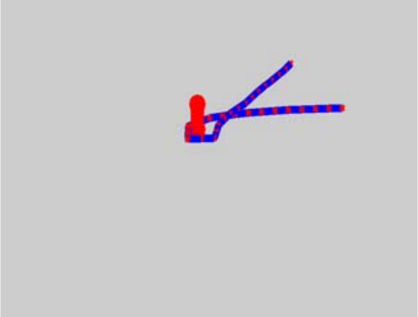

Figure (3.12-f) Snake robot continues the grasping with two contact point

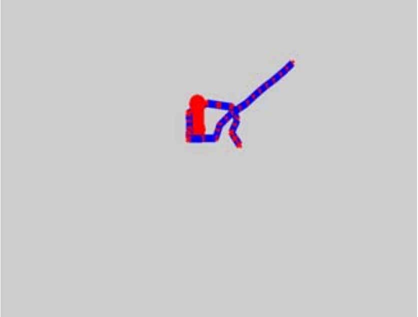

Figure (3.12-g) Snake robot continues the grasping with two contact point

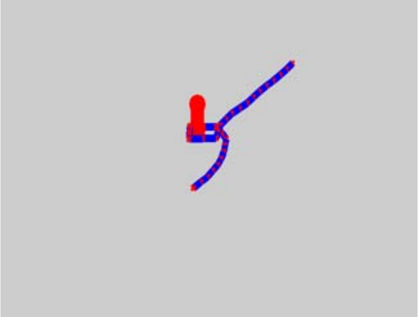

Figure (3.12-h) Snake robot continues the grasping with two contact point

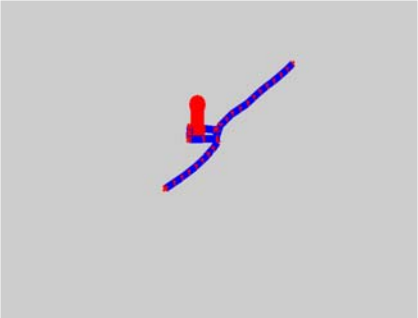

Figure (3.12-f) Snake robot concludes the grasping with third contact point

Figure (3.12) A Demonstrative Example of the Lasso-Type Grasping Procedure

Since the lasso-type grasping procedure depends on the determination of desired states w_d'' running the control system of Equation (3.15), different desired states will yield different grasping curve for the lasso-type enwrapment. Figure (3.13) shows another demonstrative example of the lasso-type grasping. Figure (3.13-a) to Figure (3.13-e) again show the lasso-type grasping with one contact point which is on the seventh link. Figure (3.13-f) to Figure (3.13-h) show the lasso-type grasping with two contact points. Second contact point is on the sixth link. Figure (3.13-i) shows the lasso-type grasping with three contact points. Third contact point is on the fifth link. Having such a possibility to autonomously construct grasping curve with different manner is important for grasping in a complex environment where the snake may be generating grasping curves that fit to passages among obstacles. As shown in Figure (3.13) different from Figure (3.12), snake-like robot has the ability of constructing the grasping curve in different manner without changing the contact point positions after their first contact on the object. By using this, snake-robot can grasp the object regardless of the head position and this provides that it is not necessary to construct the curvilinear grasping curve. Although initial states are same in Figure (3.12) and Figure (3.13), while the grasping curve is constructed by changing the head position in leftward and upward directions in Figure (3.12-b), the grasping curve can be constructed by changing the head position in rightward and upward directions in Figure (3.13-b) too. From Figure (3.12-b) to Figure (3.12-c), ssnake-robot continues to grasp the object and construct the grasping curve by changing the head position in rightward and upward directions. Similarly, from Figure (3.13-b) to Figure (3.13-c), snake-robot can continue to grasp the object and construct the grasping curve by changing the head position in leftward and upward directions too. This can be easily tracked in other figures from Figure (3.12,13-d) to Figure (3.12,13-i) and as shown in these figures, this ability result from that by using our lasso-type grasping scheme, snake-robot can control the contact points after their first contact on the grasped object and keep the contact point positions in the same position on the object during the lasso-type grasping and this provides that while the contact point positions remain same, the front part of contact points can construct the grasping curve according to head link motion.

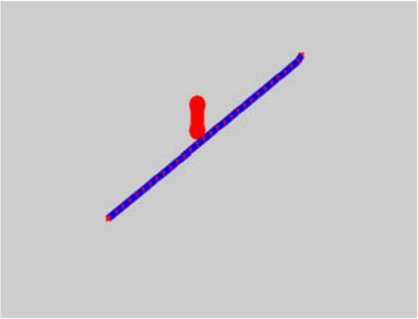

Figure (3.13-a) Initial state of snake robot

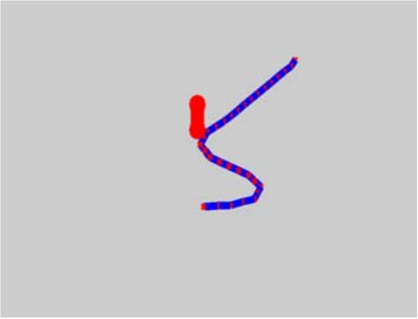

Figure (3.13-b) Snake robot starts the grasping with one contact point

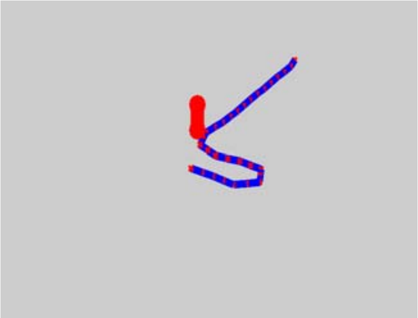

Figure (3.13-c) Snake robot continues the grasping with one contact point

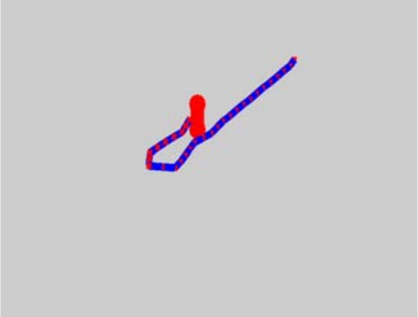

Figure (3.13-d) Snake robot continues the grasping with one contact point

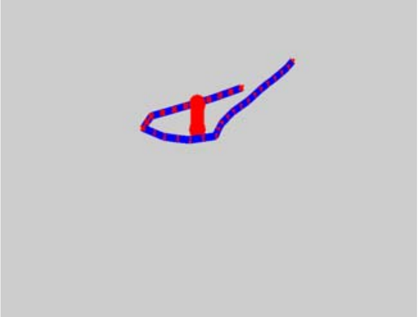

Figure (3.13-e) Snake robot continues the grasping with one contact point

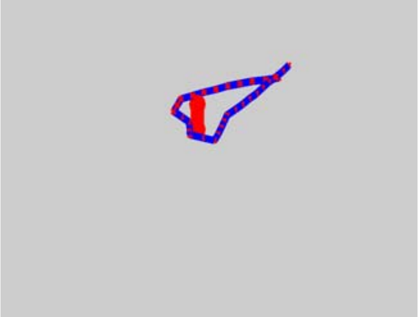

Figure (3.13-f) Snake robot continues the grasping with two contact point

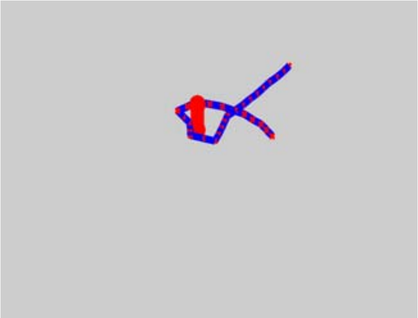

Figure (3.13-g) Snake robot continues the grasping with two contact point

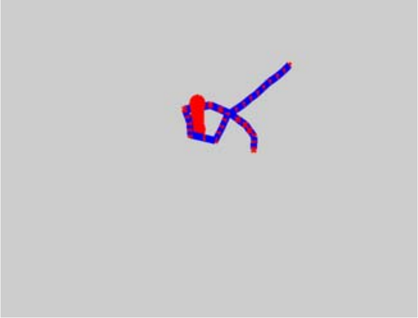

Figure (3.13-h) Snake robot continues the grasping with two contact point

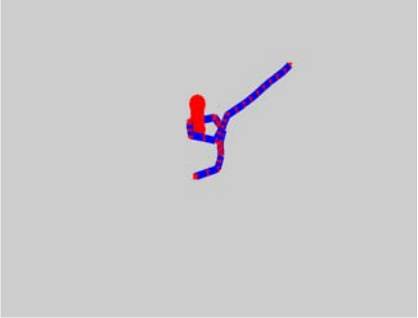

Figure (3.13-a) Snake robot concludes the grasping with third contact point

Figure (3.13) A Demonstrative Example of the Lasso-Type Grasping Procedure

3.6 Dragging the Grasped Object: Modelling the Motion

Since the snake-like robot can now control the position of contact-points as a function of the entire snake-robot motion, it can drag the grasped object to a desired position. For that, it is necessary to move the contact points harmoniously in dragging the grasped object to a desired location. For this purpose, we generate the desired state of the snake-like robot which drags the grasped object.

