Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Design of Vast DOF Artificial Muscle Actuators with a Cellular Array Structure and Its Application to a Five-Fingered Robotic Hand

Kyu-Jin Cho, Josiah Rosemarin and Harry Asada

d'Arbeloff Laboratory for Information Systems and Technology
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
kyujin@mit.edu, josiah@mit.edu asada@mit.edu

***Abstract* – This paper presents a cellular actuator design for a robotic hand. Although the motions of robotic hands are complex, a design of cellular actuator segmentation can be simplified by extracting the features of given postures and using them to design and control the actuator. A method of using captured hand posture data to design the actuator segmentation is proposed. Data from the eight most frequently used hand grips in a daily living, as defined in the sollerman hand function test, is used in this design. The gathered joint angle data is transformed into actuator displacement data and used to generate a segmentation design of the actuator. For segmentation design, feature extraction method, called non-negative matrix factorization with constraints is proposed. The actuator system has 12 SMA actuators each with eight segments, resulting in 96 configurable segments before the coupled segmentation design is applied. The coupling segmentation design effectively groups the segments into clusters which are simultaneously controlled. The segmentation design reduces 96 separately controlled segments to 8, while maintaining the ability to accomplish all desired postures. A robotic hand with five fingers, designed and fabricated using the FDM process, is driven with this actuator system, and eight hand postures are reproduced with the robotic hand.**

I. INTRODUCTION

There is an increasing need for new actuator technology in humanoids, prosthetics, and other robotic systems that have many degrees of freedom. An actuator system using new artificial muscle actuators [1] is likely to have a different architecture than those with traditional actuators.

Asada et al. [2] have proposed a new architecture for multi-axis artificial muscle actuators that assimilates the morphological structure of biological muscles. The architecture consists of a two-dimensional array of segments that are streamlined and coordinated to control multiple bones or links of a robot.

Fig. 1 illustrates an artificial muscle actuator system having a cellular array structure. The rectangular blocks represent cellular building blocks that can be actuated individually and that are connected in series and parallel to move specific points along a linkage of bones. Unlike traditional electric motors, which can rotate infinitely in the absence of mechanical stoppers, the displacement of a muscle actuator is finite. The total displacement of the artificial muscle is determined by the integral of strains distributed over the individual building blocks in series. The output force is the summation of the forces generated at the individual strands arranged in parallel. These output characteristics can be tailored to specific load conditions by arranging the cellular building blocks in various patterns and activating them in a coordinated manner.

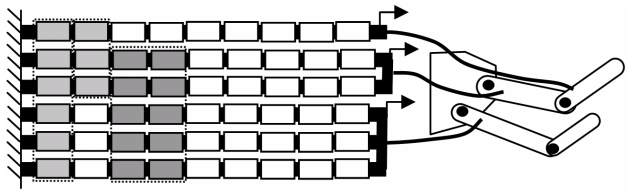

Fig. 1 Schematic of a multi-axis artificial muscle actuator array

Unlike today's robots with bulk actuators rotating specific joints, biological systems have networks of muscles, each consisting of a number of cellular building blocks forming a hierarchical structure. Although it is not realistic to exactly mimic such a complex hierarchical structure, it is worth exploring how robotic actuators can be made as an ensemble of small building blocks and how they can be coordinated to generate multi-body motion of a vast number of degrees of freedom.

In an attempt to explore an alternative to today's robot actuators, this paper is concerned with the design and architectural issues of multi-axis actuators with a cellular array structure. We attempt to build small-scale, artificial muscle actuators consisting of many segmented units; each can be activated as an individual building block, and a group of units can be coupled and coordinated to generate multi-axis motion. We will address design and architectural issues for a particular class of tasks and actuator material, with the specific goal of

0-7803-9505-0/06/$20.00 ©2006 IEEE

developing multi-axis array actuators tailored to performing coordinated posture control of a five-fingered robotic hand.

Fig. 2 Picture of the Robotic Hand Prototype used to design and test the multi-axis array segmentation architecture

II. Actuator Segmentation Design

The purpose of the segmentation design is to reduce the number of independently controlled segments and group the segments into clusters that are controlled simultaneously. After the segmentation design, the set of postures can be performed with the robotic hand by activating a combination of different sets of clusters.

