

Research Article

Biologically Inspired Perimeter Detection for Whole-Arm Grasping

David Devereux,¹ Robert Richardson,² Arjun Nagendran,³ and Paul Nutter¹

¹ Department of Computer Science, University of Manchester, M13 9PL, UK

² School of Mechanical Engineering, University of Leeds, LS2 9JT, UK

³ Institute for Simulation and Training, University of Central Florida, 32826, USA

Correspondence should be addressed to Robert Richardson; r.c.richardson@leeds.ac.uk

Received 12 June 2013; Accepted 8 July 2013

Academic Editors: J. Archibald, A. Hamzaoui, and J.-S. Liu

Copyright © 2013 David Devereux et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Grasping is a useful ability that allows manipulators to constrain objects to a desired location or trajectory. Whole-arm grasping is a specific method of grasping an object that uses the entire surface of the manipulator to apply contact forces. Elephant trunks and snakes and octopus arms are illustrative of these methods. One of the greatest challenges of whole-arm grasping in poorly defined environments is accurately identifying the perimeter of an object. Existing algorithms for this task use restrictive assumptions or place unrealistic demands on the required hardware. Here, a new algorithm (termed Octograsp) has been developed as a method of gaining information on the shape of the grasped object through tactile information alone. The contact information is processed using an inverse convex hull algorithm to build a model of the object's shape and position. The performance of the algorithm is examined using both simulated and experimental hardware. Methods of increasing the level of contact information through repeated contact attempts are presented. It is demonstrated that experimentally obtained, coarsely spaced, contact information can result in an accurate model of an object's shape and position.

1. Introduction

To grasp an object is to seize or hold it, thereby constraining it to a desired position or trajectory. Industrial robots implement finger-tip grasping (based around biological inspiration from humans) to perform tasks on objects that are normally well defined, often rigid, and always considerably smaller than the robotic manipulator. Robotic systems are increasingly being considered for new application areas where the operating requirements are more complex and less well defined; consider the use of robotic end-effectors for the assembly of complex machines with components of varying size and consistency. For applications such as these, an alternative approach to robotic grasping is required. Whole-arm grasping is an alternative to finger-tip grasping [1, 2] that is used by animals such as elephants, snakes and octopuses. Whole-arm grasping uses the entire surface of the manipulator to provide contacts between the object and manipulator. This allows the distribution of grasp forces and an increased surface area to aid the grasping of smooth

objects. For example, consider the scenario of Figure 1 where a 10-link multiple section robot arm has encircled an object and is applying grasping forces. Each link has the capability of exerting forces at various points along the objects body. The grasping control aim is to exert the minimum forces whilst grasping the object.

At each of the contacts between an object and manipulator there is a force applied to the object f_i . This force causes a torque τ_i around the centre of mass. The torque and force can be combined into a single vector called a wrench, as shown in

$$\mathbf{w}_i = \begin{pmatrix} f_i \\ \tau_i \end{pmatrix} = \begin{pmatrix} f_i \\ r_i \times f_i \end{pmatrix}, \quad (1)$$

where r_i is a vector from the centre of mass of the object to the contact point i . The set of all wrenches forms the wrench space.

The quality of a grasp is a measure of its ability to reject all possible external wrenches acting on the object, one measure

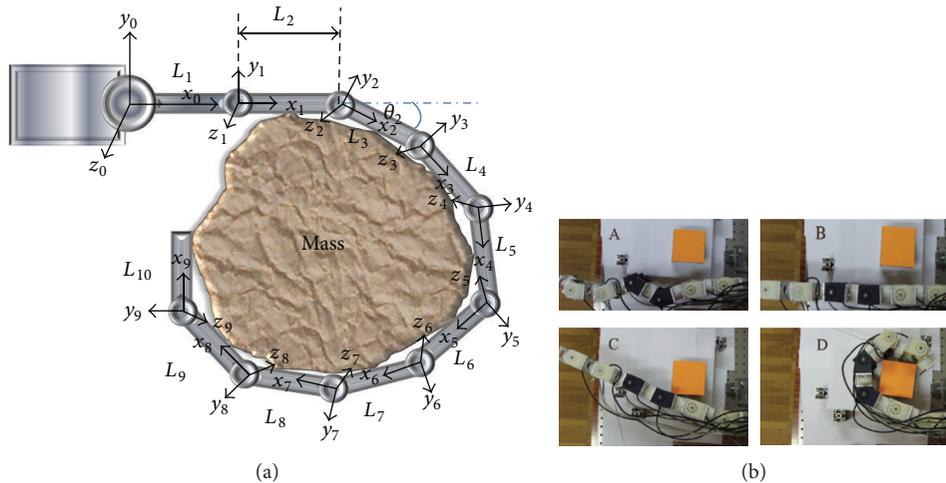


FIGURE 1: (a) Coordinate frames are assigned to a 10-link serial link manipulator, with “z” in all frames pointing out of the page. Link lengths can vary and are measured between joint centers as shown for L_2 . Counter-clockwise rotation about the z-axis is positive. Controlling the manipulator so that it encircles an object and applies the necessary forces at all contact points to ensure a stable grasp is referred to as whole-arm grasping. (b) Experimental hardware demonstrating whole-arm grasping. Stages of the whole-arm grasping include (A) starting from an initial random position, (B) straightening out prior to a “sweep-phase”, (C) proximal joints sweeping prior to contact, and (D) distal joints sweeping to complete the encirclement.

often used being the wrench ball measure [3]. The unfortunate drawback of wrench ball method is that it does not give any indication of stability. If a method simply attempts to maximise the measure, then points on extremes of facets are chosen. This risks the possibility of unstable grasps due to positioning errors, which may place contacts past edges thereby drastically changing the grasp. This problem is mitigated using whole-arm grasping as the idea is to apply lots of contact, which provides more redundancy in the grasp.

An accurate definition of the perimeter of an object is required so that the locations and magnitude of constraining forces can be identified. Existing approaches make additional demands in terms of equipment and processing power in order to be effective. In general, they require additional equipment, such as cameras, graphics cards, or force sensitive skins, and/or are computationally expensive. Current methods assume the availability of a tactile sensor skin that stretches over the surface of the manipulator; an example method is by Asano et al. [4] who use the contact location and force information to dynamically manipulate an object. Unfortunately, the fully force sensitive skin has yet to be developed, and so these methods remain simulations. This may yet change due to new material research. One such material is ionic electroactive polymers as used by Shahinpoor [5] and, more recently, Biddiss and Chau [6] who both describe using IPMCs as touch sensors. The tactile information can be processed to obtain a model of the shape and position of the unknown object. Methods for discovering the shape and location of an object through contact information alone must be able to intentionally, systematically, and reactively contact the object. The control algorithms must react to the contacts such that the positional disturbance of the object is minimised. These reactive methods can be used to fully encircle an object and gain information about the surface of that object.