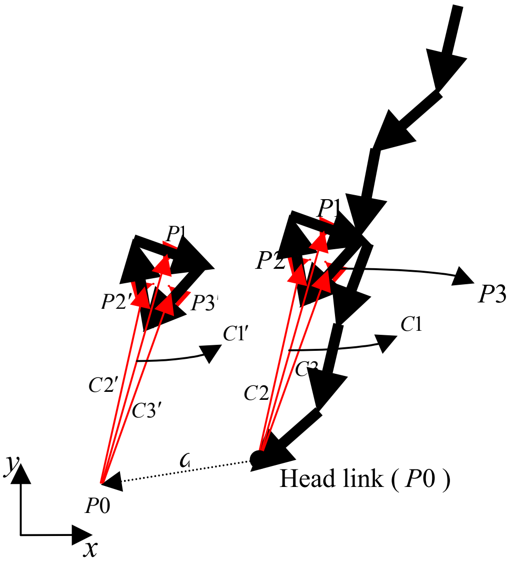

Figure (3.14) Two state of lasso-type grasping during serpentine motion.

Figure (3.14) shows the two states of lasso-type grasping during serpentine motion. This figure is considered for three contact points but, this number of contact points can be increased. In Figure (3.14), $P0 = \left[x_0 \quad y_0 \quad z_0\right]$ is the position of head link, $P1 = \lfloor x_{fi} \quad y_{fi} \quad z_{fi} \rfloor$, $P2 = \lfloor x_{f(i+1)} \quad y_{f(i+1)} \quad z_{f(i+1)} \rfloor$ and $P3 = \lfloor x_{f(i+2)} \quad y_{f(i+2)} \quad z_{f(i+2)} \rfloor$ are the contact point positions, $C1 = P1 - P0$, $C2 = P2 - P0$ and $C3 = P3 - P0$ are the difference vectors between contact points and head position. With the vectors $C1, C2, C3$ and the head position $P0$, we can determine the initial state of snake-robot as follows;

$$w'' = \lfloor x_0 \quad y_0 \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad z_{fi} - z_0 \quad z_{f(i+1)} - z_0 \quad z_{f(i+2)} - z_0$$

$$\ldots \quad x_{fi} - x_0 \quad x_{f(i+1)} - x_0 \quad x_{f(i+2)} - x_0 \quad y_{fi} - y_0 \quad y_{f(i+1)} - y_0 \quad y_{f(i+2)} - y_0 \rfloor$$

or

$$w'' = \lfloor P0(1) \quad P0(2) \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad C1(3) \quad C2(3) \quad C3(3)$$

$$\ldots \quad z_{f(n-1)} - z_0 \quad C1(1) \quad C2(1) \quad C3(1) \quad C1(2) \quad C2(2) \quad C3(2) \rfloor$$

While the snake-robot moves, appropriate contact point velocities should be calculated to drag the grasped object to a desired position without corruption of grasping. In Figure (3.14), $P0' = \left[x_0 + a \quad y_0 + b \quad z_0\right]$ is the new position of the head link, $d = \left[a \quad b \quad 0\right]$ is the difference vector between the present position of head link ($P0$) and the next position of head link ($P0'$), $P1', P2', P3'$ are the new position of contact points, $C1' = P1' - P0'$, $C2' = P2' - P0'$ and $C3' = P3' - P0'$ are the difference vectors between new contact points and new head position. Now, we can determine the desired state of the snake-robot which can drag the grasped object to a desired position as follows;

$$\begin{aligned} w''_d = \lfloor & x_0 + a \quad y_0 + b \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad z_{fi} - z_0 \quad z_{f(i+1)} - z_0 \quad z_{f(i+2)} - z_0 \\ & \ldots \quad x_{fi} - x_0 \quad x_{f(i+1)} - x_0 \quad x_{f(i+2)} - x_0 \quad y_{fi} - y_0 \quad y_{f(i+1)} - y_0 \quad y_{f(i+2)} - y_0 \rfloor \end{aligned}$$

or

$$\begin{aligned} w''_d = \lfloor & P0'(1) \quad P0'(2) \quad \theta_{y0} \quad \theta_{z0} \quad z_{f1} - z_0 \quad \ldots \quad C1(3) \quad C2(3) \quad C3(3) \\ & \ldots \quad z_{f(n-1)} - z_0 \quad C1(1) \quad C2(1) \quad C3(1) \quad C1(2) \quad C2(2) \quad C3(2) \rfloor \end{aligned}$$

As shown in the desired state, only the head position of the snake-robot changes, the other parameters remain the same as in the initial state. With this desired state, snake-robot starts the motion and while the head link reaches the desired state, the contact points can harmoniously track the head link trajectory as a result of the serpentine motion of the links having non-holonomic constraint. Consequently, contact points $P1, P2, P3$ reach the new state $P1' = P1 + d$, $P2' = P2 + d$, $P3' = P3 + d$ respectively while the difference vector $C1, C2, C3$ remain the same that is $C1' = C1, C2' = C2, C3' = C3$. With the scheme given above, the different motion patterns which can drag the grasped object to a desired position without the corruption of grasping can be generated as a result of serpentine motion of the snake-robot. Figure (3.15) shows the different motion patterns of the snake-robot that has grasped an object. Figure (3.15-a) shows the initial state of the snake robot. Figure (3.15-b,c,d) shows the dragging of the grasped object to the different directions starting from the same initial state shown in Figure (3.15-a). Figure (15-b) shows the dragging of grasped object in leftward and downward directions. In this motion, while head orientation remains the same as initial state shown in Figure (15-a), head position changes in leftward and downward directions. So the grasped object can be dragged in leftward and downward directions. Figure (15-c) shows the dragging of grasped object in upwards direction. In this motion, while head orientation and position change, the grasped object can be dragged to upward direction starting from the initial state shown in Figure (15-a). Figure (15-d) shows the dragging of grasped object in leftward and downward directions. In this motion, while head orientation changes, head position remains the same as initial state shown in Figure (15-a). So, the orientation of head pulls the object to itself and the grasping links drag the object closer to snake body as a snug fitting. This is achieved by controlling the relative position changes of contact points.

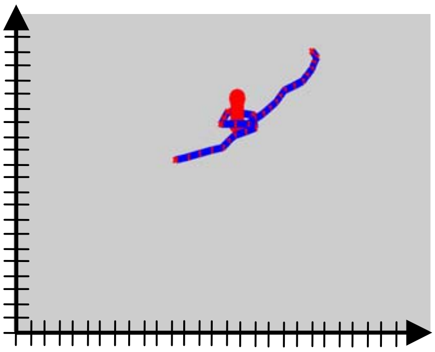

Figure (3.15-a) Initial state of snake-like robot

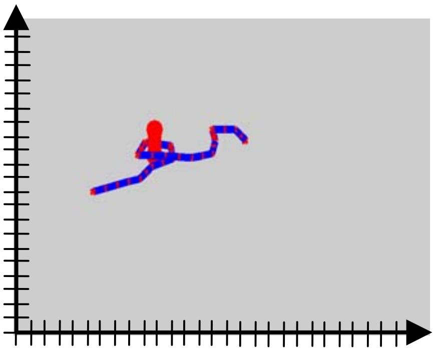

Figure (3.15-b) Dragging the grasped object in leftward and downward direction

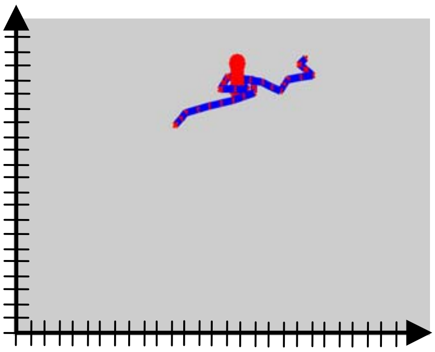

Figure (3.15-c) Dragging the grasped object in upward direction

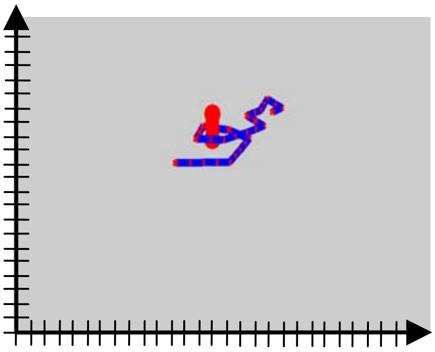

Figure (3.15-d) Snake-like robot drags the grasped object to itself as a snug fitting

Figure (3.15) Demonstrative examples of Dragging the Grasped Object

3.7 Generating the Control Strategy

During and after the lasso-type grasping we observe that some joint inputs of snake-robot may become excessively large trying to maintain the desired trajectory for example when the snake-robot becomes close to some undesired configuration, the system that drags the object while moving becomes the unstable with the control output corrupted. Figure (3.16) shows the state in which the snake robot generates a large input to maintain the desired trajectory during dragging the grasped object and the snake becomes a knot.