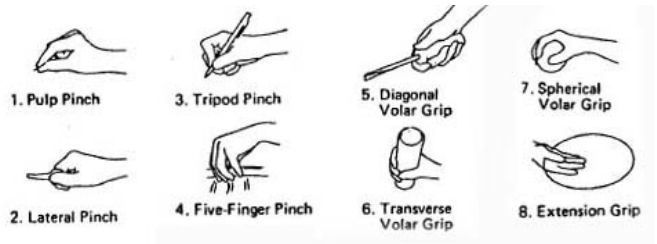

Fig. 3 Eight main hand grips into which a normal grip pattern can be divided [3]

In this paper, hand postures to be performed by the robotic hand are chosen from the grips used in a standard hand function test, which is used to assess and compare the result of a hand surgery. The eight most common hand grips used in 20 activities of daily living is defined in the sollerman hand function test [3]. These eight grips are shown in Fig. 3. The details of these grips are explained in the literature. The grips are numbered from 1 to 8, with the most frequently used grip numbered 1 and the least frequently used grip numbered 8.

A. Segmentation Design algorithm

Once the set of postures to be performed is defined, the segmentation design algorithm can be applied to the actuator displacement data. The actuator displacement data is then transformed from the finger joint angle data, which can be gathered using a data glove.

The problem of designing the segmentation can be formulated as follows:

We start with the desired output of m actuators for n different postures or modes. This can be given as a data set of m by n matrix M. Each row represents the actuator displacement of each axis, and each column represents the actuator displacement for each hand posture. The elements of this matrix are all nonnegative, since it represents the output of a one-way actuator, i.e. SMA actuator. In order to design segmentation with reduced dimensionality, the problem can be reduced to finding two matrices, W and H, such that $M \approx W*H$. The W matrix contains information about how the segmentation is done, and the H matrix contains the encoding, or control information. In Fig. 4, an example showing the relationship between the W matrix and the actual segmentation design is depicted. As shown in the figure, the W matrix has 7 rows and 12 columns, meaning that there are 7 axes and 12 segments. Each segment is labeled *W1* to *W12*, which corresponds to columns of W matrix. For example, the 12^{th} column of W, $[0\ 0\ 0\ 0\ 2\ 2\ 2]^T$, represents a segment which covers axis 5, 6, 7 with a length of 2.

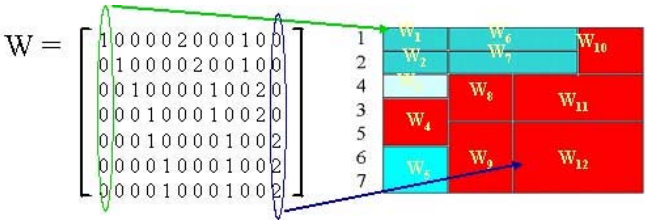

Fig. 4 Example showing a relation of a column of W matrix with an actual segmentation design

Each segment can be designed based on the values of each column. Each column of the W matrix represents which actuators and what length of each actuator the segment will be activating. On the other hand, the H matrix is a binary matrix that represents which segments should be turned on to generate a certain posture; hence it's called an encoding matrix.

The feature extraction method, called Nonnegative Matrix Factorization (NMF) with binary constraint on H matrix is applied to generate segmentation design (matrix W) and encoding (matrix H) data from the actuator displacement data (matrix M).

