The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Modeling & Characterizing Stochastic Actuator Arrays

David L. MacNair and Jun Ueda

Abstract— **This paper presents a method of modeling and subsequently characterizing stochastically controlled actuator arrays. The actuator arrays are built from cells, each containing six piezoelectric actuators and an amplification structure; however the results can be generalized to actuator arrays using solenoids, pneumatic cylinders, or any other fast acting linear actuators. The cells are controlled using an all-on all-off Bi-stable stochastic process wherein all cells are given a common input probability (control) value which they use to determine whether to actuate or relax. Laying out the cells in different networks gives different actuator array properties, which must be found before the actuator arrays can be applied to manipulators. A method is provided to calculate actuator array properties such as: travel, required actuator strength/displacement, force range, force variance, and robustness for any actuator array configuration. Finally the properties of several illustrative examples are shown and a discussion covers the importance of the properties, trends between actuator array layouts and their properties, and the course of future work in the area.**

I. Introduction

As humanoid and human force amplification robotics research progresses, the robotics community demands artificial muscles or biologically-inspired actuators with increasing force capacities, force accuracy, reaction speed, and on-board sensing/calibration capabilities. In recent years, new actuators and control techniques have been developed using shape memory alloys [1][2], pneumatic actuators [3][4], and piezoelectric materials [5][6]. Each actuator technology has strengths in particular areas, e.g. pneumatic actuators tend to have higher force capacities while piezoelectric devices tend to have faster actuation frequencies. Since applications may call for one type of linear actuator over another, control strategies which can utilize the different actuator types interchangeably can decouple higher-level controller programming from actuator design. One such control strategy places actuators in cellular units with spring-type structural elements, and then builds actuator arrays with these cells laid out in different configurations. A bi-stable stochastic control process in which cells are either fully off or fully on at any point in time [6] can then be implemented regardless of the actuator type chosen. This is implemented by placing many cells in an actuator array and giving a basic logic circuit and random number generator to each. All cells receive a common input probability value and the cells decide to actuate (constrict) or relax by comparing the input probability with the random number, as shown in Fig. 1. Over time, the actuation and relaxation of the cells gives a mean output force which is directly related to the input probability. Additionally, allowing the cells to be either fully on or off greatly reduces any hysteresis error individual actuators my otherwise exhibit.

This paper provides a novel method for calculating properties of stochastically controlled actuator arrays including: travel, required actuator strength/displacement, force range, force variance, and robustness. Since infinite configurations exist for the actuator arrays, having this direct method for calculating properties is essential. This paper focuses on actuator arrays built from piezoelectric actuators, however the same properties and calculations can be easily extended to systems using different actuator types such as pneumatic cylinders, solenoids, or other fast acting linear actuators so long as the same control method is used.

II. Actuator Array Model and Important Properties

A. Actuator Array Cell Model

Each cell in an actuator array consists of a stack of piezoelectric actuators surrounded by a nested displacement amplification structure [7]. The piezoelectric actuator stack moves from a relaxed length to a shorter actuated length when activated and back to its relaxed length when deactivated. Assuming the external force acting on the cell does not overpower the actuator, the relaxed and actuated lengths are constant. The amplification structure deforms according to the actuator displacement and forces applied externally to the cell. As long as the amplification structure is kept in the linearly elastic portion of the stress strain curve, the structure acts as a linear spring. Activating the piezoelectric actuator stack preloads the amplification structure (spring). If the cell sees no external force, this will cause it to shrink to the cell's actuated unforced length.

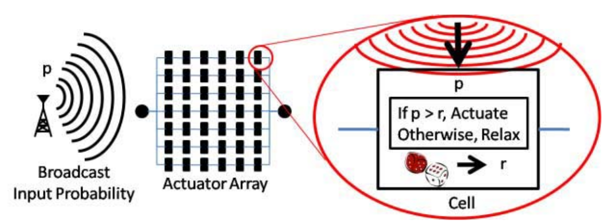

Fig. 1. Bi-Stable Stochastic Control Method.

Manuscript received March 1, 2009.