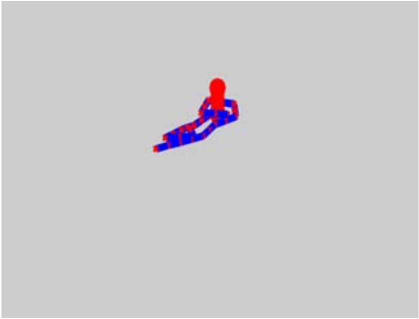

Figure (3.16) Corruption of serpentine locomotion control with excessively large inputs

To overcome this problem of destabilization, we propose an adaptively changing feedback gain to eliminate the excessively large inputs which prevent the serpentine locomotion control in the solution of Equation (3.15) which is:

$$u = \mathrm{B}''^{+}\mathrm{A}''\{\dot{w}_d'' + \mathrm{K}(w_d'' - w'')\} \qquad (3.17)$$

For the matrix $\mathrm{B}''^{+}\mathrm{A}''$, $\det\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^{T}\right)$ is a quantitative measure of loss manipulability of the system and of the singular configuration. In fact, the case in which some joints of snake-robot generate excessively large input is the state which is close to a singular configuration. We therefore have to handle such states in order to prevent excessively large inputs corrupting the serpentine locomotion control. The adaptation law of the feedback gain is defined as a logarithmic function of quantitative measures of manupulability.

$$\mathrm{K} = \mathrm{K}_f \exp\left(-\log \left| \frac{\det^p\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)}{\det^f\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)} \right| \right) \tag{3.18}$$

$$\mathrm{K}_f = \exp\left(-\log \left| \frac{\det^p\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)}{\det^f\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)} \right| \right)_f \tag{3.19}$$

where $\det^p\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)$ is the present value of $\det\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)$, $\det^f\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)$ is its former value, K_f is the former value of the feedback gain K. As seen in Equation (3.18), the feedback gain K tracks the logarithmic changes in $\det\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)$ such that when the logarithmic change in $\det\left((\mathrm{B}''^{+}\mathrm{A}'')(\mathrm{B}''^{+}\mathrm{A}'')^T\right)$ increases, the feedback gain K decreases exponentially or viceversa. Furthermore, the normalization with K_f is needed, because the exponential increase or decrease in K depends on its previous values. Having large K values in consecutive steps helps preserve values of K around previous ones so, around large ones. Similar argument would hold for small consecutive values of K. Therefore big changes in gain and inputs are prevented.

Figure (3.17-a) shows the enwrapment motion of the snake robot with adaptively changed feedback gain. Figure (3.17-b) shows the lasso-type grasping during dragging the grasped object with adaptively changed feedback gain. As shown in these figures, with the adaptively changed feedback gain, the serpentine locomotion control can not be corrupted by excessively large inputs.

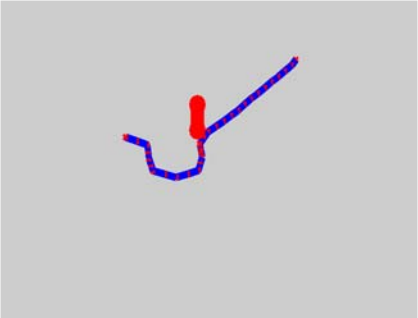

Figure (3.17-a) Lasso-type grasping with adaptively changing feedback gain

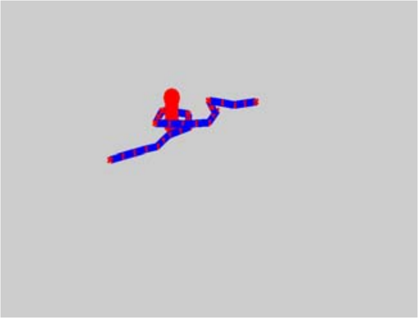

Figure (3.17-b) Dragging of grasped object with adaptively changing feedback gain

Figure (3.17) Serpentine motion during grasping with adaptively changing feedback gain

3.8 Stability of Lasso-Type Grasping

Force closed grasps are defined as grasps in which any disturbance force can be counteracted by a suitable linear combination of the applied contact forces. Force-closed grasps have the important property of being stable. Adopting the concept in [35] for our case, the contacts in the lasso-type grasping can be modelled by stiffness matrices. With this way, the lasso-type grasping stability can be analyzed and we can derive the expression for changes in contact forces as a function of the relative motion between the contacts and the grasped object.

3.8.1 Derivation of the Contact Siffness Matrix of the Snake-Robot

We first assume that the contact link has the reference frame o_A at the contact point. With respect to o_A, x_A and y_A are aligned with the principal axes of contact link and z_A with the outwardly pointing normal. We further assume that the location of contact points on the object do not change while contacts apply forces along the contact points and all contacts are considered as point contact with firiction. With these assumptions if $\lfloor \Delta x_A \quad \Delta y_A \quad \Delta z_A \quad \Delta\theta_{xA} \quad \Delta\theta_{yA} \quad \Delta\theta_{zA} \rfloor$ is the twist in the reference frame o_A, the changes of the contact forces and moments in the reference frame o_A are given by:

$$\Delta F_z = -k_n \Delta z_A \tag{3.20}$$

$$\begin{bmatrix} \Delta F_x \\ \Delta F_y \end{bmatrix} = -k_t \begin{bmatrix} \Delta x_A \\ \Delta y_A \end{bmatrix} \tag{3.21}$$

$$\Delta M_x \approx -F_{ty_0} \Delta z_A \tag{3.22}$$

$$\Delta M_y \approx F_{tx_0} \Delta z_A \tag{3.23}$$

$$\Delta M_x \approx -F_{ty_0} \Delta x_A - F_{tx_0} \Delta y_A \tag{3.24}$$

where $F_{t_0} = \lfloor F_{tx_0} \quad F_{ty_0} \rfloor^T$ are the tangential contact forces arising due to friction. The tangential force lies in the $x_A - y_A$ plane, and in general has components in both x_A and y_A directions. k_n and k_t are the stiffness coefficients of contact link. At each contact, the above equations can be combined into a single equation in the local contact reference frame ${}^i o_A$,

$${}^i\Delta F = -{}^i\kappa_c \, {}^i\Delta X_A \tag{3.25}$$

The leading subscript denotes contact i such that ${}^i\kappa_c$ is the stiffness matrix κ_c for the i-th contact and ${}^i o_A$ is the fixed frame at the ith contact, ${}^i\Delta F$ is the change in forces and moments in ith contact, ${}^i\Delta X_A$ is the rigid body motion measured in ${}^i o_A$.

$$
{}^{i}\kappa_{c} = \begin{bmatrix} k_{t} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{t} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{n} & 0 & 0 & 0 \\ 0 & 0 & F_{ty_0} & 0 & 0 & 0 \\ 0 & 0 & -F_{tx_0} & 0 & 0 & 0 \\ -F_{ty_0} & F_{tx_0} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.26}
$$

Since ${}^{i}\kappa_{c}$ does not have full row rank, ${}^{i}\Delta X_{A}$ is spanned by only the linearly independent row vectors of matrix ${}^{i}\kappa_{c}$ in the pseudo-inverse solution of Equation (3.25). As shown in the Equation (3.26), the columns of ${}^{i}\kappa_{c}$ from 4th to 6th are zero column vector and ${}^{i}\Delta X_{A}$ is spanned by three of linearly independent row vector of ${}^{i}\kappa_{c}$. In these row vectors, the last 3 components representing the orientation of contact points are zero. So, the contact forces and moments are generated by only contact position and not by contact orientation. This is due to our assumptions that the contact points on the object do not change while the contact forces are applied and that all contacts are frictional point contacts.

3.8.2 The Intrinsic Stiffness Matrix for Lasso-Type Grasping

Above equations are for the changes in forces and moments at a single contact point in response to an arbitrary motion. To find the all contributions of contacts, it is necessary to combine the all equations of contact like above for a single equation. The single equation for all system is given by:

$$
\Delta F = -\kappa_{o} \Delta x_{o} \tag{3.27}
$$

$$
\Delta F = \sum_{i=1}^{N} {}^{i}\Delta F_{o} + \Delta \overline{F} \tag{3.28}
$$

$$
\kappa_{o} = \sum_{i=1}^{N} {}^{i}T^{T} \; {}^{i}\kappa_{c} \; {}^{i}T + T_{cg}^{T} \kappa_{cg} T_{cg} \tag{3.29}
$$

where κ_{o} is the intrinsic stiffness matrix, reference to world frame O, Δx_{o} is the object twist in reference frame O. A grasp will be stable if κ_{o} is positive definite (whether or not the grasp is force closed). We will discuss stability with demonstrative example in Chapter 4. Since in the previous section we have generated ${}^{i}\Delta F$ in contact frame ${}^{i}o_{A}$, we have to represent it here in the universal frame O through appropriate coordinate transformation.

3.8.2.1 Coordinate Transformation of the Stiffness Matrix

If $\begin{pmatrix} {}^{i}x_{fi} & {}^{i}y_{fi} & {}^{i}z_{fi} \end{pmatrix}$ is the coordinates of the frame O as seen from the fixed frame for the ith contact, ${}^{i}o_{A}$, then the change in force ${}^{i}\Delta F_{o}$ at O for any infinitesimal displacement, due to the contact i, is given by

$$ {}^{i}\Delta F_{o} = -{}^{i}T^{T} \; {}^{i}\kappa_{c} \; {}^{i}T\Delta x_{o} \tag{3.30} $$

where in the spatial case

$$ {}^{i}T = \begin{bmatrix} {}^{i}R & {}^{i}D \\ 0 & {}^{i}R \end{bmatrix} \quad {}^{i}D = \begin{bmatrix} 0 & -{}^{i}z_{fi} & {}^{i}y_{fi} \\ {}^{i}z_{fi} & 0 & -{}^{i}x_{fi} \\ -{}^{i}y_{fi} & {}^{i}x_{fi} & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} {}^{i}x_{A} \\ {}^{i}y_{A} \\ {}^{i}z_{A} \end{bmatrix} = \begin{bmatrix} {}^{i}R_{3\times 3} \end{bmatrix} \begin{bmatrix} x_{0} \\ y_{0} \\ z_{0} \end{bmatrix} $$

${}^{i}D$ is a 3×3 skew-symmetric matrix of ith contact. 0 is the 3×3 zero matrix, ${}^{i}R$ is the 3×3 rotation marix that transforms vectors in frame O to the reference frame ${}^{i}o_{A}$.

