Segmentation Theory for Design of a Multi-Axis Actuator Array using Segmented Binary Control

Kyu-Jin Cho, and H. Harry Asada, *Associate Member, IEEE*

Abstract— **In this paper, we present a segmentation theory for designing a Multi-Axis Actuator Array. This array uses artificial muscle actuators, i.e. shape memory alloys, to control multiple axes in a coordinated manner. The multi-axis actuator array uses Segmented Binary Control (SBC), which is a method of controlling artificial muscle actuators in a digital manner. In SBC, actuators are segmented and each segment is controlled independently, instead of controlling the strain of the actuator as a whole. The advantage of using SBC is that it allows us to avoid the nonlinear properties of the actuator and to use a simple control for each segment. However, one problem of using SBC is the increased number of segments. The segmentation theory provides a basis for using coupled segments, or segments to be shared within the actuator array, along with independent segments, to control the multiple axes with reduced number of segments. A segmentation design procedure is developed based on the theory and the method is applied to an actuator array for driving a five-fingered robot hand capable of taking variety of postures.**

I. Introduction

Multi-Axis Actuator Array is an actuator that uses Segmented Binary Control (SBC) to control multi-axis actuators in order to activate multiple links of a robot mechanism in a coordinated way. SBC is a method of controlling artificial muscle actuators by segmenting the actuator and controlling each segment in a digital manner [1]. This Segmented Binary Control uses basic characteristics of artificial muscle actuators such as shape memory alloy actuators [2-4] and electroactive polymer actuators [5,6]. Namely, the resultant motion is the integration of the strain along the whole material, or the summation of individual displacements contributed by all the segments. As far as the value of the integral remains the same, the resultant displacement does not vary although some segments have larger strains than others, or the strain is uniformly distributed along the material. Therefore, we do not have to control the strain of the material as a whole, but can generate the same total displacement by activating local segments of the material.

One drawback of SBC is increased number of control loops and drive system. Although each control loop regulating the individual segment is rather simple, many loops are needed. Interestingly, however, there is an effective method of reducing the control loops for multi-axis SMA actuators. As illustrated in Fig. 1, *M*-axes of SMA wires are laid on a two-dimensional array of local heating and cooling units. Note that this two-dimensional array of units is segmented not only in the longitudinal direction of each bundle of SMA wires but also in the transverse direction. Therefore adjacent SMA wires lay on the same segment are heated or cooled at the same time for that particular portion of the wires. This, of course, reduces independence of the adjacent SMA wires to a certain degree, but on the other hand generates a coordinated movement among them. It is also important to note that with this two-dimensional segmentation the number of control loops is significantly reduced.

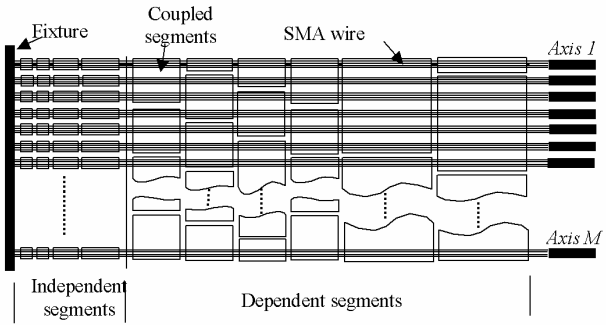

Fig. 1. Two-dimensional segmentation of multi-axis SMA actuator system

In this paper, we first define a Coupling Coefficient. This coefficient plays a major role in determining how much the multi-axis segments can be lumped together. Next, segmentation theory will be presented, which states that the independent segments can generate the given *m* trajectories with the coupled segments. A segmentation design procedure will be developed based on the theory followed by a simple example. Finally, the method will be applied to a five-fingered anthropomorphic robot hand capable of taking variety of postures.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0413242.

The authors are with the d'Arbeloff Laboratory for Information Systems and Technology, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. (email: kyujin@mit.edu, asada@mit.edu).

II. Multi-Axis Segmentation Theory

Fundamental properties of multi-axis coordination using SBC will be analyzed in this section. Consider a subset of actuator axes A_m, as shown in Fig. 2. The subset consists of m, $1 \le m \le M$, actuator axes with an equal total length:

$$A_m = \{i_j \,|\, 1 \le i_1 < i_2 < \cdots < i_m \le M\} \tag{1}$$

If these m axes are to generate "similar" output displacements, $y_i(t), \forall i \in A_M$, they should be generated with some coupled segments combined with independent segments. The goal of the analysis is to quantify the "similarity" of output trajectories and find the proper length of the coupled segments that can generate the "similar" part of the trajectories.

