2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Optimal Control of Multi-Input SMA Actuator Arrays Using Graph Theory

Leslie J. Flemming, David E. Johnson, and Stephen A. Mascaro

Abstract—**Shape memory alloy (SMA) actuators are compact and have high force-to-weight ratios, making them strong candidates to actuate robots, exoskeletons, and prosthetics. To optimize speed and energy consumption, SMA actuators have been embedded in an *NxN* vascular network that can deliver electric and thermofluidic energy to the each actuator. The scalable architecture of the vascular network allows for $2N$ control devices (valves, transistors) to be shared amongst N^2 actuators, so that as the number of actuators increases, the number of required control devices scales at a smaller rate. This Network Array Architecture (NAA) allows for each actuator to be controlled individually or in discrete subarrays. However, not all combinations of actuators can be activated simultaneously; therefore in general, a sequence of control commands will be need to be executed in order to achieve the desired actuation.**

By treating each actuator as having a binary state, the combined states of the actuator array can be represented by graph theory, where states are nodes and the transitions between states are graph edges. By properly weighting the costs of the transitions, graph search techniques can be used to find optimal sets of control commands for desired state changes. This paper formulates the control of NAA actuators systems as a graph theory problem, and characterizes the ability of search algorithms to optimize a weighted combination of speed and energy usage, while minimizing computational cost.

I. Introduction

A primary challenge to maximize the wearability of robotics, prosthetics and exoskeletons is the need to actuate many degrees of freedom (DOF) with minimal bulk. Wearability is enhanced when robotic actuation closely matches the characteristics of human actuation according to metrics such as strength, speed, range of motion, power density, and degrees-of-freedom. Wearability additionally implies implementation with a long-lasting untethered energy source.

As a prime example, consider the human hand, which has 21 degrees-of-freedom in the fingers and thumb alone. It is extremely difficult to design a robotic/prosthetic hand to mimic this range of motion of a human hand. A greater engineering challenge is to actuate it with similar power density of skeletal muscles (50 to 100 W/kg [1]). Although DC motors can produce wide range motion, their typical power density is approximately 10 W/kg. Therefore, in order for a prosthetic arm to match the capabilities of the biological counterpart, it would have to be 5 to 10 times the volume and weight of the human arm. Pneumatic and hydraulic actuators can match or exceed the power density of human muscles, but they involve the hazards of high operating pressures and require a tethered source of pressurized fluid or a heavy/noisy compressor. Piezoelectric actuators have power density comparable to human muscle, but require potentially hazardous operating voltages and create only microscale displacements, which would be impractical to leverage into useful motion for a prosthetic or exoskeleton.

One solution to advancing wearable robotics is to implement large networks of high power-density "muscle-like" actuators such as Shape Memory Alloys (SMA) and ElectroActive Polymers (EAPs) [2]. Unlike electromagnetic or hydraulic actuators, the principal dynamics of SMAs are thermomechanical. SMA actuators have high power density and are able to store thermal energy, allowing them to maintain a force without further energy input. Their dynamics are highly nonlinear and therefore difficult to control with classical control methods [3-11]. By neglecting the nonlinear elements of their dynamics and treating them as binary actuators (fully contracted/extended), and networking large numbers of these actuators mechanically in series or parallel, a more realistic means of control can be achieved [12-15]. This method of control requires novel ideas for energy delivery and removal.

Inspired by the human body, a "vascular network" has been created for delivering/removing energy to/from arrays of SMA "muscles" by thermofluidic methods [16]. Unlike hydraulic or pneumatic systems, these networks operate at relatively low pressures, similar to biological vascular systems. In order to characterize the performance and evaluate the full potential of these *Wet Actuators*, complex dynamic models of the SMA response to thermofluidic and electrical inputs are needed, as well as multi-input intelligent control algorithms tailored to on-off control of the nonlinear dynamics. Furthermore, when arrays of actuators are networked together, the fluidic impedances of the vascular networks interact in such a way that unique dynamic behaviors emerge from the overall network that would not result from a purely electrical network of similar architecture. Thus, in addition to intelligent network control algorithms, dynamic modeling is critical for characterization of the wet actuator networks, as well as the individual

Manuscript received Oct 15, 2010. This work is supported by the National Science Foundation: Award 1031848.