D. L. MacNair and Jun Ueda are with the Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: David.Macnair@gatech.edu, Jun.Ueda@gatech.edu).

Alternatively, if the length of the cell is held constant, this will cause a force on the external constraint directly proportional to the change in length of the piezoelectric actuator stack. For this reason, each cell is modeled as a spring with a pure force generator acting in parallel. Representations of the actual cell, the model of the cell, and the cell's movement are shown in Fig. 2.

B. Actuator Array Model

Any one dimensional actuator array can be represented using four different element types, each having a certain number of variables, equations, and constants. The element types are shown in Table I and Fig. 3, and are described below:

1) Node: A node (N_i for i=1,2,…,n where n is the number of nodes) is used to connect the other elements types and to track the position and force along the actuator array. Node N_1 and node N_n (the first and last nodes) must be on the ends of the actuator array and can only have one attached element each. These are used to represent the connection to the external environment. Any other node can be attached to two elements, one on the left and one on the right. Nodes have two variables. The position of N_i is $N_{i,x}$ and the force of N_i is $N_{i,f}$. Nodes have no equations or given values.

2) Cell: A cell (C_j for j=1,2,…,z where z is the number of cells) has one variable, the displacement from their relaxed unforced length (d_j). They also have three equations: one to relate the positions of the nodes on either side of the cell and two to relate the forces of the nodes on either side of the cell. Cells have three given values: the cell spring constant (k_j), the cell relaxed unforced length (X_j), and the pure force generator force (F_j). Assuming all cells have the same force capability (F), F_j = F when the cell is active and F_j = 0 when the cell is relaxed.

3) Spacer: A spacer (S_m for m=1,2,…,y where y is the number of spacers) is used to represent a constant length in the actuator array. They have no variables but have 2 equations, one to relate the positions of the connected nodes and one to make the forces of the two connected nodes equal. Spacers also have one given value, the length of the spacer (q_m) where m is the spacer number and varies from 1 to the number of spacers.

4) Expander: Expanders connect a single node on one side to multiple nodes on the other side without adding length to the system. This allows multiple cells to run in parallel, amplifying the force capacity and increasing redundancy of the actuator. Expanders have no variables or given values, but have n equations where n is the number of nodes connected by the expander. n–1 of the equations make the position of all of the nodes equal. The final equation adds the forces on the larger side of the expander and sets the result equal to the force on the smaller side. If a physical expander has a certain length, a spacer can be connected in series on the smaller side of the expander.

C. Important Actuator Array Properties

1) Number of Cells: As the number of cells increases, the cost increases, the power requirement increases, and the actuator array has a larger volume and mass. Increasing cells also generally increases the actuator array displacement and/or force capacity and decreases the normalized variance.

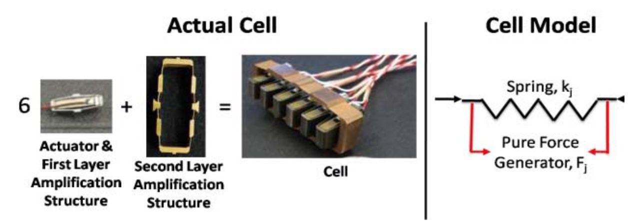

Fig. 2. Cell Model.

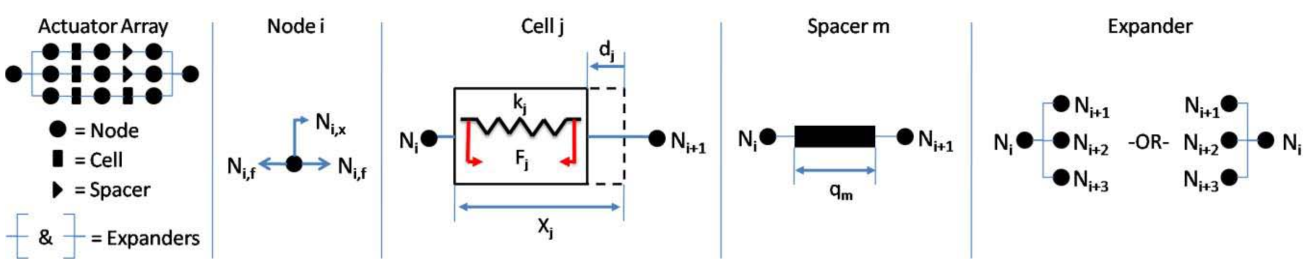

Fig. 3. Actuator Array Elements.