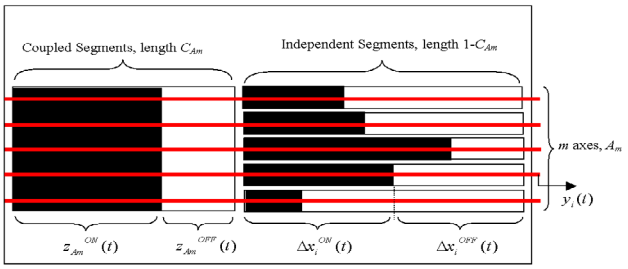

Fig. 2. Coupled and independent segments of m axis array

For the sake of simplicity let us normalize each output displacement with its maximum stroke so that it is confined within the interval between 0 and 1: $0 \le y_i(t) \le 1$. In Segmented Binary Control, the output displacement is proportional to the total length of the actuator material whose state is "ON". If the total ON-state length is normalized with its maximum length so that $0 \le x_i^{ON}(t) \le 1$, then the total ON-state length of the actuator material is the same as the normalized output displacement $y_i(t)$; $x_i^{ON}(t) = y_i(t)$. On the other hand, the actuator material length of "OFF" state is the rest of the entire actuator length. Therefore,

$$x_i^{ON}(t) + x_i^{OFF}(t) = 1 \tag{2}$$

When coupled segments are used, the ON-state length $x_i^{ON}(t)$ consists of the one involved in the coupled segments and the one in the independent segments. As shown in Fig. 2, the coupled segments are divided into the ON state of length $z_{Am}^{ON}(t)$ and the OFF-state of length $z_{Am}^{OFF}(t)$. Likewise each independent strip of segments is divided into $\Delta x_i^{ON}(t)$ and $\Delta x_i^{OFF}(t)$.

$$\begin{aligned} x_i^{ON}(t) &= z_{Am}^{ON}(t) + \Delta x_i^{ON}(t), \\ x_i^{OFF}(t) &= z_{Am}^{OFF}(t) + \Delta x_i^{OFF}(t), \quad \forall i \in A_M \end{aligned} \tag{3}$$

The question is to find the size of the coupled segments: $z_{Am}^{ON}(t) + z_{Am}^{OFF}(t)$. The maximum size of the coupled segments depends on how the trajectories of a given subset of actuator axes are similar to each other. Note that such similarity must be evaluated for the entire set of discrete output positions, or the entire trajectories.

Fig. 3-(a) illustrates trajectories of a subset of actuators A_m defined over time interval T_I:

$$0 \le y_i(t) \le 1, \quad \text{for } \forall t \in T_I, \forall i \in A_m \tag{4}$$

The dynamic range of these trajectories stays in a certain middle range, as shown by the shaded area in Fig. 3-(b). The figure also indicates the normalized length of the actuator material at ON-state, $x_i^{ON}(t)$, as well as the one of OFF-state, $x_i^{ON}(t)$, corresponding to the output trajectories, $x_i^{ON}(t) = y_i(t)$. The height below the shaded area, given by $\ell^{ON} = \underset{i \in Am}{Min}\, x_i^{ON}(t)$, implies that all the m actuators have at least ℓ^{ON} length of ON-state, while the one above the shaded area, given by $\ell^{OFF} = \underset{i \in Am}{Min}\, x_i^{OFF}(t)$, indicates that all the m actuators have at lest ℓ^{OFF} length of OFF-state. In other words, all the m actuators share $\ell^{ON} + \ell^{OFF}$ length of ON/OFF states. Fig. 3-c) shows the profile of the sum of these lengths $\ell^{ON} + \ell^{OFF}$ and its lower bound for the entire time span:

$$C_{Am} = \underset{t \in T_I}{Min}\,[\underset{i \in Am}{Min}\, x_i^{ON}(t) + \underset{i \in Am}{Min}\, x_i^{OFF}(t)] \tag{5}$$

The lower bound C_{Am}, shown by the dot-and-dash line in the figure, represents how much the subset of axes A_m can be controlled together for the entire trajectories with a block of coupled segments. The coefficient C_{Am}, referred to as a Coupling Coefficient of order m, plays the major role in determining how much the multi-axis segments can be lumped together. Now, the following theorem constitutes the principle of multi-axis segmentation.