L. J. Flemming is graduate student in Mechanical Engineering, University of Utah, Salt Lake City, UT 84112 USA(phone: 801-865-3617, fax: 801-585-9826, e-mail: leslie.flemming@utah.edu).

S. A. Mascaro, is a professor of Mechanical Engineering, University of Utah, Salt Lake City, UT 84112 USA (smascaro@mech.utah.edu).

D. E. Johnson is a researcher in the School of Computing, University of Utah, Salt Lake City UT USA 84112 (e-mail: dejohnso@cs.utah.edu).

actuators themselves. Ultimately, these methods of characterization could be adapted to fuel-powered SMA [17] and wet EAP muscles, replacing thermodynamics with chemical dynamics.

As a target application to demonstrate the impact of this research, we would specifically like to enable prosthetic hands/exoskeletons that could replicate/manipulate the actuated degrees-of-freedom similar to that of a human hand. The "muscles" themselves would fit in a human-sized prosthetic forearm or contained in a low-profile sleeve worn over a human forearm.

The goal of this paper is to present optimal control algorithms for controlling arrays of wet actuators by applying graph theory. In Section II, we first review the characteristics of SMA actuators and prior work in networking arrays of wet SMA actuators and controlling them with electric and thermofluidic inputs. In Section III, we show how graph theory can be used to formulate the multi-input control problem for an array of networked wet SMA actuators. We present how graph theory algorithms can be used to search for sequences of control commands that optimize actuation time and energy usage. Finally, in Section IV, the performance of the graph theory algorithms will be examined through simulation.

II. Background

A. Shape Memory Alloy (SMA) Actuator

SMAs are a class of smart materials that are able to return to a predefined shape after being strained up to approximately 4%. This deformation and recovery of strain is achieved by a thermomechanical process where the SMA is strained while it is cold (<~70C) and then restored when it is heated about its transformation temperature of ~70C. These characteristics allow SMA to be implemented as compact, high force-to-weight ratio actuators. Because the thermomechanical process is nonlinear and difficult to model and control, this research will treated the actuators as binary (fully contracted (1) or extended (0)) and the control strategy will only be concerned with delivering/removing energy in order to transition between these two states. In order to achieve higher resolution displacements/forces, the actuators will be bundled together and operated in series/parallel respectively. Figure 1 shows a robotic hand that is actuated by wet SMA actuators in series to produce discrete displacements of the fingers. Electric and thermofluidic inputs to the actuators are controlled by a network of transistors and valves. The intelligent control algorithms presented in this paper will be applied to this hardware in future work.

B. Wet SMA Actuators

SMA wire can be heated electrically very quickly (milliseconds); however the cooling of the SMA wire can take multiple orders of magnitude longer if unforced air convection is used. To improve the cooling process, forced convection and water baths have been used, but these add significant amount of size and mass to the actuator. To maintain a high force-to-weight ratio for the actuators, wet SMA actuators [13] were developed, in which an SMA wire is embedded in a compliant vessel. The compliant vessel allows a small amount of cold fluid to flow over the wire to improve the convection cooling and it also allows for convective heating using hot fluid.

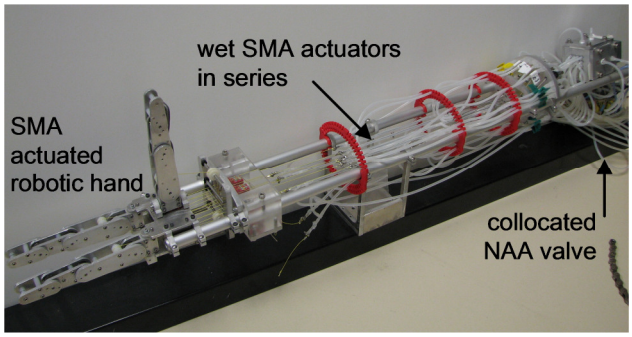

Figure 1.Robotic hand actuated by wet SMA actuators connected in series.

