The Programming and Training of a Multi-tasking Sensory Motor controlled EVA-Robot Operating in a zero-g Environment

Presented by:
Alan Rosen PhD - MCon Inc.
David B. Rosen - MCon Inc.

1

2

This presentation is about the programming and training of a robotic multitasking system to perform EVA tasks in a zero-g environment.

This is a new technology and I want to thank Rick Wagner for inviting MCon here to share this overview.

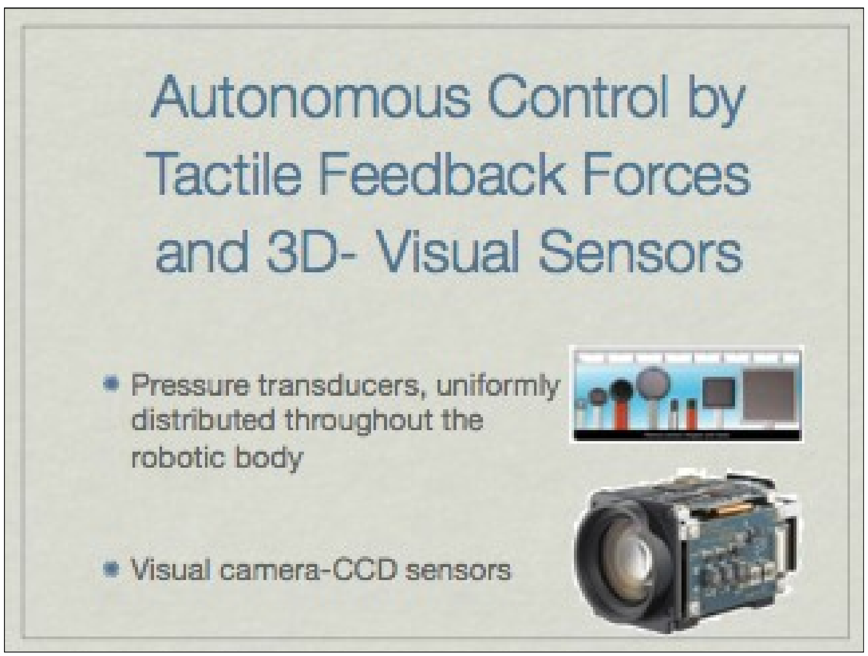
What I'm about to show you is the first practical application of the Relational Robotic Controller designed and developed by MCon inc. The RRC reverse engineers the control functions of the human brain. After publishing our theoretical detailed design approach in scientific journals for the past five years, and lecturing at various scientific meetings. We are now ready for an implementation phase.

Here we apply 2 innovative technologies to the conceptual design of a hybrid system consisting of the NGC-AWIMR and the MCon Inc. RRC robot.

3

The first thing you confront when building an autonomous multi-tasking robots. is the Controller Challenge.
that is
The design and development of a brain like control system that can autonomously control all the actuators of the robotic system for a large variety of multi-tasks with coordination and synchronization of all moving parts.

We believe that we have solved the controller challenge with the development of the Relational Robotic Controller.



