Evolving a self-organizing neuromechanical system for self-healing aerospace structures

Michael L. Rilee,*
L-3 Communications GSI, Goddard Space Flight Center, Greenbelt, MD, 20771

Steven A. Curtis† and Cynthia Y. Cheung‡
NASA Goddard Space Flight Center, Greenbelt, MD, 20771

and

John E. Dorband§
NASA Goddard Space Flight Center, Greenbelt, MD, 20771

NASA is developing a novel articulated truss built from a highly redundant, highly integrated network of actuators. Near-term implementations of this truss architecture make use of Addressable Reconfigurable Technology (ART) and can be centrally controlled as, for example, one's hand is directed by the central nervous system. Mid-to-far term implementations make increasing use of micro- and nano-technologies to allow truss systems to scale eventually through thousands of nodes and beyond into the realm of Super Miniaturized Addressable Reconfigurable Technology (SMART). Structures made from this material may take on shapes as required to meet mission requirements for deployment, storage, locomotion, shape control, and so on. One of the key applications of this technology is for nano- and pico-spacecraft.

Furthermore, the large number of redundant elements opens up many possibilities to address and mitigate faults and failures within the truss. Not only does this architecture provide physical reconfiguration, system control must be able to adapt to its new configuration. In this work, we describe a new control architecture for a synthetic neural system designed to meet this challenge. Genetic algorithmic evolution within supercomputer-based simulations allows the system components to situate themselves amongst each other and their environment. The synthetic neural system is based on composable behavioral units called Neural Basis Functions (NBF) that provide a way to unify low-level autonomic and high-level reasoning in a single operational architecture. Distributed systems fit naturally within this framework.

We describe fault modes of and recoveries enabled by the architecture and the results of our first attempts to construct a synthetic neural system based on NBFs with a focus on the self-organizing and self-healing properties of the system. We emphasize the scaling issues associated with the large number of nodes in nano-technology-based SMART structures and how distributed systems, e.g. multi-spacecraft systems, are controlled.

---

* Scientist, Space Sciences, Mail Stop 695.1.
† Branch Head, Planetary Magnetospheres Branch, Mail Stop 695.
‡ Scientist, Planetary Magnetospheres Branch, Mail Stop 695.1.
§ Computer Scientist, Department Name, Mail Stop 931.

I. Introduction

Every now and then a new tool is fashioned that changes the constraints that encompass our lives. The highly integrated neuromuscular mesh being studied by NASA at Goddard Space Flight Center is such a technology.

Many space systems concepts must be physically large in one or more dimensions. Antennas and magnetometer booms must be long; parasols, collecting optics, solar power arrays, or synthetic aperture arrays must be broad; scaffolding, habitats, and other volumes must encompass their contents. Even so, these systems must minimize the mass and volume and be robust enough to withstand the launch environment and still deploy. Deployable systems entail additional degrees of freedom that add risk to these space systems and much effort is spent accommodating and reducing this risk.

By coupling advances in materials and information processing and fabrication a dramatically new kind of structural material becomes possible. We have been pursuing the next step in the evolution of Massively Parallel Systems (MPS). Large-scale parallel supercomputers with thousands of computing nodes connected by high-speed communication fabrics are a current step. The next step is to integrate with such computing power massively parallel electro-mechanical systems (MP-EMS). With this integration we will have a physical structure that can maintain or change its shape as required for a particular application. Certainly there are challenges relating to the producibility and reliability of such MP-EMS with their multitude of moving parts. But in this work we do not address this particular issue other than to say engineering challenges posed by LCD laptop computer displays were similarly daunting in their early history. Instead we focus on the control of MP-EMS.

Our capability to process and generate information limits our ability to control systems with many degrees of freedom. Current electro-mechanical systems (EMS), and in the broader sense all engineered systems, are designed to rigidly control the degrees of freedom available to a system. That is the essence of design. With the advent of Micro-EMS (MEMS) and Nano-EMS (NEMS) the numbers of components and their interconnections can become very large. Control systems must scale well with this growth of complexity.