Busch [7] describes three reactive methods that could be employed to grasp an object by using whole-arm grasping. The first method is the same method of grasping as McMahan et al. [8], Walker et al. [9], and Desai et al. [10]; Busch calls this method *Zorro*. The method contains two competing controllers, one of which attempts to make the end-effector of the robot circle around the object, while the other attempts to keep the body of the robot away from the object. The combination of these two controllers causes the arm to tighten around the object as the end-effector circles the object. The second method is the *Follow-the-Leader* method. This method operates by initially making the end-effector contact and follow the surface of the object, while the remaining links follow the path the end-effector took along the surface of the object. This approach is based on the work by Reznik and Lumelsky [11]. Both approaches require the availability of a force sensitive skin to detect contacts, contours, and forces. The last method is called *Lasso* and requires the use of sensors covering the entirety of the arm that can detect the distance to any object normal to the sensor. The *Lasso* method encircles the object without making contact with it, meanwhile, the object’s structure is analysed for grasp planning purposes with the hypothetical sensors. All three of these methods analyse the structure of the object on the fly and reactively move to adapt the grasp to the object. Unfortunately the *Lasso* and *Follow-the-Leader* methods require additional complex sensors, whilst the *Zorro* method requires the knowledge of the object’s position and shape.

A new method that does not require additional sensors other than those needed for actuation in a closed-loop system and also does not require excessive assumptions/prior knowledge may therefore be beneficial. The contributions of this paper are in demonstrating how to interpret and combine the coarse contact information provided by such a

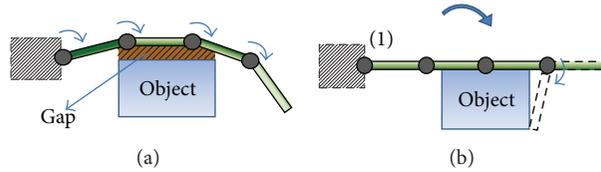


FIGURE 2: (a) All joints move simultaneously resulting in contact at link 3 and leaving a gap between object and manipulator. (b) Only base joint moves until contact is made—the result is multiple points of contact at link 2 and link 3. Link 4 then moves inward until contact is made, resulting in wraparound.

method, through a novel and systematic object exploration algorithm. The approach taken here is as follows. A snake-like serial-link manipulator is first controlled to encircle a free-moving object. This is performed using a joint-based impedance control scheme that stops the links from moving as soon as contact is detected. Multiple attempts at encircling the object are made that include placing the manipulator in a different reference position relative to the object. The contact information obtained from these attempts is preserved and added to the gathered data. This data is then passed to an algorithm called the *inverse convex hull*. The algorithm filters the contact information from the innermost links and forms a continuous shape that is representative of the object being grasped from them.

The rest of this paper is organised as follows. Section 2 details a newly proposed novel method for gathering contact information. Section 3 describes a method for analysing that contact information to model the shape and position of the object to be grasped. Section 4 describes the experimental equipment used to examine the performance of algorithms, which is demonstrated and discussed in Section 5.

2. Gathering the Contact Information

A new method inspired by the motion of an octopus’s arm has been developed that can reactively and systematically interact with the object to gain the required contact information to determine its shape for grasping. An octopus has an amazing amount of flexibility, but it only uses relatively simplistic and planar movements [12]. One method the octopus has for grasping an object begins by bringing the arm into contact with the object a small distance down from the tip of the arm. Progressively more links of the arm are then brought into contact with the object until the object is entirely encircled [9]. This method can be thought of as a combination of the *Zorro* and *Follow-the-Leader* methods. The method does not use the object’s shape information but instead relies on its octopus-like movements that can result in a partial envelopment of the object. This in turn will provide the required contact information to analyse the object’s shape, although multiple attempts at contacting the object may be required to gain enough information to accurately model the object completely. The method proposed here mimics this process and has been termed an *Octograsp* in homage to its inspiration.

In the following work, a fixed base serial link manipulator with rigid links is used to demonstrate the *Octograsp* algorithm. The algorithm initializes the serial link manipulator

by aligning all the joints to a straight line. During the wrap-around stage, only one joint of the serial link manipulator is moved at a time, until the object is completely encircled. Moving only one joint at a time may improve efficiency in terms of the amount of contacts being made, while also preventing the arm from displacing the object during the motion. This is illustrated in Figure 2(a) where all of the links have been moved simultaneously, and a gap results between the object and arm. In Figure 2(b) only the base joint has been moved causing the rest of the arm to sweep across the environment. This method results in a flat contact between the object and arm.

A feed-forward computed torque controller using the Recursive Newton Euler dynamics with a secondary PID controller for disturbance rejection is used to decouple the joints and achieve the desired torques. The manipulator itself is controlled using a joint-based impedance controller that is described in detail in Section 5. It has been shown in previous work [13] that the same methods outlined in this work operate well without the additional torque sensors; they are used here to improve the performance of the impedance controller.

2.1. Detecting the Contacts. The joint angles can provide valuable information on the perimeter of an object. At each instant in time, the desired and actual positions of the joints are known. If the error between them breaches a normal operating error threshold, then it can be assumed that a contact has occurred and react accordingly. This type of process is called a model reference controller [14].

The initial experiments using the *Octograsp* algorithm showed that although the object can be encircled with a snake-like arm, very limited information regarding the contacts is available with one attempt. This is especially the case when using the highly simplified experimental robot described in Section 4. To compensate for this, and to gain more information, multiple attempts at grasping can be made. Two methods are proposed towards this. (i) The first involves moving the base of the arm to a different initial position relative to the object, resulting in new contact information. There may, however, be errors in the global movement or measurement of the base. (ii) The second alters the contact positions of the links by limiting the rotation of the baselink during encirclement; a reduction in the base link rotation restricts the reach of the arm and thereby repositions the links. The amount the baselink which is restricted is referred to as a *back amount* in the remainder of the paper (e.g., a back amount of 0.5 radians means the base link stopped rotating 0.5 radians from an unrestricted encirclement). The