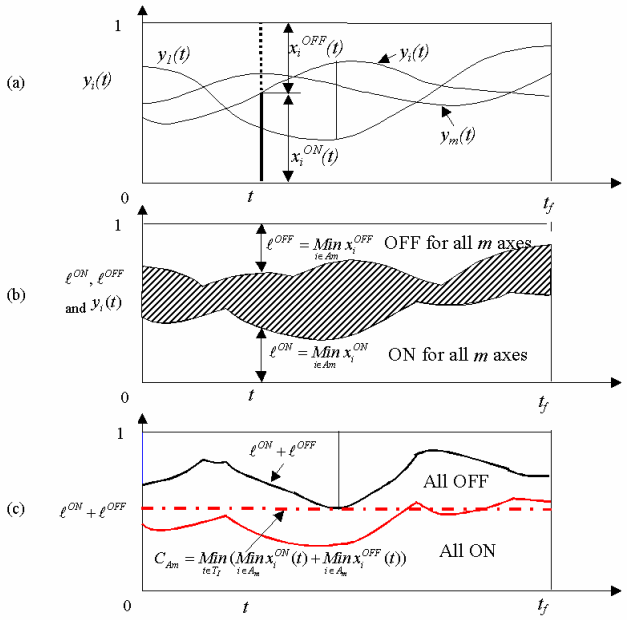

Fig. 3. Trajectories of m axes (a), all On and OFF states for all the m axes (b), and the common segments (c)

Theorem Let $x_i^{ON}(t)$ and $x_i^{OFF}(t)$ be, respectively, the lengths of ON-state segments and OFF-state segments that generate a given trajectory of the *i-th* actuator axis. Let A_m be a subset of *m* actuator axes and $z_{Am}{}^{ON}(t)$ and $z_{Am}{}^{OFF}(t)$ be the lengths of ON state and OFF state of coupled segments across the *m* axes in A_m. If the total length of the coupled segments is no larger than the coupling coefficient C_{Am},

$$\begin{aligned} & z_{Am}{}^{ON}(t) + z_{Am}{}^{OFF}(t) \le C_{Am} \\ & = \underset{t\in T_I}{Min}\,[\underset{i\in Am}{Min}\, x_i^{ON}(t) + \underset{i\in Am}{Min}\, x_i^{OFF}(t)], \quad \forall t \in T_I \end{aligned} \tag{6}$$

then the segment lengths, $x_i^{ON}(t)$ and $x_i^{OFF}(t)$, can be generated by a combination of the coupled segments $z_{Am}{}^{ON}(t)$ and $z_{Am}{}^{OFF}(t)$ and independent segments such that:

$$\begin{aligned} & x_i^{ON}(t) = z_{Am}{}^{ON}(t) + \Delta x_i^{ON}(t), \\ & x_i^{OFF}(t) = z_{Am}{}^{OFF}(t) + \Delta x_i^{OFF}(t), \quad \forall t \in T_I, \forall i \in A_m \end{aligned} \tag{7}$$

where $\Delta x_i{}^{ON}(t)$ and $\Delta x_i{}^{OFF}(t)$ are, respectively, the lengths of the independent segments at ON state and OFF state. The total length of the two is no larger than 1- C_{Am},

$$0 \le \Delta x_i{}^{ON}(t) + \Delta x_i{}^{OFF}(t) \le 1 - C_{Am}, \quad \forall i \in A_m \tag{8}$$

Hence, the independent segments can generate the given *m* trajectories although the coupled segments of length less than C_{Am} are lumped together. The proof of this theorem is given in the Appendix.

Remark 1

Among the *m* axes of independent segments, less than *m* axes of segments may further be coupled with each other. Subtracting the ON-state coupled segments, $z_{Am}{}^{ON}(t)$, from the original trajectory yields the residual trajectory to be generated by the independent segments, $\Delta x_i{}^{ON}(t)$. Treating $m'\,(2 \le m' < m)$ axes of these residual trajectories $\Delta x_i{}^{ON}(t)$ as the original trajectories, the above theorem can be applied again. If the coupling coefficient of m' axes is non-zero: $C_{Am'} > 0$, there exist coupled segments of length C_{Am} across the m' axes. This procedure can be repeated until all the residual trajectories have no coupling, i.e. $C_{Am'} = 0$.