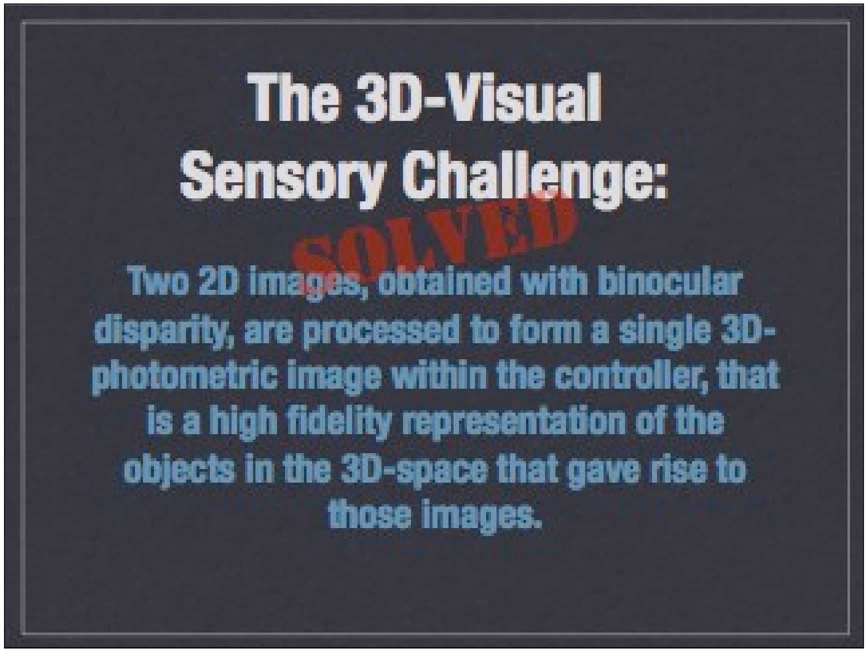
4

The RRC controls the robot autonomously with only 2 inputs:

Pressure transducers, uniformly distributed throughout the robotic body, monitor with high spatial resolution all forces exerted on each and every part the “skin-surface” of the robotic body.

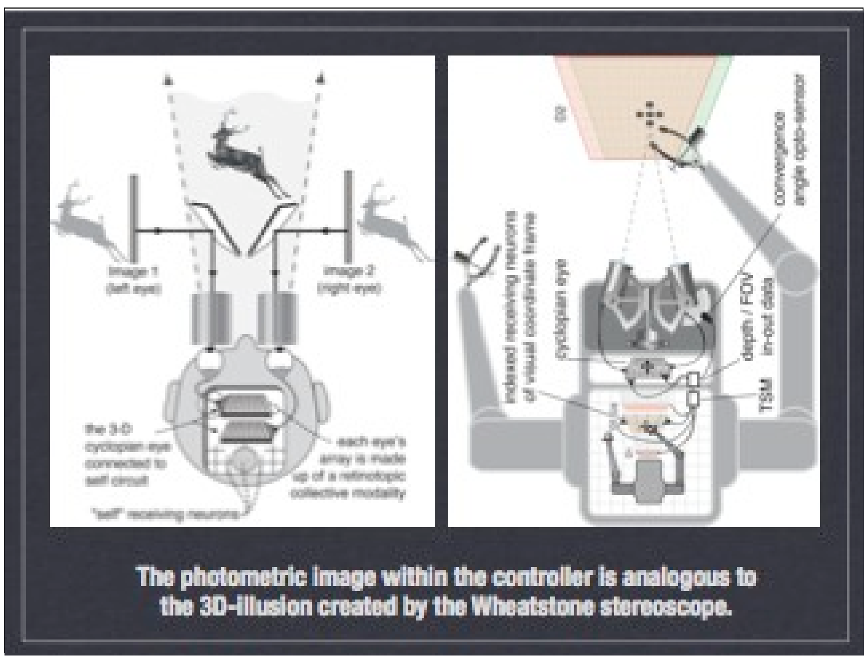
Visual camera-CCD sensors that monitor the visual 3D space in the field of view (FOV) of the robot.

The tactile and visual sensors are configured to form a tactile and visual coordinates (located within the controller), that is a reflection of the robotic body and the visual space in which the robot is operating.



5

The next challenge is 3D-visual sensory challenge: Building a robot that actually sees.
Two 2D images, obtained with binocular disparity, are processed to form a single 3D-photometric image within the controller, that is a high fidelity representation of the objects in the 3D-space that gave rise to those images. We have a solution for that one too.