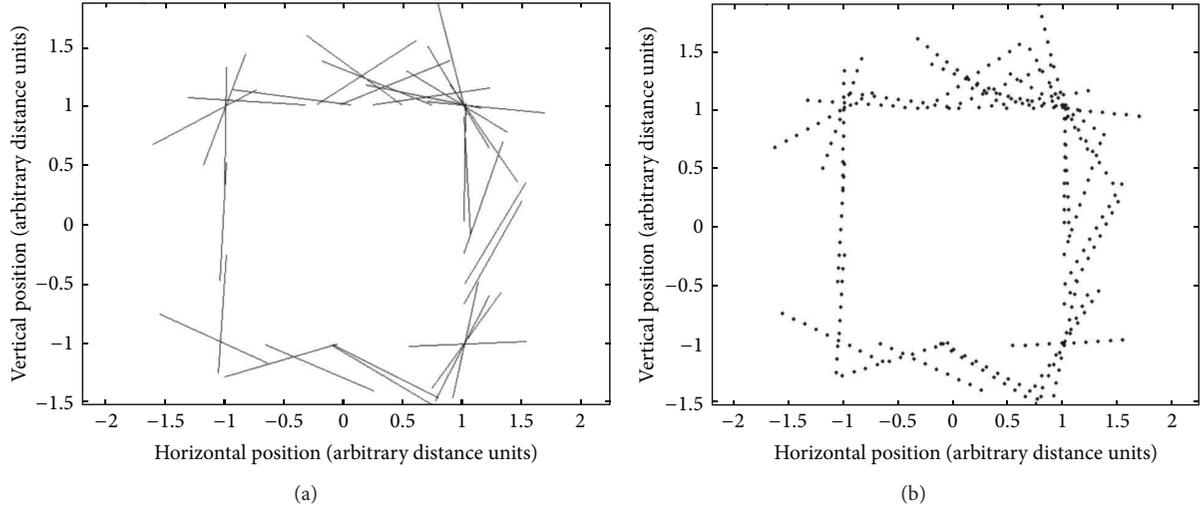


FIGURE 3: (a) Known link positions during the grasp. (b) Result of discretising link positions.

technique of restricting the baselink will be referred to from now on as the *multiple orientations* method. Once contact information has been obtained, a method is required to process this information to gain a model of the shape and location of the object.

3. Interpreting the Contact Information

A numerical simulation was created to develop algorithms to process contact information into object perimeter information. Thirty rigid independent links were randomly placed on the perimeter of a simulated square. It can be seen from Figure 3(a) that the absence of links passing through the central part of the figure suggests that the shape is square. A method will now be described that can model the shape of an object by inferring it from this absence of links.

3.1. Inverse Convex Hull. The innermost parts of the links represent the object circumference. An algorithm is required to identify the contact points and convert them into an object perimeter. The commonly used convex hull algorithm produces a perimeter that encloses a set of points. The problem described here is the opposite to a convex hull, as a perimeter that contains no points (links) is required. This problem can be solved using the inverse convex hull algorithm [15], which selects a set of points that forms a shape that contains no points. The inverse convex hull algorithm first inverts the distance of discrete points from an origin, following which the convex hull of that set is found. The origin point needs to be located within the object, ideally as close to the centre as possible. In this work, the mean of the contact locations gives the origin, since the mean returns a position within the object as long as the arm contacts the object evenly across its entire surface. The corresponding points from the original set are the inverse convex hull set. Since the inverse convex hull processes discrete points, it requires discrete link positions. This involves converting the continuous links into a set of individual points. The number

of points that each link is converted into, that is, the level of discretisation, is a user-controlled parameter. Figure 3(b) shows the discretisation of the links, using 10 points per line.

Each of the i points is represented as an n -dimensional vector (\mathbf{P}_i) that specifies its position relative to an origin; the origin needs to be located within the object. In the 3D case, \mathbf{P}_i would be represented as

$$\mathbf{P}_i = [x \ y \ z]^T. \quad (2)$$

The distance of each of the points from the origin (r_i) is

$$r_i = \sqrt{\mathbf{P}_i \cdot \mathbf{P}_i}. \quad (3)$$

To invert that distance, such that the distant points are now the closest and vice versa, the distance is raised to a negative power ($-A_{\text{exp}}$) which is another user-controlled parameter. The inverted scalar distance (d_i) is calculated as

$$d_i = r_i^{-A_{\text{exp}}}. \quad (4)$$

Finally, the direction of each point needs to be preserved such that only the distance is inverted. The convex hull is then generated from those inverted data points $\mathbf{P}_i^{\text{inv}}$, where

$$\mathbf{P}_i^{\text{inv}} = d_i \frac{\mathbf{P}_i}{|\mathbf{P}_i|}. \quad (5)$$

This process has been applied to all of the points in Figure 3(b) with the result shown in Figure 4. Each of the points are at the same angle as they originally were, but the points that were the closest to the origin are now the furthest and vice versa. The shape that this conversion forms depends upon the value of A_{exp} . Low values, such as 0.05 as used in this example, tend to squeeze the points together forming a circular shape (as seen in Figure 4).

The final step involves finding the convex hull of the inverted points and matching them to the corresponding points within the original set. Figure 5 shows the original

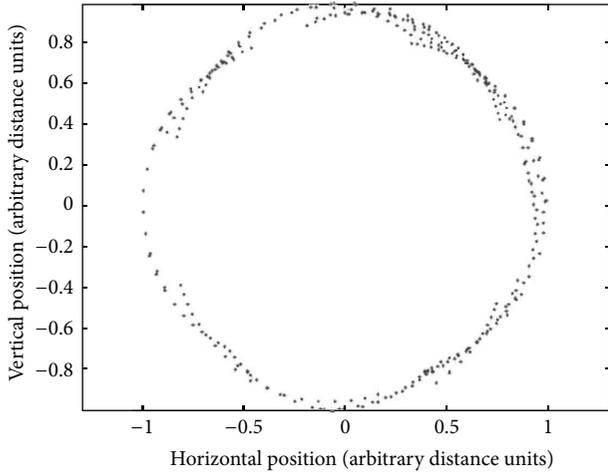


FIGURE 4: The discretised points after their distance from a centre point has been inverted.

discretised points and the polygon that is the inferred shape of the object. It can be seen that the inferred object, which is a square, is not 100% accurate, due to the lack of contacts at certain points along the length of the square. Therefore, the aim of the algorithm is to gain as much surface contact over the object as possible. However, even with full contact, the algorithm has two problems. Firstly, even with perfect information, an amount of rounding at corners will occur, the precise amount depends on the value of A_{exp} . Secondly, modelling of concave objects will be troublesome as the inverse convex hull algorithm tends to smooth over them. Further discussion of these problems and their consequences are presented later in this section.

