The Programming and Training of a Multitasking Sensory Motor Controlled EVA-Robot Operating In a Zero-g Environment

by Alan Rosen and David B. Rosen

1. INTRODUCTION

This presentation is about the programming and training of a multi-tasking robotic system designed for zero-g, EVA-locomotion, inspection, maintenance, and repair functions. The zero-g EVA-robotic system is based on a fully designed multitasking system that has been described in the Neural Network Journal and Springer's LNCS [1], [2].

[1] Sensory Motor Control by Reverse Engineering Biological Modalities"
http://www.mcon.org/submtd/nnj_nng1886_121507.pdf
[2] An Electromechanical Neural Network Model of the Human Body and Brain
http://www.mcon.org/submtd/ICON_IP_RosenRosen71506 web.pdf

To illustrate the affect of the newly-designed EVA-robot on the programming-training of the brain-controller, a conceptual design of an EVA-robot is presented. The equipment compartment of the EVA-robot is modeled after the AWIMR Project [3], and 2-arms, 4-legs and 2-cameras are modeled after the Neuronal Correlate of Modalities (NCM)-robot [1], [2], [4].

[3] Wagner, R. & Lane, H. (2007) Lessons Learned on the AWIMR Project. Presented at the Space Robotics Workshop, ICRA. April 14, 2007.

[4] "A robotic optical circuit that generates 3D-visual images: a robotic solution to the inverse optics problem of visual seeing," published in the Neural network Journal and available for viewing at
http://www.mcon.org/submtd/nnj_nng1911_121507.pdf

1.1 The EVA-Robotic Body

Figure 1 shows the conceptual design of the EVA-robot modeled after the NGC-AWIMR project and the NCM robot. The robotic motion of the arms, legs, and cameras are facilitated by the 40-joints system shown in the Figure. The motion of all 40-endjoints are controlled simultaneously by the EVA-robotic controller. Quadrilateral symmetry has been designed into the 3D-visual sensors and the 2 robotic arms with respect to each pair of 2-adjacent legs in the 4-legged system. The robotic torso, mounted on a rotating platform located on top of the NGC-AWIMR compartment, may take on any of 4-positions symmetrically located between each pair of adjacent legs mounted on the same compartment. In each of the 4-quadrilaterally symmetrical regions, the robotic torso may be associated with the 2-adjacent legs, 2-arms mounted on the torso, the robotic controller mounted on the torso and the 2-camera visual system mounted on the controller. The robotic system in each quadrilaterally symmetrical region is similar to the NCM-multitasking robotic system described in [1] and [2].

1.2 The Sensors

In each of the 4-quadrilaterally symmetrical regions of the EVA-robot the tactile sensors and the 3D-visual sensors are identical to those described in the Neural network Journal [4] and [1]. These sensors are configured to form the tactile and visual internal coordinates (located within the controller) that are a reflection of the robotic body and the visual space in which the robot is operating.

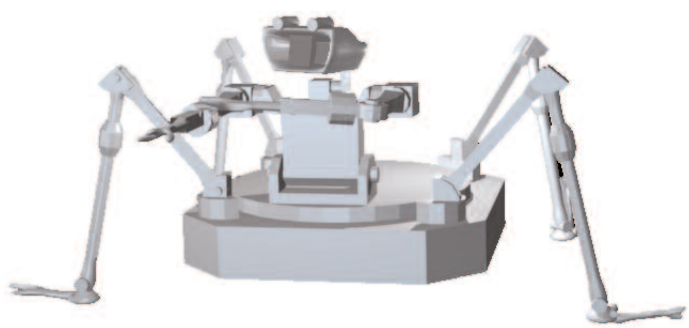

Figure 1. The conceptual design of the EVA-robot modeled after the NGC-AWIMR project and the NCM robot.

1.2.1 The Tactile Sensors

Made up of fiber-embedded pressure transducers, the tactile sensors are uniformly distributed throughout the EVA-robotic body. These sensors detect any physical contact taking place on the robotic body and measure the reaction forces exerted on the limbs during locomotion, and hand manipulation tasks. In each of the 4-quadrilaterally symmetrical regions of the EVA-robotic body, the tactile sensors that facilitate the design of the coordinates of the robotic self and the near space around the robotic self are identical in all respects to the sensors described in "Sensorimotor control by reverse engineering the biological modalities: reverse engineering the human body and brain" published in the Neural network journal and available for viewing at:
http://www.mcon.org/submtd/nnj_nng1866_121506.pdf

Coordinate Frames in the 3-D Space in Which the Robot is Operating	
S3-self	The fixed location of mechanoreceptors distributed on the total robotic body
S3-near	A set of indexed locations that cover the space surrounding the body
S3-space	Defined as S3-self plus S3-near
S3-end joint	Indexed location in a sub-space defined by the range of motion in an end-joint in the S3-near space
Coordinate Frames Within the Controller	
S-self	A reflection of the S3-self space. Indexed receiving nerons within the controller that define the total robotic body.
S-near	A reflection of the S3-near space. Indexed receiving nerons that define the space surrounding the robotic body.
S-space	Defined as S-self plus S-near
S-end joint	A reflection of S3-end joint space. Indexed locations in a sub-space defined by the range of motion of the end-joint in the s-near space (in a Nodal Map Module within the controller).

Table 1. Mathematical designation of the various coordinate spaces.

1.2.2 The Visual Sensors

In each of the 4-quadrilaterally symmetrical regions of the EVA-robotic body, the 3D-visual sensors have been designed to be similar in all respects to the sensors described in "A robotic optical circuit that generates 3D-visual images: a robotic solution to the inverse optics problem of visual seeing," published in the Neural network Journal [4] and available for viewing at:
http://www.mcon.org/submtd/nnj_nng1911_121507.pdf

1.3 Quadrilateral symmetry

The quadrilaterally symmetrical system is advantageous for 4 reasons:

1. Programming the self location and identification functions, required in order to define the external coordinate frame within the controller, need be performed for only one of the 4 quadrants.
2. The design of the binocular visual system and the calibration procedure for one quadrant is identical to the calibration procedure described in the Neural network Journal [4]. (It is fully applicable to each of the remaining 3-quadrants).
3. The training-programming procedure performed in any one quadrant is applicable to the programming performed on each of the 3 remaining quadrants.
4. The EVA-robot has a 360 degree surveillance capability and may perform locomotive functions in any one of 8 directions without re-orienting the equipment compartment.

2. THE EVA-ROBOTIC CONTROLLER

The EVA-robotic controller is a modified NCM-controller [1], [2], re-configured only to conform to the range of motion of all the moveable parts of the robot in the coordinate frame in which the EVA robot is operating. The coordinate frames within the controller, which are reflections of the coordinate frames in the 3D-space in which the robot is operating, are shown in table 1. In the following sections the various coordinate spaces that are reflected into the robotic controller will be referred to by the mathematical designators given in table 1. A block diagram of the EVA-robotic controller is shown in Figure 2.

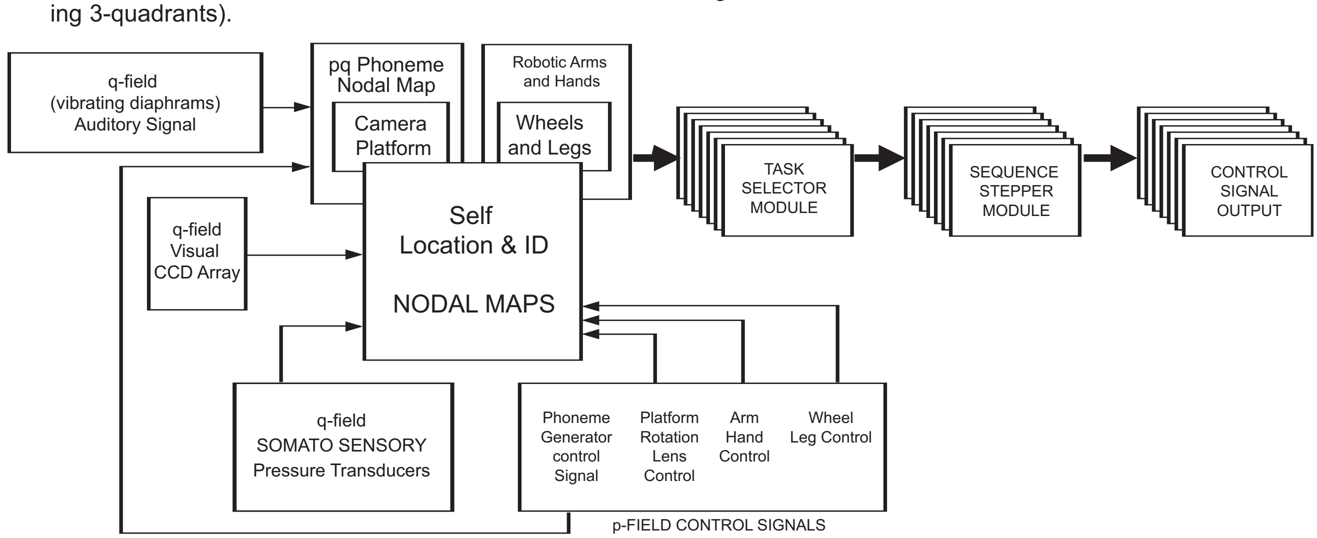

Figure 2. A hierarchical array of modules. All the Nodal Map Modules, Task Selector Modules, Sequence Stepper Modules and Control Signal-output Modules (associated with each joint in the body), operate simultaneously during each frame period.

2.1 The Operation of the EVA-Robotic Controller

The robotic controller is made up of modules that operate in the various endjoint subspaces shown in table 1.

2.1.1 The tactile neural networks

A set of neural net works located within the controller are used to determine the indexed coordinate location of the robotic self in the S-self space, and the indexed coordinate location of flailing limbs in the S-near space. The origin of the coordinates, identified as the center of Mass (CM) of the system, is located on top of the NGC-AWIMR platform, at the center of the torso-yoke shown in Figure 1.