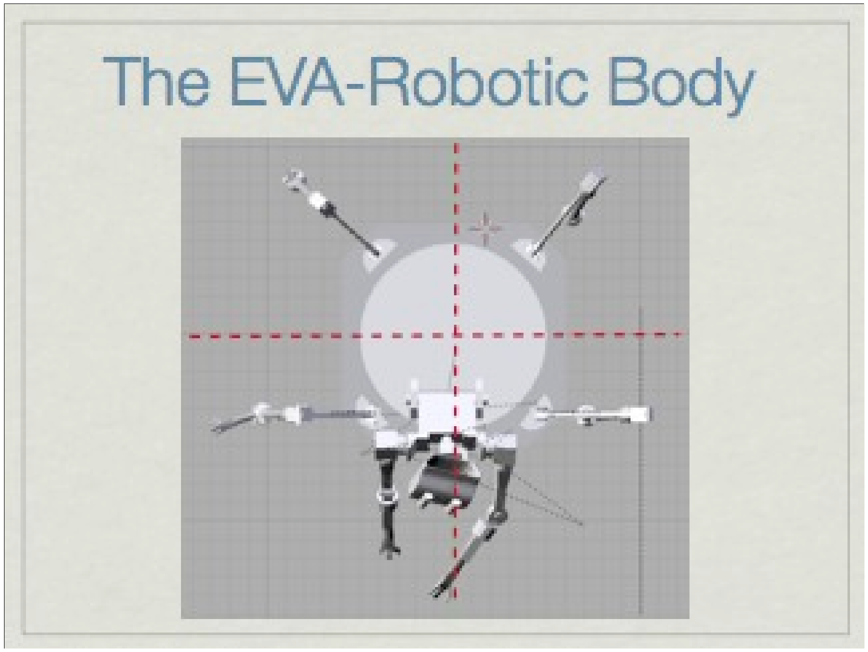


6

We reverse engineered the neurophysiology of the human eyesight and built on the principals of psychophysics to create vision depth perception in the RRC.
Image formation in the RRC is based on the Wheatstone stereoscope.

The 3D-illusion that is demonstrated by the Wheatstone stereoscope is also generated, by processing data within the controller, as a “real” 3D-photometric image.

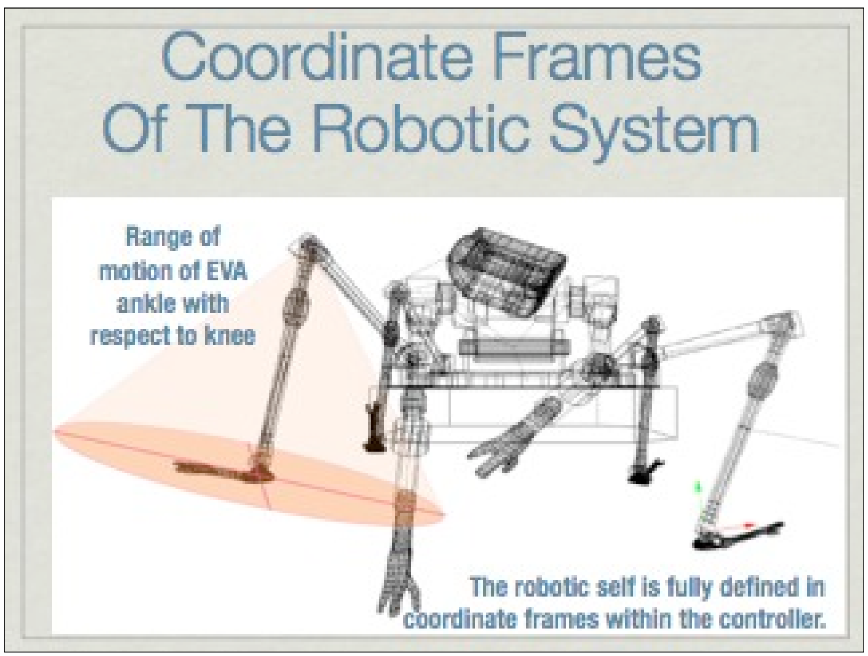
Each individual CCD-sub-image (a portion of some real sub-object), is calibrated with, and located at a depth position, in a coordinate frame within the controller, that is a high fidelity representation of the sub-object located in the 3D-space in which the robot is operating.



7

This figure shows the hybrid system modeled after NGC-AWIMR project and the MCon RRC project.

• Quadrilateral symmetry has been designed into the 3D-visual sensors and the RRC portion
• Each quadrilaterally symmetrical region is similar to the RRC-multitasking robotic system described in the references.