Table I. Actuator Array Elements

Element Type	Variables	Equations	Constants
Node i	Position ($N_{i,x}$) Force ($N_{i,f}$)	None	None
Cell j	Displacement (d_j)	$N_{i+1,x} - N_{i,x} - d_j = X_j$ $N_{i,f} - k_j(d_j) = F_j$ $N_{i+1,f} - k_j(d_j) = F_j$	Spring Constant (k_j) Unforced Length (X_j) Pure Force Generator Force (F_j)
Spacer m	None	$N_{i+1,x} - N_{i,x} = q_m$ $N_{i+1,f} - N_{i,f} = 0$	Length (q_m)
Expander	None	$N_{i,x} - N_{i+1,x} = 0$ $N_{i,x} - N_{i+2,x} = 0$ $N_{i,x} - N_{i+3,x} = 0$ … $N_{i,f} - N_{i+1,f} - N_{i+2,f} - N_{i+3,f} - \ldots = 0$	None

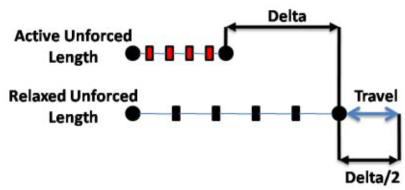

Fig. 4. Actuator Array Travel.

2) Actuator Array Travel: For the purposes of this paper, the actuator array displacement is considered to be half the difference between the relaxed unforced length and the active unforced length. The relaxed unforced length is the length of the actuator array when all cells are relaxed and no external force is applied; this is also considered the minimum length. Similarly, the actuated unforced length is the length when all cells are active and no external force is applied. This length is shorter than the minimum length. The actuator array travel spans from the relaxed unforced length (minimum actuator array length) to this plus the displacement, as shown in Fig. 4. This assumption is explored further in the discussion use of actuator arrays in antagonistic pairs.

3) Force Function: This is a function of the probability input and the current actuator array length which yields the force an actuator array will provide. For a given length, a command of 0% input probability will give the minimum possible force in the actuator array, a command of 100% will give the maximum possible force, and the command to mean-force relationship is linear between the two. A controller uses this function to achieve a desired mean force output.

4) Required Actuator Force/Displacement: Each piezoelectric or other type of linear actuator must have at least a certain force capacity, or required actuator force, in order to ensure it is able to actuate fully when loaded. Each actuator must also have a certain displacement which it will move to, but not beyond, whenever activated. For the purposes of this paper, it is assumed that there are no compression forces in the actuator array. This is also considered good design practice.

5) Variance Function: This is a function of the input probability value and the current actuator array length which yields the expected variance in the output force for an actuator array. While the mean force output will remain constant for a constant input probability value, the actual force output will vary over time. The larger the force variance, the farther from the mean value the force is likely to be at any point in time. Variance will lead to a greater potential for positioning error with open-loop control.

6) Robustness: The worst case failure of a cell is a break; meaning F_j and k_j for the cell are always zero. This renders the broken cell and all cells connected in series to the broken cell useless. Two robustness measures were developed to characterize actuator arrays in terms of robustness. The "Minimum Cell Loss to Uncontrollability" defines, in the worst case scenario, how many cells would have to break to have zero controllable force capacity. The "Worst Failure Force Function" is the force function an actuator array can achieve after breaking a given number of its most critical cells (the cells which results in the lowest achievable total force once lost). These two measures are shown in Fig. 5. So long as forces on the actuator array are below this value, the actuator array will be able to function and have time to cope after the break of any cell.