3.8.2.2 Constant External Force

In the lasso-type grasping, because of the gravitational forces, there exists an external force on the grasped object acting through the center of gravity and a coordinate system, o_{cg} can be attached to it whose axes are aligned such that the external force acts in the z_{cg} direction. A rigid body motion of the grasped object will not change the force, but will result in changes in the moment of the external force about the origin of o_{cg}. If the external force is denoted by mg, the following matrix representation can be written:

$$ \begin{bmatrix} \Delta F_{x} \\ \Delta F_{y} \\ \Delta F_{z} \\ \Delta M_{x} \\ \Delta M_{y} \\ \Delta M_{z} \end{bmatrix} = - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -mg & 0 & 0 & 0 & 0 \\ mg & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{cg} \\ \Delta y_{cg} \\ \Delta z_{cg} \\ \Delta \theta_{x_{cg}} \\ \Delta \theta_{y_{cg}} \\ \Delta \theta_{z_{cg}} \end{bmatrix} $$

$$\Delta\overline{F} = -\kappa_{cg}\Delta x_{cg} \tag{3.31}$$

The same type of coordinate transformation in Section 3.8.2.1 can be applied here yielding $\Delta\overline{F} = T_{cg}^{T}\kappa_{cg}T_{cg}\Delta x_{o}$.

In this chapter, we have derived the kinematic and control model for lasso-type grasping and dragging grasped object to a desired position. The demonstrative results using these derived models will be obtained in Chapter 4. In Chapter 4, firstly we will show that the results of the redundancy controllable system derived in Chapter (3.2) with demonstrative examples. Secondly, we will demonstrate the results of adaptively changing feedback gain in serpentine motion during grasping with and without adaptively changing feedback gain. Thirdly, we will demonstrate the results of grasping stability and show that it is possible to regrasp the object when the grasped object stability is corrupted by a external force. Finally, we will evaluate the performance of system derived in Chapter 3.

CHAPTER 4

SIMULATION RESULTS

4.1 Results of Lasso-Type Grasping

In the simulation, we consider a 15-link snake robot, with 6 wheeled links $(m = 6)$ that is, their non-holonomic constraints are active and other links whose non-holonomic constraints are removed and are not wheeled. We assume that the size of object to be grasped does not involve more than 10 numbers of links in the snake robot. We set the cost function as:

$$V = a\det(\mathrm{A}'^{T}\mathrm{A}') + b\det(\mathrm{B}'\mathrm{B}'^{T}) \qquad (4.1)$$

The first term in Equation (4.1) implies the measure of the singular configuration while the second term is related to the manipulability of the system. The manipulability measure can be consider as a quantitative measure of the capability of a manipulator to move its end effector freely in any direction. This measure can also be regarded as an index of the distance from a singular configuration. Hence the manipulability measure can be used as the cost function when the second subtask is the avoidance of singularities [37].

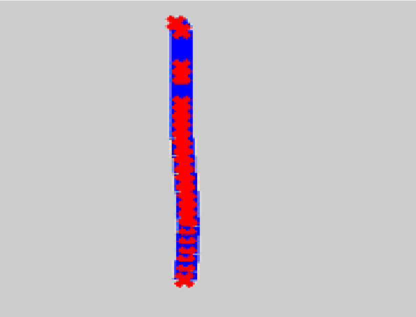

Figure (4.1) Initial configuration of the snake robot

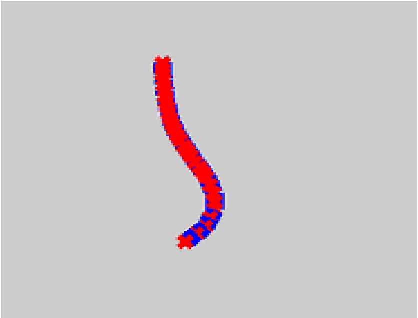

Figure (4.2) Desired configuration of the snake robot

We run the simulation with and without singular configuration avoidance to demonstrate its importance in the efficiency in our modified model. The initial configurations given below belong to a robot posture given in Figure (4.1).

$$w'(0) = [0 \quad 0 \quad 0 \quad \pi/2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\theta_y(0) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\theta_z(0) = [\pi/2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \pi/15]$$

$$K = 2.$$

The desired configuration corresponds to a snake posture is given in Figure (4.2).

$$w'_d = [0.5 \quad 0.5 \quad 0 \quad \pi/3 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3 \quad -0.3]$$

$$\dot{w}'_d = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Figure (4.3) shows the simulation result of the control of the robot head position/orientation and link rise without the singular configuration avoidance $(\alpha = 0)$. In this case, the controller does not use the redundancy. Figure (4.3-a,b,c,d) shows the link rise error, head position error and head orientation error respectively. As shown in these figures, the snake robot is able to track the desired state because all of the errors convergence to zero. Figure (4.3-e) demonstrates the input which becomes zero, that is the snake robot converges to the straight line which is known as singular configuration. This implies reaching a singular configuration for the case of not putting a singular configuration avoidance in the control system $(\alpha = 0)$.

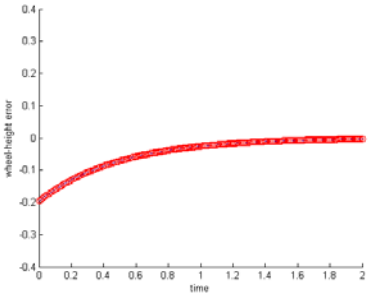

Figure (4.3-a) Link rise error

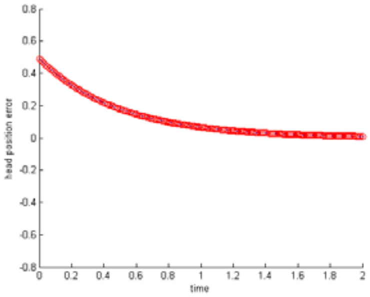

Figure (4.3-b) Head-position error

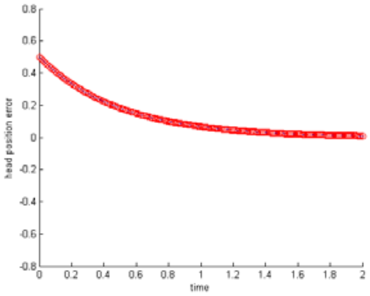

Figure (4.3-c) Head-position error

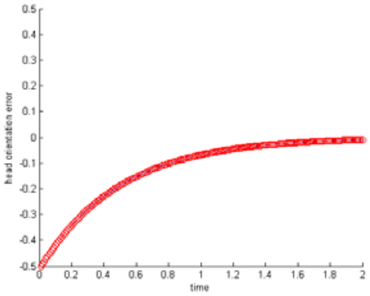

Figure (4.3-d) Head-orientation error

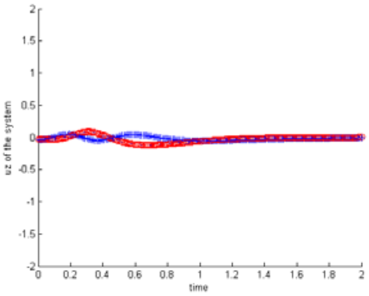

Figure (4.3-e) The input of two joints $(u_{z12} \Rightarrow oo, u_{z14} \Rightarrow ++, \alpha = 0)$

Figure (4.3) Simulation results without singular configuration avoidance

Figure (4.4) shows the simulation result of the control of the robot head position/orientation and link rise with the singular configuration avoidance $(\alpha = 15)$. In this case, the controller uses the snake redundancy. We set $a = 10, b = 10^{-6}$. The figure shows that the snake head and the link rise track each their desired states and the input does not become zero. Figure (4.4-d,e,c,b) gives the link rise error, head position error and head orientation error respectively. As seen from these figures, the snake robot tracks the desired state because all of the errors convergence zero. Figure (4.4-a) demonstrates that the input which fluctuates around zero, never becomes zero during motion. While the error decreases, the fluctuations around zero decrease too. Consequently, when the error is zero, the fluctuations finish and input becomes zero. This implies that the singular configuration is avoided $(\alpha = 15)$.