Remark 2

In general, the common trajectories of $z_{Am}{}^{ON}(t)$ and $z_{Am}{}^{OFF}(t)$ are not uniquely determined. Although the upper limit of the total length of the coupled segments is unique, $z_{Am}{}^{ON}(t) + z_{Am}{}^{OFF}(t) \le C_{Am}$, each of the trajectories, $z_{Am}{}^{ON}(t)$ and $z_{Am}{}^{OFF}(t)$ is not uniquely determined.

Remark 3

Let m_1 and m_2 be two integers, $M \ge m_1 > m_2 \ge 2$, and A_{m1} and A_{m2} be two subsets of axes: $A_{m1} \supset A_{m2} \ne \phi$. Then, for arbitrary normalized trajectories,

$$C_{Am1} \le C_{Am2} \tag{9}$$

Namely, the more axes one wishes to lump together, the shorter the maximum length of the coupled segments becomes.

As defined in (5), the coupling coefficient can be computed for various numbers of axes, $2 \le m \le M$. For *m* = 2, (5) reduces to

$$\begin{aligned} C_{ij} = \underset{t\in T_I}{Min}\,[& Min\,(x_i^{ON}(t), x_j^{ON}(t)) \\ & + Min\,(x_i^{OFF}(t), x_j^{OFF}(t))], \qquad 1 \le i, j \le M \end{aligned} \tag{10}$$

Note that when $C_{ij} = 0$, axis *i* and axis *j* cannot be coupled at all, whereas the two actuators are totally coupled if $C_{ij} = 1$. Arranging $C_{ij}\ (1 \le i, j \le M)$ in matrix form yields a *M*-by-*M* Coupling Matrix of order 2.

$$\mathbf{C} = \{C_{ij}\} \in \Re^{M\times M} \tag{11}$$

This symmetric matrix provides useful insight as to how the multiple axes can be coupled together.

III. Segmentation Design Procedure

Based on the theoretical analysis described in the previous section a design procedure for finding multi-axis segmentation with reduced control loops will be presented in this section. Coupling coefficients play the major role in finding which axes to lump together and how long the coupled segments can be.

Step 1. Decoupling and Merger

The original *M* axis problem can be reduced to reduced axis problems using the following algorithm:

1.1 Compute the *M*-by-*M* Coupling Matrix given by (10) for all the combinations of two trajectories: $y_i(t), y_j(t) \,|\, i, j \in A_M, t \subset T_I$.

1.2 Search for 1's in the matrix. If 1 is found at the $(i,j)^{th}$ element, actuators *i* and *j* are totally coupled. Merge axes *i* and *j* to reduce the number of axes. If there are multiple 1's in a row, merge all of them into a single axis.

1.3 Search for 0's in the Coupling Matrix. Swap the actuator numbers *1* through *M* for shifting 0's to off-diagonal blocks so that the Coupling Matrix may be transformed to a block-diagonal matrix. Resultant diagonal blocks determine subgroups of axes that are mutually independent. Segmentation design may be performed for each subgroup of axes that are coupled.

Step 2 Gross Segmentation

For each of the axis subgroups determined in Step 1, the following gross segmentation is performed.

2.1 Enumerate all the possible, exclusive partitions of the axes involved in the subgroup. Let M_o be the total number of the axes involved in a subgroup, and A_{Mo} be the set of the M_o axes. Each possible partition of the axes is represented with a set of axis subsets given by

$$P_k = \{A_{m1}, A_{m2}, \cdots, A_{mk} \mid A_{Mo} = A_{m1} \cap A_{m2} \cap \cdots \cap A_{mk}\} \quad (12)$$
$$, k = 1, \cdots, K$$

where $A_{m1}, \cdots, A_{mk}$ are axis subsets that are exclusive to each other and that cover the whole M_o axes. The axis subset A_{mi} may be a singleton, having only one axis as a component.

2.2 Eliminate partition P_k, if it includes an axis subset A_{mi} that is not a singleton and whose coupling coefficient C_{Ami} is zero. For such an axis subset, segments cannot be lumped. Hence it is subsumed by other partition having finer partitions.

2.3 Compute the coupling coefficient of each A_{mi}, and construct coupled segments of length C_{Ami} and independent segments of length 1- C_{Ami}.

Step 3 Fine Segmentation

The independent segments created in Step 2 may further be coupled.