3.2. Inverse Convex Hull Parameters. The two user-controlled parameters of the inverse convex hull algorithm, level of discretisation and A_{exp} , will each have an effect on the quality of the resulting object shape generated. Investigations were conducted to examine the effects of each parameter and to determine their optimal values. Three shapes were used in each of the investigations: an equilateral triangle, a square, and a regular pentagon. A circle of diameter 2 units encloses the shapes and therefore lengths of their sides were 1.7321, 1.4142, and 1.1756 arbitrary distance units, respectively. Equation (6) is used to measure the accuracy by returning a percentage error (Err) with zero being the best and lowest possible outcome as follows:

$$\text{Err} = 100 * \frac{A_{gen} + A_{orig} - 2A_{int}}{A_{orig}}, \quad (6)$$

where A_{orig} is the area of the actual shape, and A_{gen} is the area of the generated shape, A_{int} is the area of the intersection between the actual and generated shapes. The area of intersection between two shapes is difficult to calculate for the case where the two shapes can be any size and shape. Therefore, an algorithm [16] was used that breaks the objects into very small discrete elements that are analysed for whether the shapes overlap, intersect, or do not overlap in that

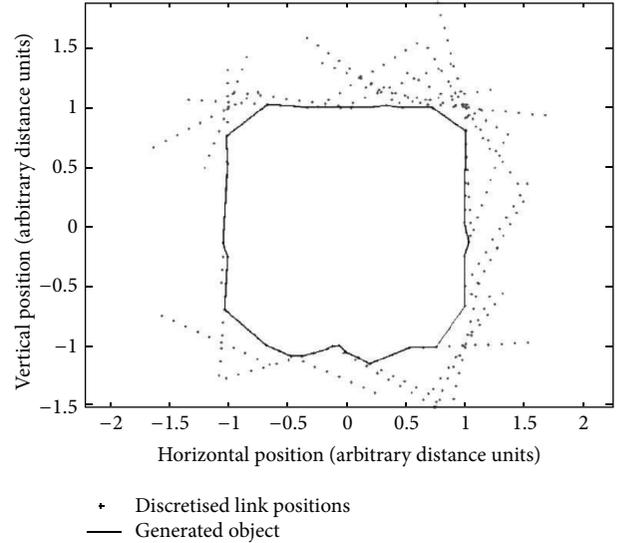


FIGURE 5: The points making up the object's shape are determined by finding the corresponding points making-up the convex hull of the points in Figure 4.

element. The sum total of the area of those that overlap and intersect approximates the area of overlap. Decreasing the size of the discrete elements increases the accuracy of the resulting intersection calculation.

3.2.1. Investigating the Discretization Parameter. The set of previously mentioned shapes was analysed with the inverse convex hull algorithm several times with varying levels of discretisation. To isolate the discretization parameter from errors in measured input data, the shapes themselves were used as the input to the algorithm instead of the data obtained from robot contact in simulation. A_{exp} was held at a constant value of 0.05 for the duration of this experiment.

The percentage area error for the three shapes when the level of discretisation varies between 10 and 1000 points per segment of the shape is shown in Figure 6. The area error is very low for all shapes and discretisation levels since perfect information is being fed to the inverse convex hull algorithm. It can be seen that the error is quite variable in the region between 10 and 200 points but settles down to a consistent low value beyond that region. A value of more than 100 and less than 300 points per segment is therefore recommended since lower values are not consistent, whilst higher values increase the amount of data that needs to be processed. Figure 7 shows an approximately linear correlation between the level of discretisation and the computation time. This is confirmed by the fact that the objects with fewer sides consume lesser time during computation.

3.2.2. Investigating the A_{exp} Parameter. The effect of varying A_{exp} (see (4)) while keeping discretization level constant at 100 was investigated, with the same three objects serving as input to the inverse convex hull algorithm. Figure 8 shows this effect on the area error for values between 0.05 and 1. The graph's trend in general shows an increase in error as A_{exp} is

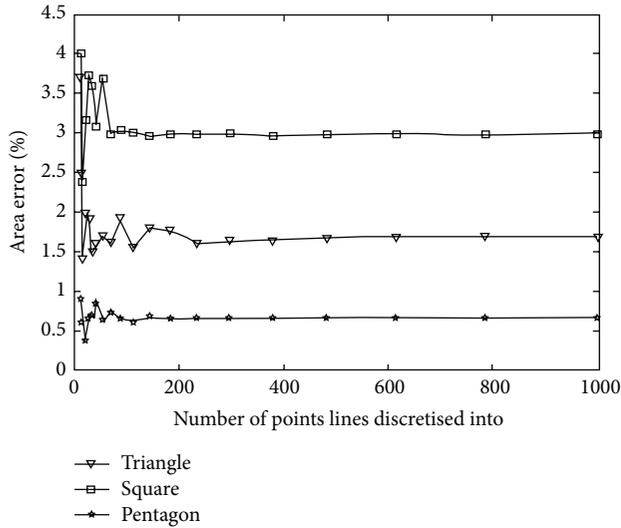


FIGURE 6: The effect of discretisation level on area error when A_{exp} is fixed at 0.05.

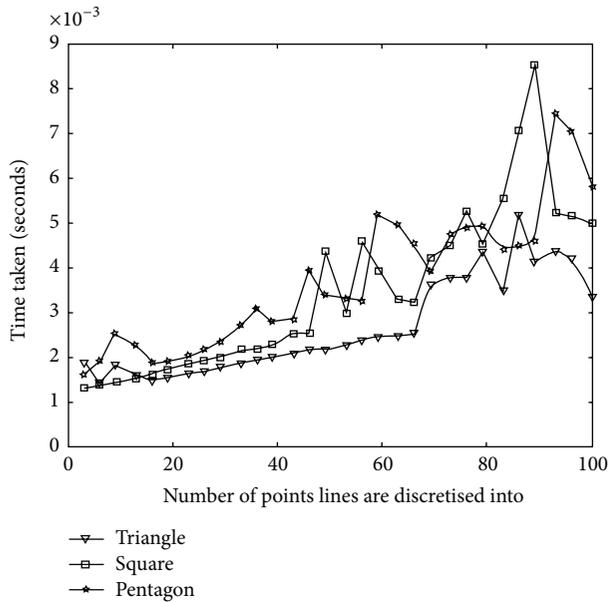


FIGURE 7: The effect of discretisation level on computation time.

increased. This is due to an increased amount of rounding at corners.