Non-negative Matrix Factorization (NMF) has been proposed as a tool to find a set of basis functions for representing non-negative data.[4] The notion is particularly applicable to image articulation libraries made up of images showing a composite object in many articulations and poses. When used in the analysis of such data, NMF would find the intrinsic 'parts' underlying the object being pictured. NMF has also been applied to extract invariant spatiotemporal components, or synergies, from the simultaneous recordings of the activity of many muscles [5]. The algorithm is as follows:

The cost function used to quantify the quality of the

approximation can be constructed using the square of the Euclidean distance between M and WH. This is lower bounded by zero and becomes zero when M = WH.

$$\|M - WH\|^2 = \sum_{ij} (M_{ij} - (WH)_{ij})^2 \qquad (1)$$

The following update rules are used to update the W matrix and H matrix iteratively. The Euclidean distance is non increasing under these update rules.

$$W_{ia} \leftarrow W_{ia} \frac{(MH^T)_{ia}}{(WHH^T)_{ia}} \qquad H_{aj} \leftarrow H_{aj} \frac{(W^T M)_{aj}}{(W^T WH)_{aj}} \qquad (2)$$

The Euclidean distance is invariant under these updates if and only if W and H are at a stationary point of the distance. The above update law is derived by choosing the step size of the gradient descent method, such that the update law of the gradient descent method becomes a multiplicative one. The proof of convergence is also given in the literature. The update laws have been directly applied to the current problem of designing actuator segmentation.

In order to apply the above algorithm to the actual prototype of the actuator system, additional constraints that meet the design constraints must be considered and implemented.

The flow chart of the algorithm with constraints is shown in Fig. 5. Multiple iterations are done with different initializations of W and H matrices, since different initializations can lead to different solutions. The additional constraint of making the H matrix a binary matrix is applied after each update of the W and the H matrix according to the update rules (2). The idea is to filter the H matrix so that the values of the matrix elements go to either one or zero. For this purpose, sigmoidal function shown below is used as a filter.

$$S(x) = \frac{1}{1 + e^{-100(x-0.5)}} \qquad (3)$$

Even though elements of the W matrix can be any positive number in general, there are constraints involved in the actual design. One is that since we have used off-the-shelf parts for each segment of thermoelectric devices, only certain values can be allowed. In our design, we have used multiples of same size elements, hence a multiple of 1.2 *mm* stroke is allowed. Another constraint is that since the length of segments for each axis is constrained, the sum of segments that cover the same axis should be smaller than that length. These constraints are implemented in the algorithm.

Multiple update iterations are done until the solution converges.

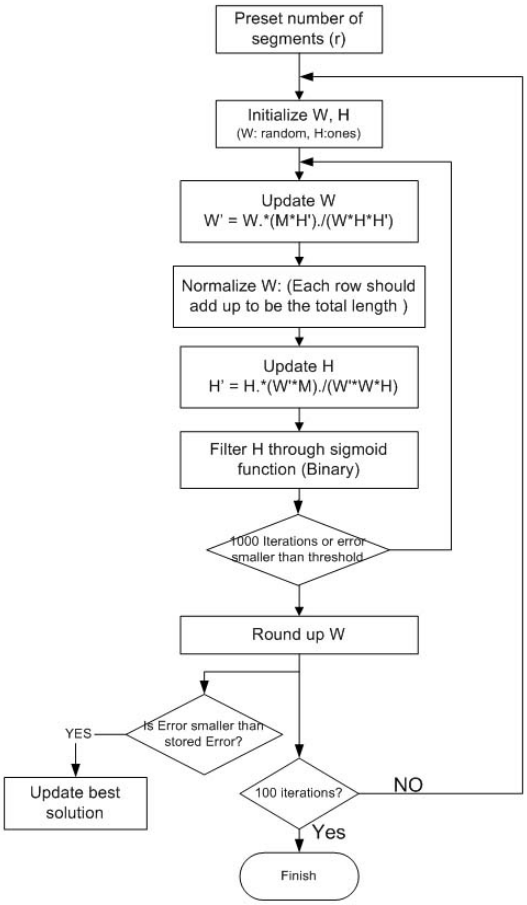

Fig. 5 Flow chart of segmentation design algorithm

B. *Discussion of the algorithm*

There is no guarantee that this algorithm will generate the optimal solution, but the generated design can be accepted with a certain amount of error. Due to the discrete size of the actuator segments, there is an error caused by the resolution of the actuator segments. The smaller the segments are, the better the actuator accuracy is. These errors, however, can be compensated by decoupling the actuators for the thumb. Since thumb plays a major role in controlling the grasps, giving it more independence and high resolution will improve the performance of the actuator system without changing the actuator design for other fingers.