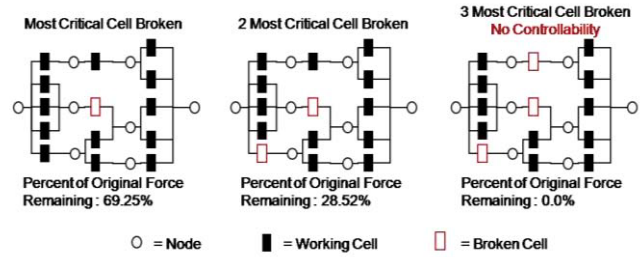

Fig. 5. Robustness Measure. "Minimum Cell Loss to Uncontrollability" = 3.

III. Actuator Array Analysis

A. Relationship Generation

To analyze the properties of the actuator arrays, the element equations are arranged into three systems of linear equations, or relationships, which are then used to solve for internal variables. Each relationship consists of an A matrix containing the coefficients of the internal variables in the element equations, a B vector containing the internal variables themselves, and a C vector containing the given values (the right hand side of each equation). The system of linear equations is then solved using (1).

$$B = A^{-1}C \tag{1}$$

1) Force Relationship: The force relationship identifies all node forces given that certain cells are active and the actuator array has a given overall length (X_{tot}). This relationship uses the element equation exactly as they appear in Table I but also adds (2) and (3), where node n is last node in the system. In the cell equations, the values of constant F_j are filled in as either F or 0 based on whether the cell is currently active or relaxed. Equation (1) solves for B, the internal variables of the system. The force output for the actuator array is equal to the force in node 1, a component of vector B.

$$N_{1,x} = 0 \tag{2}$$

$$N_{n,x} = X_{tot} \tag{3}$$

2) Controllability Relationship: The controllability relationship is used to determine if an actuator array has any controllable force capacity. The relationship is the same as the force relationship except for two differences.

First, F_j the pure-force generator force in the cell equation is moved to the left side of the equation for all cells. Second, (4) is added setting the force in node 1 and thus the actuator array output force equal to 1. If the A matrix is full rank, the actuator array can change the exerted force by activating or relaxing cells. Fig. 5 shows examples of an uncontrollable actuator arrays.

$$N_{1,f} = 1 \tag{4}$$

3) Displacement Relationship: The displacement relationship is used to determine the unforced relaxed length and unforced actuated length of the actuator array. The relation is the same as the force relationship except that (2) is changed to (5) which sets the force in node 1 and thus the actuator array output force equal to 0. When run with all cells relaxed, this will give the unforced relaxed length and when run with all cells activated it will give the unforced activated length of the system.

$$N_{1,f} = 0 \tag{5}$$

B. Property Calculations

The *Actuator Array Travel* is the first property calculated for each actuator array. This is generated by using the *Displacement Relationship* to calculate the unforced relaxed length and unforced activated length. As stated above, the actuator array travel spans from the relaxed unforced length to this plus the displacement, as shown in Fig. 4.

The *Force Function* and *Force Variance Function* are calculated using A_f, B_f, and C_f which are the A, B, and C matrices from the *Force Relationship*. C_f can be separated into a vector containing the force components (G) and a vector containing all of the other components (H) as shown in (6). G contains two duplicate force entries for each cell, and each cell's pure-force can be modeled as an independent Bernoulli trial multiplied by the pure-force capacity of the cell (F_j) as shown in (7), where r is the random value generated by the cell and p is the input probability as well as the expected value of the Bernoulli trial. The mean, or expected value, of B is calculated using (8) [8].

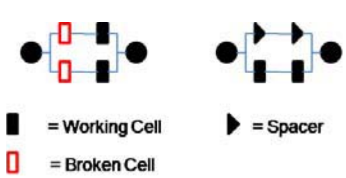

Fig. 6. Uncontrollable Actuator Arrays.

$$C_f = \overset{C_f}{\begin{bmatrix} c_1 \\ \vdots \\ c_j \\ F_1 \\ F_1 \\ c_{j+3} \\ \vdots \\ c_n \end{bmatrix}} = \overset{G}{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ F_1 \\ F_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}} + \overset{H}{\begin{bmatrix} c_1 \\ \vdots \\ c_j \\ 0 \\ 0 \\ c_{j+3} \\ \vdots \\ c_n \end{bmatrix}} \tag{6}$$

$$F_j = \begin{Bmatrix} 1 \cdot F & if\,(p > r_j) \\ 0 & if\,(p \le r_j) \end{Bmatrix} \tag{7}$$

$$\mathrm{E}[B_f] = A_f^{-1} \cdot \begin{bmatrix} F_1 \cdot p \\ \vdots \\ F_j \cdot p \\ \vdots \\ F_z \cdot p \end{bmatrix} + A_f^{-1} \cdot H \tag{8}$$