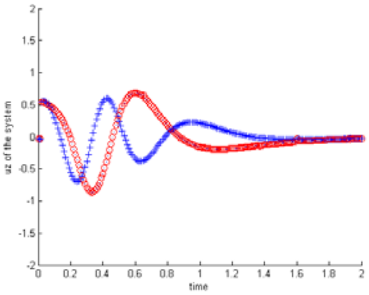

Figure (4.4-a) The input of two joints $(u_{z12} \Rightarrow oo, u_{z14} \Rightarrow ++, \alpha = 15)$

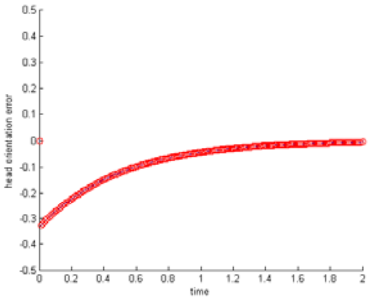

Figure (4.4-b) Head orientation error

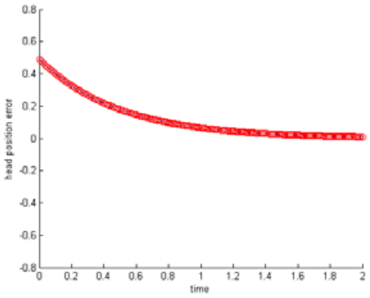

Figure (4.4-c) Head position error

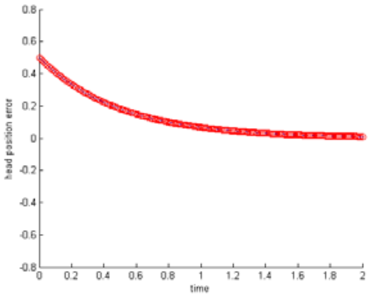

Figure (4.4-d) Head position error

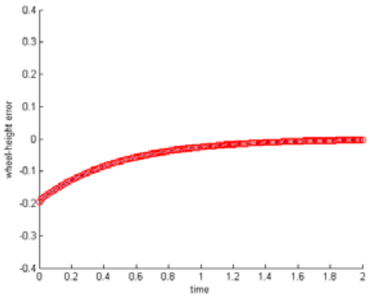

Figure (4.4-e) Link rise error

Figure (4.4) Simulation results with singular configuration avoidance

These simulations given by Figure (4.3) and Figure (4.4) deal with low-error performance where snake head makes small amount of position and orientation. To show the effectiveness of our control strategy we can also evaluate the high-error performance. For high error performance simulations, we consider high amount of orientation error $\left(w_d(4) - w(4)\right)$ which is also seen in the parameters of simulation given in Figure (4.5) and the initial and desired state of snake robot which we take as:

$$w = [6.85 \quad 8.08 \quad 0 \quad -3.66 \quad -0.05 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.25 \quad -0.3 \quad -0.35 \quad -0.4 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6]$$

$$w_d = [6.85 \quad 8.08 \quad 0 \quad -9.42 \quad -0.05 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.25 \quad -0.3 \quad -0.35 \quad -0.4 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6]$$

$$\alpha = 10, a = 20, b = 10 .$$

Figure (4.5) shows the head orientation error and the inputs which belong the two joints with and without singular configuration avoidance. From Figure (4.5-c) to Figure (4.5-d), we find out that the snake robot tracks the desired state with and without singular configuration avoidance because the error converges to zero in either case. But Figure (4.5-a) demonstrates that in lack of the singular configuration avoidance $(\alpha = 0)$, the input becomes zero after 0.6 seconds, that is the snake robot converges to the straight line which is known as singular configuration. In Figure (4.5-b), however, the input fluctuates around zero, and never becomes zero during motion, thus singular configuration is always avoided $(\alpha = 10)$.

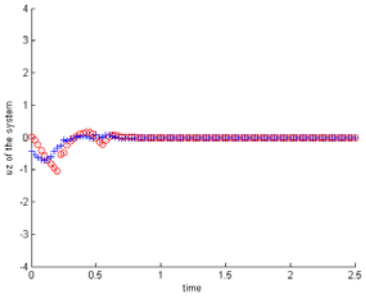

Figure (4.5-a) The input of two joints $(u_{z26} \Rightarrow oo, u_{z27} \Rightarrow ++, \alpha = 0)$

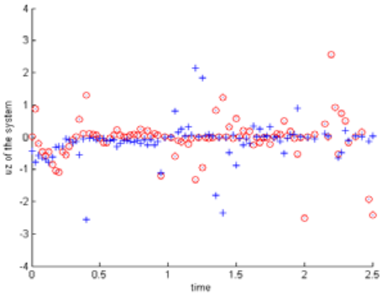

Figure (4.5-b) The input of two joints $(u_{z26} \Rightarrow oo, u_{z27} \Rightarrow ++, \alpha = 10)$

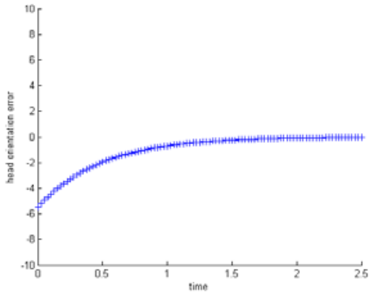

Figure (4.5-c) Head orientation error $(\theta_{z0d} - \theta_{z0}), \alpha = 0$

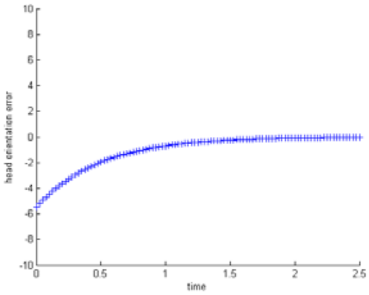

Figure (4.5-d) Head orientation error $(\theta_{z0d} - \theta_{z0}), \alpha = 10$

Figure (4.5) Simulation results of the high-error performance with and without singular configuration avoidance

4.2 Results of Serpentine Motion During Grasping

In Equation (3.15), it is important to control the position of contact points, position and orientation of head link and link rises of the links simultaneously for dragging the object which is wrapped in lasso-type. As said in Chapter 3, the position of contact points should move as a result of the entire snake body motion to drag the grasped object. Furthermore, since the entire snake-robot motion results from the motion of last links having non-holonomic constraint, the contact points motion and so, the grasped object motion is also generated from the motion of last links having non-holonomic constraint. So, while the head link moves, the contact points and grasped object can move as a result of the head link motion and so, the entire snake body motion. Let us recall that in the simulation, we consider a 15-link snake robot, with 6 wheeled links $(m = 6)$ that is, their non-holonomic constraints are active. The other links have non-holonomic constraints removed (are not wheeled). We assume that the perimeter of the object to be grasped does not surpass the length of 8 links in the snake robot. Firstly we set the feedback gain K as a constant $\mathrm{K} = 1$ to test the system in Equation (3.15) for constant gain and we set the initial and desired state as follows.

$$w'' = [2.0139 \quad 5.5077 \quad 0 \quad -5.7572 \quad -0.05 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.25 \quad -0.3 \quad -0.35 \quad -0.4$$

$$-0.45 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad 2.4154 \quad 1.8029 \quad 2.1689 \quad 3.6418 \quad 3.2926 \quad 2.7967]$$

$$w''_d = [0.0139 \quad 4.5077 \quad 0 \quad -5.7572 \quad -0.05 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.25 \quad -0.3 \quad -0.35 \quad -0.4$$
$$-0.45 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad 2.4154 \quad 1.8029 \quad 2.1689 \quad 3.6418 \quad 3.2926 \quad 2.7967]$$

$$\mathrm{K} = 1, \ m = 6 .$$

This desired state is determined by the scheme given in Chapter (3.6). Figure (4.6-a) shows the initial state of the snake-robot and Figure (4.6-b) shows the final state in which some joints have excessively large input which prevents the serpentine locomotion control during dragging the grasped object as seen in a knotted snake confguration.

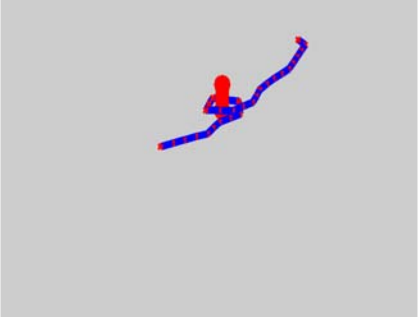

Figure (4.6-a) Initial state of snake-like robot

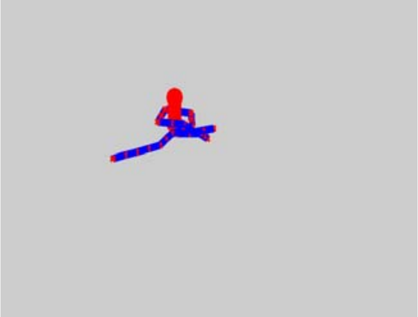

Figure (4.6-b) Final state of snake-like

Figure (4.6) Serpentine motion during grasping without adaptively changed feedback gain

Figure (4.7) shows that the joint angles of the last four links having non-holonomic constraint, have discontinuities in which these joints exhibit excessive large input for the controller, and corrupt the serpentine locomotion control during dragging the grasped object.

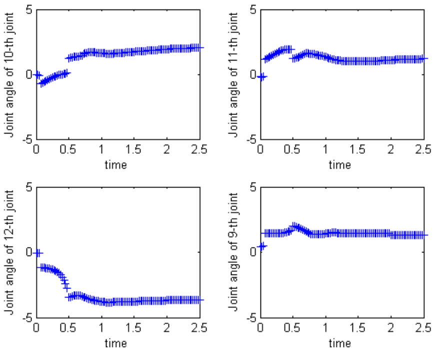

Figure (4.7) Joint angle of the four links having non-holonomic constraint

Now let us consider the effect of adaptively changing feedback gain. Figure (4.8) shows that the joint angles of last four links having non-holonomic constraint do not exhibit the discontinuities that was seen in Figure (4.7). Figure (4.9-a) shows the initial state of snake-robot being the same as in Figure (4.6-a), Figure (4.9-b) shows the final state of the snake-robot. As shown in Figure (4.9-b), the serpentine motion of last six links having non-holonomic constraint can not be corrupted with excessively large inputs as it was previously.