III. IMPLEMENTATION

In order to implement the segmentation design to design of a robotic hand system that performs specific tasks, a few points need to be addressed. First, the data of finger joint angles for each hand posture, or specific task, needs to be reliably collected. Second the collected data should be processed with an automatic algorithm that generates the actuator segmentation design when given a set of hand posture data that the robotic hand is to perform. The authors believe that the automated process of collecting data and generating actuator design enables an actuator system to be

programmable. This concept can be applied to many different applications where a large number of artificial muscle actuators are used with a specific set of tasks. Fig. 6 shows the overview of the design procedure for multi-axis cellular actuator array, which starts with motion capture of the hand postures, segmentation design, programming or hard-wiring the multi-axis actuator system according to the segmentation design and finally attaching it to the robotic hand.

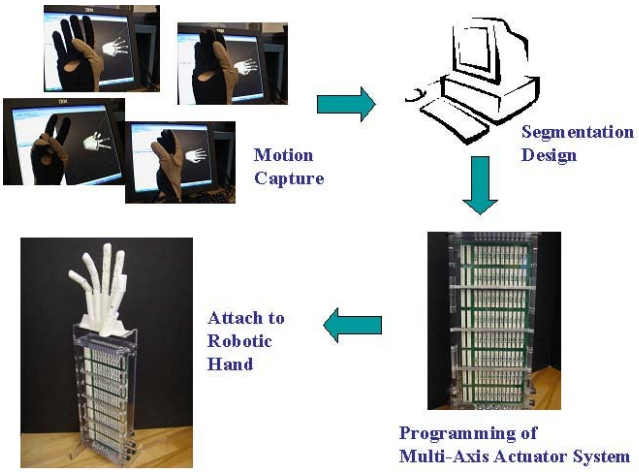

Fig. 6 Overview of the design procedure of multi-axis actuator array for driving robotic hand

A. Data collection of Finger joint angles

The design of the actuator segmentation starts with gathering the data of finger joint angles using a data-glove. A data glove from Immersion Corporation (CyberGlove®) is used to capture the joint angle data of five fingers. The data glove provides 22 joint-angle measurements to transform hand and finger motions into real-time digital joint-angle data. Data from three flexion sensors per each finger is used for the actuator design.

Each hand grip is performed after wearing the data glove, and finger joint angle data is gathered. The captured images from the data glove are shown in Fig. 7.

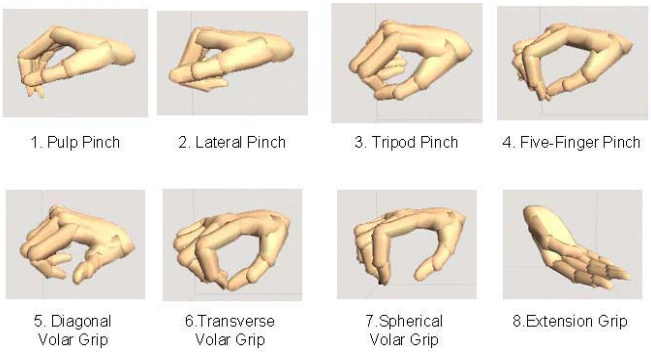

Fig. 7 Eight main hand grips captured with a data glove

B. Transformation of joint angles to actuator displacement

In order to use the collected data for the actuator segmentation design, the joint angles are transformed into actuator displacements. Fig. 8 shows the cross-sectional view of the robotic finger along with the tendons of SMA actuator and bias springs that are connected to the distal phalange (the bone close to the finger tip).