Since each of the Bernoulli trial is independent, the variance of B can be calculated using (9) [8].

$$\mathrm{Var}[B_f] = A_f^{-1} \cdot \begin{bmatrix} F_1 \cdot p \cdot (1-p) \\ \vdots \\ F_j \cdot p \cdot (1-p) \\ \vdots \\ F_n \cdot p \cdot (1-p) \end{bmatrix} \tag{9}$$

The *Force Function* is $\mathrm{E}[N_{1,f}]$, an entry in E[B]. Likewise the *Variance Function* is $\mathrm{Var}[N_{1,f}]$, an entry in Var[B]. The *Force Function* is a linear function which shows that the mean force will increase linearly with increasing input probability and/or actuator array length. The *Variance Function* is a quadratic function with roots at p = 0 and p = 1, and a maximum at p = 0.5. Given that a certain actuator array topology has already been chosen, a designer can find the *Required Actuator Force/Displacement* for a desired force function (τ) by first finding a unit force function (σ) for the actuator array given all $F_j = 1$. A scaling factor (υ) can be found using (10). The *Required Actuator Displacement (RAD)* for each cell is then calculated using (11). The scaling factor can then be plugged in for all F_j in the force relationship to find the force in the nodes connected to each cell, which is the *Required Actuator Force* for each cell.

$$\upsilon = \frac{\tau}{\sigma} \tag{10}$$

$$RAD_j = \frac{\upsilon}{k_j} \tag{11}$$

The “Minimum Cell Loss to Uncontrollability” and the “Worst Failure Remaining Force” are calculated using a combination of the *controllability relationship* and the *force relationship*. First the system is input into the *force relationship* with all cells active and with the actuator array at its minimum travel. The most critical cell is determined as the cell carrying the highest force. This cell is then broken and the resulting actuator array is checked for controllability using the *controllability relationship*. If the actuator array still has a controllable force, it is once again fed into the force relationship with all cells active to determine the “Worst Failure Force Function.” This process is repeated until the actuator array is uncontrollable. The number of breaks required to make the actuator array uncontrollable is the “Minimum Cell Loss to Uncontrollability.”

IV. Examples

A. Relationship Generation

In order to normalize the results and reduce computation time, several constraints were placed on the actuator arrays.

1) The force output for each actuator array was constant across all simulated topologies. This allowed the variances of the different topologies to be compared directly since the variance would otherwise scale with the output force.

2) All cells in all actuator arrays had the same cell spring constant ($k_j = 1$) and cell relaxed unforced length ($X_j = 2$).

The solution methods presented above are valid when these constraints are not present, however the constraints help determine the correlation between network topologies and actuator array properties.

B. Results and Graphs

Fig. 7 shows five actuator arrays which are analyzed below. Table II shows the Number of Cells, Actuator Array Travel, and Required Actuator Force/Displacement for the five different actuator arrays. Fig. 8 shows the force PDFs for actuator array D. This graph shows the different forces which can be immediately achieved along with the probability that each point is reached given an input probability. Fig. 9 shows the force function, which is the same for all of the actuator arrays due to constraint 1. The actuator arrays will have a force capability from 0.0 to 1.0 when at their minimum travel and a force capability from 0.5 to 1.5 when they are at their maximum travel.

Fig. 10 shows the force variance curves for all of the actuator arrays.

Table III shows the minimum cell loss to uncontrollability and the percentage reduction between the original force function and the “worst failure force function” after breaking one cell.