4. Experimental Platform

The proposed algorithm was evaluated using experimental equipment. This stage is crucial as a large proportion of work in the area of multilink snake-like robots for manipulation is restricted to simulations due to ideal assumptions and technology demands. The experimental equipment is designed to demonstrate whole-arm grasping with coarse contact information. The manipulator consists of four links and contains minimal sensors. The hardware has been designed to be modular such that it can be reconfigured with a minimal

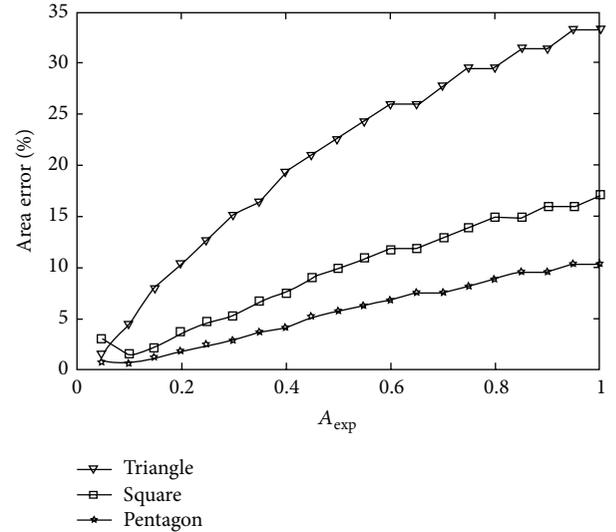


FIGURE 8: The effect of A_{exp} on area error.

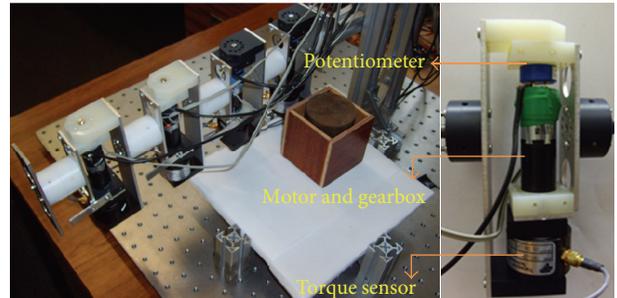


FIGURE 9: The experimental setup.

effort and allows for the possibility of correction if a particular configuration is found to be lacking. The experimental manipulator (shown in Figure 9) consists of four identical modules with each module consisting of a motor, gearbox, potentiometer, and torque sensor. The control of the arm and the acquisition and processing of contact data are performed using LabView on a standard desktop PC.

The results obtained from the experiments are 2D, although the algorithm can process 3D information by exploring multiple planes using one of the two methods explained in Section 2. 2D information may be sufficient for grasping purposes; the only disadvantage is that potentially stronger grasps in alternative planes will not be discovered. However, there is generally little difference between the strength of good and optimal grasps [17].

5. Joint-Based Impedance Control

Individual joints are controlled by an impedance controller. Impedance control attempts to implement a dynamic relationship between manipulator variables such as force and end point position, rather than controlling each of these variables alone [18]. A conventional impedance controller [19] defines a relationship between the position and velocity

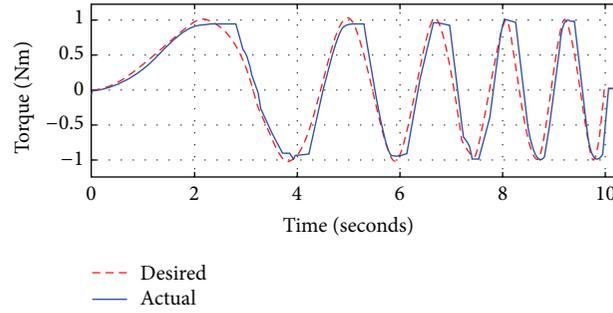


FIGURE 10: Closed-loop torque control.

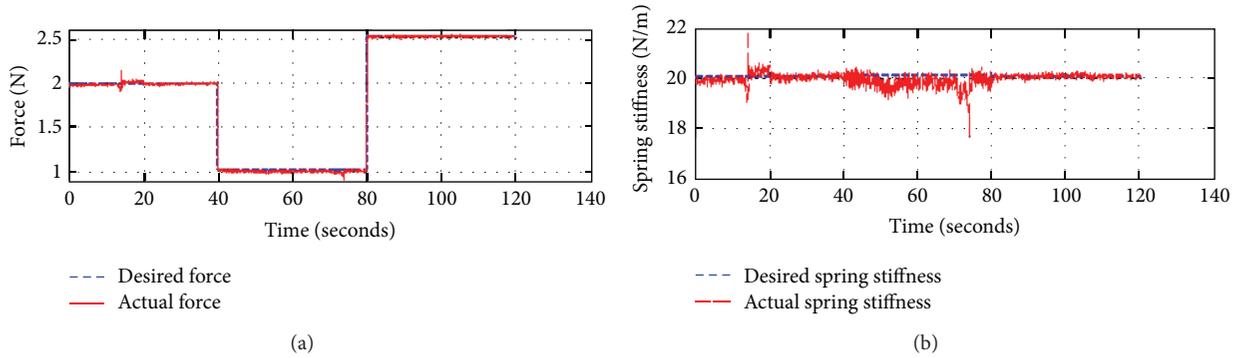


FIGURE 11: Experimental verification of the target impedance with spring constant of 20 N/m.

error of the end-effector and the force the end-effector applies (and therefore the torques at each joint). The joint-based impedance controller operates instead in the joint space allowing for a dependable contact behaviour at the cost of having to manually calculate a trajectory. This is a biologically inspired scheme that is capable of mimicking muscular compliance. As the error in tracking a certain response deviates from the expected behaviour, the impedance scheme reduces the amount of torque applied, resulting in reduction of overall torque of the joint. Hence less force is exerted by the link tip, and this reduces the risk of displacing the object. Furthermore, having such a scheme in place allows for smooth transition between the contact and not contact states and enables smoother actuator loads with minimum jerk.

Under ideal conditions, the computed torque controller enables joint space trajectories to be tracked very closely. Closed-loop torque control response of one of the joints to a chirp signal of magnitude 1 Nm, varying in normalized frequency between 0 and 0.05 Hz, is shown in Figure 10. The desired torque is accurately tracked with some phase lag and friction effects.

A closed-loop torque controller for a spring constant of 20 N/m was implemented at the joint tip. End-effector force for a single link was measured using a sensor and compared to the desired force generated by the impedance scheme. The corresponding stiffness from the relationship of force and position was computed. The results show a good agreement between the theoretical and experimental responses as seen in Figure 11.

6. Evaluation of Performance

The performance of the *Octograsp* algorithm at analysing the shape of an object was examined by varying the algorithm parameters and comparing the generated model of the object to the actual object. It must be noted that due to the limited number and size of the links, the objects being examined could not be fully encircled and touched. As a result, the evaluation of modeling will only focus on the sides being contacted. The accuracy of the generated model when multiple attempts at contacting the object are made from the same relative position (method 2 from Section 2.1(ii)) was examined. The parameters that were controlled were (i) the number of attempts at encircling and (ii) the amount the base joint of the robot moves back after each attempt (back amount). A square object was positioned and orientated the same for each run of the experiment with camera-based tracking used to verify the exact location. The inverse convex hull algorithm was used to generate a model of the object from the obtained contact information and compared to the known location and shape of the object to find the accuracy error.