C. Network Array Architecture

The wet SMA actuator assembly does resolve the issue of cooling rate; however the control devices (valves, transistors) can be many times heavier than the SMA actuator itself. Configuring the actuators in a Network Array Architecture (NAA) [18] allows for an *NxN* array of actuators to be controlled by 2*N* control devices, reducing the total weight of the system. Figure 2 shows a schematic of the electrical network. The actuators in a common row are connected to a constant voltage source and on the sink side (ground) and the actuators in a common column are connected to the sink (ground). In order to send energy to the actuator, the row and column switch must be closed to complete the circuit. The diodes in the circuit ensure that the current cannot take an undesired route to ground. Each actuator can be activated individually, or multiple actuators can be activated simultaneously by activating multiple rows and/or columns. However actuators on the diagonal (e.g. A1

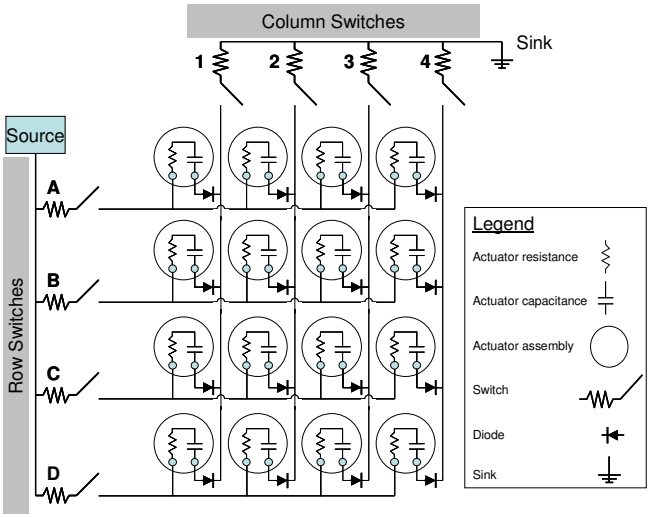

Figure 2. Network Array Architecture

& B2) cannot be activated simultaneously without permitting energy to flow into adjoining actuators as well (e.g. A2 & B1).

The fluidic network is controlled in a similar manner to the electrical network. However, due to the compliance (capacitance) of the fluidic vessel, the fluidic domain must be constructed using a collocated NAA [19]. This architecture eliminates parasitic responses observed in a fluidic system using the standard NAA.

D. *Control Logic for NAA*

The control logic for NAA is described in detail in [20]. Each control device of NAA is treated as a binary device which is either connected (1) or disconnected (0). The N control devices on the source and the sink side of the array can be in any of 2^N states (e.g. 1000, 0100, 1100, ...). When the set of control devices are all disconnected (0000), there is no flow (electric/fluidic) through the system, which will not produce any change in the system. This leaves 2^N-1 configurations each for the sink and source side of the array. In order for energy to flow from the source through the actuator(s), at least one source side control device and one sink side device must be connected (closed). Since the two sets of control devices are orthogonal to one another, there are $(2^N\text{-}1)^2$ configurations of the control devices that will connect a subarray (e.g. 1x1, 1x2, 2x2, …, *NxN*) of actuators from the source to sink. Figure 3 shows a few examples of control configuration for both fluidic and electric NAA.

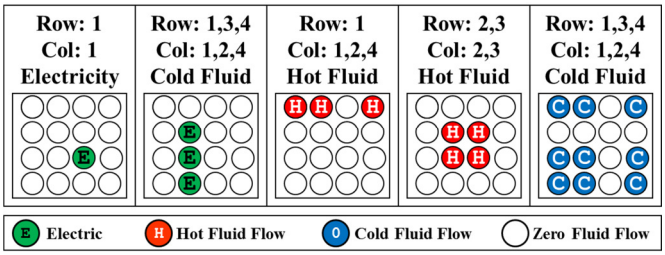

Figure 3. Control commands examples.

Because of the architecture of NAA, not all the actuators can be controlled simultaneously and it may take a sequence of control commands to produce the desired actuation. The fluidic network allows either hot or cold fluid to be delivered to the actuator array at any point in time. The fluid and electrical networks are independent and can operate simultaneously, but a complete algorithm that takes full advantage of this ability has not been developed and this task with be discussed in the future work section.