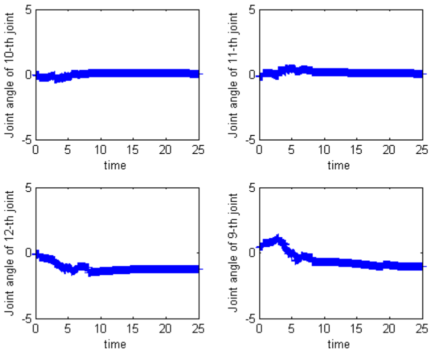

Figure (4.8) Joint angle of the four links having non-holonomic constraint

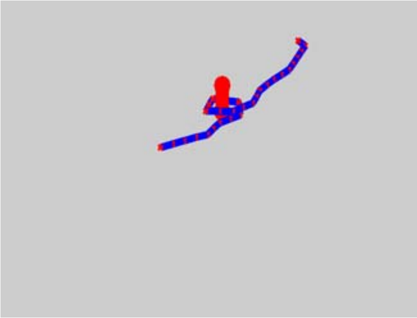

Figure (4.9-a) Initial state of the snake-robot

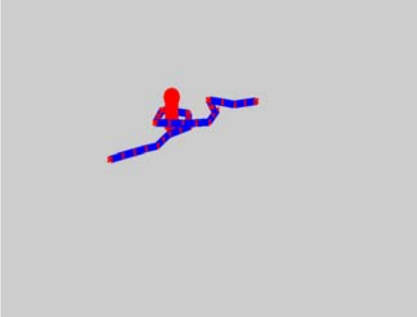

Figure (4.9-b) Final state of the snake-robot

Figure (4.9) Serpentine motion during grasping with adaptively changed feedback gain

Figure (4.10-a) shows that the head position errors converge to zero. Because of the adaptively changed feedback gain, it does not have homogeneous convergence in which snake-robot converges the desired state with the step having the same amount of displacements. Figure (4.10-b,c,d) shows the position error of the contact points. As shown in Figure (4.10-b,c,d), $\left((x_{fid}-x_{0d})-(x_{fi}-x_{0})\right)$, $\left((y_{fid}-y_{0d})-(y_{fi}-y_{0})\right)$ and $\left((z_{fid}-z_{0d})-(z_{fi}-z_{0})\right)$ are equal to zero for all contact points, that is, all contact points are able to track the head link movement. Furthermore, as shown in Figure (4.10-e,f,g), all contact points track the head-link movement which was given in Figure (4.10-a). Although in Figure (4.10-e,f,g), the contact points do not have homogeneous convergence, all contact points have same trajectory. This situation is verified by Figure (4.10-b,c,d) because $\left((x_{fid}-x_{0d})-(x_{fi}-x_{0})\right)$, $\left((y_{fid}-y_{0d})-(y_{fi}-y_{0})\right)$ and $\left((z_{fid}-z_{0d})-(z_{fi}-z_{0})\right)$ are equal to zero for all contact points during the snake-robot motion. So, all contact points and the grasped object track the head link movement. Since the head link movement results from the serpentine motion of the last six links which have the non-holonomic constraint, the contact points and the grasped object movement also result from serpentine motion of the last six links having the non-holonomic constraint and the grasped object can not be corrupted by the motion of contact points.

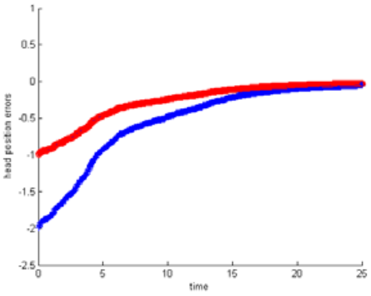

Figure (4.10-a) The head position errors,

$(x_{0d}-x_{0}) \Rightarrow$ blue graph, $(y_{0d}-y_{0}) \Rightarrow$ red graph.

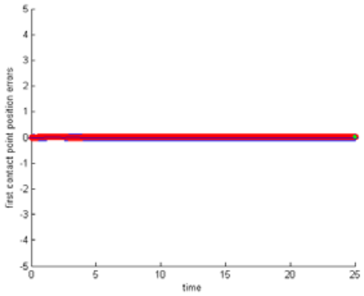

Figure (4.10-b) The position errors of first contact point,

$(x_{f5d} - x_{0d}) - (x_{f5} - x_0) \Rightarrow$ blue graph, $(y_{f5d} - y_{0d}) - (y_{f5} - y_0) \Rightarrow$ red graph,

$(z_{f5d} - z_{0d}) - (z_{f5} - z_0) \Rightarrow$ green graph.

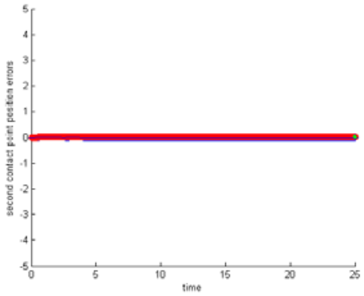

Figure (4.10-c) The position errors of second contact point,

$(x_{f6d} - x_{0d}) - (x_{f6} - x_0) \Rightarrow$ blue graph, $(y_{f6d} - y_{0d}) - (y_{f6} - y_0) \Rightarrow$ red graph,

$(z_{f6d} - z_{0d}) - (z_{f6} - z_0) \Rightarrow$ green graph.

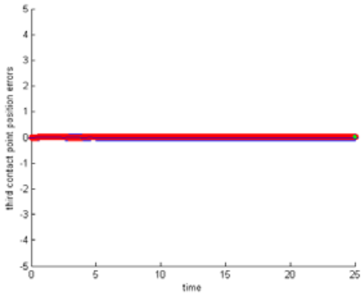

Figure (4.10-d) The position errors of third contact point,

$(x_{f7d} - x_{0d}) - (x_{f7} - x_0) \Rightarrow$ blue graph, $(y_{f7d} - y_{0d}) - (y_{f7} - y_0) \Rightarrow$ red graph,

$(z_{f7d} - z_{0d}) - (z_{f7} - z_0) \Rightarrow$ green graph.

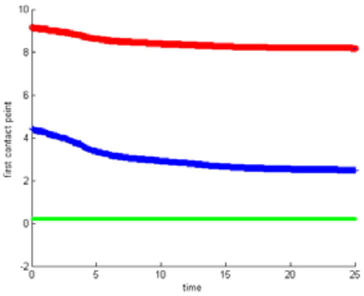

Figure (4.10-e) The movement of first contact point,

$x_{f5} \Rightarrow$ blue graph, $y_{f5} \Rightarrow$ red graph, $z_{f5} \Rightarrow$ green graph.

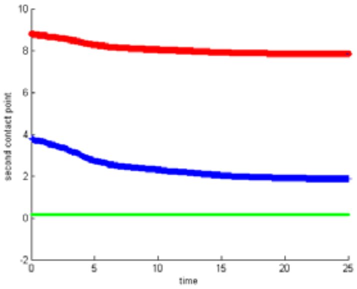

Figure (4.10-f) The movement of second contact point,

$x_{f6} \Rightarrow$blue graph, $y_{f6} \Rightarrow$red graph, $z_{f6} \Rightarrow$green graph.

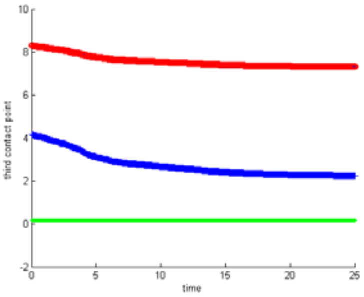

Figure (4.10-g) The movement of third contact point,

$x_{f7} \Rightarrow$ blue graph, $y_{f7} \Rightarrow$ red graph, $z_{f7} \Rightarrow$ green graph.

Figure (4.10) Simulation results of serpentine motion during grasping with adaptively changed feedback gain

4.3 Results of Grasping Stability

While the grasped object is dragged by the snake-robot, the stability of grasping is an important quantity to counteract under disturbance forces. Since the stability of the grasped object can be corrupted by the external disturbance forces, regrasping is necessary to restore stability. To show the regrasping of the object, we set the (κ_c) of the contact points as follows;

$$k_t^1 = k_n^1 = 50,\ k_t^2 = k_n^2 = 50,\ k_t^3 = k_n^3 = 50$$

$$F_t^1 = [20 \quad -10], F_t^2 = [20 \quad -10], F_t^3 = [20 \quad -10], mg = 40.$$

Since κ_o derived with these parameters is a positive definite matrix, the grasping is a stable grasping. If we apply a small external force to the object, the change of grasped object position can be found by solving the pseudo-inverse solution of Equation (3.27). If we select the small external force, F_{ex}, as $F_{ex} = [1.5 \quad 2 \quad 0]$, we can find the change of the grasped object position, d_{ch}, as $d_{ch} = [0.1823 \quad 0.0012 \quad 0.0918]$. If we give this velocity to contact points of the snake robot by using the scheme given in Chapter (3.6), the snake-robot is faced to regrasp the object to restore stability if possible. Figure (4.11-a) shows the unstable grasped object whose position is corrupted by a small external force F_{ex}. As shown in Figure (4.11-b), the snake robot regrasps the object to obtain again a stable grasp.

