

clocks, timers, interrupts, controls, a full duplex serial port, and +5V, +12V, and -12V power supplies.

The host interface to each processor card supports 8 Mbaud serial communications for downloading software and other system functions. The host can select which processor it connects with by writing to an 8-bit card-select register in the NBbus master. Once every millisecond, the NBbus master writes this processor card address out to the NBbus. The addressed card then enables serial communications to that processor until another card is selected. In order to allow processor cards to request a connection with the host, a single host request line runs to all of the processors. If this line is asserted, the host individually polls each card until it finds the one requesting a connection. Because the processor card address is only 8 bits wide, the system is limited to 256 cards. For implementations requiring a more than 10 cards, electrically isolated groups of cards can be logically connected by bidirectional bus repeaters. The overall system configuration is shown in Fig. 6.

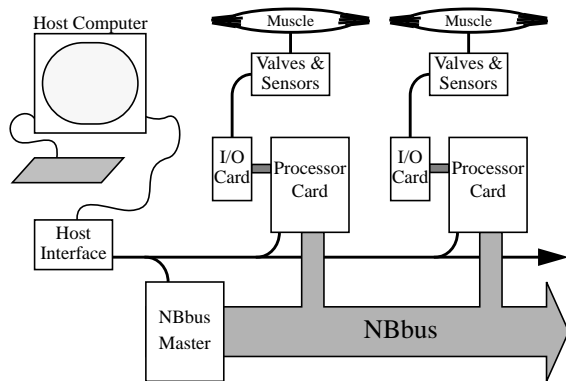


Fig. 6. SYSTEM BLOCK DIAGRAM

High level control generated by a host computer is passed to the Anthroform Neural Controller through a serial connection to the NBbus master. In addition, this serial interface allows for downloading meta-neuron programs into the processors and configuring the address sequence generated by the NBbus master.

## VI. CONCLUSION AND FUTURE WORK

The Anthroform Neural Controller is a distributed computation system based on the human moto-sensory system residing in the spinal cord. At the time of this writing, the controller hardware design was complete and approximately half of the processor card logic had been fully tested. With this tool complete, we plan to interface it to the Anthroform Arm Manipulator and begin our research of neural control theories relevant to the reflexive functions of spinal circuits. In addition to its intended use, we are also considering using this parallel computing architecture for controlling more typical robotic manipulators or for doing feedforward readout of standard neural network architectures.

## ACKNOWLEDGMENT

We gratefully acknowledge the efforts of Ching-Ping Chou, Pierre-Henry Marbot, and Professor Arun Somani for their help in defining the NBbus. We also thank Texas Instruments for their generous contributions.

## REFERENCES

- [1] S.C. Jacobsen, E.K. Iversen, D.F. Knutti, R.T. Johnson, and K.B. Biggers, "Design of the Utah/MIT dexterous hand," Proc. IEEE Intl. Conf. on Robotics & Automation, pp. 1520-31, San Francisco, April, 1986.
- [2] Cover of "Presence: Teleoperators and Virtual Environments," T.B. Sheridan, and T.A. Furness III, Ed., MIT Press, Winter 1992.
- [3] B. Hannaford, and J.M. Winters, "Actuator properties and movement control: biological and technological models," In "Multiple Muscle Systems," J.M. Winters, Ed., Springer Verlag, 1990.
- [4] H.F. Schulte, "The Characteristics of the McKibben Artificial Muscle," Appendix H of "The Application of External Power in Prosthetics and Orthotics," National Academy of Sciences - National Research Council, Washington, D.C., 1961.
- [5] E. Kandel, and J.H. Schwartz, "Principles of Neural Science," Elsevier, New York, 1981.
- [6] A.L. Hodgkin, and A.F. Huxley, "A Quantitative Description of Membrane Currents and its Application to Conduction and Excitation in Nerve," J. Physiology, vol. 117, pp. 500-544, 1952.