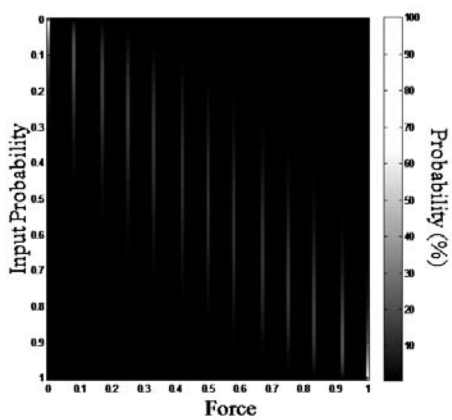

Fig. 8. Force PDF for Actuator Array D.

Table II. Example General Properties

Topology	A	B	C	D	E
Number of Cells	4	4	4	4	10
Actuator Array Travel	2 – 2.125 units	8 – 10 units	4 – 4.5 units	4 – 4.667 units	6 – 6.5 units
Required Actuator Displacement	0.250	1.000	0.500	0.667	0.333
Maximum Required Actuator Force	0.250	1.000	0.500	1.000	0.500

Table III. Example Robustness Properties

Topology	A	B	C	D	E
Minimum Cell Loss to Uncontrollability	4	1	2	1	2
% Reduction After Losing Most Critical Cell	25.0 %	100.0%	33.3%	100.0%	33.3%

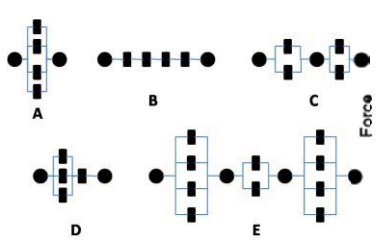

Fig. 7. Example Actuator Arrays.

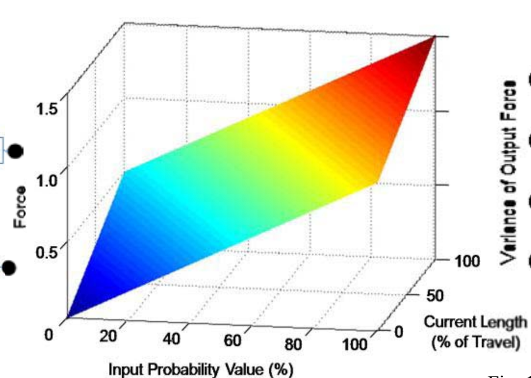

Fig. 9. Example Force Function.

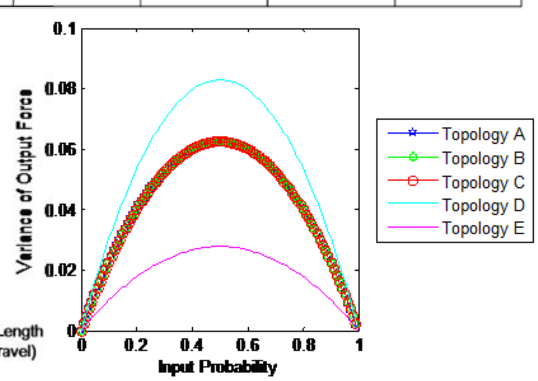

Fig. 10. Example Variance Functions.

V. Discussion and Conclusion

A. Force Function

The linearity and constant slope for each actuator array allows a controller directly command a desired mean force as long as the minimum and maximum forces at the endpoints of the actuator array's travel have been identified and the controller has knowledge of the actuator array's current length. Additionally, combining a displacement sensor with the actuator array, the current belief of a manipulator and its payload can also be continuously updated by higher level controllers for more accurate manipulation. When combined with a force sensor, the actuator array can also be continuously calibrated so that it remains robust and accurate even despite multiple cell failures.

B. Variance

The variance of the muscle is a measure of the difference between the force commanded and the force delivered by an actuator array at any given point in time. Since the cells change position rapidly, this each instantaneous error provides only a small error in the overall impulse delivered and averages out quickly. Increasing the number of cells in an actuator array generally lowers the normalized variance; however variance also scales up with increasing force capacity. Cells which uniformly carry the internal forces generally have a lower variance.