2.1.2 The Nodal Map Modules (NMM)

A NMM is made up of an array of microprocessor based storage slots known as nodes. The set of nodes that define the S-space within the controller, are a reflection of the coordinates of the 3D-space, the S3 space, in which the robot is operating. Each storage slot-node represents an indexed location of a coordinate located in the 3D-space, and is used to store q-input signals and a set of p-output signals, known as table line entries. The symbol q is used to denote either tactile or visual sensory signal inputs. The symbol p denotes a control signal emitted from the controller, that may be applied to any motor-joint in order to generate a motor action, generally a single nodal transition of the endjoint.

2.1.3 The qi-position of an endjoint in each of the 40-Nodal Map Modules

The EVA robotic system is made up of 40-NMMs, with one NMM assigned to each joint in the robotic system.The position of the end of each limb with respect to the position of the joint is called the qi-position of the endlimb. In the 40-joint NMM-system, 40 qi-positions are recorded, one position in each NMM (The 3D-position of the end joint in the S3-endjoint space is reflected to an indexed storage slot in the S-endjoint space within the controller). As the end-joint moves in the S3-endjoint space, the qi position is recorded via an inverse kinematics intermediate circuit, onto the exact indexed locations of the S-endjoint space. The number of nodes present in the S-endjoint space is determined by the range of motion of the endjoint in the S3 endjoint space. There are 2-additional input signals that may be stored within the nodes present in any NMM. The first is a Task-initiating Trigger (TT), an activation signal related to the pre-planned final position or the goal, q-final, of the controlled trajectory of motion of q-initial. That is, the robotic controller may be required to generate a controlled trajectory of motion of any given endjoint from q-initial to q-final. The second parameter is the magnitude of the sensory signal stored at any indexed location. For example a tactile signal that forms a TT may also store the magnitude of the pressure on any given pressure transducer, as well as its location. Visual sensors may store all the parameters of the photometric signal in an indexed storage slot and a photometric pattern, stored in an array of indexed locations, may be detected either as TT-patterns or obstacle avoidance TT-patterns.

2.2 Inverse kinematics and the intermediate circuit for determining qi-positions: Primary and secondary NMMs.

The primary NMMs are associated with free-ending joints. These are joints where the qi location of the endjoint interacts only with the environment. The secondary NMMs are associated with intermediate joints. These are joints where the qi location of the intermediate joint interacts with another joint in addition to the environment. The distinction between primary and secondary NMMs is important in the training process. Generally the free-ending joints, associated with primary NMMs, are trained explicitly to move the free-ending joint to some pre-determined final position, qf-position. All the intermediate joints, the secondary NMMs, are trained implicitly by inverse kinematics, to take on intermediate qi-positions that are consistent with the trained primary qi to qf-location. The position of the free-ending finger endjoint with respect to the center of mass of the equipment compartment is shown in Table 2. For example, the position of the index finger endjoint relative to the first index joint, shown in Figure 1, is recorded in the primary NMM. Simultaneously, the positions of all the intermediate joint-qi's are recorded in the secondary NMMs shown in Table 2.

Free Ending and Intermediate End-Joints in the Nodal Map Modules
Free Ending Finger Joint in the Primary NMM
• qi-indexed location of the finger end joint with respect to the first joint
Intermediate Endjoints Positions in the Secondary NMM
• The position of the second joint with respect to the wrist
• The position of the wrist with respect to the elbow
• The position of the elbow with respect to the shoulder
• The position of the shoulder with respect to the torso
• The orientation of the torso with respect the the center of mass at the torso pivot (just above the platform)
Note: The torso can take on 3-positions with respect to the center of mass of the system: A folded position (zero degrees), mid 45 degree orientation and a verticle position (90 degrees) Thus, the inverse kinematics of the arm positions are dependant of the torso position. However, the foot postions my be independant of the torso postion, and depend only on the position of each leg with respect to the equipment platform (as determined by the the postion of the center of mass).

Table 2. Location of the endjoints in the primary and secondary Nodal Map Modules. The angular displacements of the motors at each joint determines the secondary location of the endjoint in the secondary Nodal Map Modules.

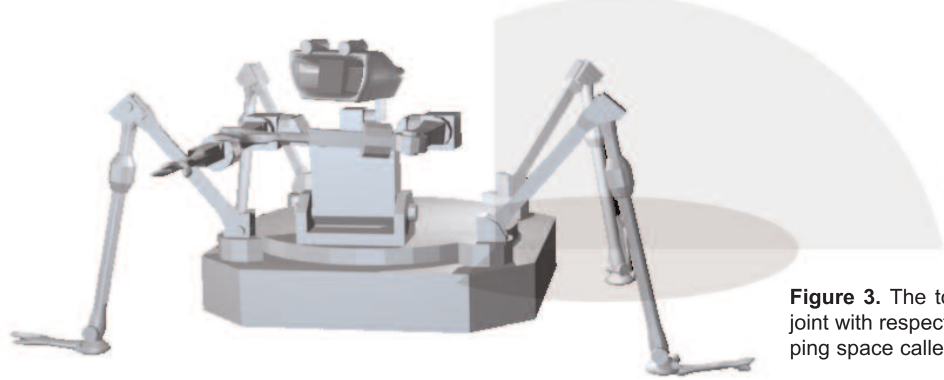

Figure 3. The total angular displacement of each end joint with respect to the joint determines the nodal mapping space called the S3-end joint space.

2.3 Tracking the qi-position of the free ending joint: qi in the primary and secondary NMMs

In order to connect a signal originating at a q-initial position at the tip of a robotic finger to its corresponding point in the near space of the internal nodal map, one must convert the signals received from a set of angle measuring transducers at all the intermediate joints into the qi position which is stored into the corresponding storage-slot (indexed location) within the controller. In the EVA-robotic model the motion of each joint may have a maximum of 3-degrees of freedom. If we assume one motor per degree of freedom, then an angle measuring transducer may be used to measure the torque-generated angular displacement at the shaft of each motor. The 3-angular displacements at each joint determine the position of the robotic endjoint with respect to that joint. The position of the tip of the finger is a function of the angular displacements of all the intermediate joints between the tip of the finger and the Center of Mass of the EVA robot (see Table 2). The intermediate joint-J position is a function of all the angular displacements at all the joints between the intermediate joint-J and the center of mass of the system. Table 2 shows the corresponding positions of secondary intermediate joints as a function of the angular displacements of all the joints between robotic finger and the center of mass of the system.

In the design of the robotic arm, an intermediate circuit, associated with each joint on the robotic body, is required in order to convert all the angle measurements to a q-initial location in the internal coordinate frame. The intermediate circuit then transmits the q-initial signals to the indexed nodal locations in the S-intermediate joint space.

A complete and separate intermediate circuit is associated with each robotic joint. The range of motion of each end limb, and the possible q-initial locations covered by the end limb depend on the number of joints between the end limb and the center of mass of the system. For example, Figure 3 shows a robotic limb end-joint that defines a conical region with apex at the hip. The length of the cone is determined by the total length of the leg from the hip to the foot, and conical angle is determined by the angular range of motion of the leg with respect to the hip. The robotic elbow end-joint covers a smaller conical region that is determined by the length of the upper arm from the shoulder to the elbow. The angular inputs from the angle measuring transducers located at all intermediate joints between the end-joint and the center of mass are applied to the intermediate circuit that is associated with the end-joint. The q-initial position of a robotic end-joint is a function of all the inverse kinematics angular positions of all the intermediate circuit between the end joint and the center of mass of the system [5].

In a fully designed system, the designer of the system may measure, calculate, or test (by inverse kinematics techniques) the angular positions of all intermediate joints, when the robotic finger end-joint is moved through the itch-goal directed trajectory. Thus all the intermediate circuits and positions of end-joint in the associated topographic mappings may be simultaneously determined and programmed in the process of moving the primary end-joint.

2.4 The Task Selector Module (TSM)

There are 2-TSM's in the EVA-robotic system; one associated with the tactile sensory system, the itch scratch and collision system, and the other associated with the visual system. The tactile TSM, shown in Figure 2, generates the q-final goal location (The "itch" location in [1]) and motivates the robot to perform a "scratch"-type trajectory aimed at the "itch"-point. For training purposes, the q-final "itch" location may be artificially generated by the TSM and applied to the appropriate q-initial-defined Nodal Map Module.

2.5 The Sequence Stepper Module (SSM)

There are 40 SSM's in the EVA-robotic system. The SSM is a scanning circuit that scans the NMM region between qi and qf and generates a sequence of p-values taken from the values stored in the indexed locations between qi and qf. The TSM-activated q-final "itch"-location becomes the Task-initiating Trigger (TT) that activates the SSM to examine the region in the NMM, between q-initial and q-final. The SSM then transmit the sequence of pq-values to the Control-signal Output Module

2.6 The Control-signal Output Module (COM)

There are 40-COMs in the EVA-robotic system, one COM for each joint in the EVA-robot. During each frame period all 40 joints may be controlled by the 40-COMs. The COM controls the speed of motion of a robotic part. During each frame period, the output of the SSM that represents the pre-planned trajectory of motion between qi and qf, is applied to the COM. If there are no obstacles in the pre-planned trajectory, then one and only one, namely the first control signal of the sequence may be transmitted to the joint-motor during that frame period. The speed of motion, including the possibility of a pause, is determined by the COM. If the pre-planned trajectory is to be implemented at maximum speed, then the COM will transmit to the motors the pre-planned sequence of p1….pn at the frame rate of one nodal transition per frame period, until n frames have elapsed and the trajec-