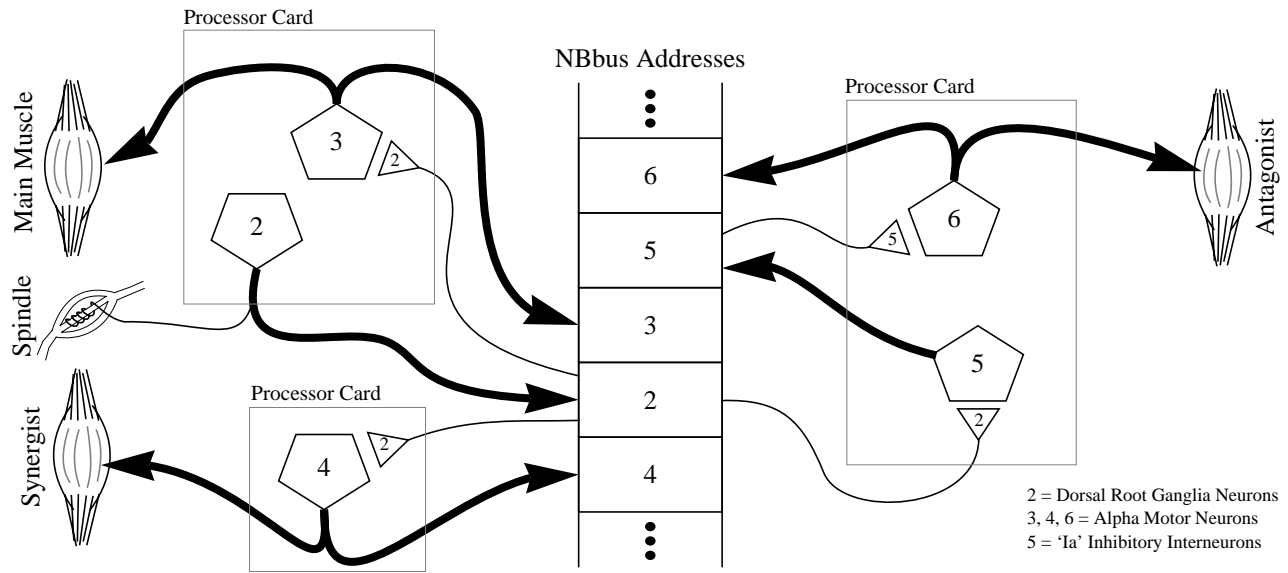


Fig. 4. THE NBbus ABSTRACTION OF THE STRETCH REFLEX

The stretch reflex presented in Fig. 2 is shown here as it would be implemented on the Anthroform Neural Controller. The specialized hardware interface to the NBbus allows the bus to be viewed as a single common storage space for meta-neuron outputs. The meta-neuron addresses in this figure correspond to the numbers in Fig. 2. The partitioning of meta-neurons among processor cards is somewhat arbitrary, the limitation being that sensory and motor meta-neurons must run on a processor card that offer the correct I/O connections to the Anthroform Arm Manipulator. For example, since meta-neuron #5 has no I/O connections, it could be hosted by any processor.

can be increased proportionally. In addition, the address sequence can be altered to increase the update rate of some connections over others. For instance, by using a sequence like 1,9,2,9,3,9,4,9, etc., address 9 would be updated every 2 microseconds while the others were updated every 1 millisecond.

## V. HARDWARE DETAILS

A circuit card on the NBbus contains a processor, NBbus interface hardware, and ports for connecting to input and output devices and a workstation for high level control and system functions. In this section we describe these interfaces as well as the important features of the processor we chose and how they all work together to form a complete system.

The TMS320C30 Digital Signal Processor (DSP) was chosen as the main processing unit for this system. It is specialized for performing high-speed floating-point calculations appropriate to our expected meta-neuron algorithms, and is supported by a C compiler and an extensive debugging system. The processor runs at 16 MIPS, allowing roughly eight typical meta-neurons (with 100 inputs) to be hosted by each processing card. The TMS320C30 also has three independent external busses allowing easy interfacing to local memory, the NBbus, and I/O devices as is shown in Fig. 5. The DSP also interfaces to a PC based software development system through a dedicated emulator port, and has another high-speed (8 Mbaud) serial port appropriate for communicating with a host computer.

The NBbus itself supports no I/O functions, so all sensory input and motor output must be interfaced through a meta-neuron on a processor card. To prepare the processor cards to provide a wide range of I/O formats, a generic I/O interface connector and bus were defined. It allows separately designed I/O daughter cards (e.g., analog to digital converters, digital to analog converters, PWM generators, or quadrature decoders) to be easily connected to a processor card.