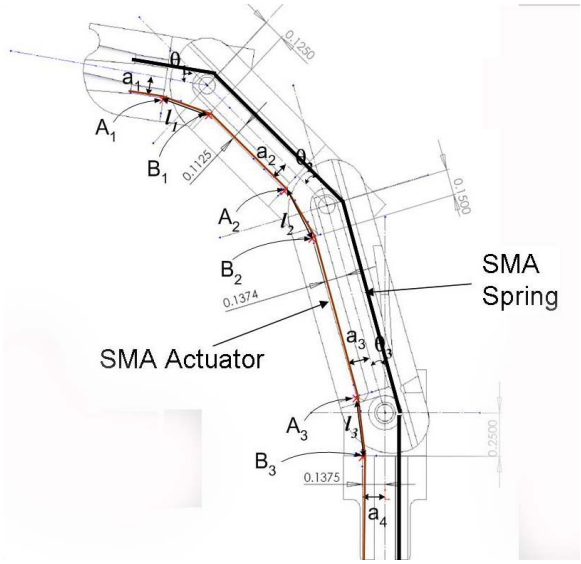

Fig. 8 Cross-section of a index finger of the robotic hand: SMA actuator is used as a flexor to bend the finger and SMA spring is used as an extensor to provide bias force that pulls the finger back to full extended posture when the SMA actuator is not activated.

The angle of each joint is related to distance between points A_i and B_i, shown in fig. 8. Given a bending angle of θ_i, the actuator displacement required is the amount of distance change between points A_i and B_i, between the case when the bending angle is 0 and when the bending angle is θ_i. The total actuator displacement is given by sum of these distance changes for all three joints. Using a rotation matrix $R_i(\theta)$, the rotated position of A_i, denoted A_i' is $R_i(\theta)*A_i$. The distance between Ai' and Bi, is l_i. Therefore, the total actuator displacement required to produce a bending angle of (θ_1, θ_2, θ_3) is given by $L_i - (l_1+l_2+l_3)$, where L_i is the sum of distance between A_i and B_i for all i's. Similar calculation can be done for the actuator controlling the other joints.

There are a few things to note about the design of the robotic finger. An SMA spring is used as an extensor to provide bias force. The finger is in a fully extended posture when the SMA actuators are not activated. SMA actuators and springs run from the end of the finger tip all the way to the end of the multi-axis actuator array. The DIP (distal interphalangeal) joint and the PIP (proximal interphalangeal) joints are coupled and under-actuated, thus there is only one actuator for driving both joints. The actuator, however, drives the DIP joint only after the actuator has driven the PIP joint to its maximum bending angle or when there is something blocking the PIP joint from rotating.

Table 1 Actuator displacement data transformed from the collected joint angle data for eight hand postures (Unit: mm)

		1	2	3	4	5	6	7	8
thumb	1st	3.30	3.03	4.44	4.97	0.97	4.77	3.77	4.65
	2nd	3.73	2.80	4.14	5.30	1.98	4.66	3.67	3.32
index	1st	3.65	4.86	4.53	3.37	2.28	2.89	2.31	4.49
	2nd	1.87	1.39	1.81	2.95	0.60	2.35	1.99	3.01
middle	1st	4.87	8.10	7.06	4.71	6.37	5.43	4.24	3.26
	2nd	1.05	2.09	2.75	1.18	0.59	2.22	1.90	2.68
ring	1st	4.49	7.98	6.46	4.73	7.22	5.59	4.00	1.88
	2nd	4.21	5.56	4.13	2.70	4.13	1.43	1.05	3.08
pinky	1st	4.61	7.93	6.49	5.09	7.47	5.82	4.23	2.12
	2nd	3.71	6.36	5.76	4.92	6.44	6.36	5.07	0.68

The transformed actuator displacement data for all eight hand postures are given in table 1. Two actuator displacements for each finger are designated due to the under-actuated nature of the finger.

C. Segmentation Design

The algorithm explained in the previous section was applied to the set of actuator displacement data in table 1. The matrices W and H in Fig. 9 are the result of the segmentation design. The W matrix represents the segmentation design, and the H matrix represents the encoding needed to produce the outputs. The H matrix is a binary matrix, containing either one or zero.

The error, defined as a frobenius norm of matrix *M-WH*, from the design was 7.6099. A total of 8 clusters of segments were used. The number of segments used was determined by iterating the process for different number of segments and selecting the design with the smallest error. The H matrix shows which cluster of segments to turn on for each posture. For example, for posture 1, segments 1 and 3 should be turned on, and for posture 2, segments 1, 3, 5, 6, and 7 should be turned on.