E. *Optimization of the Wet SMA Actuator Array*

Previous work [21-22] has characterized the performance of wet SMA actuators using both fluid and electricity to deliver thermal energy to produce actuation. Figure 4 shows the characteristics (speed, efficiency) of a single wet SMA actuator with different rates of energy input. These results show that the electrical network is up to 2.5 times more efficient than and twice as fast as the fluidic network. However, the energy source to heat the fluid (e.g. propane)

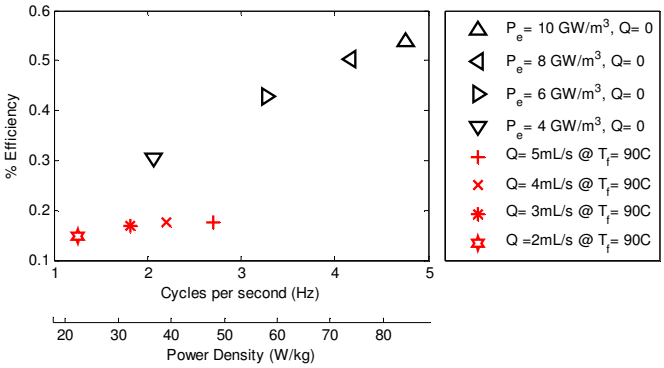

Figure 4 Speed vs. Efficiency of a single wet SMA actuator activated by electricity or hot fluid.

can have up to 100 times the energy density than that of an electric battery. Therefore although fluidic heating is not as fast or efficient, more total actuations can completed per unit mass of energy storage. For arrays of actuators, identifying the optimal set of control commands to yield a desired state will be achieved through graph theory discussed in the following section.

III. Intelligent Control of NAA using Graph Theory

A. *Graph Theory Structure of NAA*

In the previous sections, the SMA actuators and NAA have been defined by discrete states and control commands. Graph theory can be applied to the discrete components of NAA. NAA arranges the binary actuators (contracted (1) and extended (0)) in an *NxN* array that can be in one of 2^{N*N} states. These states will be treated as nodes in the graph. The NAA also defines $(2^N\text{-}1)^2$ discrete control command configurations that can address various combinations (subarrays) of actuators. These control commands are potential edges between nodes. Each of these control commands can be applied to any node of the graph; however some control commands will not produce any actuation. Commands that produce actuation will be defined as edges in the graph. Because there are three input modes (hot fluid, cold fluid, or electricity) for each control command configuration, there can be multiple edges between two specific nodes, where such a graph is defined as a *multigraph*. Therefore, the total number of possible control commands expands to 3 (2n-1) 2 from each nodes.

B. *Constructing the graph*

The 2^{N*N} nodes and $3(2^N\text{-}1)^2$ control commands can be represented by N^2 bit numbers. Valid edges of the graph can be easily identified by applying bitwise operations between the nodes and control commands. A complete graph for the NAA can be constructed with the following algorithm:

1) *Hot fluid or electrical heating control commands*
   test = bitor (node, control command)
   if test ~= node then control command = edge
2) *Cold fluid control command*
   test = bitor (~node, control command)
   if test ~= (~node) then control command = edge

This algorithm identifies all the edges that cause at least 1 actuator to change states. If the control command forming the edge delivers energy to an actuator of a subarray and there is no resulting actuation (e.g. hot fluid is delivered to a subarray of actuators, one of which is already hot & contracted), then this energy flow is termed *superfluous*. As long as at least one actuation occurs, the edge is still valid, but not as energy efficient as one resulting in full actuation (e.g. 4 out of 4). Figure 5 shows four examples of determining if a control command is an edge. The control commands for examples A & C actuate all the actuators that are addressed by the control command. Examples B & D have superfluous flow. B is an edge because it produces a single actuation, while D does not produce any actuation, and would not be included in the graph. There are redundant edges that can also be removed. These edges produce the same actuation as a smaller control subarray and are therefore redundant. For example, a 4x1 control command that heats 3 actuators can be removed from the graph because a 3x1 control command/edge will produce the same actuation.

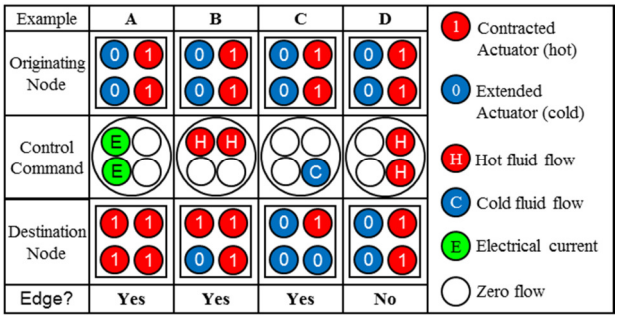

Figure 5 Examples of identifying edges in a NAA graph

An edge with superfluous flow has lower energy-efficiency, but it may produce the desired actuation at a faster rate. Figure 6 shows a small portion of the 2x2 graph (not all nodes and edges are shown). Path ABC has no superfluous flow and a cost of [2.8τ, 3ε], where τ is a time unit and ε is an energy unit. Path AC has an accumulated time and energy cost of [2.3τ, 4ε]; where the desired actuation is completed faster with superfluous flow at the penalty of being less efficient. To express a preference for energy or time, a weighted time-energy cost (C_{te}) function with a performance weighting factor (w_{et}) has been defined as:

$$C_{te} = w_{te} * Time + (1 - w_{te}) * Energy \qquad (1)$$

When $w_{et} = 1$, the total actuation time is minimized and when $w_{et} = 0$, the total actuation energy is minimized.

C. Graph Theory Algorithms

Now that NAA has been formulated as a graph, we will examine three of the basic graph search algorithms: Best First Search (BFS), Dijkstra's, and A* [23]. The *multigraph* representation of the actuator array has some interesting characteristics, such as high connectivity between nodes,

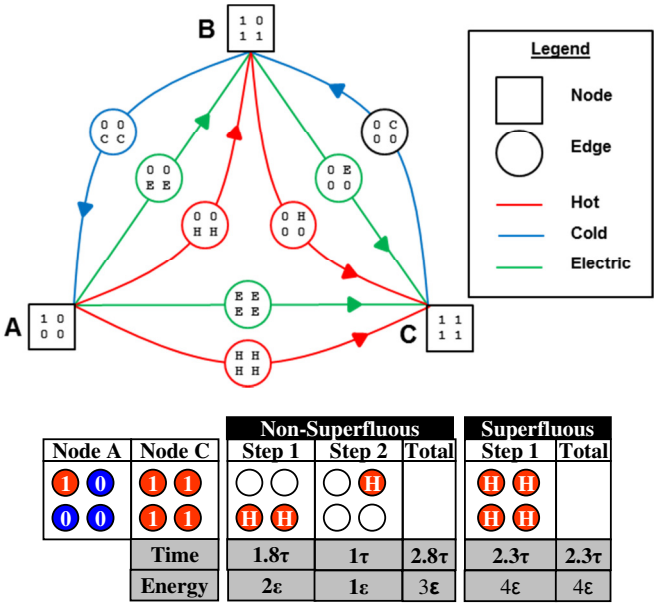

Figure 6 Partial graph of a 2x2 array with desirable superfluous flow.

large state spaces, and redundant paths between start and goal nodes.

Best First Search (BFS): is a greedy search technique which follows an estimated cost-to-goal heuristic. While the path is not optimal, BFS tends to have a low computational cost. The heuristic value for this binary SMA actuator array is equal to the number of actuators (bits) that need to be flipped from the intermediate node to destination node.

Dijkstra's: is a search technique that identifies a path with minimum path cost. However, Dijkstra's tends to be computationally costly, because it explores the graph with an uninformed perspective of the destination. The cost to traverse an edge in our graph is C_w.

A*: is a search algorithm that identifies a minimum cost path like Dijkstra's but it includes an estimated cost-to-goal heuristic like BFS to take a directed approach of exploring the graph. This results in lower computational cost than Dijkstra's and a more optimal path than BFS.

IV. Simulations and Analysis

A. Graph Scalablity

For a given array size N, an NAA graph can be built and implemented in MATLAB using the algorithm in section III.B. Table 1 shows the dimensions of each NAA Graph and the number of bytes it takes to represent the adjacent nodes of that graph. The table shows that the preprocessed

Table 1 Dimensions of Preprocessed NAA graph

N	Nodes 2^{N*N}	Control Cmds $3(2^N-1)^2$	# of Total Edges / Adjacent Nodes	# of Bytes to represent the Adjacent Nodes
2	16	27	110	288
3	512	147	19,070	50,176
4	65536	675	13,847,870	29,491,200

graph does not scale well, but the number of control commands does scale well. Therefore, in future work the graph will be constructed as an expanding wave front problem.