The connector definition allows for daughter cards to be stacked in order to have more than one connected to each processor card. The daughter card I/O interface supports 13 bits of address, 32-bit data,

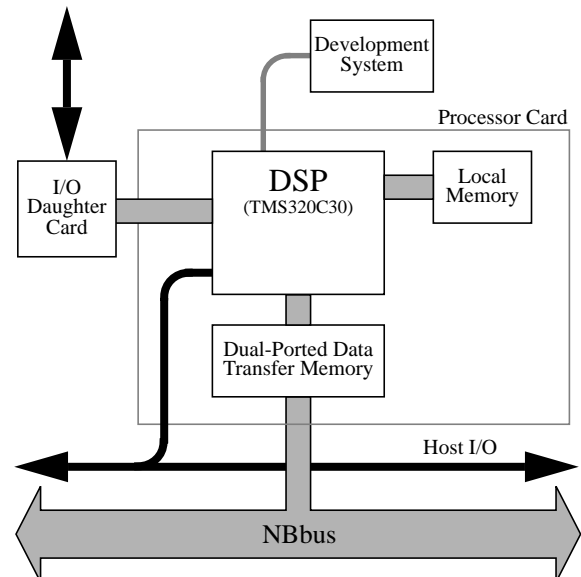


Fig. 5. PROCESSOR CARD BLOCK DIAGRAM

The TMS320C30 allows the user to develop meta-neurons in the C programming language. The processor also supports three independent busses allowing easy interfacing to local memory, the NBbus, and I/O devices.

degree-of-freedom, a total of 42 meta-neurons will be needed. If we allow for some additional spinal circuits and a few meta-neurons to interpret higher level control that would bring the total to around  $N=100$ . This works out to about 1 MIPS per meta-neuron for a system total of 100 MIPS. Although this is a little beyond the power of today's readily available processors, the computations can be easily partitioned among several processors since the computational load *per meta-neuron* is quite reasonable. The fully-connected network of 100 meta-neurons, for example, could be realized on just 10 processors running at 10 MIPS

This leaves us with the nontrivial problem of distributing all the meta-neuron output values to the proper processors where they are needed as inputs. A digital bus is appropriate, but based on the constraints developed in this section and the previous one it should be specialized to:

- support a fully connected network that is expandable to hundreds of meta-neurons distributed over a smaller number of processors;
- allow easy reconfiguration if a meta-neuron needs to be moved to different processors;
- allow each connection to be updated every 1 millisecond;
- support a programmable delay for each connection; and
- require a minimum amount of processor overhead.

These constraints lead to the definition of the Neural Broadcast Bus in the following section, and (in combination with the processor requirements above) to the final definition of the processing hardware in Section V.

#### IV. THE NEURAL BROADCAST BUS

The Neural Broadcast Bus, or NBbus, is a special purpose digital bus which forms a fully connected network of up to 1024 meta-neurons and allows them to be distributed arbitrarily over a maximum of 256 processors. The data values passed by this bus are 32-bit floating point numbers which represent the average activity of the individual neurons that a meta-neuron is modeling (Fig. 3).

In order to limit the number of physical wires required to implement the bus, a broadcast scheme was used. The scheme functions like this: a single bus master consecutively generates all the meta-neuron address numbers between 0 and 1023, repeating this cycle once every millisecond. Each address is sent out over a common set of address lines to every processor in the system. The processor that hosts<sup>6</sup> the addressed meta-neuron then places the meta-neuron output value on a common set of data lines. Any processor that needs the output of the addressed meta-neuron (in order to use it as an input for meta-neurons that *it* hosts) just reads the data bus to acquire the out-

put value. As long as a complete cycle of all the addresses is accomplished every millisecond, then all 1,047,552 individual input-output pairs are updated every millisecond resulting in a system capable of over 1 billion connections per second. In addition, the system is very flexible since a meta-neuron may be moved to a different processor without reconfiguring the bus. To relocate a meta-neuron, the new host processor need only know that meta-neuron's address and the addresses of its inputs.

There are only two major drawbacks to this scheme:

1. there is no provision for emulating variable delays between meta-neurons; and
2. each processor must be constantly monitoring the NBbus address lines so it can read and write the appropriate values for the meta-neurons it hosts.