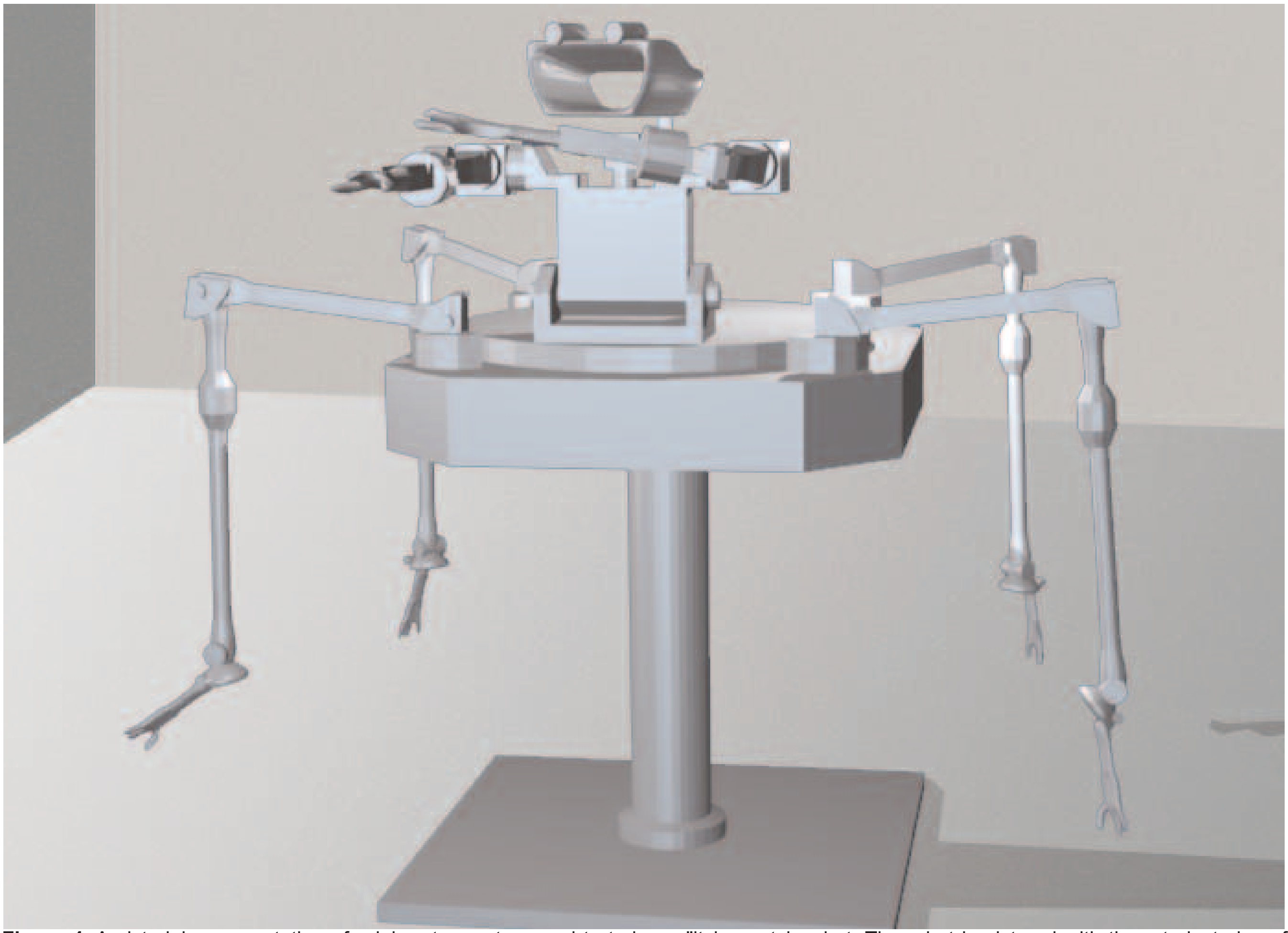

Figure 4. A pictorial representation of a laboratory setup used to train an "itch-scratch robot. The robot is pictured with three trajectories of motion: scratching the elbow with the left hand while extending the right arm.

tory of motion from qi to qf is implemented. Generally, in addition to priority shifts to other COMs, the speed of motion determined by the COM depends on the number of re-planning shifts that may take place in the pre-planned trajectory, during the progression from qi to qf.

3. TRAINING AND PROGRAMMING THE EVA-ROBOTIC SYSTEM

Training the EVA-robotic system is a 4-step process:

a) Training-programming the nodes of each NMM with the corrected set of p-values

b) Training all NMM with self location and identification. The robot must have motion-knowledge of the location of all parts of the body and flailing limbs relative to any and every other part of the robotic body and limbs.

c) Training the visual near space for reaching and touching spots of light

d) Hierarchical Task Diagram (HTD)-training:

- d1. Training the tactile near space for reaching, touching, grasping and blind ambulation in a 1g environment (with visual TTs)
- d2. Simple Hierarchical Task Diagram (HTD) multi-tasking: 1) Goal directed ambulating with visual obstacle avoidance, 2) Goal directed hand manipulation: picking up and setting down objects. 3) Goal directed ambulation in a zero g environment,
- d3. HTD-multitasking: Surveillance, inspection, simple repairs and replacement of parts.

A pictorial representation of a laboratory set-up to train the itch-scratch robot is shown in Figure 4 The robot is attached at its center of mass, and all itch-scratch trajectories are performed relative to the center of mass.

3.1 Training Each Nodal Map Module

Training each Nodal Map Module is accomplished by use of a modified “Hebbian” learning rule [5] Figure 5 is a training flow diagram of the p-vectors and q-vectors through the controller during one frame period (40-Nodal Maps Modules are trained simultaneously during each frame period). Two paths are shown in the figure, a training path and an operational path. Training is performed on the Nodal Map Modules and on the Sequence Stepper Modules. The Nodal Map training consists of the tabular assignment of a correct set of p-value-table line entries assigned to each nodal location of the Nodal Map Module. The correct p-value, a table-line-entry, is that p-signal that causes an exact motor displacement of a robotic part, to an adjacent node. At each node there are 27-p signal transitions to adjacent nodes in a three dimensional nodal map, and 8-p signal transitions to adjacent nodes in a two dimensional nodal map. A More detailed description of the training of all the nodes of an RRC-circuit, to perform a

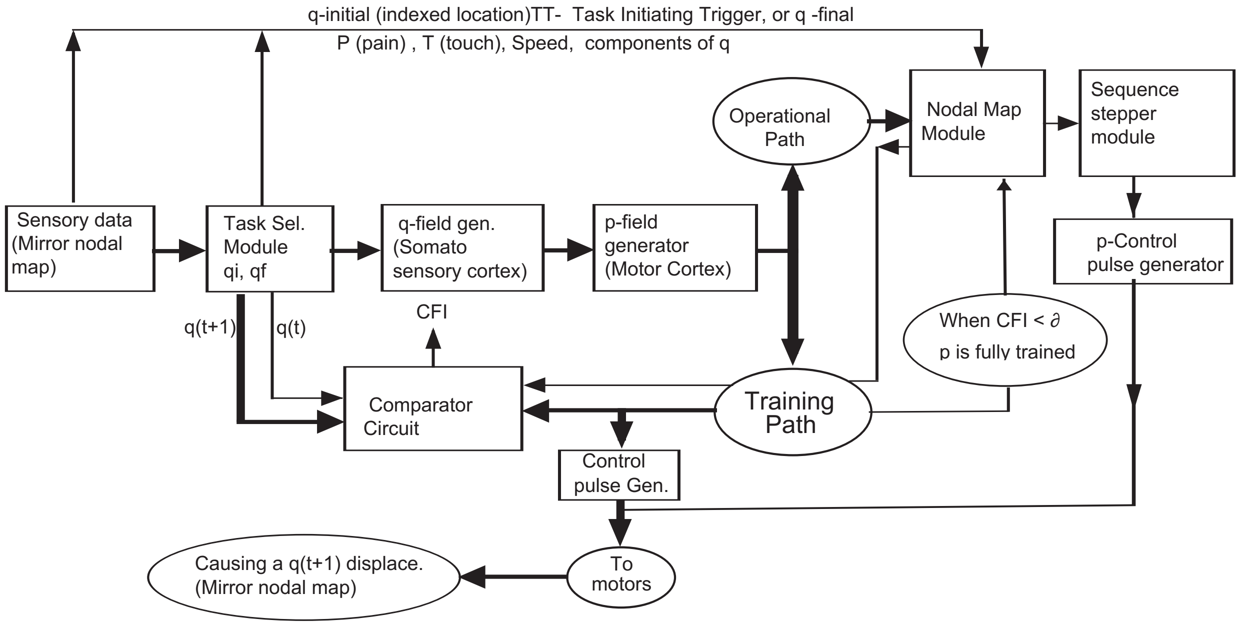

Figure 5. Training the nodal map module. Two paths are shown in the figure, a training path and an operational path. The set of p-signals are trained repeatedly until the CFI is less than delta. When CFI ≤ ∂, the p-signal is assigned to a node in a Nodal Map Module as a table-line entry. A pictorial representation of the training path flow is also shown in Figure 5.

complete set of "itch-scratch" motor control tasks is described in an article titled "The Engineering Design of a NCC-circuit for the Sensory-motor Control of a Robotic Arm" [5].

3.1.1 The Task Selector Module and the Sequence Stepper Module

The Task Selector Module (TSM) generates the q-final "itch" location and motivates the robot to perform a "scratch" action aimed at the "itch"-point. For training purposes, the q-final "itch" location may be artificially generated by the TSM and applied to the appropriate q-initial-defined Nodal Map Module. The TSM-activated q-final "itch"-location becomes a Task-initiating Trigger (TT) that activates the Sequence Stepper Module to examine the region in the Nodal Map Module, between q-initial and q-final, and select a pre-planned trajectory between q-initial and q-final.

3.1.2 The Corrected Set of p-values Assigned to Each Nodal Map Module

The method of selecting the correct set of control signals (p-values for all the motors associated with the 40-robotic joints) required to move a robotic part from q-initial to q-final, is shown in Figure 5. The p-vector assigned to each node of each Nodal Map Module is first calculated, estimated and tested by the designer (the magnitude is designed to generate an exact transition to an adjacent node, and the directions, often determined by inverse kinematics, are calculated for each of the end-joints). Figure 5 shows that the estimated p-values are then corrected repeatedly until the correction factor increment (CFI<∂) approaches zero.