Information processing is not the only limiting factor in such systems. Limits also arise in the very coupling of the information component of a system to its more overt physical state through sensors and actuators. In particular, among such limits are the uncertainties and errors arising at the interface between the physical and information aspects of the system. For example, even if one has an excellent and detailed model of a physical system to be controlled, the degree to which the model can act as a predictive proxy for the real model decays with time. Roughly, the more complicated the system, the more rapidly reality departs from model prediction. From the field of data assimilation, we know that errors and uncertainties in sensor measurements guarantee discrepancies and inconsistencies between measurements and models. A goal of our work is to develop a system that operates, for the most part, without recourse to models, but which can still seek to achieve abstract goals formulated in terms of them. In short, we aim at low-level reliability achieving high-level purpose.

Components of our reconfigurable structures are addressable for the same reasons of control and function that devices in integrated circuits are addressable, hence the name for the most readily realizable form of our structural material, Addressable Reconfigurable Technology (ART). As MEMS and NEMS technologies come on line, Miniaturized ART (MART) and Super Miniaturized ART (SMART) entail.

Many of the standard MPS concepts apply to these ART/EMS. Are system components to be controlled via a Multiple Instruction – Multiple Data (MIMD) protocol or are they to operate via a Single Instruction – Multiple Data (SIMD) protocol? The chief difference between parallel computing systems and ART is that the physical side effects of our ART systems go beyond heating the environment – to changing their configuration and physically interacting with their environment.

As stated above, current practice in industrial robots is to severely constrain the degrees-of-freedom available to mechanical systems. One reason why industrial robots are built so robustly is to passively minimize oscillations and sway so that system state can be readily assessed. ART and its descendants, MART and SMART, aim to achieve the same level of precision and control but in an active instead of a passive fashion. Because an ART system is actively controlled to maintain system constraints, and function, it is inherently flexible and provides new options to adapt to changes in its environments or system capabilities.

In this paper, we describe possible applications that help motivate some of our architectural considerations. The architecture and implementation are discussed. An innovative synthesis of control, sense, and actuation are presented. Because the system actively controls itself and must adapt as new functions are added, we note that the step to self-healing systems is not large.

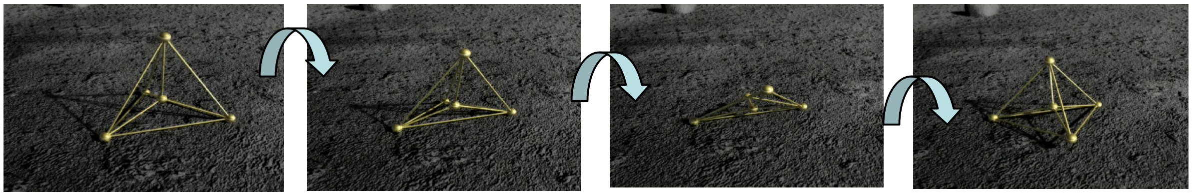

Figure 1. Sequence illustrating motion of the Tetrahedral Walker

II. Architecture

The ART, MART, and SMART set of architectures are the subject of six NASA patent applications. These three approaches to the construction of reconfigurable structures differ mainly in the fabrication technology used to construct their components with EMS, MEMS, and NEMS being the focus of ART, MART, and SMART respectively. ART is realizable in the near-term with prototype components and systems being developed as this is being written and is the focus of this work.

The fundamental element of the ART architecture is a strut that may be individually addressed and whose length may be controlled. Several extension/retraction schemes are possible and are currently being developed. The ends of these reversibly extensible struts are connected via swiveling mechanisms at nodes. Thus nodes become vertices and struts become edges in a geometrical structure. One can go through the entire set of tessellations of one, two, and three-dimensional space. The simplest is a single segment consisting of one strut between two nodes. For two-dimensions a triangle with three nodes and three struts is the simplest structure, for three-dimensions it is the Tetrahedron. For most applications, only the lengths of the struts are controlled, leaving the swivel-connections at nodes free to accommodate the constraints thereby set. However, the control system must still be careful not to over-constrain the system and push the components beyond their capabilities.

At this point we note that an ART structure consisting of a single Tetrahedron, or a 1-Tet, can display interesting behavior, including locomotion, see Figure 1. The packing scale of a 1-Tet is set by the physical limits of the retractability of the struts and the size of the nodes, see Figure 2. A payload can be mounted at the center of the frame provided by a 1-Tet by struts connected to the vertex nodes by four struts. Such a configuration is a 4-Tet because the original volume of the 1-Tet has been divided into 4 tetrahedral areas. A 12-Tet consisting of 12 tetrahedral sub-volumes and 26 struts and 9 nodes forms a duo-decahedron with remarkable reconfigurability. In fact, nodes and struts may be connected together in a tetrahedral mesh to form columns, sheets, or space-filling volumes that are actively reconfigurable.