3.1 Compute the residual trajectories

$$\Delta y_j(t) = \Delta x_j^{ON}(t) = x_j^{ON} - z_{Ami}^{ON}(t), \quad (13)$$

and compute the M_o-by-M_o Coupling Matrix

3.2 Repeat Step 2 for the residual trajectories, and construct coupled segments in the independent segment area generated by the gross segmentation.

3.3 Repeat Steps 3.1 and 3.2 until all the residual trajectories have zero coupling coefficients, no more axes can be coupled, or the number of segments cannot be reduced by coupling the segments.

3.4 Select a segmentation design having the minimum number of segments among all the partitions evaluated above.

TABLE 1 (a) EXAMPLE OF TRAJECTORIES FOR SEVEN AXIS ACTUATOR ARRAY (b) COUPLING COEFFICIENT MATRIX OF THE EXAMPLE

(a)

Axis	Normalized Trajectories						
1	0	0.5	0.75	1	0	0.25	0
2	0	0.5	0.75	1	0.75	0.75	0
3	0	0.5	0.75	1	0	0.25	1
4	0	0.5	0.75	1	0	0.5	1
5	0	0.5	0.75	1	0.75	0.75	1
6	0	0.5	0.75	1	1	1	1
7	0	0.5	0.75	1	1	1	1

(b)

	1	2	3	4	5	6	7
1	1	0.25	0	0	0	0	0
2	0.25	1	0	0	0	0	0
3	0	0	1	0.75	0.25	0	0
4	0	0	0.75	1	0.25	0	0
5	0	0	0.25	0.25	1	0.75	0.75
6	0	0	0	0	0.75	1	1
7	0	0	0	0	0.75	1	1

Example Table 1-(a) shows an example of a discrete set of normalized outputs for the seven-axis actuator array to generate, i.e. trajectories to generate. Fig. 4 shows the process of multi-axis segmentation design using these trajectories, leading to the minimum segmentation, i.e. the minimum number of ON-OFF control segments needed for generating the trajectories.

Step 1 Decoupling and Merger First, the coupling matrix is constructed using step 1.1 and shown in Table 1-(b). The first row and column show the actuator numbers, 1 through 7. From the coupling matrix, two sets of mutually independent axis subsets {1,2} and {3, 4, 5, 6, 7} are identified. The segmentation of these two groups can be performed separately as shown in Fig. 4-(a). Note that the coupling coefficient $C_{6,7}$ is one, hence they are totally coupled.

Step 2 Gross Segmentation C_{12} is 0.25; hence 25% of the length of the actuator axes 1 and 2 can be lumped together. There are eleven different ways of exclusively partitioning the actuators 3 to 7. Among these only meaningful partitions are

{{3,4}, {5,6,7}} and {{3,4,5}, {6,7}}. Other cases are pruned out in Step 2.2. In Fig. 4-(b), the former partition, {3,4} and {5,6,7}, is shown. With $C_{3,4}$ = 0.75 and $C_{5,6,7}$ = 0.75, both groups can have 75% length of common segments shown by shaded areas. A similar result can be obtained for the latter partition, too.

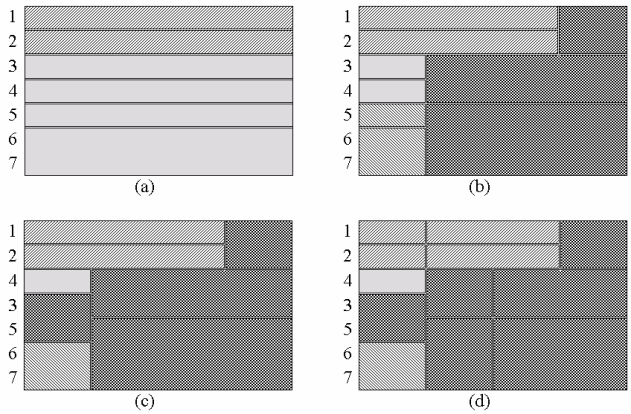

Fig. 4. Procedures of segmentation design for the trajectories given in table 1.(a) Actuators divided into two decoupled groups and actuators {6, 7} are totally coupled. (b) Common segments for actuators {1,2}, {3,4} and {5, 6, 7} are designed. (c) Common segments for independent trajectories of actuators {3, 5} are designed (d) Minimum number segmentation found for each segment