The correct table-line p-value at the initial node is that value that generates an exact displacement from the initial node at q-initial to the adjacent node at q-final. Correction of the table line entry p-value proceeds by noting the displacement error generated by the set of p-initial and applying a correction factor so that the set of p-initial (corrected) leads to an exact transition to the final node defined by q-final (see Figure 5).

A Nodal Map Module is said to be "trained" if each nodal table, assigned to each node, is made up of a complete set of p-signal line entries. Each p-signal line entry causes an exact motor displacement to an adjacent external nodal position. The set of table line entries assigned to each nodal table consists of all the p-signals that lead to exact transitions to all adjacent nodal positions.

3.2 The Training Procedure. Self Location and Identification

In order to train the robot to perform all possible itch-scratch trajectories, the q-final-itch point is initially selected at nodes that are immediately adjacent to a q-initial. For each nodal map, and for each node defined in that map, the training proceeds with the q-final location placed at distances of two, three, four, and more nodal distances from q-initial. When all the nodes of all the nodal maps associated with all the joints of the robotic body are fully trained, the RRC-circuit is said to exhibit "robotic self knowledge." The robot may have "learned" by this programming methodology how to move every robotic limb towards any and every part of the robotic body.

3.3 Training the Visual space

The visual sensory system for the EVA-robot has been described in the Neural Network Journal and in IEEE-xplore [4], [7], [8]. A multi-tasking robot uses its visual system as a search engine that searches the space in the vicinity of the robot. In order to program the robot to locate and respond to a spot of light appearing in the FOV coordinate frame, each spot of light (shown in Figure 6) may be viewed as a Task-initiating Trigger (TT) activation point (similar to the itch-TTs). The robot is trained to respond to each TT-spot of light by means of a sensory motor controlled trajectory that is goal directed towards the spot of light.

The total FOV-coordinate space may be defined by placing the image planes shown in Figure 6 at various convergence-depths along the midline-LOS. Table 3 illustrates the number of convergent locations that may be incorporated into a prototype design of the EVA-robotic visual system. For each FOV in the prototype system, the number of indexed receiving neuron (forming coordinate points in the FOV-coordinate frame) is 144,000 (24,000 CCD-sensors x6). The total number of receiving neurons for the 18-FOVs shown in table 3 is about 2.6 million receiving neurons (144,000 x 18). Each receiving neuron represents the location of a spot of light located on one of the 6-image planes associated with each FOV. Note that the indexed location of the visual receiving neurons that make up the FOV-coordinate space, is a physically significant parameter that defines the location of the spot of light in the FOV of the robotic system.

3.3.1 The Search Engine: Programming-teaching the robot to locate and respond to spots of light on each image plane. TTs and the TSM of the visual system.

A visual multi-tasking robot with the visual system described in Table 3, may be programmed to search for spots of light in a region surrounding the robot. Each FOV (18-FOV in the prototype system) may be searched by moving the point of convergence of the two cameras from a depth of infinity to a depth of 9 meters, 1 meter, 60, 30, 10, and 6 centimeters (see Table 3) Programming-teaching the robot to locate and respond to spots of light is a process of training the robot to move its head and body to determine a FOV-midline, and then sweep the depth of convergence from infinity to 6-centimeters in front of the robot.

The search engine is used to search the external environment for Task-initiating Triggers (TTs) and obstacles that may be present along the trajectory [2]. The Task Selector Module (TSM) and the Sequence Stepper Module (SSM) are the primary circuit elements used to detect TTs present in the input sensory signals.

A visual prototype robot may be trained with a single spot of light (variable color with parameters determined photometrically) similar to the ones shown in Figure 6 and 7. A spot

Number of FOV's (18 total)	**Number of Convergent Depth Positions per FOV (6 depths or 6 image planes)**		**Number of Spots per Image Plane (equals no. of CCD sensors in the arrays)**
1. Torso verical- Head-0 deg nod 1.1 Head 0deg 1.2 Head 45 deg Right 1.3 Head 45 deg Left	Depth	Angle	2x100x120=24000 spots on each image plane
	6 cm	45°	
	10cm	30.9°	
	30cm	11.3°	
	60cm	5.7°	
	100cm	3.4°	
2. Torso Vertica- Head 45 deg nod 2.1 Head 0 deg 2.2 Head 45 deg Right 2.3 Head 45 deg Left	900cm	0.4°	
3. Torso 45 deg- Head 0 deg nod 3.1 Head 0 deg 3.2 Head 45 deg Right 3.3 Head 45 deg Left			
4. Torso 45 deg- Head 45 deg. Nod 4.1 Head 0 deg 4.2 Head 45 deg Right 4.3 Head 45 deg Left			
5. Torso 90 deg- Head 0 deg nod 5.1 Head 0deg 5.2 Head 45 deg Right 5.3 Head 45 deg Left			
6. Torso 90 deg- Head 45 deg. Nod 6.1 Head 0deg 6.2 Head 45 deg Right			

Table 3. The total visual field of view coordinate space is calibrated with tactile indexed locations in the Nodal Map modules.

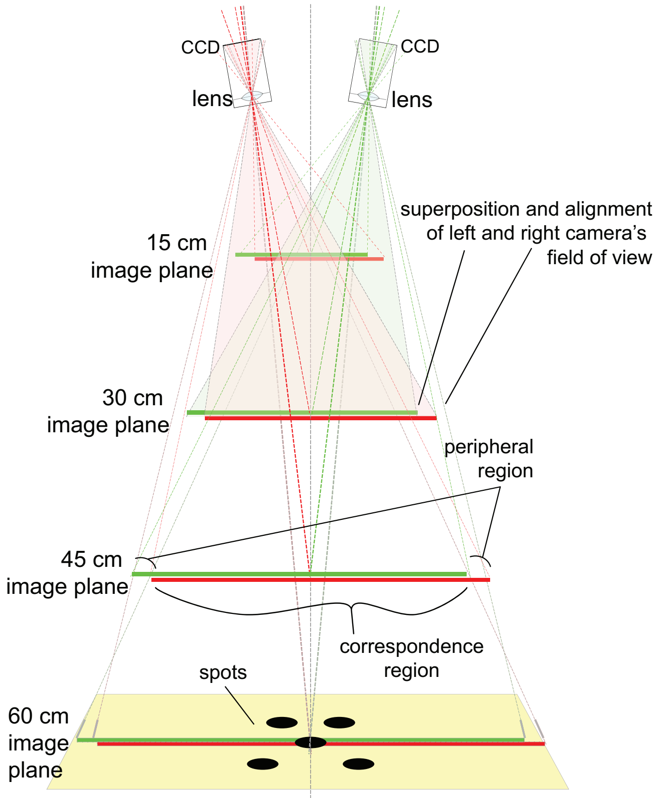

Figure 6 Four image planes. Each plane is determined by the binocular disparity and FOV of a 2-camera system. Each camera's LOS converges at convergent depths of 15, 30, 45, and 60 centimeters intersecting on the midline LOS. The FOV and areas of the right and left camera CCD-array determine the image-areas on the image plane. The two image-areas on the image plane of the right and left camera are superposed and aligned so that the central portions correspond to one another and peripheral portions are unique to the right and left camera.

of light of any color-hue at any location is identified (photometrically) as a Task Initiating Trigger (TT) by the TSM. For the prototype robot, if a TT-spot of light activates any of the 2.6 million FOV-receiving neurons, located and indexed relative to the tactile receiving neurons that define the near space, then the sensory motor control system is trained to generate a finger trajectory of motion that is goal directed towards the spot of light. The training process is identical to the process described for the tactile robotic system activated by itch-type TTs [10], [11]. (Note that the TSM locates obstacles and TT-spots of light at indexed locations of the Nodal Map Module, and that the Sequence Stepper Module may generate an obstacle-avoiding trajectory towards a TT-spot of light. During each frame period, a spot of light, detected by the Task Selector Module (TSM), is prioritized and may become the Task-initiating Trigger (TT)-activation point in the near space of the robot (similar to the itch-activated mechanoreceptors used as itch-TTs to activate an itch-scratch trajectory)).

3.3.2 The Calibration Procedure

The problem is one of calibrating the 3D-coordinate space defined by the tactile sensors, with the 3D-image defined by the superposed CCD-array image planes. It is a problem of scaling the FOV-image so that it corresponds to the scale size measured by the tactile receiving neurons in the near space. The locations of the visual receiving neurons that correspond to the tactile receiving neurons in the near space regions (light spots in the FOV-space) have been described in the previous section and are shown in Figure 7.

3.3.3 The depth of the neuron at the fixation point

The signals must be indexed or related to the sensory motor control system of the eyes, head, body, and limbs in order to view a 3D-image in the coordinate frame in which the system is operating. In the NCM-system indexing of the visual neurons is a function of the head and torso orientation (3 torso orientations, plus 3 head orientations, plus z-noding orientations per torso orientation) and the convergence angle of the opto-sensor that determines the fixation point of the converging cameras. The locations of all the receiving neurons of each 2-binocular layers located at a given fixation point (see Figure 6) are indexed to the 3D-neurons of the self location and identification coordinate frame. The fixation point and the two superposed image planes associated with it are placed at the depth of the fixation point in the self location and identification coordinate frame. For example, the two image planes, shown in Figure 6 at a depth of 60 centimeters, are defined in Figure 7 as plane D2. The depth of the fixation point of D2 is determined by the convergence angle opto-sensor and may be projected to the exact indexed location of the self location and identification coordinate frame (the Nodal Map Module). However, only the visual fixation point in D2 is indexed to the proper depth of the self location and identification coordinate frame. The depth of offset spots shown in D2 may be determined and learned by the system by shifting the fixation point to those spots.