Control becomes an increasingly interesting problem as the number of elements increase. New materials and fabrication methods will allow the fabrication of smaller and smaller ART tetrahedral, though we emphasize that ART structures can be made as large and robust as materials, actuation, and power allow. Current prototype studies are being developed using open-loop direct commands sent via wireless communication links. Wireless communication is convenient as long as the number of devices and their requirements for control do not overburden the communication spectrum: a variety of network protocols and architectures are conceivable. Commands drive behaviors, i.e. the lengthening and shortening of struts, that when performed with the correct timing produce a desired effect, for example, deployment of a structure as in Figure 2.

In the near-term, control of an ART structure is by commands sent as described above. In the most basic, centrally controlled schemes, the commands from the central processor must be consistent with capabilities of the underlying structure. For example, transformations that push nodes or struts through other nodes or struts are not allowed. An ART structure forms a geometrical pattern of interlocking constraints that must be respected as the structure reconfigures itself, lest it tear itself apart. This is a constraint resolution problem that has analogs in the design of Finite Element (FE) meshes and other fields.

This brings up a well-weathered problem in robotics, namely the consistency of the control system and the current state of the structure. For example, if an FE-like model of an ART structure is used to develop commands to reconfigure the structure, the validity of those commands depends on the validity of the model. As stated above, in many applications, robots are built to such high standards of rigidity, precision, and predictability that one has essentially total confidence that control and structure are consistent and that open-loop commanding suffices until something gets out of or crosses over a line. These point towards implementing closed-loop control, or even system autonomy, in which discrepancies are detected, isolated, and repaired.