The first problem could be solved by placing a variable depth FIFO (First In First Out) memory between the processor and the NBbus. However, to support 1024 inputs using 8-bit-wide memories would require 128 chips per processor! Our solution is to use software ring buffers as part of each meta-neuron processing algorithm. This will require roughly 10 additional instructions per input; doubling the processing power required per meta-neuron to:

$$\frac{(N-1) \frac{\text{inputs}}{\text{meta-neuron}} \times \left( 10 \frac{\text{calc-instr}}{\text{input}} + 10 \frac{\text{delay-instr}}{\text{input}} \right)}{1 \text{ msec}} \cong \frac{N}{50} \left( \frac{\text{MIPS}}{\text{meta-neuron}} \right)$$

However, emulating input delays in software will provide substantial flexibility for tuning meta-neuron circuits.

To reduce processor overhead in maintaining the NBbus we designed a special hardware interface to automate the reading and writing of meta-neuron output values. A dual-ported memory<sup>7</sup> is inserted between the processor and the NBbus. This memory contains a 32-bit word for each of the 1024 possible meta-neuron addresses and a 1-bit flag that notes which of these addresses are being hosted by the processor. As NBbus addresses arrive, a state machine implemented in a programmable logic array (PLA) first checks the flag bit, and then reads or writes the NBbus appropriately. All the processor needs to do is read the dual-port (without any timing constraints) and it will have the most recent update of a meta-neuron's output value. The processor can then run a meta-neuron algorithm on this and other inputs and write the new output value to the proper word in the dual-port. The state machine will take care of writing that output to the NBbus when the proper meta-neuron address next appears. From a software point of view, the entire NBbus can be thought of as a simple storage space for meta-neuron outputs. This abstraction is shown graphically in Fig. 4.

During neural emulations, the NBbus master will simply increment through all 1024 meta-neuron addresses, completing the cycle every millisecond. If the NBbus were to be used for other computation tasks this format can be changed easily. By reducing the total number of NBbus addresses the cycle rate

5. One alpha motor meta-neuron to drive the muscle, one dorsal root ganglion meta-neuron for feedback, and one 'Ia' inhibitory interneuron to ensure reciprocal inhibition of antagonist alpha motor meta-neurons.

6. To "host" a meta-neuron is to run that meta-neuron's algorithm and generate its output value.

7. A dual-ported memory has two independent sets of address, data, and control lines that each access the same memory words. If both ports try to access the same word at the same time then one port is required to wait until the other finishes its access.

human motor system. Each one processes a large number of inputs to generate an output signal that it passes on to many other neurons as their inputs. The number of these interconnections is very large, and this high connectivity is believed to be an important feature of biological control systems.

The interneuron communication medium imposes additional constraints on biological control systems. A complex electrochemical process is involved in a neuron firing its output and in propagating this signal to another neuron [6]. The time it takes to reset this process after firing is called the refractory period, and it lasts about 1 millisecond. Therefore, a fully activated neuron will transmit a 1 kHz pulse stream. This frequency reduces considerably during inhibition. Since a neuron can only fire once every millisecond, and since the individual firings are identical, the maximum rate of information flow in any neural connection is therefore about one value per millisecond. Another feature of neural communication is the propagation delay of neuron outputs. This delay is proportional to the length and type of the neuron, and can vary from nearly zero to as much as several tens of milliseconds for long sensory fibers. These connectivity and information flow constraints were central to the definition of our Neural Broadcast Bus (see Section IV).

The representation we chose for neuron emulation had to deal with yet another important feature of the human nervous system: its massive parallelism. Although thousands of neurons may be involved in even the simplest human reflex, our limited knowledge of their individual functions requires us to lump them into far fewer emulation units. For example, it is beyond our current understanding to model the many alpha motor neurons driving the many muscle fibers of a single muscle. We therefore combine these neurons into a single emulation unit<sup>3</sup>. We call these emulation units *meta-neurons*; they may model any number of actual neurons that work in parallel. In order to minimize the loss of information in the abstraction, the values passed between meta-neurons should represent the average activity of the individual neurons they emulate. The meta-neuron concept is illustrated graphically in Fig. 3.

### III. ARCHITECTURAL ISSUES

The design philosophy we have adhered to in the development of this project is

1. to design a human arm replica and control system that is based on known facts from human biomechanics and neurophysiology; and
2. to create engineering solutions to fill gaps that exist in this knowledge and make the system tractable.

The known facts and engineering solutions presented in the previous section go a long way toward defining the Anthroform Neural Controller hardware. In this section we outline the remaining system constraints; namely the end uses and resulting computational requirements of our system.