All the neurons of the 2-superposed image planes defined by a fixation point are located on the 2D-surface at the corresponding fixation point within the controller (see layers in Figure 7 labeled "indexed receiving neurons of the visual coordinate frame). Only the indexed receiving neurons that

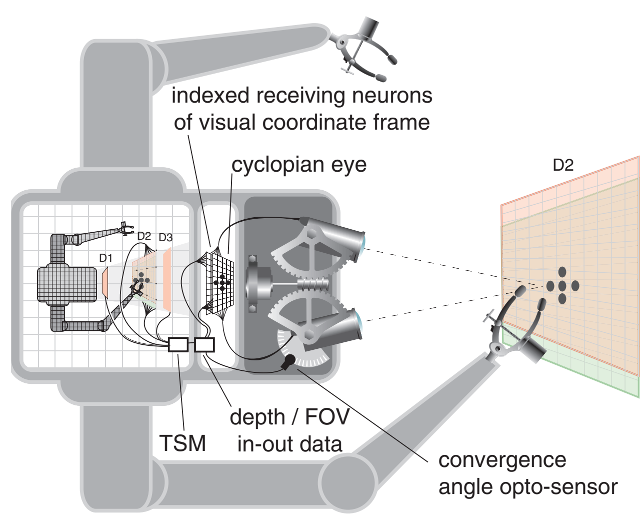

Figure 7 A visual coordinate frame within the controller. A visual coordinate frame within the controller is formed by all the indexed receiving neurons that locate spots of light on the image planes within a given FOV, and all FOVs designed into the system (see Table 1). The cyclopean eye is connected to the tactile "self identification and location"-circuit. During each frame period, the convergent angle opt-sensor determines the convergent-depth of the image plane and the TSM transmits the cyclopean eye data to the indexed locations (shown in the figure at plane position-D2).

are at the fixation point will register and be analyzed so that it exhibits the depth derived from the absolute disparity. Nearby neurons that are offset from the fixation point are not only non-corresponding but may represent images that are located at large distances further or nearer than the corresponding fixation point. The robot must learn, using all the visual quos listed by Marr [12] and Poggio [13] (e.g. size, continuity, obscuration), whether nearby neurons are located at greater, smaller, or the same depth as the fixated depth (In the next section, it is noted that this may correspond to learning to find nearby non-corresponding points that exhibit artificially learned correspondence. The only way the robot has of checking the depth of a nearby point is to converge and fixate on the nearby point and thereby determine the depth of the nearby spots. Once the depth of nearby points is established, by fixating on them, their location may be projected to the appropriate indexed locations of the Nodal Map Module. The sensorimotor control system of the robot then has the capability to move all its body and limbs relative to the location of the visual image in the indexed visual coordinate frame. It has self location and identification knowledge not only with respect to all body parts, but also with respect to the observed object located in the common coordinate frame.

3.3.4 The depth of neurons that are offset from the fixation point

An internal retinotopic depth collective modality may be formed in the microprocessor based portion of the controller. During each frame period, the internal depth collective is formed by following the design of Marr and Poggio [13] for a neural network that learns to determine retinotopic depth based on 10 photometric visual queues applied to the system. The process includes a) fixating on the offset neurons, b) determining their disparity depth, c) correlating their disparity depth with any of the 10-visual quos specified by Marr and Poggio [13], d) depending on either the correlated visual quo or the measured disparity depth, the offset neurons are now indexed into a single disparity depth selective neuron, located on the newly formed image plane of a retinotopic depth collective modality, and e) This newly formed retinotopic set of disparity depth selective neurons, located on a retinotopic depth-collective modality, may now be indexed to the correct corresponding 3D-position of the self location and identification coordinate frame (the indexed locations of the tactile 3D- coordinate frame in the Nodal Map Module).

The internal depth collective is designed to learn the correct depth-location of offset neurons by measuring visual quos detected during a given frame period, rather than shifting the fixation point to each offset neuron in order to determine its depth-location. Note that in the engineered 3D-visual-system, if a retinotopic depth collective modality is formed , then all the data necessary to form a 3D-image which is calibrated with the 3D-tactile near space coordinate frame now resides in the Nodal Map Modules. This data may be correlated with sensorimotor control that includes obstacle avoidance as well as detection of objects, shapes, forms, color, and motion that may be identified as visual-TT-patterns that may initiate the sensory motor control tasks of a multi-tasking system.

3.3.5 Training/programming the system to avoid obstacles

Obstacle avoidance is one of the primary characteristics of locomotive behavior. The visual q-field data may be used to locate obstacles in the Nodal Map Module that mirrors the Euclidean space in which the robot is operating. In the following discussion it is assumed that both visual-q field data and somatosensory q-field data is applied to the topographic ordering of neurons of the Nodal Map Module. The visual-q field sensors (cameras) and somatosensory q-field sensors (tactile) are always active, while the robot performs the tasks that it is designed to perform. Therefore, the robot is constantly monitoring the visual and tactile q-field signals with respect to the “self” nodal map, and is sensitive to visual and tactile stimulation. The monitoring function is performed by the Nodal Map Module, upon which the visual and tactile q-field signals are impressed. This nodal map, located within the controller, may be viewed as a recording monitor of visual and tactile q-field data, since it mirrors the data present in the 3-dimensional Euclidean space surrounding the robot.

A pictorial representation of a laboratory set-up to train the itch-scratch robot for obstacle avoidance is shown in Figure 4. The robot is attached to its center of mass, and all itch-scratch trajectories are performed relative to the center of mass. In the engineered 3D-visual system, all the data necessary to form a 3D-image of the obstacle resides in the Nodal Map Modules that represent the 3D-tactile coordinate frame in which the EVA-robot is operating. The internal depth collective, described above, may transmit the photometric data of all detected obstacles to the indexed locations of the Nodal Map Module. The Sequence Stepper Module then detects those obstacles and generates a pre-planned trajectory so as to avoid the photometrically detected obstacles [4], [7], [8].

The response of a multitasking NCM-robot is determined by a Hierarchical Task Diagram (HTD), the top level specification of the system, and the Task Selector Module (TSM) that prioritizes and selects during each frame period the Task initiating Triggers (TTs) present in the incoming signal [1]. The tasks on the HTD are activated by the TSM that may apply the highest priority-TT to the Nodal Map Module. The prioritized Task-initiating Triggers (TTs) are used to select the top level tasks and the lower level subtask on the HTD. For example, a visual, itch-NCM robot may be designed with two top level tasks, an itch-monitoring set of tasks and a color-hue spot-monitoring set of tasks (see section 3.3.1). Each itch-TT is location-indexed and assigned a priority level that is a function of the pressure applied to the mechanoreceptor. Each color-hue spot-TT is location-indexed and assigned a priority level that is a function of the photometric color-hue output of the CCD-sensor (receptor). However, the priority levels of all itch-TT are greater than the set of priority levels assigned to the color-hue spot-TTs. When high priority itch-type activations are absent, the robot is programmed to operate in the color-hue spot monitoring state and to discriminate and respond to different color-hues with different TT-trajectories. If an itch activation-TT is detected by the TSM during any frame period, the robot interrupts the color hue-discrimination task, switches into an itch-activation state, and proceeds to plan an itch-scratch trajectory aimed at the itch-point as the goal of the trajectory. The EVA-robot is also be programmed to avoid obstacles that may photometrically appear along the path of any pre-planned trajectory (either color-hue discrimination path or the itch-scratch path).

The priority levels of visual obstacles (generated by the

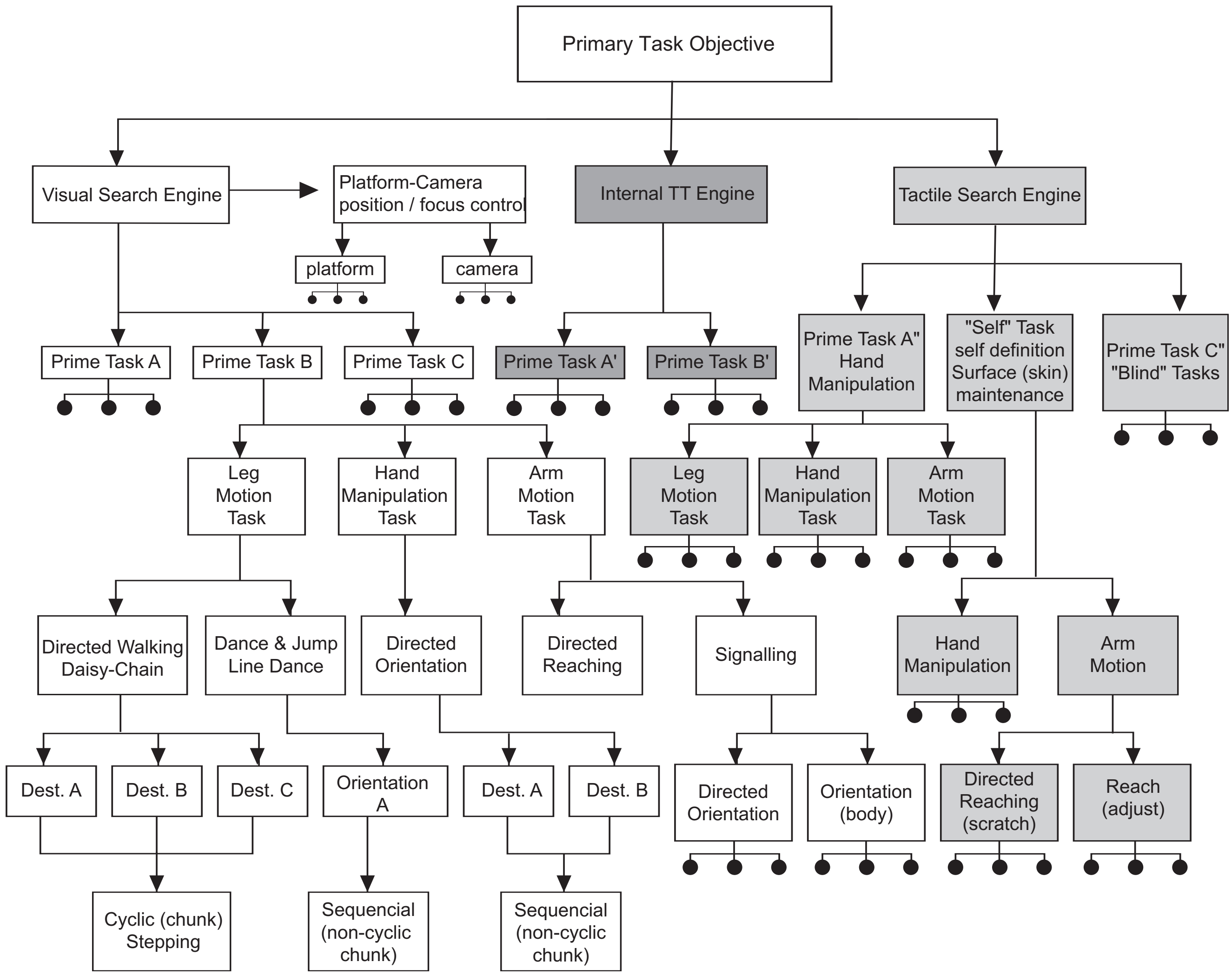

Figure 8 A hierarchical task diagram.