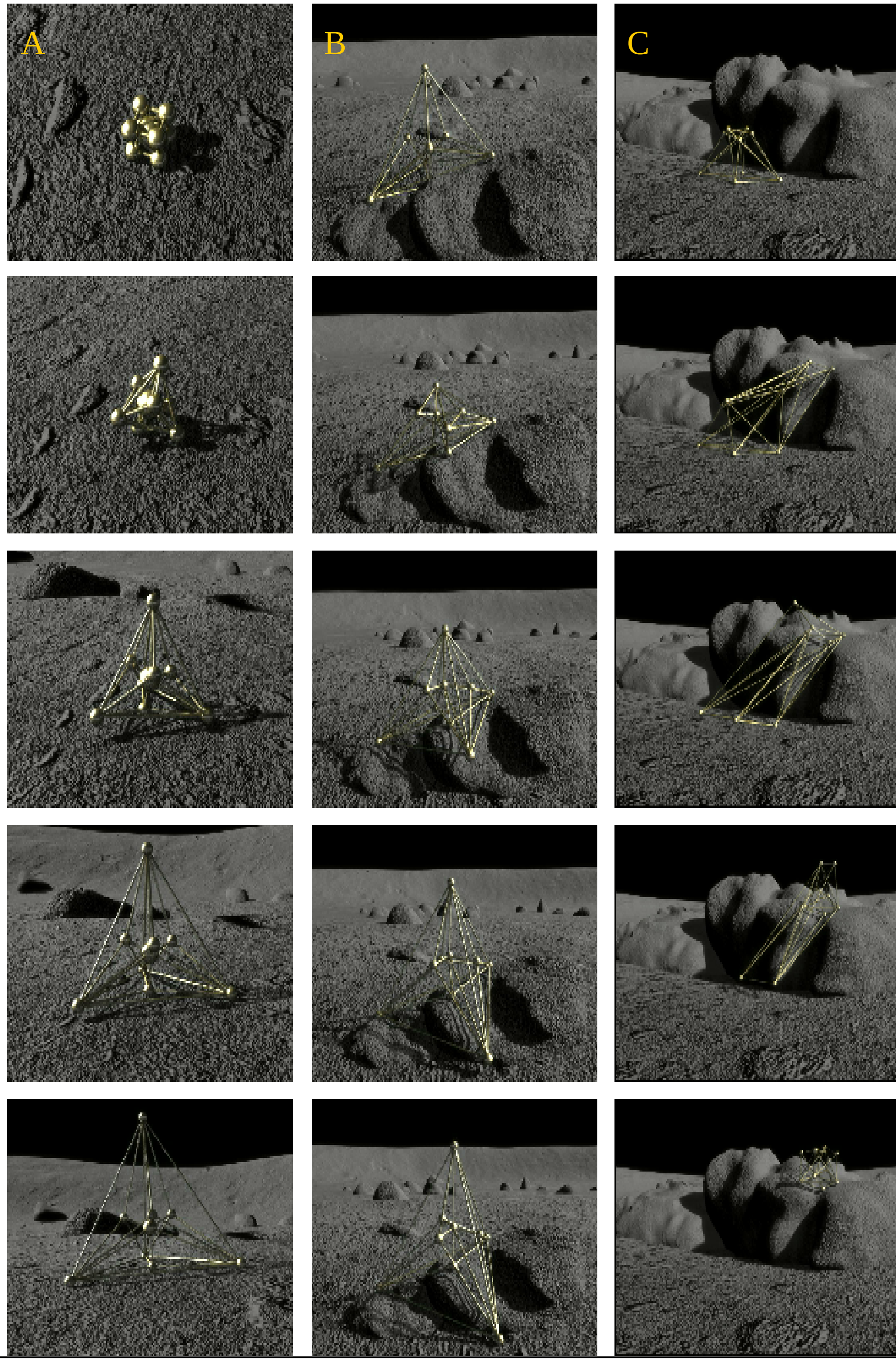

Figure 2. 12-Tet in action: A. Deployment, B, C. Encountering Obstacles

A hallmark of unintelligent behavior is the inability of a system to do just that: that is, detect, isolate, and repair discrepancies between a control system's actual and desired effects, between its understanding and reality. Indeed, there is a theoretical question as well that centers on what elements should make up a control system and how should they be determined. Furthermore, if one determines that a control system should contain an FE-like model of the ART structure, can it be said that the control system understands the ART structure well enough to come up with meaningful solutions to problems as they arise? There is a touch of pragmatism in this question, because there is a limit to the fidelity of modeling that a control system can maintain for diagnosis, prognosis, and control.

System designers, particularly of the software of the control system, can provide some level of detail but the consistency of this detail at the start and throughout the lifetime of a system has been a serious problem in Artificial Intelligence (AI) research for decades. Furthermore, there are real questions concerning whether the software, models, and related constructs that a system designer embeds in a control system are important or necessary for adaptable intelligent operation. An FE-like model for an ART structure seems like a reasonable way to represent and embed information about such a system's dynamics, but it does not scale well to NEMS-enabled SMART architecture or large-scale distributed systems such as the Autonomous Nano-Technology Swarm. Particularly when component and subsystem failures are accounted for, it becomes difficult to know a priori what sorts of models and capabilities are required. In short, specification of the system via explicit models and their dynamics is a huge problem that scales very badly. It works best when there are a few rather rigid degrees of freedom operating in controlled environments.

III. Hybrid Neural Systems for Control

These requirements for precision and reduction fit with a "traditional" symbolic approach to AI control. Patterns of symbols that represent the system and its future are manipulated until a satisfactory prediction of the future is achieved. Another approach eschews such explicit symbol manipulation and instead builds systems of actuators, sensors, and controls that are strongly, even physically, coupled to their environment resulting in robust smart servomechanisms. The dynamics of these servomechanisms are designed to maintain desirable states, or behaviors, in a kind of mechanical homeostasis. Tilden developed a series of robots on these principles that are simple, efficient, and robust. His controllers effectively govern the degrees of freedom of their systems, not by rigidly clamping them down and keeping them in well-controlled environments, but by nonlinearly coupling the degrees of freedom together so that healthy system states are sought or maintained by the system's own dynamics.

The nonlinear dynamical approach works at a "low-level" of abstraction whereas the symbolic approach operates at a "high-level". For simple systems, one can implement each approach using the tools of the other, however, for real systems with many degrees of freedom operating in real, irregular environments with uncertain futures each control approach suffers a scaling problem. The low-level approach does not scale up well to tasks where symbolic methods, i.e. expert systems, planning & scheduling conflict resolution, work. The high-level approach does not scale well down to systems with many degrees-of-freedom: the detail, fidelity, and consistency required for these models requires great investment and effort.

We have been studying a hybrid approach that seeks to couple both low- and high-level approaches to control. Restricting ourselves to strut actuation for the moment, low-level homeostatic control is maintained by smart servomechanisms acting as actuators in the structure. A single low-level controller might be linked to several struts that it drives in order to perform a particular behavior. This behavior may be moderated by inhibitory or biasing signals sent by other controllers, sensors, or high-level components. These signals are transmitted by an Evolvable Neural Interface (ENI), which acts as an active communication medium for system components. By active, we mean that the ENI may change either the destination or content of signals sent through it.