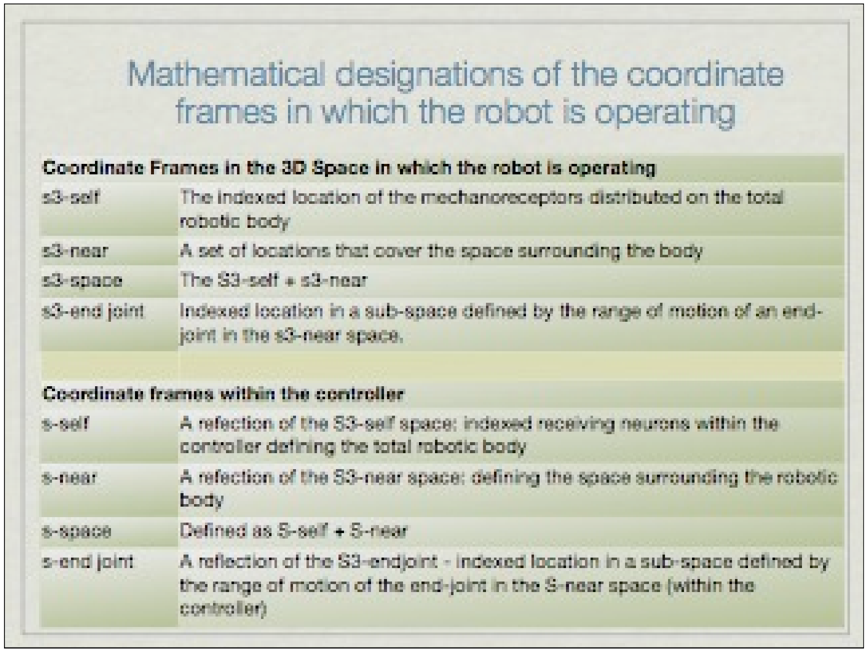
8

The 3D-coordinate space that is reflected internally INTO THE RRC, is a unique feature of our design approach.

The total coordinate space, defined by indexed locations called nodes, is divided into subspaces, one subspace for each joint in the robot.

Each subspace is incorporated into a NODAL MAP MODULE or NMM

For example, the figure depicts one NMM-subspace showing the range of motion of the ankle with respect to knee.



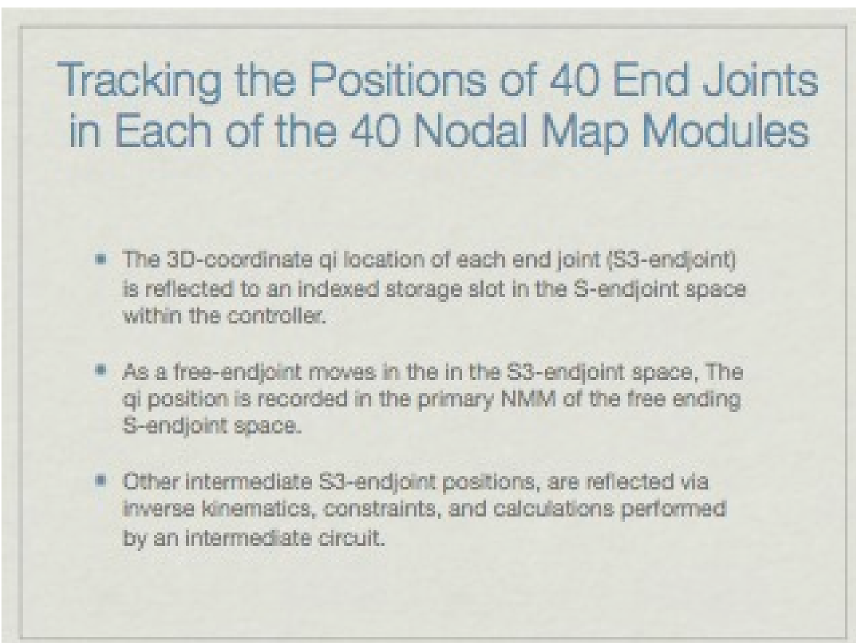
Mathematical designations of the coordinate frames in which the robot is operating

Coordinate Frames in the 3D Space in which the robot is operating	
s3-self	The indexed location of the mechanoreceptors distributed on the total robotic body
s3-near	A set of locations that cover the space surrounding the body
s3-space	The S3-self + s3-near
s3-end joint	Indexed location in a sub-space defined by the range of motion of an end-joint in the s3-near space.
Coordinate frames within the controller	
s-self	A refection of the S3-self space: indexed receiving neurons within the controller defining the total robotic body
s-near	A refection of the S3-near space: defining the space surrounding the robotic body
s-space	Defined as S-self + S-near
s-end joint	A reflection of the S3-endjoint - indexed location in a sub-space defined by the range of motion of the end-joint in the S-near space (within the controller)

9

This is a table showing the mathematical designations of the coordinate frames in which the robot is operating:

- 4-coordinate frames in the Euclidean 3D-space.
- 4 coordinate frames within the controller