TSM) do not change the state of the system (itch-activating state or color-hue spot discriminating monitoring state). Instead, the TSM transmits the photometric data of all detected obstacles via the internally generated retinotopic depth collective, to the indexed 3D-locations of the Nodal Map Module. The Sequence Stepper Module then detects those obstacles and generates a pre-planned trajectory so as to avoid the obstacle [1], [5], [9].

3.4 HTD Training: Chunking, daisy chains and line dances

The top level specification for a multi-task robotic system is a Hierarchical Task Diagram (HTD) that describes the tasks that the robot is designed and programmed to perform. It identifies the primary task objective of the robot and is the top level programming and training specification for the system. A hierarchical task diagram is a mechanism for decomposing a conceptual task-goal representation, the primary task objective, into constituent parts. Figure 8 presents a HTD for a generic multi-tasking robotic controller. The primary task objective at the top of the hierarchy must encompass all the “trained” tasks performed by the robotic system. The sub-tasks at the bottom of the hierarchy are simple sequences of motor actions which are called daisy chains, line dances or chunks. The daisy chain is a repetitive array of simple sequences called “chunks”. A daisy chain is made up of a large number of repeated “chunks”. A line dance, on the other hand, is made up of preprogrammed chunks (sequences), with each preprogrammed chunk differing from the other preprogrammed chunks.

All tasks, shown on the HTD, must have associated with each of them a Task-initiating Trigger (TT) and priority level that is keyed to the task objective. The TT is a mechanism for selecting the particular real time task that is to be performed by the robotic system. The task selection process is performed from the perceived q-field nodal mapping that is the recording monitor of the robotic system. During normal operations of a robot, and during the training process, the q-field is applied to the TSM, that generates the TT, that activates the Sequence Stepper Module, that initiates a sequence of actions that navigate the robot from a q-initial to a q-final position.

3.4.1 Navigating through the internal coordinate frame via the HTD

The sequential control characteristic of the robotic controller provides a simple mechanism for “chunking” and for building hierarchical structures (Gazzaniga, 1998, p. 386). At mid levels of the HTD, a sequence of simple motor actions may be “chunked” and identified as a task or sub-task, by a rep-

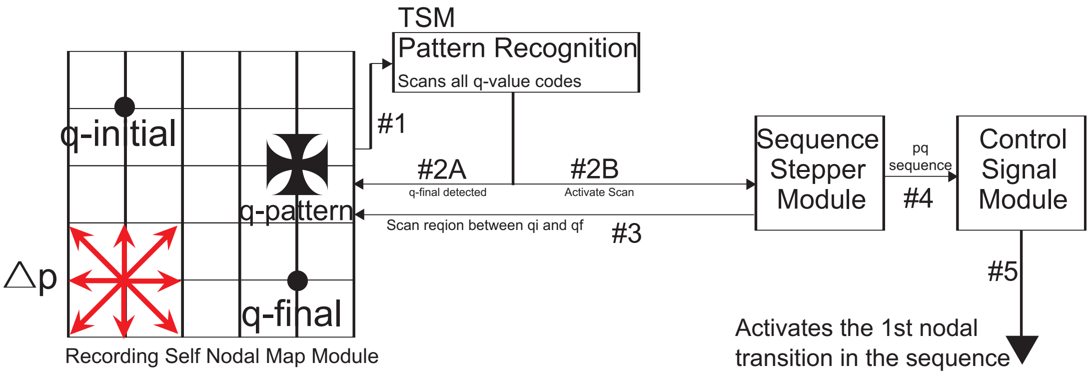

Figure 9 The functional flow throughout the TSM pattern recognition circuit. The TT priority levels are programmed offline by use of the HTD.

resentation of the destination-goal of the action. The next higher priority level, controls the "chunks" with another sequence of motor actions that aims the "chunks" towards another goal. Again, the representation of the goal identifies the task as the next higher level of action. For example, the sub-task designation of "walking" in a 1g environment may be represented by a daisy chain (repetitive sequence) of sequential chunks where each sequential set of chunk may represent one step of leg #1 and #3 of a 4-legged system. Lifting leg #1 and #3 is one chunk, bringing both legs forward approximately fifty-centimeters is the second chunk, setting it down and shifting the body weight to leg #1 and #3 represents the third chunk. The same set of 3 chunks is repeated for leg #2 and #4. An alternating sequence of three chunks per leg represents the daisy chain sequence of "walking.

The robot must "learn" or be "trained or programmed" to perform all the tasks identified in the HTD shown in figure 8. In order to be proficient in the performance of a mid level task, the robot must learn, or be "trained" to perform the lower level "chunks" that comprise the lower levels of the HTD.

3.4.2 Programming the TSM is achieved by means of the HTD

The HTD is the top level programming and training specification for the TSM. The TSM may be programmed by the system designer (supervised programming). To facilitate supervised programming, every task listed on the HTD must have a q-final-TT and a priority level associated with it. The function of the TSM is to recognize and prioritize the pattern of activation.The TSM therefore consists of a pattern recognition circuit that is programmed to recognize the patterns recorded on the nodal map module. The input to the pattern recognition circuit includes the location codes and characteristics of the pattern. Pattern recognition circuits that are applicable to the TT-search task have been designed and innovated by Stephen Grossberg [14] and Gail Carpenter [15].

3.4.3 Supervised Programming of the TSM: Programming a volitional multi-tasking robot

The TSM-pattern recognition circuit must be programmed to recognize and prioritize all q-final-TT tasks listed on the HTD. Figure 9 shows the functional flow through the TSM-pattern recognition circuit. The high priority output of the pattern recognition circuit, in the form of a q-final TT signal, goes directly to the Sequence Stepper module and initiates the sequence of pre-planned actions associated with the TT. The TSM module of any RRC robot may be programmed off-line to perform any sequence of HTD-TT tasks that are recognized and prioritized by the pattern recognition circuit.

3.4.4 The Functional flow through the TSM and SSM

The Sequence Stepper Module is activated by the q-final-TT output of the pattern recognition circuit of the TSM. During each frame period, the q-initial, q-final and a group of visual and tactile q-patterns are recorded on input world space ("Self" nodal map module). The recorded patterns are the input to the pattern recognition circuit of the TSM. The pattern recognition circuit is programmed to recognize and prioritize some of the q-patterns. If the q-final is the highest priority TT-task detected during that frame period, then the q-final is applied to the Sequence Stepper Module. The Sequence Stepper Module is then activated to scans the region between q-initial and q-final.): For the design of an obstacle-avoiding robot, the Sequence Stepper Module is designed to search the intermediate nodes between q-initial and q-final. The search is generally restricted to the lower priority TTs that define solid obstacles. The Sequence Stepper Module generates a pre-planned navigational path made up of a sequence of q-positions and p-control signals at each q-position that is dependent on the q-pattern detected at each node. If low priority level code values are assigned to all solid obstacles, then the Sequence Stepper may be designed to generate a curved path that avoids the lower code value obstacle locations. And during that same frame period, the Control Signal Module may activate the first nodal transition in the pq sequence generated by the Sequence Stepper Module.

3.4.5 A Procedural Memory Circuit in the controller

The HTD, the "self" nodal map recording monitor and the TSM-pattern recognition circuit, are analogous to a procedural memory system in the brain. The Hierarchical Task Diagram (HTD) is the basic specification for a multi-task robotic system. The HTD describes the tasks and the priority level TT that is assigned to each task that the robot is designed and programmed to perform. The pattern recognition circuit must be taught (programmed) to recognize the total set of TT-priority levels that have been designed into the HTD. During each frame period the pattern recognition circuit of the TSM examines the priority levels of all TTs that are recorded on the "self" nodal map module. Depending on the real time recording on the "self" nodal map module, the robot performs and remembers a complex sequence of obstacle avoiding tasks all aimed at fulfilling the prime tasks shown on the HTD (figure 8).

3.4.6 Low priority Obstacle Avoidance

One of the fundamental design constraints on the pattern recognition circuit of the TSM is the volitional constraint applied to obstacle avoidance. The volitional constraint gives the robot a re-planning capability of a pre-planned trajectory, at intermediate points between q-initial and q-final. Obstacle-avoidance may be programmed in the Sequence Stepper Module. In the supervised programming mode, It is possible to program obstacle avoidance in the pattern recognition circuit as well as the Sequence Stepper Module.

Low priority and high priority Obstacle Avoidance in the Pattern Recognition Circuit

For a robot performing multiple tasks, with a multiple set of trained nodal maps, the TSM must generate TTs that select the nodal map as well as the tasks defined by the HTD. The search engines shown as the top level of the HTD (figure 8) operating in conjunction with the “self” nodal map, is the primary source of operational TTs. The function of the search engines is performed by the pattern recognition circuit. The task selecting-pattern recognition circuit, shown in figure 8, may generates Task-initiating Triggers (TT) that initiate the performance of top level tasks. For example in figure 8, the search engine initiates the tasks labeled Prime Task A, B, C, and one task at each level below the top level prime task, all the way down to the bottom level, where “chunking” of the simple motor actions, occur.