C. Antagonistic Pairs

Since actuator arrays are used to provide tension forces, antagonistic muscle pairs are needed to provide forces in opposite directions. When an actuator array has the travel stated above, the force capability at the end of its travel is from 0.5 to 1.5. If the same actuator array is used in an antagonistic pair, the other muscle will have a force capacity at the same point of 0 to 1. When combined, this gives an overall force capability of 1.5 toward the stretched actuator array and 0.5 toward the shorter actuator array, as shown in Fig. 11. If the actuator arrays is allowed a farther travel, the force capability toward the contracted muscle would go to zero. Since multiple combinations of input probability values in the antagonistic pair can lead to the same force output, a least squares optimization can be applied between the two muscles to arrive at the input probability values which will result in the least overall variance. The application of this is left to future research.

D. Robustness

Two robustness criteria were introduced in this paper, the minimum cell loss to controllability failure and the worst failure remaining force. Both of these measures become useful when an actuator array failure is critical to the application of the manipulator. A simplified example is a

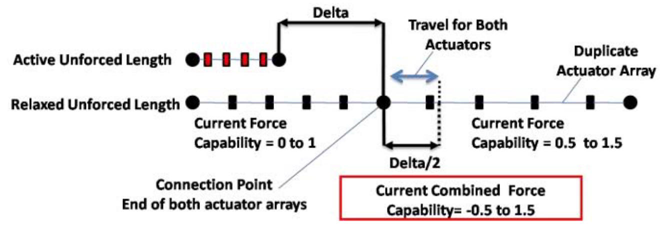

Fig. 11. Antagonistic Pairs.

manipulator hammering an object inside a radioactive area. The arm must provide large forces to swing the hammer. If cells are lost, the arm may not be able to continue swinging the hammer, but will still be fully capable of putting the hammer down, cleaning up the area, and moving to where it can be easily repaired. Generally, parallel structures increase the number of redundant force paths in a manipulator which in turn increases both robustness properties.

E. Future Research

Future research aims of this project are split into short-term and long-term goals. Short-term goals include applying the muscles to dynamic simulations of manipulators with various degrees of freedom, developing higher level controllers for these manipulators, and experimentally testing cells and muscles to ensure they match calculated results. Long-term goals include applying learning algorithms and sensor fusion techniques to give a robot an accurate representation of the external environment and its internal configuration, and applying the overall system to high degree of freedom robotic applications such as humanoid robotics and exoskeletons.

References

[1] Srinivasan, A., McFarland, D., Canistraro, H., and Begg, E. (1997). Multiplexing embedded NiTi-NOL actuators to obtain increased bandwidth in structural control. *Journal of Intelligent Material Systems and Structures*, 8(3):202-214.

[2] Fu, Y., Du, H., Huang, W., Zhang, S., and Hu, M. (2004). TiNi-based thin films in MEMS applications: a review. *Sensors and Actuators. A, Physical*, 112(2-3):395-408.

[3] Sanchez, A., Mahout, V., and Tondu, B. (1998). Nonlinear Parametric Identification of a McKibben Artificial Pneumatic Muscle Using Flatness Property of the System. *Proceedings of the 1998 IEEE International Conference on Control Applications*, 1998. 1.

[4] Caldwell, D. (2002). Biomimetic Actuators in Prosthetic and Rehabilitation Applications. *Technology and Health Care*, 10(2):107-120.

[5] Selden, B., Cho, K., and Asada, H. H. (2006). Segmented Shape Memory Alloy Actuators Using Hysteresis Loop Control. *Smart Materials and Structures*, 15(2):642-652.

[6] J. Ueda, L. Odhner, and H.H. Asada, "Broadcast Feedback of Stochastic Cellular Actuators Inspired by Biological Muscle Control," *The International Journal of Robotics Research, Special Issue on Bio-Robotics,* pp. 100-109, February 2007.

[7] J. Ueda, T. Secord, H.H. Asada, "Design of PZT cellular actuators with power-law strain amplification," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 1160-1165, November 2007.

[8] William W. Hines, Douglas C. Montgomery, David M. Goldsman, Connie M. Borror; *Probability and Statistics in Engineering (Forth Edition),* Wiley, 2003.