An example of the contact information provided by a typical *Octograsp* is shown in Figure 12(a). The example shows four attempts at grasping the object with a back amount of 0.1 radians. The four-end configurations of the arm (one sequence of lines with dots for each attempt) are shown in relation to the actual object (dashed lines). The path of the centre of the free-moving square object (obtained from camera-based tracking) shows the displacement of the object

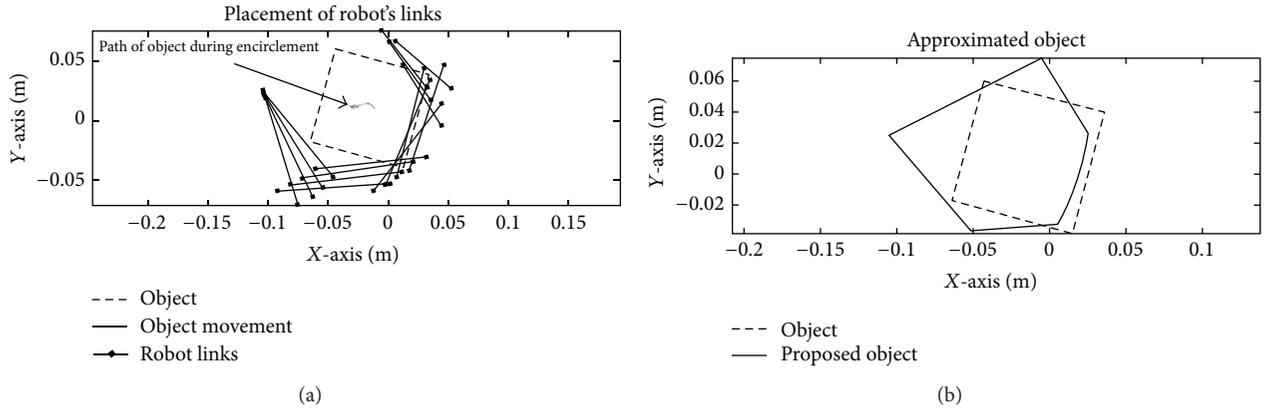


FIGURE 12: Example result from a run where 4 attempts were made with back amount of 0.1 radians.

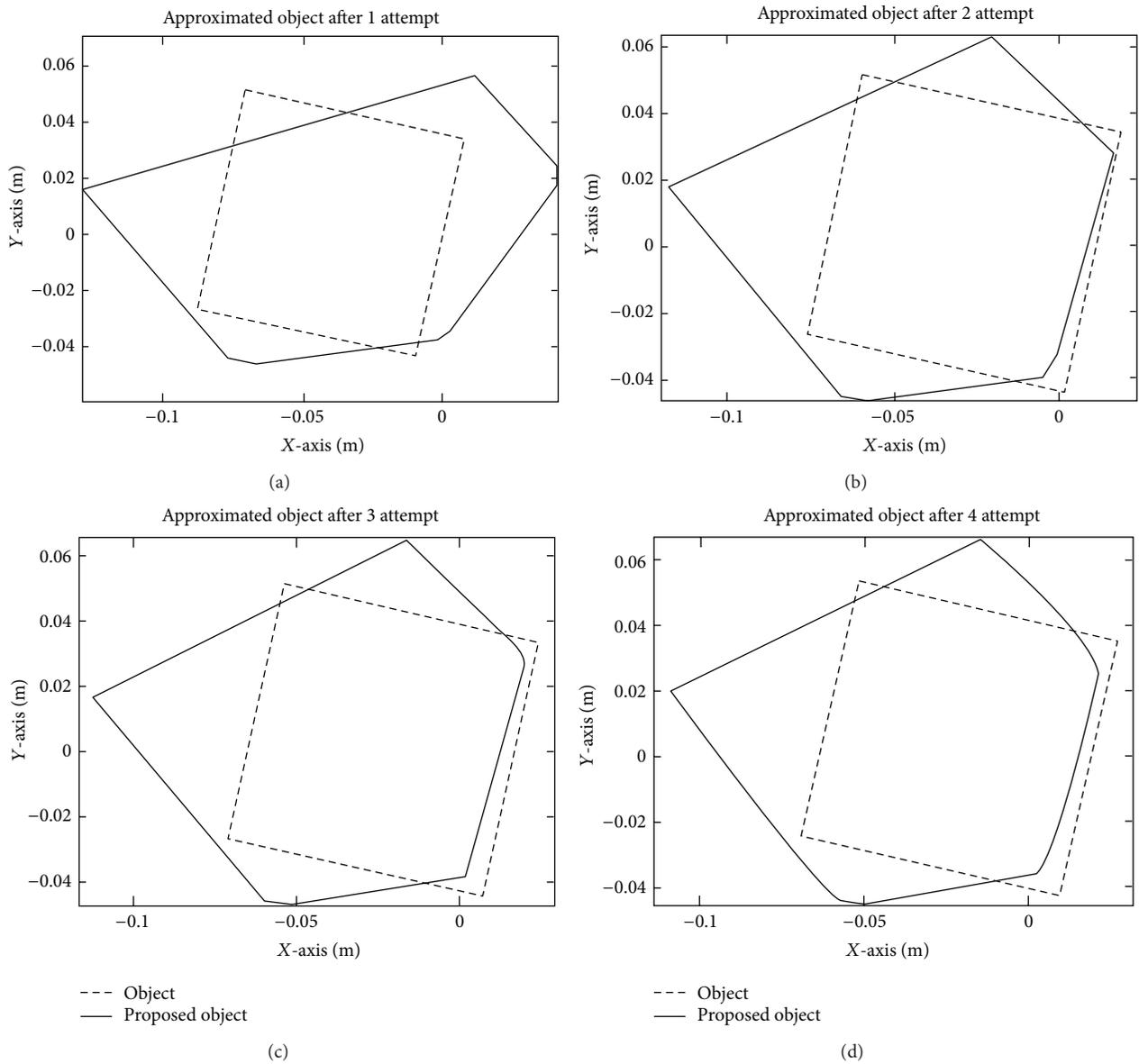


FIGURE 13: The shape being more accurately represented as more attempts are made.