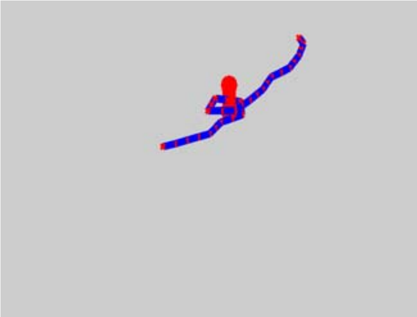

Figure (4.11-a) Corruption of grasping with the external disturbance force

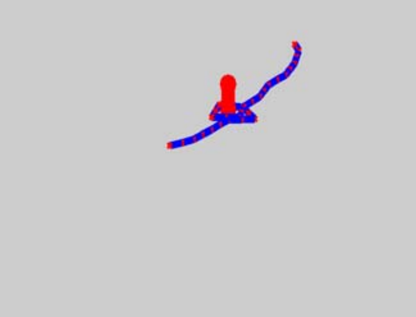

Figure (4.11-b) Regrasping of object for again gaining the grasping stablity

Figure (4.11) Regrasping of the snake-robot

4.4 Performance Analysis

4.4.1 Performance Analysis of Redundancy Controllable System

The performance analysis in robotic applications is needed to select the system parameters most effectively. In this thesis, the system parameters are evaluated to determine their appropriate values. Firstly, the system parameter α given in Equation (3.10) for the solution of redundancy controllable system will be evaluated. α should be a positive constant according to the concept given in [28]. But α should be selected for effectively running of our lasso-type grasp system. For this purpose, we set the initial and desired state of the system and the coefficients of cost function given in Equation (4.1) as follows.

$$w' = [0 \quad 0 \quad 0 \quad \pi/3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]',$$

$$w'_d = [2.5 \quad 7 \quad 0 \quad -2\pi/3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]',$$

$$a = 100, \quad b = 100.$$

We run the redundancy controllable system for two different values of α. Firstly, we set the α as $\alpha = 1$. Figure (4.12) shows the two joint inputs of redundancy controllable system for $\alpha = 1$. As shown in this figure, although the redundancy controllable system run to avoid the singular configuration, the inputs converge zero, that is, the snake robot converges to straight line which is known as singular configuration.

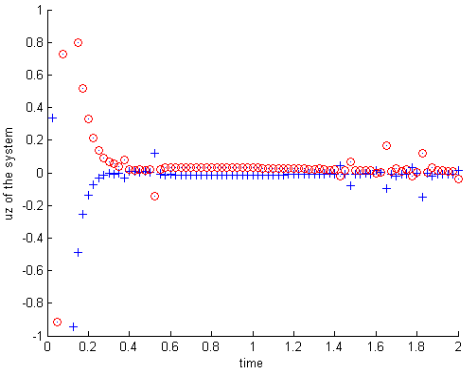

Figure (4.12) Joint inputs $u_z(25)$ and $u_z(26)$ of redundancy controllable system for $\alpha = 1$.

When we change α as $\alpha = 15$, we observe that the redundancy controllable system can avoid singular configurations. Figure (4.13) shows the two joint inputs of redundancy controllable system for $\alpha = 15$. As shown in this figure, the inputs never becomes zero. This implies the singular configuration avoidance of snake robot.

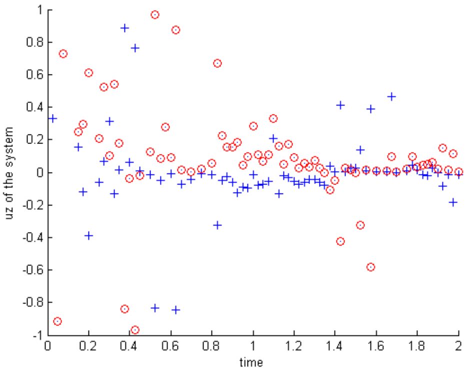

Figure (4.13) Joint inputs $u_z(25)$ and $u_z(26)$ of redundancy controllable system for $\alpha = 15$.

As a result, α for the solution of redundancy controllable system given in Equation (3.10) should be big enough as that the redundancy controllable system can avoid the singular configurations.

Secondly, the system parameter a, b given in Equation (4.1) as a cost function coefficients of redundancy controllable system will be evaluated. a, b is given in [27] as a positive constant. But a, b should be selected for effectively running of our lasso-type grasp system. For this purpose, we set the initial and desired state of system and the coefficient α given in Equation (3.10) as follows.

$$w' = [0 \quad 0 \quad 0 \quad \pi/3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]',$$

$$w_d' = [2.5 \quad 7 \quad 0 \quad -2\pi/3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]'$$

$$\alpha = 15.$$

We run the redundancy controllable system for two different values of a and b. Firstly, we set the a and b as $a = 1$, $b = 1$. Figure (4.14) shows the two joint inputs of redundancy controllable system for $a = 1$, $b = 1$. As shown in this figure, although the redundancy controllable system run to avoid the singular configuration, the inputs converge zero, that is, the snake robot converges to straight line which is known as singular configuration.

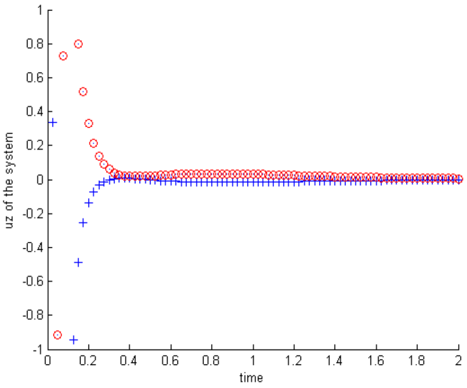

Figure (4.14) Joint inputs $u_z(25)$ and $u_z(26)$ of redundancy controllable system for $a = 1$, $b = 1$.

When we change a and b as $a = 50$, $b = 100$, we observe that the redundancy controllable system can avoid from singular configurations. Figure (4.15) shows the two joint inputs of redundancy controllable system for $a = 50$, $b = 100$. As shown in this figure, the inputs never becomes zero. This implies the singular configuration avoidance of snake robot.

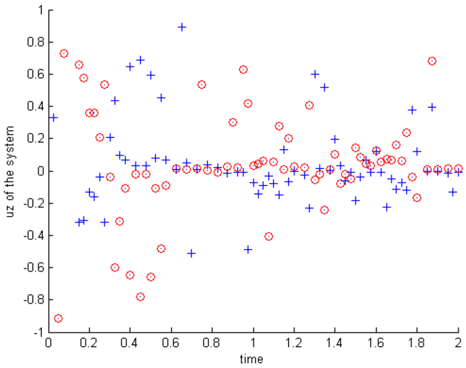

Figure (4.15) Joint inputs $u_z(25)$ and $u_z(26)$ of redundancy controllable system for $a = 50$, $b = 100$.

As a result, a and b for the cost function of redundancy controllable system given in Equation (4.1) should be selected as large values around $20 \le a, b \le 100$ so that the redundancy controllable system can avoid the singular configuration.

4.4.2 Performance Analysis of Serpentine Motion During Grasping

As a performance analysis of serpentine motion during grasping, we analyse the high error performance evaluation of the control scheme based on the adaptively changing feedback gain given in Equation (3.18). Low error performance of this control scheme has been given in Section (4.2). To analyse the high error performance, we set the same initial state given in Section (4.2) and the desired state including the high error as follows.

$$w'' = [2.0139 \quad 5.5077 \quad 0 \quad -5.7572 \quad -0.05 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.25 \quad -0.3 \quad -0.35 \quad -0.4$$

$$-0.45 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad 2.4154 \quad 1.8029 \quad 2.1689 \quad 3.6418 \quad 3.2926 \quad 2.7967]$$

$$w''_d = [-3.0139 \quad -2.5077 \quad 0 \quad -6.2832 \quad -0.05 \quad -0.1 \quad -0.15 \quad -0.2 \quad -0.25 \quad -0.3 \quad -0.35 \quad -0.4$$

$$-0.45 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad -0.6 \quad 2.4154 \quad 1.8029 \quad 2.1689 \quad 3.6418 \quad 3.2926 \quad 2.7967]$$

$$, \ m = 6 \, .$$

This desired state having high head position errors is determined by the scheme given in Chapter (3.6). Figure (4.16) shows that the joint angles of the last four links having non-holonomic constraint, have discontinuities in which these joints exhibit excessive large input for the controller, and corrupt the serpentine locomotion control during dragging the grasped object.

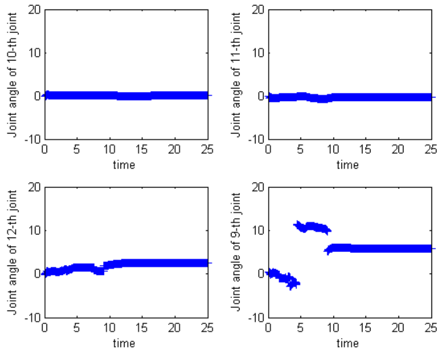

Figure (4.16) Joint angle of the four links having non-holonomic constraint

As a result, we can say that the control scheme based on the adaptively changing feedback gain given in Equation (3.18) may generate the excessively large inputs which corrupt the serpentine locomotion control for the high error performance.

4.4.3 Performance Analysis of Grasping Stability

Stability is a important quantity to analyse the grasping performance. We analyse the lasso-type grasping stability by means of matrix κ_o given in Equation (3.29). κ_o is a positive definite matrix for a stable grasp. By changing the weight of grasped object, we will analyse the lasso-type grasping performance. Firstly we give a stable grasping configuration as follows.

$$k_t^1 = k_n^1 = 1, \ k_t^2 = k_n^2 = 2, \ k_t^3 = k_n^3 = 5$$

$$F_t^1 = [1 \quad 2], F_t^2 = [2 \quad 3], F_t^3 = [-0.5 \quad 0.3], mg = 4 \, .$$

The matrix κ_o is a positive definite matrix having positive and real eigen values for stable grasping [35]. For this configuration, since the matrix κ_o has the positive and real eigen values (719.8844, 787.5844, 0.0949, 2.9273, 8.2893, 7.1409), the lasso-type grasping is a stable grasping.