In a multitasking robot, a hierarchy of priority level-TTs must be programmed into the pattern recognition circuit. Most obstacles are observed as low priority TT-patterns occurring in the path of a higher priority q-final. In this case the obstacle avoiding path is generated by the Sequence Stepper Module. However when an obstacles become a high priority TTs, it is detected by the pattern recognition circuit. For example, the priority level assigned to a visual pattern of an obstacle must be a function of distance and speed. Obstacles at large distances and velocities that do not pose a danger to the “self” are assigned low priority in the pattern recognition circuit. In this case (large distance, benign velocity), the pattern recognition circuit may respond to a higher priority TT, and the Sequence Stepper may respond to the obstacle only if it is in the path between q-initial and q-final. In successive frames, as the robot gets close to the obstacle, the priority level of the pattern-code value is programmed to increase as a function of distance to the obstacle (in the pattern recognition circuit). At a sufficiently high priority-code value the pattern recognition circuit is programmed to generate a task Interrupt-TT that activates the Sequence Stepper Module with a q-final that avoids the obstacle, (stop or turn). In case of a speeding obstacle (for example a projectile), in successive frames, as an obstacle or projectile gets closer to the “self” circuit, the priority-value code is programmed to increase as a function of speed and distance. Again, at sufficiently high priority-code values the pattern recognition circuit is programmed to generate a task Interrupt-TT that avoids the projectile, (blink or duck). This programming of the pattern recognition circuit is related to an innate biological-TT associated with the speed and distance of a projectile or obstacle relative to the “self” circuit (associated with the “blink” or “duck” response).

3.5 Programming the priority levels of TT-tasks recognized by the TSM

The basic specification for a multi-task robotic system is a Hierarchical Task Diagram (HTD) that describes the tasks and the priority level-TT of each task that the robot is programmed to perform. During each frame period the pattern recognition circuit of the TSM examines the priority level of all TTs that are recorded on the “self” nodal map module. The input to the pattern recognition circuit of the TSM consist of the q-pattern-code values recorded on the “self” nodal map recording monitor. All input TT-pattern-are assigned (by the designer) one of a multiple number of priority levels. For example, three priority levels and an obstacle detection level may be assigned to certain sets of q-pattern value codes. An emergency-TT is assigned the highest priority, whereas other TT-patterns are assigned mid-level and low level priority. Obstacles are assigned a lower priority level than the low level-TTs. The mid-level and high-level TTs are task interrupt TTs. That is, a high level TT will interrupt all tasks being performed by the controller. A mid-level TT will interrupt only the low-level TT tasks. And the low level TTs will control the robot whenever mid-level and high level TTs are absent.

There are a number of constraints on the generation of Task-initiating Triggers (TTs), that arise from the fact that there are certain tasks that the robot can and cannot perform simultaneously. Starting at the top of the HTD shown in Figure 8, and working downward, the following constraints are noted:

A. At the top level, 3-TTs may be generated simultaneously, one by each of the engines shown at the top of the figure. However, with one exception, only one prime task may be performed at any time (because different ‘’training” or learning is performed at all lower levels of different prime tasks shown on the HTD). The one exception is that hand manipulation tasks performed with the visual search engine, always require pressure transducer input (touch feel data), performed with the tactile search engine, to assure that the object being manipulated is not damaged. Thus whenever a hand manipulation task is activated, the tactile search engine detects it, and immediately activates tactile search engine prime task A” which operates simultaneously with the prime task under which the hand manipulation task is performed.

B. The internal-TT search engine is an emergency-TT generator. All TTs generated by the Internal-TT Engine are high priority, task interrupt triggers that put the robot into a protective task sequence. High priority TTs generated by the visual search engine, and the tactile search engine, go to the Internal TT Engine for implementation. If a TT is generated by the internal engine, that TT overrides all other TTs, and the robot is shifted either immediately, or with a fixed time delay, to performing one of the emergency protective tasks Prime task A’ or Prime task B’ shown in the diagram. Some sample emergency TTs that initiate the primed tasks are

1) A crash, or any sharp (damaging collision) blow to the robotic body (“self”).
2) A threatening, fast moving object or projectile that is about to collide with the robot.
3) An unprovoked attack (by animals or humans).
4) Sudden sharp blinding light impinging on the camera lens.
5) A malfunction of any robotic part that interferes with proper accomplishment of the prime task. Or

6) The detection of maintenance requirements, or malfunctions that do not interfere with the performance of the prime task, but are precursors or predictors of future interference. (Time delayed emergency -TT)

C. The Tactile Search Engine generates, in addition to high priority and low priority triggers, 2-TTs that are assigned an intermediate level priority. They are

1) The Prime task B" "Surface (skin) Maintenance and Repair" task. And

2) One intermediate priority trigger in Prime task C" associated with leg motion tasks.

The intermediate priority task B" triggers are used to alleviate minor surface (skin) irritations, such as scratching an itch or adjusting some robotic part that has become misarranged. The Task C" intermediate priority trigger activates a "blind walking" task, when there is not sufficient light to implement a "Leg Motion" task under the visual search Engine category.

Both triggers are short term mid-level priority, task interrupt triggers that will interrupt the performance of low priority tasks, repair or alleviate the environmental condition that gave rise to the trigger, and then return the control to the task that was interrupted.

D. All other TTs generated by the Visual Search Engine, and the Tactile search Engine, are low priority TT.

1. There are three lower level tasks shown immediately below the Prime Task level, shown in figure 8. They are Leg motion tasks, Hand manipulation tasks, and Arm Motion tasks. Those three tasks may be performed simultaneously, or each task may be performed individually, independent of the other two. Generally the three tasks are temporally coordinated with one another.

2. There are a multiplicity of tasks below each of the Leg Motion, Hand Manipulation, and Arm Motion tasks. At each of the lower levels, only one sub-task may be performed at any given time. That is, under the Leg Motion Tasks category, the robot may perform either "Directed walking" or "Dancing", but it cannot perform those two tasks simultaneously. Similarly, when it does directed walking, it could walk towards destination A, B, or C, but only one destination, at a time.

4 EVA-ROBOTIC SYSTEMS

Ambulating in zero-g inside the space lab compartment and EVA inspection and surveillance.

Some primary task objectives that the multi-tasking EVA-robotic controller may be trained to perform are:

A. Ambulating in a 1g environment
B. Repair and replacement of parts in 1g
C. Ambulating in a zero-g environment (inside a spacelab/space station environment)
D. Inspection of the exterior surface of a spacecraft for micrometeorite hits in a zero-g environment

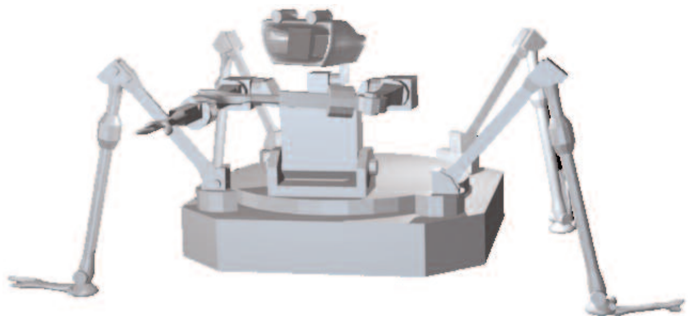

Figure 10. Ambulating in a 1g environment.

4.1 Ambulating in a 1g environment

Daisy chain walking of the EVA-robot was outlined in section 3.4.1. Figure 10 is a pictorial representation of an ambulating EVA-robot. A motion picture video of the walking robot has also been generated. The HTD shown in Figure 8 shows the directed walking task at the third level of prime task B. All the tasks must be listed in an HTD and a priority level must be assigned to each task. The sequence of TTs generated by the TSM for the directed walking daisy chain is as follows:

a) A destination-A q-final is applied to the directed walking task (3rd level) of prime task B (See Figure 8).
b) Destination A-q-final TT is applied to the NMM thereby activating cyclic chunking
c) Cyclic chunking activates lifting leg #1 and #3.
d) Zero reaction forces on legs 1,3 generates a TT to "stop lifting".
e) "Stop lifting" generates a TT to bring legs 1,3 forward (50 centimeters)
f) Legs at 50 centimeters leads to shifting body weight to legs 1,3.
g) Reaction forces on legs 1,3 activates lifting of legs 2,4.
h) Etc

4.2 Robotic repair or replacement of parts

An ambulating robot may also be trained to perform repair or replacement of parts. All the tasks must be listed in an HTD and a priority level must be assigned to each task. In this case the robot performs directed walking towards a destination TT-pattern that may be painted on the part to be replaced.

a) The destination TT activates a cyclic walking towards the destination.

b) The robot is programmed to stop when the distance-TT=50 centimeters

c) The stop is a TT that activates the prime task B hand manipulation task, and the search for a TT-pattern identifying the part to be replaced.

d) The TT pattern activates an index finger reaching task.

e) Touching the object may generates either a grasping task, followed by an attempt to move the part. Failing to move the part may trigger the task of reaching for a screw driver and unscrewing a set of screws, etc etc

4.3 Ambulating in a zero-g environment

Figure 11 shows the EVA robotic system "monkey climbing" in a zero-g environment. In this case the EVA robot may use both arms or legs for climbing in an environment where grasping bars are readily available. All the tasks must be listed in an HTD and a priority level must be assigned to each