A variety of algorithms and implementation methods are available for the low- and high-level system, but their communication is mediated by the ENI, and thus their input/output interfaces are essentially neural. We speak of the High-Level Neural System (HLNS) and Low-Level Neural System (LLNS) even though the HLNS may have a significant non-neural component, and the LLNS may be mainly implemented in physical hardware. An HLNS may implement a Fuzzy Controller to emit commands to one or more LLNSs that in turn govern the behavior of the physical system itself. Together these three elements, HLNS, LLNS, and ENI, form a Synthetic Neural System (SNS). The ENI acts as a middleware layer that can allow different SNSs to be merged or even nested to operate together forming new SNSs. Thus Scalable SNSs (SSNSs) may be formed.

The fundamental unit within the SSNS framework is the Neural Basis Function (NBF), which is a software construct that spans the minimum software system required for autonomous adaptive control within this framework. The NBF is the embodiment of a goal and the behavior associated with its attainment. One HLNS and one LLNS connected by an ENI forms a simple NBF. The LLNS controls a number of physical elements of the system and maintains system health as in Tilden's work. The HLNS monitors the progress of the system towards system goals and sends commands to the LLNS through the ENI to perform actions that will make that progress.

The goal of our work is for the LLNS to control the many degrees of freedom that compose the physical system, in this paper, this means the ART structure. The LLNS produces information, e.g. sensor information, status, that it puts into the ENI. The HLNS picks up this information and uses it to monitor progress and to make plans. The HLNS will put its commands, data, and status information back into the ENI that are then picked up and used by the LLNS implementing closed-loop control.

IV. Solar Sail Example

To make this more concrete, consider the example of an NBF that seeks to bring a solar sail propelled spacecraft to a deep space rendezvous. A LLNS descendant from Tilden's heliostat is a natural means to control a Solar sail propulsion system consisting of a thin reflective stretchable membrane mounted to a large MART frame. Furthermore suppose an HLNS composed of a Fuzzy Expert System (FES) is linked to the LLNS through an ENI and is responsible for managing operations and executing correction maneuvers. The LLNS maintains the attitude and configuration of the sail to maintain the acceleration requested by the HLNS. Furthermore, suppose as time goes on, segments of the sail reflector wear out and fail or get stuck pointing in the wrong direction. The LLNS would automatically compensate for these by reconfiguring the sail: this is built into the smart servomechanism and it's coupling to the sail. Some status feedback may be communicated through the ENI to the HLNS, but this need not be extensive. In essence, the HLNS would not necessarily need to account for the degradation of the sail, a fact that simplifies HLNS construction and improves its performance.

If the FES required a course correction on its way to the rendezvous, it would send a new acceleration request to the LLNS through the ENI. It does not need to make detailed plans about which segments to open and close and how to configure the MART sail frame: those are low-level duties that are handled by the lower-level. Before we wrote that the ENI was an active medium and might change the request: this is a major difference from current practice with command protocols, but it is quite important. This change opens up possibilities for fault tolerance that do not exist today.

Suppose for example that our solar sail spacecraft also has a reaction wheel system controlled by its own LLNS that is also connected to the ENI and thus is also part of this attitude & propulsion NBF. Today, we would spend a great deal of time carefully constructing a command sequence for both the sail configuration and the reaction wheels to affect the required change in acceleration. With the NBF system, the request from the HLNS is fed to the ENI where it is transformed as it is transmitted to the two LLNSs, one for the sail frame (LLNS-SF) and the other for the reaction wheels (LLNS-RW). Both low-level systems can respond to the HLNS request with their status feedback going back onto the ENI. Thus the low-level control response to the HLNS request actually constructs a low-level feedback loop through the ENI, coupling the two LLNS systems. If something goes wrong with either low-level system, they try to compensate for themselves and each other, in real-time and without requiring high-level intervention, latencies, or detailed, explicit models. The system is thus designed to handle variance and faults by continuous monitoring and control, particularly by the LLNS. Because the LLNS are intimately and nonlinearly coupled with the physical system, they are not based on a small-perturbation linear control framework and can handle large deviations. Of course, stresses to the system would show up with the status feedback information that would likely drive the HLNS to initiate diagnostic functions and further remedial actions.