B. *Analysis of Search Algorithms / Computational Cost*

Of the three search algorithms, A* can both identify an optimal path and complete the search quickly. The challenge in using A* is correctly identifying the heuristic cost-to-goal, which is used to estimate the remaining cost to get from an intermediate node to the destination node as the search progresses through the graph. Our current metric for the heuristic cost-to-goal (h) is to calculate the number of actuators (bits) that need to change state (flip between 0 and 1) from an intermediate node to the destination node (goal). To ensure that h is weighted appropriately compared to C_w, we define the estimated total cost-to-goal, C_{total}, evaluated at any intermediate node:

$$C_{total} = w_h * h + (1 - w_h) * \sum C_{te} \qquad (2)$$

where w_h is the heuristic weighting and $\sum C_{te}$ is the summation of C_{te} from the origin node to the intermediate node. The A* algorithm uses C_{total} to perform a directed search through the graph. Figure 7 shows the characteristics of the A* algorithm on a 4x4 array with w_h ranging from 0 (equivalent to Dijkstra's) to 1 (equivalent to BFS). For the first example, the w_{te} has been set to zero to minimize energy. Figure 7A shows the mean number of edges the algorithm explored (computational cost of the algorithm) and 7B shows the mean energy cost for 100 randomly generated simulation sets (origin, destination). BFS quickly finds a path between the origin and the destination, but the path is always not at a minimum energy. The Dijkstra's algorithm identifies a path with minimum C_{te}, but on average explores multiple orders more edges than BFS. An optimal value of w_h is estimated to be 0.55 in order to find a path with minimum cost C_{te}, without unnecessary computation. On average, the algorithm only explores 9000 edges for this size graph. The optimal value of w_h may vary depending on the performance characteristics of the wet SMA actuator and the NAA array. In Figure 7B, there is superfluous energy being sent to the array when w_h is greater than 0.6. Although the array is less efficient in these cases, the total actuation time (Figure 7C) decreases and results in faster actuation rates similar to the case in Figure 6.

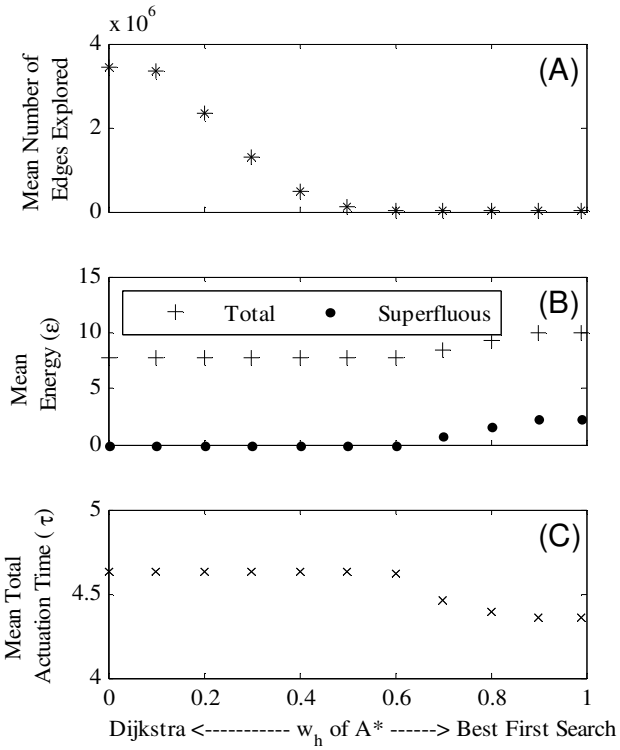

Figure 7. Mean # of edges explored, mean energy (ε) and mean total actuation time with varying Heuristic Weighting (w_h) for A* search algorithm for 4x4 array

C. *Analysis of the Time-Energy Weighting*

Now using the optimal $w_h = 0.55$ in the A* algorithm, the performance of the array can be quickly evaluated by varying the w_{te} (averaging 25 ms per origin and destination set using MATLAB). In Figure 8A and 8B, we see that the performance of the array transitions from slow and more energy-efficient to fast and less energy-efficient. Figure 8C plots the breakdown of energy costs associated with different inputs to the array. As w_{te} is increased, electricity and superfluous flow are increasingly utilized, resulting in greater speed at the expense of more energy usage. The