Step 3 Fine Segmentation To further couple the independent segments, residual trajectories of the individual axes are obtained from (13), and a new coupling matrix is derived from the residual trajectories. As stated in Remark 2, the residual trajectories are in general not unique. For the gross segmentation given by Fig. 4-(b), there are 128 different sets of residual trajectories. When the largest possible ON-state length is used for generating the common trajectories, $z_{Am}^{ON}(t)$ and $z_{Am}^{OFF}(t)$, the coupling matrix of Table 2 for axes 1 to 6 (now denoted *1i* to *6i*) is obtained. In this coupling matrix, $C_{3i,5i}$ is 0.25, hence the independent segments of axes 3 and 5 can be lumped together up to 25% of the total length of the actuator, which is the whole length of the independent segments of axes 3 and 5. As shown in Fig. 4-(c), axis 3 is swapped with axis 4 to make axes 3 and 5 adjacent to each other so that they can be coupled together. Finally, Fig. 4-(d) shows the final segmentation including the ones inside the common segment blocks.

TABLE 2 COUPLING COEFFICIENT MATRIX OF INDEPENDENT TRAJECTORIES

	1i	2i	3i	4i	5i	6i
1i	0.75	0	0	0	0	0
2i	0	0.75	0	0	0	0
3i	0	0	0.25	0	0.25	0
4i	0	0	0	0.25	0	0
5i	0	0	0.25	0	0.25	0
6i	0	0	0	0	0	0.25

The above computation is repeated for all the possible common trajectories as well as for different gross segmentations in Step 2. The total of twelve segments involved in the segmentation of Fig. 4-(d) is the minimum among these. If the segments were not coupled and only a single size segments were used for the segmentation design, twenty-eight segments should have been used; Hence 57.14% savings in the number of segments have been achieved.

IV. APPLICATION TO AN ANTHROPOMORPHIC ROBOT HAND

The design concept of multi-axis actuator array with two-dimensional segmentation architecture is applied to an anthropomorphic robot hand driven by SMA actuators. A single human hand has 19 joints at the five fingers alone. Despite numerous degrees of freedom, many of them are coupled. Such coupled motion can be generated effectively with use of multi-axis segmented binary control.

Behavior of human fingers has been studied in the robotics and biomechanics communities for years. Okada [7] characterized human grasp and manipulative behavior and classified various poses into seven groups. Cutkosky [8] also shows an example of grasp taxonomy for manufacturing tasks. More recently, human finger postures have been studied extensively in the field of human-computer interaction (HCI) [9]-[10]. To better understand hand gestures from visual images and other sensor data, anatomical and kinesiological constraints among multiple joint axes are used for eliminating unnatural poses and physiologically infeasible poses [10]. Such constraints greatly reduce the search space. Like-wise, the reduction in the space of natural poses helps us reduce the number of segments of a multi-axis actuator array for driving an anthropomorphic robot hand.

We defined fourteen hand postures that span the useful range of human hand motions, and the actuator array will be designed to perform these postures. For these postures, finger joint angles can be derived. These joint angles are transformed into displacements of the actuators driving the individual joints. To implement segmented binary control, the displacements must be discretized with resolution equal to the resolution of the actuator. The resolution of the actuator is the output displacement that can be produced by the smallest segment. The discretized displacements are then normalized and used to design segmentation.

SMA actuators with length equal to N smallest discrete segments drive each finger having a maximum bending angle of α_i. The smallest actuator segment can produce an output displacement of η. Angle θ_i of the *i-th* finger joint is transformed into normalized displacement y_i of the actuator axis i, using the following parameters:

$$y_i = \frac{N \cdot \eta}{\alpha_i} \theta_i \qquad (14)$$

Note that $N \cdot \eta$ is the maximum output displacement that each axis can produce.

We have chosen to design an actuator array consisting of five *30 mm* segments per axis, with a maximum joint bending angle of 90°; hence $N=5$ and α =90°. The displacements are quantized into five levels, and normalized by the maximum output displacement. Quantized and normalized values of the actuator displacements, $y_i(\theta_i)$, are directly derived from the defined joint angles as follows.

$$y_i(\theta_i) = \frac{k}{N} \quad \text{for} \quad k\frac{\alpha}{N} - \frac{1}{2}\frac{\alpha}{N} \le \theta_i < k\frac{\alpha}{N} + \frac{1}{2}\frac{\alpha}{N} \qquad (15)$$

where k is an integer between 0 and 5.