V. Evolution and Learning

Tolerance of variance, low-level active control, and closing feedback loops through the ENI are key to synthesizing a single NBF control system from the four elements in the preceding paragraph, the HLNS, LLNS-SF, LLNS-RW, and the ENI. However, it would seem that we have exchanged the problem of model specification and detailed control of a single monolithic HLNS (the current approach) with a host of distributed unspecified interconnections. In the current approach, a spacecraft controller, human or machine, would determine how the sail frame and reaction wheels are to behave and command them each appropriately: for the NBF approach it seems that the ENI knows how to do this by itself. The answer to this statement is that the ENI learns how to do this through a two-stage process, first during development and second during an initial fitting out during deployment of the actual system.

During the developmental stage, an NBF is embedded in a high fidelity simulation. The NBF components contain the expertise and design elements described above; for example, the FES of the HLNS contains the rules and procedures required to direct the progress of the system. The HLNS can use the latest state-of-the-art in planning & scheduling technology, however the symbols used in its processing are not matched precisely with the real-world constructs of the system. Instead, the ENI mediates the communication between the HLNS and the LLNS. The ENI is trained so that during operations this mediation is effective, efficient, and reliable. The simulations provide a safe sandbox for the NBF components to learn the meaning of the various inputs and signals moving through the ENI. An evolutionary approach, via Genetic Programming that modifies the structure of the ENI, is the key instrument of change during the developmental stage. A training regimen is provided to allow the NBF components to learn the details of the particular simulation in which it is embedded, an analogous process occurs when the NBF is deployed and fitted out to a physical system. The training and evolution regimen, as well as the high fidelity modeling required to train deployable systems, point towards very high capability computer systems. As stated above, the NBF SSNS when embedded in deployable systems would undergo another regimen of test and training, however with the benefit of many generations of development in simulation.

This approach seems to require a level of modeling rivaling that being proposed for the autonomous control of spacecraft and other systems. This is likely correct, but through the course of the NBF control system development it learns what it needs to do to get the job done. The NBF does not need to take a full scale, highly detailed, high fidelity model of the deployed system with it to achieve remote autonomous operations. Though if such a model is required, say for diagnostic purposes, it could be integrated with the system within a HLNS that could be linked to the NBF's ENI through the appropriate communications link, i.e. deep space communications for our solar sail spacecraft example.

Note that the components of an NBF need not be located in the same computer, nor even the same spacecraft. The system is naturally distributed with ENI forming the medium of communications. Thus the architecture admits a wide variety of communication rates and latencies. Part of this flexibility comes from the protection from the continuous control by the layer of LLNSs connected through the ENI.

The NBFs are also inherently flexible: they are learning systems, therefore they have internal degrees-of-freedom that are allowed to vary during training and evolution, but are generally only slowly varying during operations. These degrees-of-freedom are allowed to vary during operations to allow the system to adjust to changes in the system and its environment. In more drastic situations, components of NBFs may within certain bounds reconfigure themselves to adjust to drastic changes in system capabilities or functions. In this way we expect NBF-based SSNS to show a high degree of behavioral plasticity when placed in severe conditions. This "budding" of new behaviors is an outgrowth of the system's capability to self-situate during development and training.

VI. Conclusion

In this work we have described a new structural architecture and a new kind of control system to go along with it. The ready availability of small motors, microcontrollers, extensible struts, and low power high performance computers have led to encouraging preliminary results of our initial experiments with components a reconfigurable tetrahedral mesh: we are well on our way to realizing our first physical prototypes this year.

The individual components of these meshes are addressable and may be individually controlled from a central processor. In the near term, the systems will be simple enough that reasonably complete, explicitly instantiated, either by hand or with computer assistance, models can be used to determine command sequences. This explicit addressing is the core of ART technology. Such modeling works best for simple systems in uncomplicated and controlled and well-understood environments. For real systems, such modeling runs head-on into the representation and truth-maintenance problems that have challenged robotics for years. The impact of these issues is hardened when autonomous systems are considered. The level of detail required to accurately model systems well enough to predict and plan for their behavior can become excessive. To limit the size and scale of these models one turns to approximations, but deciding which details are most important can be daunting and will reflect the biases of the system implementer.