The Anthroform Neural Controller must operate as a test bed for the development of meta-neuron algorithms. This re-

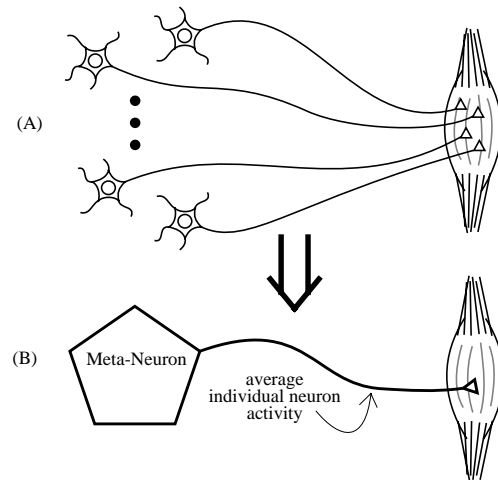


Fig. 3. THE META-NEURON ABSTRACTION

Hundreds of individual neurons often work in parallel to perform a single macroscopic function. The hundreds of individual alpha motor neurons which drive a muscle (A), for instance, can be modeled by a single meta-neuron (B) that is responsible for driving that muscle. To minimize the loss of information in the abstraction, the output of a meta-neuron represents the average activity of the individual neurons it emulates.

quires that the processor itself:

- be capable of running complex algorithms that generate new meta-neuron output values at a rate of 1 kHz (approximating the upper bound of the biological system);
- be readily programmable in a high level language (for development of meta-neuron algorithms); and
- be able to interface with the Anthroform Arm Manipulator (for sensory inputs and motor outputs).

In order to understand the computational requirements of the above constraints we must first evaluate roughly how much computation per meta-neuron is required. We estimate that meta-neurons will require on the order of 10 “typical” processor instructions per input in order to calculate the output value. If the system is to fully interconnect  $N$  meta-neurons<sup>4</sup>, each one will have  $N-1$  inputs. Since each meta-neuron output must be calculated from these inputs once every 1 millisecond, the processing load will be approximately:

$$\frac{(N-1) \frac{\text{inputs}}{\text{meta-neuron}} \times 10 \frac{\text{instr}}{\text{input}}}{1 \text{ msec}} \cong \frac{N}{100} \left( \frac{\text{MIPS}}{\text{meta-neuron}} \right)$$

$$1 \text{ MIPS (Millions of Instructions Per Second)} = 10^6 \left( \frac{\text{instr}}{\text{sec}} \right) = 10^3 \left( \frac{\text{instr}}{\text{msec}} \right)$$

In order to implement the stretch reflex of Fig. 2 we will need three meta-neurons for every muscle<sup>5</sup>. Assuming 7 degrees-of-freedom, and an antagonist pair of muscles for each

3. A similar abstraction was used in Fig. 2 where a single white pentagon represented many alpha motor neurons working in parallel.

4. To prevent artificially limiting the number of interconnections, we must support a fully-connected network in which each node has a direct connection to every other.

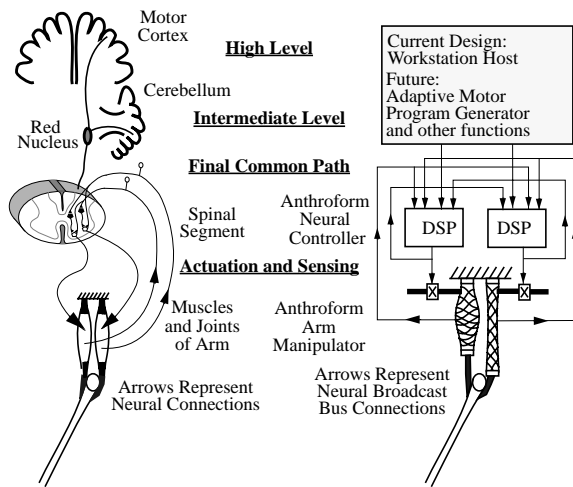


Fig. 1. FROM BIOLOGY TO THE ANTHROFORM BIROBOTIC ARM  
The current design of the Anthroform Neural Controller will model the activity of the human nervous system residing in the spinal cord. Higher level control will be provided by an interface to a workstation.

the physical structure of the biological system.