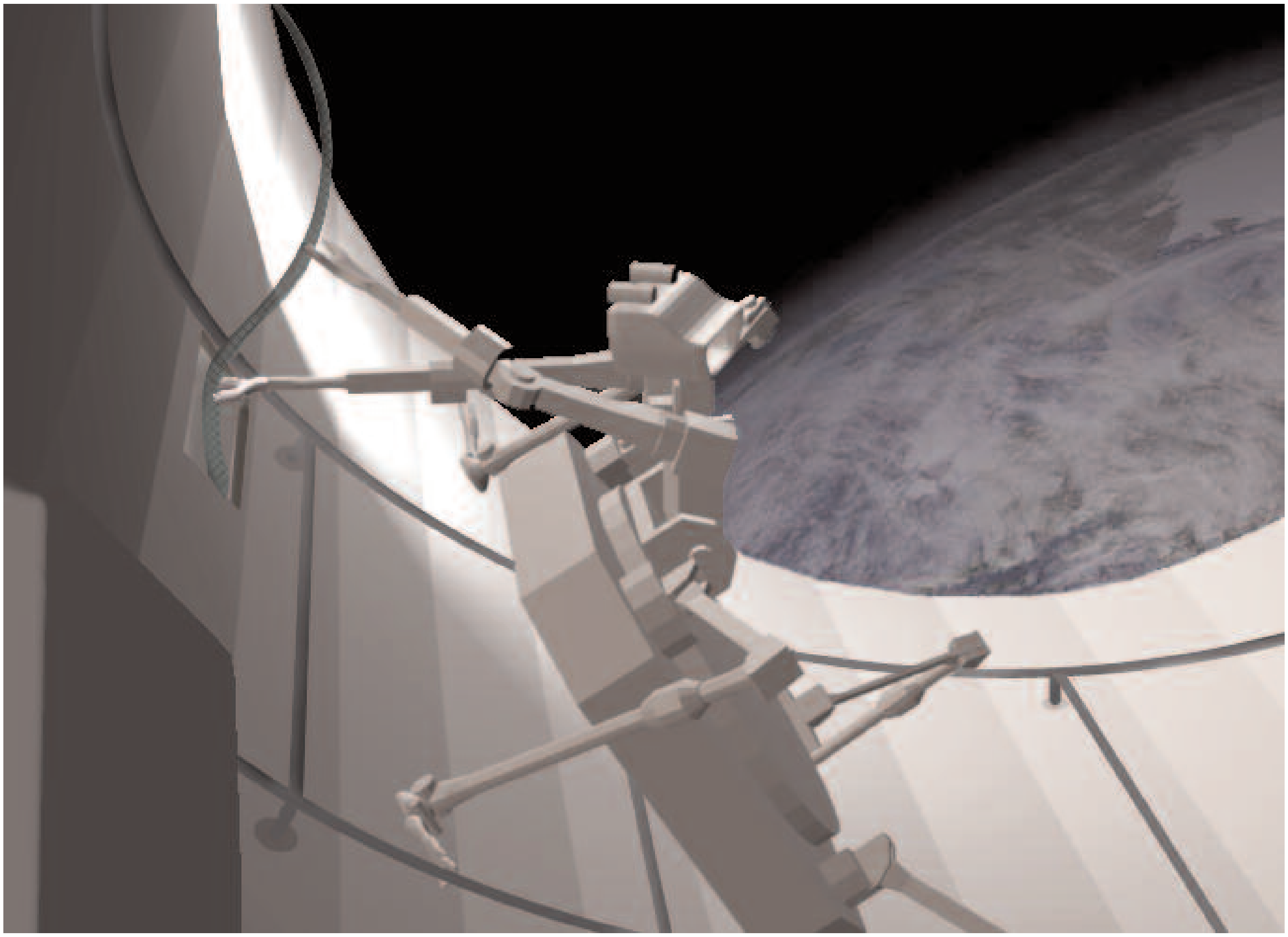

Figure 11. Zero-g “monkey climbing” in the interior of the spacecraft.

task. The EVA is trained to perform the line dance shown in the HTD.

a) The image of a grasping bar-TT leads to a reaching motion of an arm or leg.

b) Touching the bar with an index finger leads to the activation of the hand manipulation task.

c) The hand is extended and then grasps the grasping bar.

d) A solid grip on the grasping bar is a TT that leads to grasping with other arms or legs. Etc.

4.4 Inspection of the exterior surface of a spacecraft

Figure 12 is a pictorial representation of an EVA robot inspecting the exterior of a spacecraft in a zero-g environment. The robot may be designed with electrostatic foot pads, sticky foot pads or grasping feet (shown in the Figure). The robot may ambulate as described in 4.1, 4.2, or 4.3, and inspect the surface visually or by sliding the tactile hand palms and finger sensors over the smooth surface of the spacecraft. The robot may be trained to perform repair and replacement of parts by adding those prioritized task to the HTD.

REFERENCES

[1] Rosen A., & Rosen D. B. (2007a). Sensorimotor Control by Reverse Engineering the Biological Modalities. Submitted for publication in the Neural Network Journal of the INNS http://www.mcon.org/submtd/nnj_nng1886_121507.pdf

[2] Rosen, A., & Rosen, D. B. (2006b) An Electromechanical Neural Network Robotic Model of the Human Body and Brain: Proceedings of the ICONIP-2006 Hong Kong; King et al (Eds) LNCS Part 1 pp. 105-116, Berlin Heidelberg: Springer-Verlag. http://www.mcon.org/submtd/ICON_IP_RosenRosen71506web.pdf

[3] Wagner, R. & Lane, H. (2007) Lessons Learned on the AWIMR Project. Presented at the Space Robotics Workshop, ICRA. April 14, 2007.

[4] “A robotic optical circuit that generates 3D-visual images: a robotic solution to the inverse optics problem of visual seeing,” published in the Neural network Journal and available for viewing at http://www.mcon.org/submtd/nnj_nng1911_121507.pdf

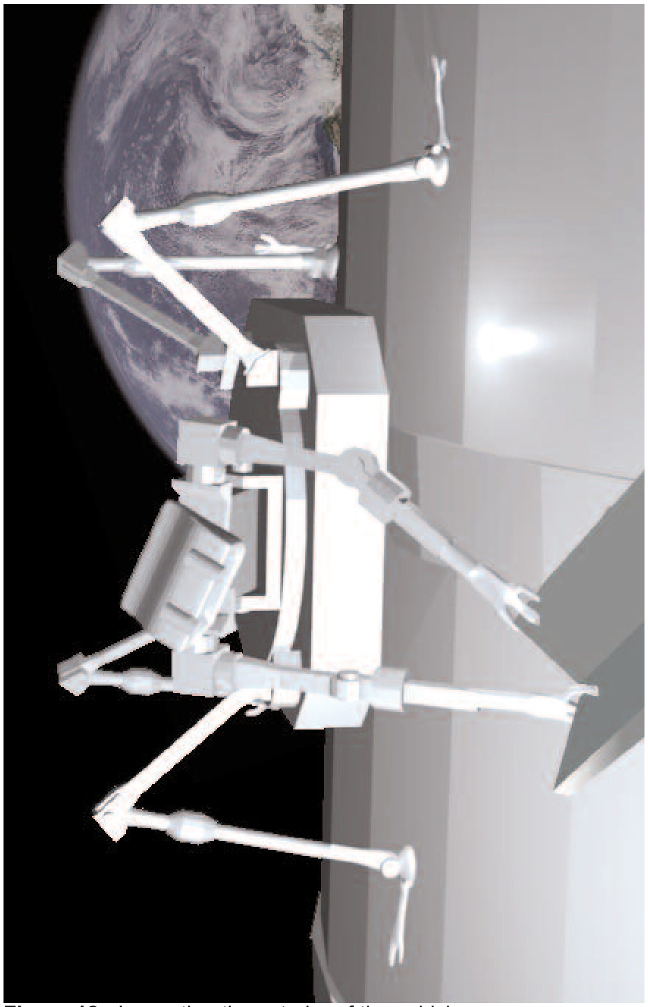

Figure 12. Inspecting the exterior of the vehicle.

[5] Rosen, A., Rosen, D.B. (2003b). The Engineering Design of an NCC circuit for the Sensory-motor Control of a Robotic Arm. Machine Consciousness Technical Journal. 1:57-69 (see www.mcon.org)

[6] Hebb, D.O. (1949). The Organization of Behavior: A Neurophysiological Theory. New York:Science

[7] Rosen, A. & Rosen, D. B. (2007b) A Robotic Optical Circuit that that generates a 3D-visual image: Solution to the inverse optics problem for robotic visual seeing by reverse engineering the neurophysiology of the modalities of the retinal receptors. IJCNN-2007 ieeexplore/servlet/opac

[8] Rosen, D. B. & Rosen, A. (2007c) A robotic neural net based visual sensory motor control system that reverse engineers the motor control functions of the human Brain. IJCNN-2007 ieeexplore/servlet/opac

[9] Rosen, A., & Rosen, D. B. (2006a) A Neural Network Model of the Connectivity of the Biological Somatic Sensors. Neural Networks, IJCNN'06: International joint conference on. ieeexplore.ieee.org/servlet/opac?punumber=11216 (ISBN: 0-7803-9490-9 pp 299-306)

[10] Rosen D.B., & Rosen, A (2003a) The Design of a Volitional, Obstacle Avoiding Multi-tasking Robot. Machine Consciousness Technical Journal. 1:41-56

[11] Rosen, A., & Rosen, D. B. (2006b) An Electromechanical Neural Network Robotic Model of the Human Body and Brain: Proceedings of the ICONIP-2006 Hong Kong; King et al (Eds) LNCS Part 1 pp. 105-116, Berlin Heidelberg: Springer-Verlag.

[12] Marr, D. & Poggio T. (1976) Cooperative Computation of Stereo Disparity. Science 194; 263-287. See also Marr, D. (1962) Vision. San Francisco: W.H. Freeman

[13] Marr, D. & Poggio T. (1979) A computational theory of human stereo vision, Proc. R. Soc. Lond. B 204:301-328

[14] Grossberg, S. (1988). Neural Networks and Natural intelligence. Cambridge Mass:MIT Press

[15] Carpenter, G.A., Grossberg, S. (Eds) (1991). Pattern Recognition by Self-Organizing Neural networks. Cambridge Mass: MIT Press

ACKNOWLEDGEMENTS

The authors acknowledge the support of machine consciousness inc. (mcon inc.) in the publication of this article. all the data and figures for this article are based on patents and publications relating to the relational robotic controller (RRC)-circuit that have been published in the mcon inc. website www.mcon.org. the authors are particularly grateful for the financial support and permission to publish the mcon data.