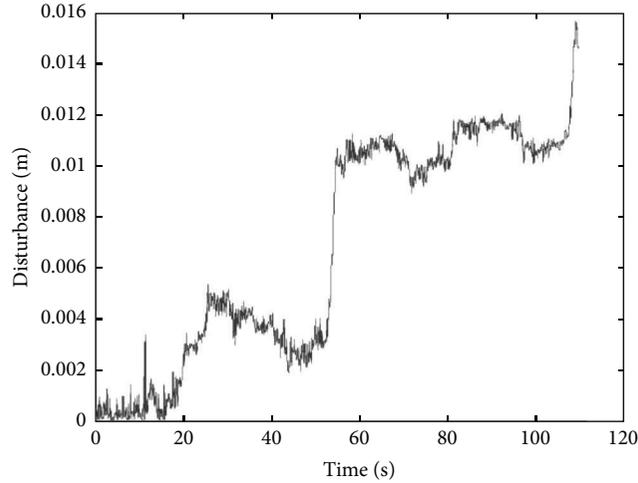


FIGURE 14: Displacement of object during first four encircling attempts from the first position.

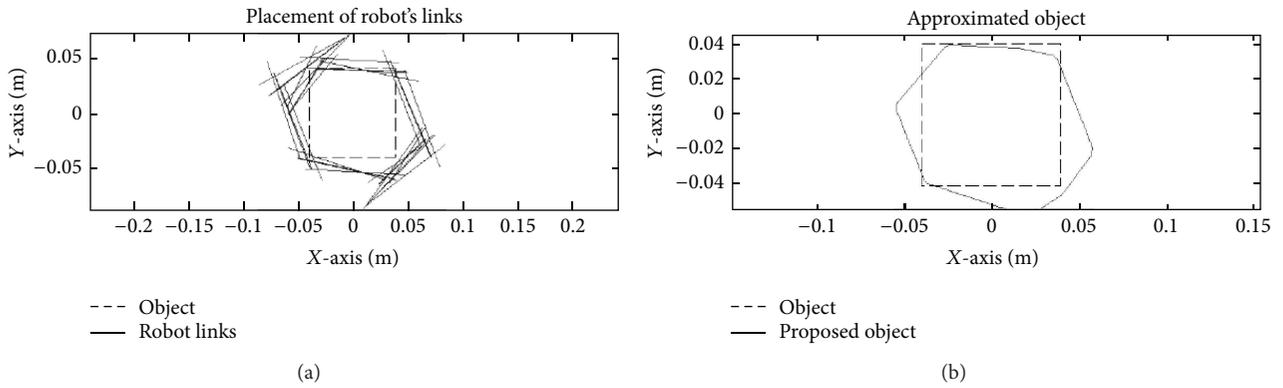


FIGURE 15: Link positions for all attempts during the grasping experiment.

over time due to contact. The major disturbance to the object is caused when the link of the arm fully coincides with a face of the square. As a result, later attempts by link 4 pass through where the square shape should be positioned, since the square is no longer present having been pushed by previous attempts. The resulting model generated by the inverse convex hull algorithm is shown in Figure 12(b).

Figures 13(a)–13(d) show how the shape is progressively built up and improved upon after each successive attempt. The displacement of the object causes the right hand side to be underestimated while the lack of contacts on the other sides causes those sides to be overestimated in size; the overestimation arises from a lack of information that could be readily corrected through a longer snake arm or multiple robot positions. The important result is that the side of the square encircled accurately represents the objects original shape.

The object was encircled from two different base positions, with the second position rotated by π radians from the first position. There were four attempts per position with a back amount of 0.1 radians. It was found that the accuracy error reduced significantly after each position change attempt, dropping by about 30%. Analysing the object from

the second position greatly improves upon the accuracy and compensates for the displacement of the object.

The object disturbance over time for the first four encircling attempts is shown in Figure 14. Each contact produces a small but noticeable increase in object disturbance. At roughly 55 seconds, there is a prominent spike that is the cause of the increase in model error. The overall disturbance is however less than 16 mm after four encircling attempts. This is a significant result for the proposed joint-based impedance control scheme given that the robot weighs 1.672 Kg and the object only weighs 200 g.

The information provided by the eight attempts at contacting the object and the resulting model produced by that contact information are shown in Figure 15. This result illustrates that the accuracy of the produced model can be increased (in comparison to results from Figure 12) by contacting the object multiple times from different positions and orientations. Object displacement can be limited by making the manipulator lightweight in comparison with the object and also by reducing the manipulator's speed of operation. The amount of contact information gathered can be improved by increasing the number of links and reducing their size.

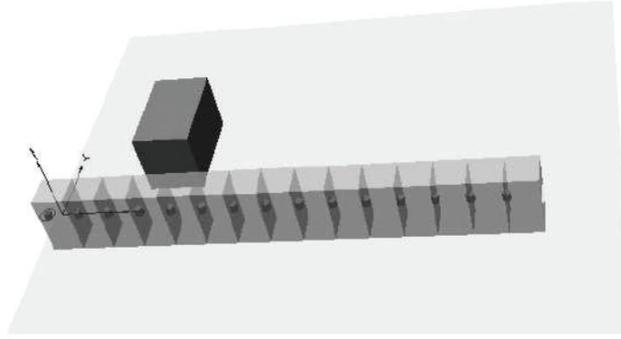


FIGURE 16: Simulation setup of manipulator with several small links grasping an object.

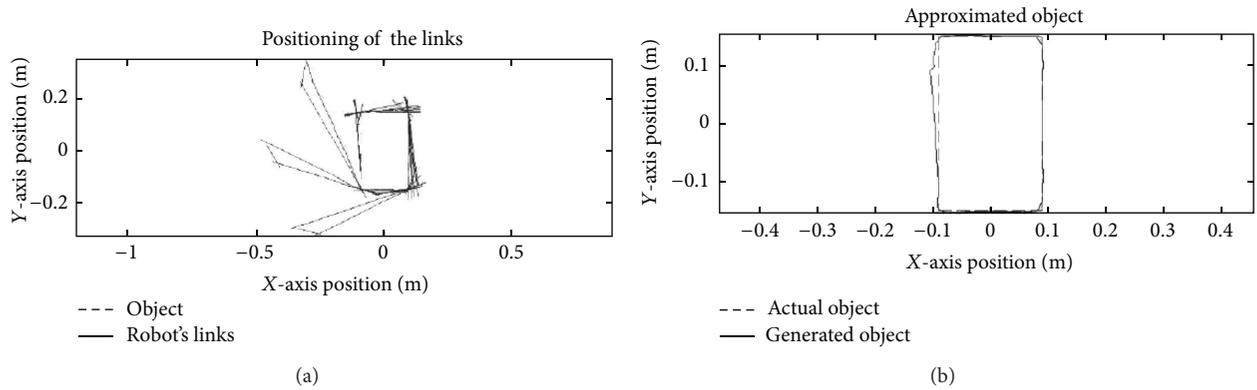


FIGURE 17: Placement of links and object representation in simulation.