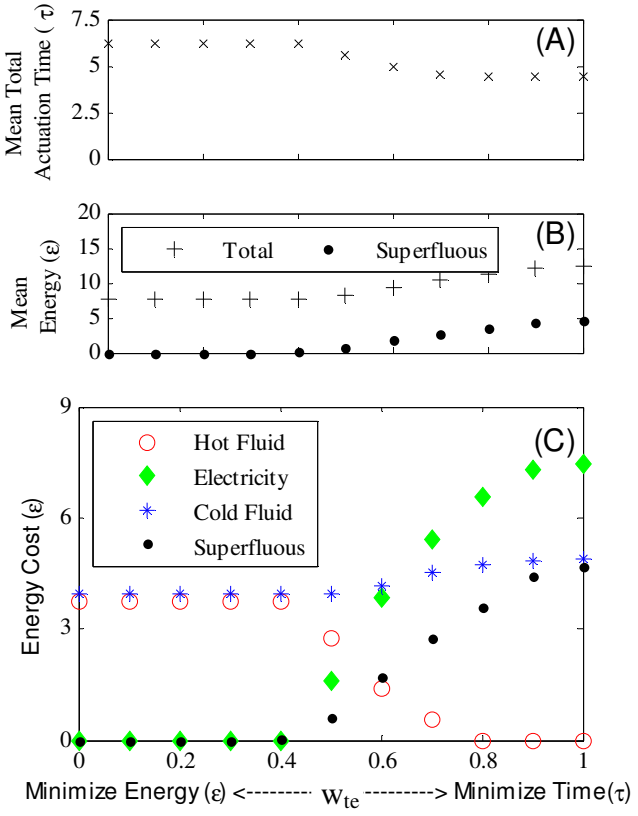

Figure 8 Performance Characteristics of Actuator Array with vary Cost Function (w_{te}).

system transitions between energy-efficiency and speed when w_{te} is approximately 0.5. The optimal value of w_{te} will vary based on dimensions and characteristics of the actuators and the vascular network.

V. Conclusion

This paper has formulated the optimal control of a networked array of multi-input SMA actuators as a graph theory problem, and has shown that it is a viable method to determine an optimal set of control commands to transition the actuator array from one node to another node. As expected, the A* algorithm found the path of least cost based on performance characteristics while only exploring a small percentage (0.3%) of the graphs. Although A* was able to find a minimum cost path by only exploring a small percentage of the graph, this graph becomes too large to be preprocessed and stored for actuator arrays larger than 4x4.

Future work will need to examine the control of larger arrays and must work without a preprocessed graph. Because of the scalability of the control commands, an expanding wave solution is a possibility for implementing the A* algorithm. For example in a 4x4 array, starting from the origin node, the graph would explore the 675 control commands and identify a maximum of 225 new nodes. This is because NAA results in a multigraph and some edges are redundant while others do not produce any actuation. The search algorithm would then identify one node with the minimum cost and expand up to 225 additional nodes to the wave front. This will continue until the search algorithm finds a path to the destination node with minimum cost. From the time it takes to construct the preprocessed graph and the average number of edges explored, our current estimate of the search algorithm is 50 ms for a 4x4 array.

In this work, the A* algorithm can select a path using either fluid or electricity (fluid followed by electricity sequentially or vice versa). However, since the electric and fluidic and control hardware are independent of one another, future algorithms will allow both electricity and fluid to be delivered simultaneously to different subarrays, reducing the total time from origin to destination.

Finally algorithms will need to be developed to identify control command sequences for arrays where rows of actuators are mechanically connected in series. The challenge with this task is that there are multiple array configurations (nodes) that result in the same total row displacements.

References

[1] I. W. Hunter and S. Lafontaine, "Comparison of muscle with artificial actuators," presented at IEEE Solid-State Sensors and Actuator Workshop, Hilton Head Island, SC, USA, 1992.

[2] T. Mirfakhrai, J. D. W. Madden, and R. H. Baughman, "Polymer Artificial Muscles," *Materials Today*, vol. 10, pp. 30-38, 2007.

[3] S. M. Dutta and F. H. Ghorbel, "Differential hysteresis modeling of a shape memory alloy wire actuator," *IEEE/ASME Transactions on Mechatronics*, vol. 10, pp. 189-197, 2005.

[4] S. Govindjee and E. P. Kasper, "Computational aspects of one-dimensional shape memory alloy modeling with phase diagrams," *Computer Methods in Applied Mechanics and Engineering*, vol. 171, pp. 309-326, 1999.

[5] K. Ikuta, M. Tsukamoto, and S. Hirose, "Mathematical model and experimental verification of shape memory alloy for designing micro actuator," presented at Proceedings of the IEEE Micro Electro Mechanical Systems, Nara, Jpn, 1991.