In order to free the processors for execution of neuron models, connections between simulated neurons are maintained in hardware over a special purpose bus that we call the Neural Broadcast Bus, or NBbus. At first, these neuron models will provide only low-level reflexive control; higher level control signals will be provided by an NBbus interface to a workstation. However, in the future we hope to better model high level control with an adaptive motor program generator. The overall system structure and its mapping to the biological system is shown in Fig. 1. This paper focuses on the development of the Anthroform Neural Controller hardware. The software neuron models that run on this hardware will be the subject of future research.

## II. NEUROPHYSIOLOGY BACKGROUND

In this section we introduce some important features of a typical neural circuit<sup>1</sup> in order to gain an understanding of the end use of the Anthroform Neural Controller. We then go on to outline the structural characteristics of the human nervous system and how they guided the development of our design.

The stretch reflex<sup>2</sup> shown in Fig. 2 is an important neural circuit, and one we plan to emulate on the Anthroform Neural Controller. This spinal reflex contributes to posture maintenance by resisting muscle length perturbations. It functions as follows: when the joint is displaced by an external force, the resulting elongation of the many spindles within a muscle causes the associated *afferent* dorsal root ganglia neurons to fire. These signals are propagated up to a spinal cord segment where they activate the hundreds of *efferent* alpha motor neu-

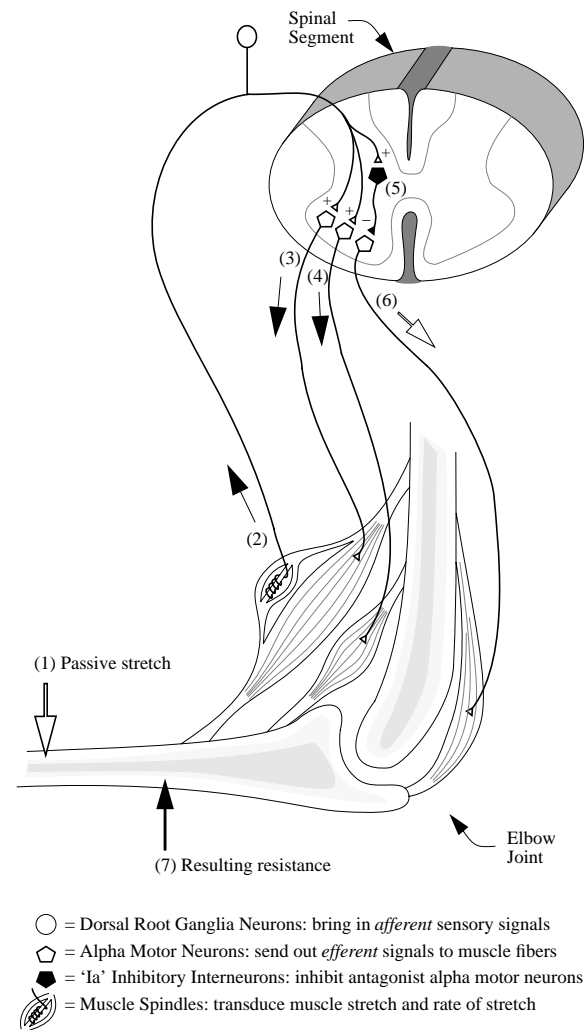


Fig. 2. THE STRETCH REFLEX

Stretching a muscle with an external force (1) causes the spindles to activate neurons in the dorsal root ganglia (2). These neurons then activate the alpha motor neurons of the same muscle (3) and those of the synergist muscle(s) (4) as well as activating the 'Ia' inhibitory interneurons (5). The interneurons then inhibit the alpha motor neurons of the antagonist muscle(s) (6). The stretch is thereby resisted (7) as the flexor muscles tighten and the extensor muscle relaxes. Each of the neuron symbols in the figure represents hundreds of individual neurons acting in parallel.

rons associated with the perturbed muscle. In addition, the *afferent* signals activate the motor neurons of synergist muscles and the 'Ia' inhibitory interneurons. The interneurons in turn *inhibit* the hundreds of alpha motor neurons of the antagonist muscles. The activated and inhibited motor neurons then pass their signals to the appropriate muscle fibers within the muscles. The net result is to contract the stretched muscle and its synergists, and relax the antagonist muscles, thereby resisting the perturbation. As a result of being processed locally in the spinal cord, the reaction time of this reflex is on the order of 40 milliseconds (as opposed to 120+ milliseconds for signals passing through conscious control in the cortex) [5].

The neurons described above are the basic foundation of the

1. We use "neural circuit" to mean a biological network of neurons since "neural network" now connotes a computation method.