Segment / Axis	1	2	3	4	5	6	7	8
1	2	1	0	1	1	0	0	3
2	3	0	0	2	0	0	0	3
3	2	0	1	1	1	0	0	3
4	2	0	0	2	0	0	0	4
5	3	0	1	0	1	1	0	2
6	1	1	0	0	1	0	0	4
7	3	0	1	0	1	1	0	1
8	1	0	3	1	1	0	0	2
9	3	0	1	0	1	1	0	2
10	5	0	0	0	0	1	1	1

Posture / Segment	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	0
2	0	0	1	0	0	1	1	0
3	1	1	1	0	1	0	0	0
4	0	0	0	1	0	1	0	0
5	0	1	1	0	0	0	0	0
6	0	1	0	0	1	1	0	0
7	0	1	1	0	1	1	0	0
8	0	0	0	0	0	0	0	1

Fig. 9 (Top)W matrix of the final design, where each column represents a segment and (bottom) H matrix of the final design, where each column represents which segments to turn on for each posture

D. Robotic Hand

The designed actuator system is attached to a robotic hand with five fingers. Each finger has three joints, where two joints, the PIP and the DIP are coupled. The thumb has two joints that bend toward the palm and one joint that rotates the thumb perpendicular to the palm. Fig. 2 shows the picture of the assembled robotic hand with actuator system attached. Two SMA actuators are connected to each finger, one at the end of the finger tip, and another at the end of the bone that is closest to the palm. These SMA actuators are pulled to bend the finger at the PIP and the metacarpophalangeal (MCP) joint, the joint that is closest to the palm. The PIP joint and the DIP joint is coupled, as mentioned before. As shown in Fig. 8, an SMA spring, that utilizes super-elasticity of SMA, is used to apply bias force, instead of regular springs which would have taken up more space. When the SMA actuator is deactivated, the SMA springs pull the fingers back to its neutral position, which is all opened posture. All the parts of the robotic hand are fabricated with a rapid prototyping machine (FDM) and assembled with a steel rod that connects each joints.

E. Multi-axis Actuator System

The actuator array uses the Segmented Binary Control (SBC) [6], which segments an artificial muscle actuator into many independently controlled, spatially discrete volumes, and applies simple On-Off control to each segment. Thermoelectric modules (TEM) are used to rapidly heat and cool the segments of SMA actuators [6]. By changing the direction of the current flow, a TEM can either heat or cool a surface. Fig. 10 shows the schematic of one segment of an actuator array. An SMA actuator is sandwiched between a thermoelectric module and a thermally insulating substrate board [7].

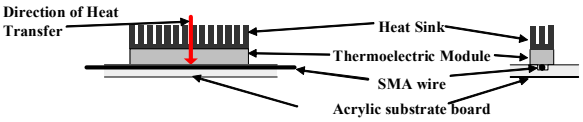

Fig. 10 Schematic of one segment of an actuator array. Thermoelectric module on a substrate board drives a segment of SMA actuator wire.[7]

Fig. 11 shows three views of the actuator system. From the front view, the bed of thermoelectric modules can be seen, which is embedded in a PCB board for easy wiring. Electric connectors are located at one side of the actuator that has one to one electrical connection with all the thermoelectric modules. 12 SMA actuators are sandwiched between transparent acrylic substrate board and thermoelectric modules. One large water-cooling heat sink is attached at the other side of the thermoelectric modules as can be seen from the back view. SMA springs run on top of the acrylic substrate board. All of the SMA actuators and SMA springs are connected at the bottom plate of the actuator system, tied to a hole on the head of the screw. Screws are used to fine tune the tension of

the SMA actuator and SMA spring. Aluminum bars are used to stiffen the acrylic substrate board, and springs between the heatsink and the outside plate apply pressure on to the thermoelectric modules, since it is important to make sure that the SMA wires are firmly in contact with the thermoelectric modules.