[6] K. Kuribayashi, S. Shimizu, M. Yoshitake, and S. Ogawa, "Mechanical properties and control of shape memory alloy thin film actuator," *Seimitsu Kogaku Kaishi/Journal of the Japan Society for Precision Engineering*, vol. 64, pp. 413-417, 1998.

[7] C. Liang and C. A. Rogers, "One-dimensional thermo-mechanical constitutive relations for shape memory materials," *Journal of Intelligent Material Systems and Structures*, vol. 8, pp. 285-302, 1997.

[8] Y. Liu, D. Favier, and L. Orgeas, "Mechanistic simulation of thermomechanical behaviour of thermoelastic martensitic transformations in polycrystalline shape memory alloys," presented at 7th European Mechanics of Materials Conference on Adaptive Systems and Materials: Constitutive Materials and Hybrid Structures, Frejus, France, 2004.

[9] A. Bhattacharyya and D. C. Lagoudas, "Stochastic thermodynamic model for the gradual thermal transformation of SMA polycrystals," *Smart Materials and Structures*, vol. 6, pp. 235-250, 1997.

[10] M. H. Elahinia and H. Ashrafiuon, "Nonlinear control of a shape memory alloy actuated manipulator," *Journal of Vibration and Acoustics, Transactions of the ASME*, vol. 124, pp. 566-575, 2002.

[11] J. Jayender and R. V. Patel, "Modelling and Gain Scheduled Control of Shape Memory Alloy Actuators," IEEE Conference on Control Applications, Toronto, Canada, 2005.

[12] B. Selden, K.-J. Cho, and H. H. Asada, "Segmented binary control of shape memory alloy actuator systems using the peltier effect," IEEE International Conference on Robotics and Automation, New Orleans, LA, United States, 2004.

[13] K. J. De Laurentis, A. Fisch, J. Nikitczuk, and C. Mavroidis, "Optimal design of shape memory alloy wire bundle actuators," presented at IEEE International Conference on Robotics and Automation, Washington, DC, United States, 2002.

[14] R. Mukherjee, T. F. Christian, and R. A. Thiel, "An Actuation System for the Control of Multiple Shape Memory Alloy Actuators " *Sensors and Actuators A: Physical*, vol. 55, pp. 367-382 1996.

[15] K. T. O'Toole, M. M. McGrath, and D. W. Hatchett, "Transient Characterisation and Analysis of Shape Memory Alloy Wire Bundels for the Actuation of Finger Joints in Prosthesis Design," *Mechanika*, vol. 68, pp. 65-69.

[16] S. A. Mascaro and H. H. Asada, "Wet shape memory alloy actuators for active vasculated robotic flesh," IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003.

[17] V. H. Ebron, Z. Yang, D. J. Seyer, M. E. Kozlov, J. Oh, H. Xie, J. Razal, L. J. Hall, J. P. Ferraris, A. G. MacDiarmid, and R. H. Baughman, "Fuel-powered artificial muscles," *Science*, vol. 311, pp. 1580-1583, 2006.

[18] S. Mascaro and H. Asada, "Vast DOF wet shape memory alloy actuators using matrix manifold and valve system," ASME International Mechanical Engineering Congress, Dynamic Systems and Control Divison, Washington, DC., United States, 2003.

[19] L. Flemming and S. Mascaro, "Wet SMA actuator array with matrix vasoconstriction device," ASME International Mechanical Engineering Congress and Exposition, Dynamic Systems and Control Division, Orlando, FL, United States, 2005.

[20] L. Flemming and S. Mascaro, "Control of a scalable matrix vasoconstrictor device for wet actuator arrays," IEEE International Conference on Robotics and Automation, Rome, Italy, 2007

[21] J. Ertel and S. Mascaro, "Thermomechanical Modeling of Wet Shape Memory Alloy Actuator," ASME International Mechanical Engineering Congress and Exposition, Dynamic Systems and Control Division, Chicago, IL, United States, 2006.

[22] L. Flemming and S. Mascaro, "Analysis of Hybrid Electric/thermofluidic Control for Wet Shape Memory Alloy Actuators," ASME Dynamics Systems and Controls, Hollywood, CA, 2009.

[23] S. Russel and P. Norvig, Artificial Intelligence – A Modern Approach, 3rd ed., Upper Saddle River, New Jersey: Pearson Education Inc., 2010.