TABLE 3 DISCRETIZED AND NORMALIZED DISPLACEMENTS USED FOR SEGMENTATION DESIGN

Actuator #	Assigned Joint	Open	Survey	Envelope1	Envelope2	Ball1	Ball2	Fist1	Fist2	Pinch1	Pinch2	Pinch3	Pinch4	Point	Write
1	Thumb(DIP)	0	0.2	0.2	0.4	0.4	0.6	0.8	1	0.4	0.4	0.4	0.4	1	0
2	Thumb(MP)	0	0.4	0.4	0.6	0.4	0.6	0.8	1	0.6	0.6	0.6	0.6	1	0.4
3	Index(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0.6	0.6	0.6	0.6	0	0.6
4	Index(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0.6	0.8	0.6	0.8	0.4	0.6
5	Middle(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0.4	0.6	0.6	0.6	1	0.8
6	Middle(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0.2	0.4	0.6	0.8	1	0.6
7	Ring(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0.2	0.4	1	1	1	1
8	Ring(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0	0.2	1	1	1	1
9	Pinkie(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0	0.2	1	1	1	1
10	Pinkie(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0	0.2	1	1	1	1

Table 3 shows the discretized and normalized displacements used for segmentation design, where each column represents a specific hand posture and each row represents the displacements of each actuator. Now using these data, segmentation is designed using the procedure given in the previous section.

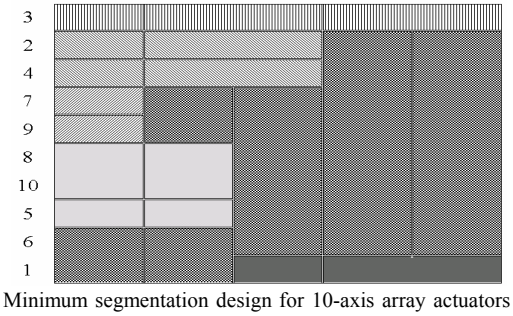

Fig. 5. Minimum segmentation design for 10-axis array actuators driving robotic hand

Fig. 5 shows the final segmentation design achieving the

minimum number of segments. Twenty-one segments are used in the final design, which is 58% savings in the number of segments compared to when the actuators are not coupled at all.

V. CONCLUSION

The segmentation theory for designing a multi-axis actuator array has been presented, and the segmentation design procedure has been developed based on the theory. The method has been applied to an actuator array for driving a five-fingered robot hand capable of taking variety of postures, and we were able to reduce the number of segments to 58% of the original amount. With this design, the control of ten-axis actuator array is reduced to a simple on-off control of twenty-one segments and each hand posture can be represented with twenty-one bits of 0's and 1's. This segmentation design has been implemented with a shape memory alloy actuator driven by thermoelectric modules and will be presented in another conference [11]

APPENDIX

Proof of the theorem

Consider a block of coupled segments of length C_{Am}, consisting of ON-state segments with length $z_{Am}{}^{ON}(t)$ and OFF-state segments with length $z_{Am}{}^{OFF}(t)$:

$$z_{Am}^{ON}(t) + z_{Am}^{OFF}(t) = C_{Am}, \quad t \in T_I \tag{A-1}$$

Let us design the coupled segments such that $z_{Am}{}^{ON}(t)$ is given by

$$z_{Am}^{ON}(t) = Min\left[C_{Am}, \underset{i \in A_m}{Min}\, x_i{}^{ON}(t) \right] \tag{A-2}$$

and that the OFF segment length $z_{Am}{}^{OFF}(t)$ is given by

$$z_{Am}^{OFF}(t) = C_{Am} - z_{Am}^{ON}(t) \tag{A-3}$$

We want to show that, although a fraction of $y_i{}^{ON}(t)$ is determined by the coupled segments, the remaining independent segments whose length is 1 - C_{Am} are large enough to generate the rest of $x_i{}^{ON}(t)$:

$$\Delta x_i{}^{ON}(t) = x_i{}^{ON}(t) - z_{Am}{}^{ON}(t) \le 1 - C_{Am} \tag{A-4}$$