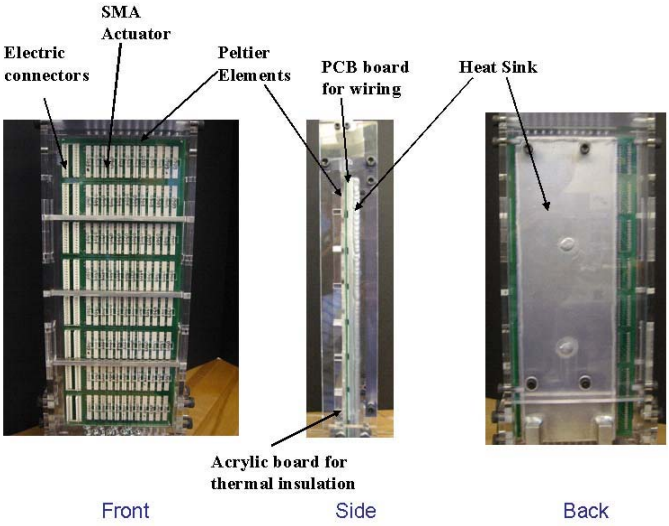

Fig. 11 Different views of multi-axis actuator array system that drive 12 SMA actuators.

Using the robotic hand prototype, the eight grips that the actuator segmentation was designed to perform can be produced. Fig. 12 shows the eight grips implemented with the robotic hand. Due to the limitation of the thumb, the postures are not exactly human-like. For example, the thumb does not rotate so that the face of the thumb can face the bottom of the other fingers, which will provide a stable grip.

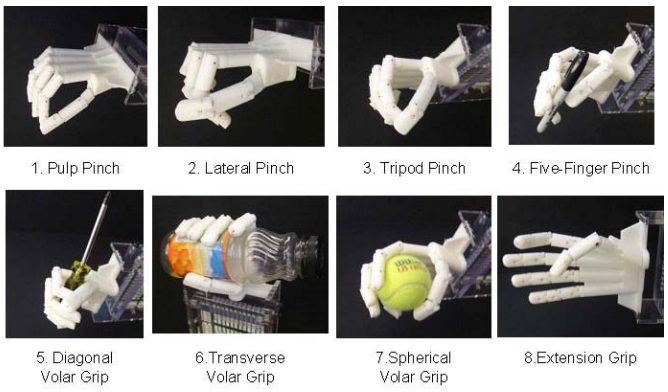

Fig. 12 Eight handgrips from the sollerman hand function test performed by the robotic hand

IV. Conclusion

In this paper, the method of using captured hand posture data to design the actuator segmentation has been presented. Data from the eight most frequently used hand grips in a daily living is used to design the actuators for the robotic hand. For segmentation design, a feature extraction method, called non-negative matrix factorization with constraints is used. With the design, the actuator can drive all five fingers into the designated postures by controlling just eight clusters of segments. A robotic hand with five fingers, designed and fabricated using the FDM process, is driven with this actuator system, and the hand postures are reproduced with the robotic hand.

References

[1] J. D. W. Madden, N. A. Vandesteeg, P. A. Anquetil, P. G. A. Madden, A. Takshi, R. Z. Pytel, S. R. Lafontaine, P. A. Wieringa, and I. W. Hunter, "Artificial muscle technology: Physical principles and naval prospects," *IEEE Journal of Oceanic Engineering*, vol. 29, pp. 706-728, 2004.

[2] K. J. Cho, H. H. Asada, "Segmentation architecture of multi-axis SMA array actuators inspired by biological muscles," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and System, pp.254-259*

[3] C. Sollerman and A. Ejeskar, "Sollerman hand function test. A standardised method and its use in tetraplegic patients," *Scand J Plast Reconstr Surg Hand Surg*, Vol.29, pp. 167-176, June 1995.

[4] D. Lee and S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401:788-791, 1999.

[5] d'Avella A, Santiel P, and Bizzi E. "Contributions of muscle synergies in the construction of natural motor behavior." Nat Neurosci 6: 300–308, 2003.

[6] B. Selden, K. Cho, H. Asada, "Segmented Binary Control of Shape Memory Alloy Actuator Systems Using the Peltier Effect, " in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 4931-4936, 2004.

[7] K. J. Cho, H. H. Asada, "Multi-Axis SMA actuator array for driving anthropomorphic robot hand," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*