2. Informally referred to as the "knee jerk" reflex.

# The Anthroform Neural Controller: A System for Detailed Emulation of Neural Circuits

Ian MacDuff, Steven Venema, and Blake Hannaford

Biorobotics Laboratory, Dept. of Electrical Engineering, FT-10

University of Washington, Seattle, WA 98195

Email: blake@ee.washington.edu

**Abstract**—Existing robotic manipulator and controller designs compare unfavorably to the human arm when performing tasks in unstructured environments. So-called “anthropomorphic” designs have tried to improve robot performance in these domains by replicating the kinematic structure of the human arm while continuing to use traditional actuation and control techniques. In this paper we describe a versatile parallel computing architecture for emulating the spinal circuits of the human nervous system. When used in conjunction with a dynamically realistic replica of the human arm, this controller will provide a versatile tool for studying human moto-sensory control. The design is based on the structural constraints of the nervous system, and consists of a special purpose digital bus which implements connections between simulated neurons running on TMS 320C30 digital signal processors (DSPs). The system supports up to 1024 individual neuron models, each connected to every other at least once every millisecond. These neuron models may be distributed over as many as 256 processor circuit cards, each supporting an interface for high level control from a host and another for input and output functions.

## I. INTRODUCTION

Standard industrial robot manipulator arms have been designed for precise positioning in highly structured and constrained environments. The development of robotic and telerobotic systems for use in unstructured environments has therefore been a significant challenge, and not altogether a success. Because the human arm performs so well in these domains, “anthropomorphic” designs have become increasingly popular. Such designs have traditionally focused on replicating the kinematic relationships of manipulator links and joints, and have neglected actuator, dynamic, and control aspects [1][2].

The Anthroform Biorobotic Arm project is an attempt to build an anthropomorphic robot such that both the manipulator and its controller are based on current knowledge of human biomechanics and neurophysiology. The project is divided into two subprojects; the Anthroform Arm Manipulator and the Anthroform Neural Controller. The primary goal in the development of these subprojects is to produce an accurate test bed for studying theories of neural control. In addition, however, the Anthroform Biorobotic Arm will represent a very humanlike manipulator with “natural” kinematics and dynamics, allowing possible applications ranging from teleoperation to prosthetic limbs.

Successful control of the Anthroform Arm Manipulator by a neuron model implemented on the Anthroform Neural Con-

troller will be a necessary (but not sufficient) condition to show that the model is indeed descriptive of a function of the human motor control system. However, this is only true to the extent that the arm’s dynamics accurately model the human musculoskeletal system. For the Anthroform Biorobotic Arm to be useful as a test bed for neural control models, it is crucial that it have a high level of biomechanical accuracy. To attain this accuracy, Prof. Jack Winters of Catholic University of America [3] is using the elements of Table 1 below to develop our hu-

Biological System	Anthroform Arm Manipulator	Comments
Bones	Fiberglass links	Cast from real human bones
Joints	Joint replacements	Developed for reconstructive surgery
Tendons	Nylon/fabric braid	Specially tuned for dynamic properties
Muscles	“McKibben” pneumatic artificial muscles	Shorten and widen when air pressure inflates a rubber bladder surrounded by a helical nylon mesh [4]

Table 1: ELEMENTS OF THE ANTHROFORM ARM MANIPULATOR

man arm replica. All of these elements are attached at anatomically appropriate positions, and each has static and dynamic characteristics which make it a good model of the biological equivalent. Together they form a highly anthropomorphic 7 degree-of-freedom manipulator.

The Anthroform Neural Controller is the control technology counterpart to the Anthroform Arm Manipulator. Through a combination of specialized hardware and software, it serves to simulate the activity of spinal neurons and their interconnections. The major elements of the controller architecture are shown in Table 2 below. The hardware design constraints were

Biological System	Anthroform Neural Controller	Comments
Neuron cell bodies	Software neuron models	Run on TMS320C30 digital signal processing (DSP) chips
Axons and dendrites	Neural Broadcast Bus	Special purpose digital bus which forms fully connected network of neuron models

Table 2: ELEMENTS OF THE ANTHROFORM NEURAL CONTROLLER

defined by the known (and relatively invariant) structural constraints of the human nervous system (e.g., the connectivity and rate of information flow). In this way, the constraints imposed by the hardware in our system match those imposed by