We address this old problem in robotics by synthesizing two important approaches to constructing control systems. Low-level nonlinear control systems are used to maintain active low-latency control over essentially the system as it acts in its irregular environment. High-level control systems based on more deliberative techniques and explicit, though perhaps more approximate, models deal with high-level goals and the overall progress of the mission. The high-level control systems are the domain of high performance statistical estimation and ruleprocessing software, whereas the low-level lives on neural signal processing and bandwidth with some analog system effects thrown in the mix. Rather than specifying, or even possibly overspecifying, the interactions of low- and high-level components, the interactions between these two levels are discovered through the Genetic sampling of the space of possible interfaces. There is still room for the experience of designers to play a role in the construction of NBFs, but our approach relaxes the constraints on the precision required to make such experience fit and reduces the consequences when it doesn't. NBFs feature internal degrees of freedom, a core genetic code and other parameters, that are allowed to vary during development, training, and system initialization. This built-in adaptability should translate to adaptability beyond development and training and into operations as well.

Generally, our approach draws from the arena of Genetic Programming (GP), and to a lesser extent artificial life research. Our architecture depends heavily on a library of low- and high-level functions to generate important behaviors in the physical system. This is possible because we do have some knowledge of the system and effective ways to manage certain aspects of the system. We have rather specific goals for the control of systems such as the ART mesh. These goals mainly concern control of ART mesh shape and movement. Due to our focus on interfaces and linking existing functions, our work resembles more the work of those whose GP focuses on the function level rather than GP from the bottom-up starting at the machine instruction level.

In addition to the adaptability of the neural control system, the physical structure, the ART mesh with its possibly large numbers of nodes and struts, also has a large number of degrees of freedom. These can provide physical flexibility to allow for the ART mesh as a whole to compensate for problems with the mesh, i.e. broken struts, missing nodes, stuck elements, etc.

Neuromuscular devices: future synthesis of hardware & control

The first experimental implementations of NBF-based control systems will be on computer workstations and Beowulf-class cluster supercomputers. For the developmental stage, high capability computing is required to generate the populations so that evolutionary dynamics and learning occurs.

The first deployable systems will also require high performance computing, because current AI techniques suitable for HLNS require computing performance that is not readily available onboard space systems. However, advanced computing systems such as the Fault Tolerant Commercial-Off-The-Shelf (COTS) Experiment proposed for NMP Space Technology 8 would readily allow NBF-based techniques to be employed. The parallel and distributed structure of the SSNS are an excellent match for advanced non-von Neumann computers such as the radiation hard Field Programmable Processor Array (FPPA) which is well suited for neural computation.

As stated above, our current prototypes use a central computer to remotely address and command actuators to reconfigure the systems. With Radio Frequency ID (RFID) technology, we can address large numbers of nodes. So for EMS and many MEMS-based systems, wireless control of the structures is adequate.

In the far-term, the control system and the ART mesh will be even more highly integrated, with computing elements built into the ART structure itself. Eventually we foresee extensions of something like FPPA technology being used to develop reconfigurable computing hardware that implements a parallel system of interconnected NBFs. These hardware NBFs would provide an undifferentiated computing capability that would specialize during the development stage mentioned above. EMS-based ART structures can be implemented today. In the future, ART, MART, and SMART structures may provide a programmable building material with applications varying from full-function and form limb replacement to planetary rovers.

References

Beiu, V. C., J. R. Frigo, and K. R. Moore, "On the reliability of Nervous (Nv) Nets," *in Proc. of Computational Intelligence for Modeling Control and Automation*, Vienna, Austria, 12-15 February 1999

Flynn, A. M. and R. A. Brooks, "Battling Reality," *AI Memo 1148*, MIT AI Laboratory, October 1989

Hasslacher, B. and M. W. Tilden, "Living Machines," *Robotics and Autonomous Systems: The Biology and Technology of Intelligent Autonomous Agents*. Editor: L. Steels. Elsevier Publishers, Spring 1995. (LAUR – 94 – 2636)

Koza, J.R. *Genetic Programming II, Automatic Discovery of Reusable Programs*, MIT Press, 1994

Russell, S. and P. Norvig, *Artificial Intelligence: a modern approach*, Prentice Hall, 1995

Sethson, M., M. Karlsson, and P. Krus, "Neuro-Mechanical Networks as an Architecture for System Design," in *Computational Synthesis: From Basic Building Blocks to High Level Functionality Papers from 2003 AAAI Spring Symposium, H. Lipson, E.K. Antonsson, and J.R. Koza, Cochairs.* Technical Report SS-03-02