While the weight of grasped object is increased for the same configuration, we can observe that the lasso-type grasping stability goes to instability. If we select the weight of grasped object as $mg = 9$, the matrix κ_o has the non-real eigen values 1.0e+002 *(7.0826, 7.9665, 0.0018, 0.0801 + 0.0035i, 0.0801 - 0.0035i, 0.0480) and so, the lasso-type grasping is a instable grasping.

As a result, we can say that an appropriate weight of grasped object should be selected for a lasso-type grasping in which grasped object can be dragged without the corruption of grasping stability because of the grasped object weight.

CHAPTER 5

CONCLUSION

Although the concepts are fully extendible to three dimensional using the snake's head to grasp an intended object, grasping studies of snake-like robot in the literature have been performed as either the planar examples or the fixed base hyper-redundant manipulators. In this thesis, we extend the snake robot grasping to 3-D with a grasping ability using not only just the head, but any body links while undertaking a serpentine motion. This is the novelty of our approach where the snake robot grasps an object with any of its body link which is at close proximity to the object while undergoing its serpentine motion with the remaining links and dragging the grasped object. Since our snake robot has the pitch motion for every link, we can ensure that the links do not run onto each other as they wrap around the object. A lasso-type power grasp is then possible for our 15-link snake robot as seen in the simulation results of this thesis.

Furthermore we develop the kinematic and control models for lasso-type grasping and for dragging the grasped object to a desired state. This control model includes an adaptively changing feedback gain which prevents excessively large inputs to corrupt the serpentine locomotion control. According to our lasso-type grasping model, while the snake robot can grasp the object beginning with the any body link at close proximity of the object, the contact points can be controlled to generate the curvilinear grasping curve by using our lasso-type grasping procedure. For dragging the grasped object, we define a scheme which can determine the appropriate desired state to drag the grasped object to a desired position.

Since the stability of the grasped object is important to resist the disturbance forces as well as the force closure grasping is important to counteract the disturbance force, to analyze the stability of the lasso-type grasping, we introduce a stability model of lasso-type grasping based on contact stiffness matrices that faces the snake to regrasp when gone unstable.

As the general results, we can say that snake-like robot can grasp the any objects which should have the appropriate weight providing the positive and real eigenvalues in κ_o and have the perimeter which can be covered by the number of links constrainted with the inequality $m + 2r \leq n - 1$. For dragging of grasped object, we can say that snake-like robot can drag the grasped object which should have the appropriate weight providing the positive and real eigenvalues in κ_o and appropriate desired state consisting of low error performance have to be defined to drag the grasped object to a this desired state by using our dragging procedure.

5.1 Future Work

As the future works, the intelligent approaches can be proposed for the lasso-type enwrapment and dragging of the grasped object during the motion planning. Our grasping and dragging procedures can generate the appropriate desired to grasp or drag the object or grasped object respectively in a known environment but in a unknown environment snake robot have to learn generating the appropriate desired state providing the object grasping and dragging of grasped object to a desired position without collision. Furthermore, since snake-like robot can handle the low error performance during the dragging of grasped object to a desired position, the intelligent gait selection can be attached to our system to determine the appropriate gaits of last links having non-holonomic constraint for transporting the grasped object to any position because the last links having non-holonomic constraint have the redundant degree of freedom to generate the appropriate gaits.

REFERENCES

[1] Worst R., Linnemann R., " Contruction and Operation of a Snake-Like Robots", IEEE-1996

[2] Klaassen B., Paap L. K., "GMD-SNAKE2: A Snake-Like Robot Driven by Wheels and a Method for Motion Control", IEEE/ International Conference on Robotics and Automation, May 1999.

[3] Paljug E., Ohm T., Hayati S., " The JPL Serpentine Robot: a 12-DOF System for Inspection", IEEE/ International Conference on Robotics and Automation, 3143-3148, 1995.

[4] NEC Corporation, "Orochi 12-DOF Snake-Like Robot", 1996.

[5] Erkmen I., Erkmen M. A., Matsuno F., Chatterjee R., Kamegawa T., "Snake Robots to the Rescue!" IEEE Robotics and Automation Magazine, 2002

[6] Sensor Based PlanningLab– Search and Rescue Applications, Internet WWW page, at URL:. http://voronoi.sbp.ri.cmu.edu/projects/

[7] Pettinato J. S., Stephanou H. E., "Manipulability and Stability of a Tentacle Based Robot Manipulator", IEEE /Computer Society, 458-463, 1989.

[8] Desai R., Rosenberg C. J., Jones J. L., "Kaa: An Autonomous Serpentine Robot Utilizes Behavior Control", IEEE/IROS, 250-255, 1995.

[9] Chirikjian G., Burdick J., "Kinematics of Hyper-Redundant Robot Locomotion with Application to Grasping", IEEE/ International Conference on Robotics & Automation,720-725, 1991.

[10] Dowling K., "Limbless Locomotion: Learning to crawl with a Snake Robot", Phd thesis, Carnegie Mellon University, Pittsburg. PA, 1997

[11] Hirose S., "Biologically Inspired Robots: Snake-Like Locomotor and Manipulators" ,Oxford University Press, 1993.

[12] Ikeda H., Takanashi N., "Joint Assembly Moveable Like a Human Arm", NEC Corporation, 1987.

[13] Chirikjian G., Burdick J., "Design an Experiments with a 30-DOF Robot" IEEE Transactions on Robotics and Automation, 1993.

[14] Chirikjian G., Burdick J., "A Modal Approach to Hyper-Redundant Manipulator Kinematics", IEEE Transactions on Robotics and Automation, 1993.

[15] Chirikjian G., Burdick J., "An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators" IEEE-, 1990.

[16] Chirikjian G., Burdick J., "Hyper-Redundant Robot Mechanism and Their Applications", IEEE/RSJ-IROS, 1991.

[17] Munerato F., Miallachi D., Laurent C., "Flexible Mini-Robot with Autonomous Moiton", PDF Document from Internet.

[18] Immega G., Antonelli K., "The KSI Tentacle Mnipulator", IEEE/ International Conference on Robotics and Automation, 3149-3154, 1995.

[19] Crespi A., Badertscher A., Guignard A., Ijspeert J. A., "Amphibot I: an Amphibious Snake-Like Robot", Elsevier/ Robotics and Autonomous Systems, 163-175, 2004.

[20] Ma S., Araya H., Li L., "Development of a Creeping Snake-Robot", IEEE International Symposium on Computational Intelligence in Robotics and Automation, 77-82, 2001.

[21] Prautsch P., Mita T., "Control and Analysis of the Gait of Snake Robots", IEEE/ International Conference on Control Applications, 1999.

[22] Date H., Hoshi Y., Sampei M., "Dynamic Manipulability of a Snake-Like Robot with Consideration of Side Force and its Application to Locomotion Control", Tokyo Institute of Technology, PDF Document from Internet.

[23] Ostrowski J., Burdick J., "Gait Kinematics for a Serpentine Robot", IEEE/ International Conference on Robotics and Automation, 1996.

[24] Hirose S., Mori M., "Development of Active Cord Mechanism ACM-R3 with Agile 3D Mobility", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.

[25] Chirikjian G., Burdick J., “Parallel Formulation of the Inverse Kinematics of Modular Hyper-Redundant Manipulators” IEEE/ International Conference on Robotics & Automation, 708-713 1991.

[26] Migadis G., Kyriakopoules K. J., “Design and Forward Kinematic Analysis of a Robotis Snake”, IEEE/ International Conference on Robotics & Automation, 3493-3498, 1997.

[27] Ma S., Ohmameuda Y., Inoue K., Li B., “Control of a 3-Dimensional Snake-Like Robot”, IEEE/ International Conference on Robotics & Automation, 2067-2072, 2003.

[28] Matsuno F., Mogi K., “Redundancy Controllable System and Control of Snake Robots Based on Kinematic Model”, IEEE/ Conference on Decision and Control, 4791-4796, 2000.

[29] Matsuno F., Suenaga K., “Control of Redundant Snake-Robot based on Kinematic Model”, IEEE/SICE, 1481-1486, 2002.

[30] Yamakita M., Hashimoto M., Yamada T., “Control of Locomotion and Head Configuration of 3D Snake Robot”, IEEE/ International Conference on Robotics & Automation, 2055-2060, 2003.

[31] Hirose S., Umetani Y., “The Kinematics and Control of the Soft Grippers”, Web page of Tokyo Institute of Technology.

[32] Pollard S. N., Perez T., “Grasp Stability and Feasibility for an Arm with Articulated Hand”, IEEE/ International Conference on Robotics and Automation, 1581-1585, 1990.

[33] Park C. Y., Starr P. G., “Grasp Synthesis of Polygonal Objects”, IEEE/ International Conference on Robotics and Automation, 1574-1580, 1990.

[34] Hsu P., Li Z., Sastry S., “On Grasping and Coordinated Manipulation by a Multifingered Robot Hand”, IEEE/ International Conference on Robotics and Automation, 384-389, 1988.

[35] Howard S. W., Kumar R. V., “On the Stability of Grasped Objects”, IEEE/ Transactions on Robotics and Automation”, 1996.

[36] Saito M., Fukaya M., and Iwasaki T., “Modeling, analysis, and synthesis of serpentine locomotion with a multilink robotic snake”, IEEE Control Systems Magazine.

[37] Yoshikawa T., “Foundations of Robotics-Analysis and Control”, MIT Press, 1990.