Let us first assume that

$$\underset{i \in A_m}{Min}\, x_i{}^{ON}(t) \le C_{Am} \tag{A-5}$$

Then, from (A-2),

$$\begin{aligned} \Delta x_i{}^{ON}(t) &= x_i{}^{ON}(t) - \underset{j \in Am}{Min}\, x_j{}^{ON} \\ &\le 1 - \underset{j \in Am}{Min}\, x_j{}^{OFF}(t) - \underset{j \in Am}{Min}\, x_j{}^{ON}(t) \\ &\le 1 - \underset{t \in T_I}{Min}\left[\underset{j \in Am}{Min}\, x_j{}^{ON}(t) + \underset{j \in Am}{Min}\, x_j{}^{OFF}(t) \right] \\ &= 1 - C_{Am} \end{aligned} \tag{A-6}$$

From the first line of the above equation it is clear that $\Delta x_i{}^{ON}(t)$ is non-negative, hence

$$0 \le \Delta x_i{}^{ON}(t) \le 1 - C_{Am} \tag{A-7}$$

If (A-5) does not hold, then from (A-2), $z_{Am}^{ON}(t) = C_{Am}$. Therefore,

$$\Delta x_i{}^{ON}(t) = x_i{}^{ON}(t) - C_{Am} \le 1 - C_{Am} \tag{A-8}$$

Also,

$$0 \le \underset{j \in A_m}{Min}\, x_j{}^{ON}(t) - C_{Am} \le x_i{}^{ON}(t) - C_{Am} = \Delta x_i{}^{ON}(t) \tag{A-9}$$

Therefore, (A-4) has been derived, i.e. the independent segments are large enough to generate $x_i{}^{ON}(t)$, so that $x_i{}^{ON}(t) = z_{Am}{}^{ON}(t) + \Delta x_i{}^{ON}(t)$. Similarly, we can show that the independent segments are long enough to generate $x_i{}^{OFF}(t)$. From (A-3) and (13),

$$\begin{aligned} \Delta x_i{}^{OFF}(t) &= x_i{}^{OFF}(t) - z_{Am}{}^{OFF}(t) \\ &= (1 - x_i{}^{ON}(t)) - (C_{Am} - z_{Am}{}^{ON}(t)) \\ &= 1 - C_{Am} - (x_i{}^{ON}(t) - z_{Am}{}^{ON}(t)) \\ &= 1 - C_{Am} - \Delta y_i{}^{ON}(t) \le 1 - C_{Am} \end{aligned} \tag{A-10}$$

Using (A-8) in the above equation yields

$$\Delta x_i{}^{OFF}(t) = 1 - C_{Am} - \Delta x_i{}^{ON}(t) \ge 0 \tag{A-11}$$

Therefore,

$$0 \le \Delta x_i{}^{OFF}(t) \le 1 - C_{Am} \tag{A-12}$$

Q.E.D.

REFERENCES

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

[2] Ikuta, K., Tsukamoto, M., and Hirose, S., "Shape Memory Alloy Servo Actuator System with Electric Resistance Feedback and Application for Active Endoscope," *International Conference on Robotics and Automation*, Vol. 1, pp.427-430, 1988.

[3] K. K. Safak and G. Adams, "Modeling and simulation of an artificial muscle and its application to biomimetic robot posture control," *Robotics and Autonomous Systems*, vol. 41, pp. 225-243, 2002.

[4] B.-H. Park, M. Shantz, and F. Prinz, "Scalable rotary actuators with embedded shape memory alloys," in *Smart Structures and Materials* 2001.

[5] Bar-Cohen, Y., edit, "Electroactive Polymer Actuators as Artificial Muscles – Reality, Potential, and Challenges", *SPIE*, 2001.

[6] J. D. Madden, P.G. Madden, I.W. Hunter, "Conducting Polymer Actuators as Engineering Materials", *Proc. of SPIE Conference on Electroactive Polymer Actuators and Devices*, pp.176-190, 2002.

[7] T. Okada, "Analysis of Finger Motion and Hand tasks," *Biomechanism* 3, University of Tokyo press. pp. 133-144, 1975.

[8] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 269-79, 1989.

[9] J. Lin, Y. Wu, and T. S. Huang, "Modeling the constraints of human hand motion," in *Proc. Workshop on Human Motion*, 2000.

[10] J. Lee and T. L. Kunii, "Model-based analysis of hand posture," *Computer Graphics and Applications*, IEEE, vol. 15, pp. 77-86, 1995.

[11] Kyu-Jin Cho, H. Harry Asada, "Multi-Axis SMA Actuator Array for Driving Anthropomorphic Robot Hand," in *Proc. of 2005 IEEE International Conference on Robotics and Automation*