6.1. Improvements through Increased Number of Links. To demonstrate the effectiveness of the algorithms under representative conditions, a simulated robot that consists of greater number of smaller links was analysed. The simulation setup is shown in Figure 16. The dynamic simulation environment was created in *MSC Visual Nastran 4D* and controlled using *Matlab-Simulink*.

The *Octograsp* algorithm was used to encircle the cuboid object from three positions, each rotated by $(\pi/4)$ degrees from each other. Two attempts at encircling the object with a back amount of 0.7 radians were made in each position for a total of 6 attempts. As before, the resulting contact information was processed using the inverse convex hull algorithm to determine the object's shape and compared with the known shape, position, and orientation of the object. The results were similar to that of the experimental situation in that there is a significant decrease in representation accuracy error when encircling from the second position (reduced from 19.9% to 9% on the third attempt). The final error for this experiment is only 4% and demonstrates that an increased number of links of reduced size provides for higher resolution data.

Figure 17 shows the contact information provided by each encircling attempt and the resulting object representation from the simulation. Having several small links in comparison to the length of the side of the object allows the arm to contact much more of the straight sides of the object in parallel, which in turn provides a large amount

of useful contact information that can produce an accurate representation of the object.

7. Conclusion

In this paper, methods for building a model of an object from tactile information have been presented. The *Octograsp* algorithm allows snake-like robots to gain contact information regarding the shape of an object. The inverse convex hull algorithm can then process that contact information in order to determine the object's shape and position. Both of the methods require no external sensors, such as touch sensors, but also do not preclude their use. The method is generic enough to work on any kinematic serial chain (snake-like) robot. In general, the *Octograsp* and inverse convex hull algorithm can generate accurate models of objects without the need for sophisticated sensors. The algorithms were implemented experimentally as well as in simulation, with the experimental kit providing coarse information and demonstrating the feasibility of the method for real-world application, while the simulation was used to demonstrate the accuracy of the method based on drawbacks identified with the experimental equipment.

It was found that the best approach was to select as many positions around the object as possible but only make two attempts in each position, one where direct contact is made and another with as large a shift in the base joint as possible.

This ensured that the largest amount of different surface contact is made, with minimal object disturbance.

A drawback of the proposed approach is that concave objects cannot be accurately represented with these methods. However, these concave elements are often not required to obtain a good grasp and may in some cases be too small in comparison to the robot's size to fully utilise. The second issue is that of object disturbance; a minor disturbance has a negative effect on the resulting accuracy of the object representation.

References

- [1] J. R. Napier, "The prehensile movements of the human hand," *The Journal of Bone & Joint Surgery—British Volume*, vol. 38, no. 4, pp. 902–913, 1956.
- [2] K. Salisbury, "Whole arm manipulation," in *Proceedings of the 4th Symposium on Robotics Research*, pp. 183–189, 1988.
- [3] D. Kirkpatrick, B. Mishra, and C.-K. Yap, "Quantitative Steinitz's theorems with applications to multifingered grasping," *Discrete & Computational Geometry*, vol. 7, no. 1, pp. 295–318, 1992.
- [4] F. Asano, Z.-W. Luo, M. Yamakita, and S. Hosoe, "Dynamic modeling and control for whole body manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, vol. 4, pp. 3162–3167, October 2003.
- [5] M. Shahinpoor, "Ionic polymer-conductor composites as biomimetic sensors, robotic actuators and artificial muscles—a review," *Electrochimica Acta*, vol. 48, no. 14–16, pp. 2343–2353, 2003.
- [6] E. Biddiss and T. Chau, "Electroactive polymeric sensors in hand prostheses: bending response of an ionic polymer metal composite," *Medical Engineering & Physics*, vol. 28, no. 6, pp. 568–578, 2006.
- [7] C. Busch, *Whole-arm grasping with hyper-redundant planar manipulators using neural networks [Ph.D. dissertation]*, Vorarlberg University of Applied Sciences, 2002.
- [8] W. McMahan, B. Jones, I. Walker, V. Chitrakaran, A. Seshadri, and D. Dawson, "Robotic manipulators inspired by cephalopod limbs," in *Proceedings of the CDEN Design Conference*, Montreal, Canada, July 2004.
- [9] I. D. Walker, D. M. Dawson, T. Flash et al., "Continuum robot arms inspired by cephalopods," in *Unmanned Ground Vehicle Technology VII*, vol. 5804 of *Proceedings of SPIE*, pp. 303–314, March 2005.
- [10] R. Desai, C. Rosenberg, and J. Jones, "Kaa: an autonomous serpentine robot utilizes behavior control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems: Human Robot Interaction and Cooperative Robots*, vol. 3, pp. 250–255, August 1995.
- [11] D. Reznik and V. Lumelsky, "Multi-finger "hugging": a robust approach to sensor-based grasp planning," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, pp. 754–759, May 1994.
- [12] Y. Y. Yekutieli, S. G. German, F. T. Tamar, and H. B. Binyamin, "How to move with no rigid skeleton? The octopus has the answers," *Biologist*, vol. 49, no. 6, pp. 250–254, 2002.
- [13] D. Devereux, P. Nutter, and R. Richardson, "Reactive and pre-planned encirclement for whole arm grasping," in *Proceedings of the 7th IEEE International Conference on Industrial Informatics (INDIN '09)*, pp. 668–673, June 2009.
- [14] P. J. McKerrow, *Introduction To Robotics*, Addison Wesley, 4th edition, 1998.
- [15] F. D. Snyder, D. D. Morris, P. H. Haley, R. T. Collins, and A. M. Okerholm, "Autonomous river navigation," in *Mobile Robots XVII*, vol. 5609 of *Proceedings of SPIE*, pp. 221–232, October 2004.
- [16] P. Koprowski, "areaintersection.m," <http://www.mathworks.com/matlabcentral/fileexchange/15557-areaintersection-m>, 2007.
- [17] D. Devereux, *Control strategies for whole arm grasping [Ph.D. thesis]*, The University of Manchester, Manchester, UK, 2010.
- [18] N. Hogan, "Impedance control: an approach to manipulation," in *Proceedings of IEEE American Control Conference*, pp. 304–313, 1984.
- [19] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated hands," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 105, no. 1, pp. 33–41, 1982.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