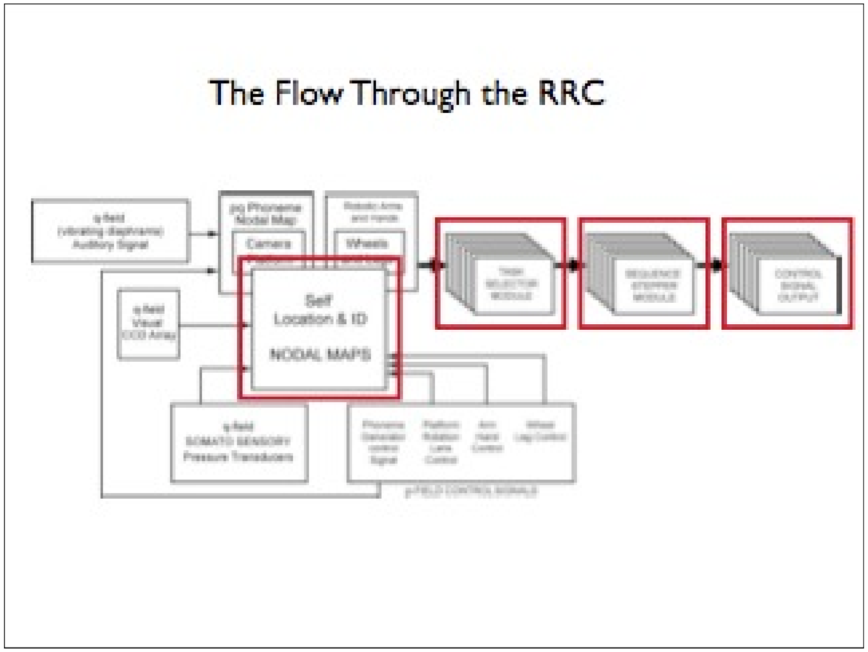


10

Each sub-space, reflected into each of the 40-NMMs, tracks the end joint positions, designated as qi-positiion.

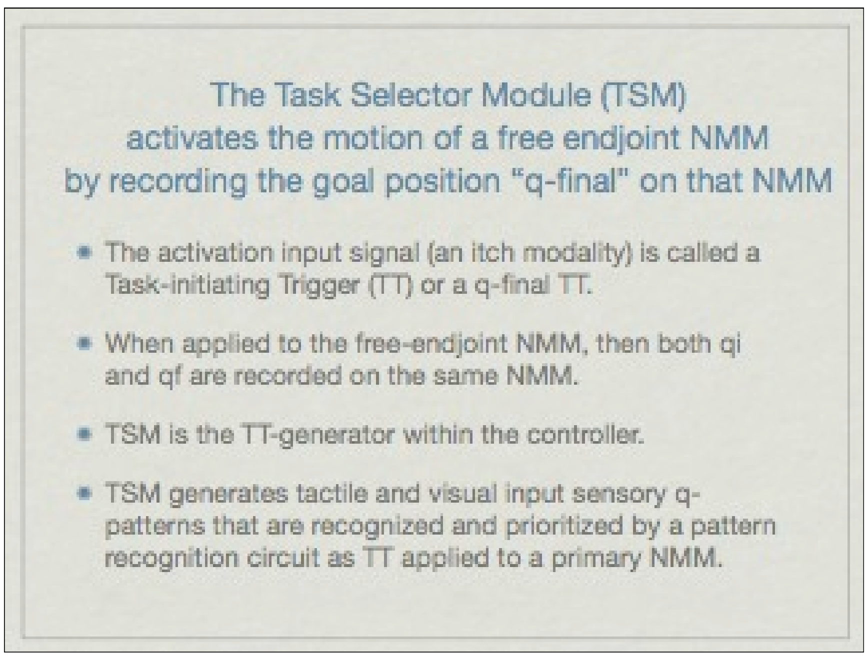
• As the free-endjoint moves in the in the S3-endjoint space, The qi position is recorded in the primary NMM of the free ending S-endjoint space.

• All other intermediate S3-endjoint positions, are reflected via inverse kinematics constraints and calculations performed by an intermediate circuit, to a qi position recorded (stored) in a secondary NMM's for the S-endjoint spaces.



11

This block diagram introduces you to the modules present in the RRC-system. Namely : The TSM NMM, SSM, and COM.

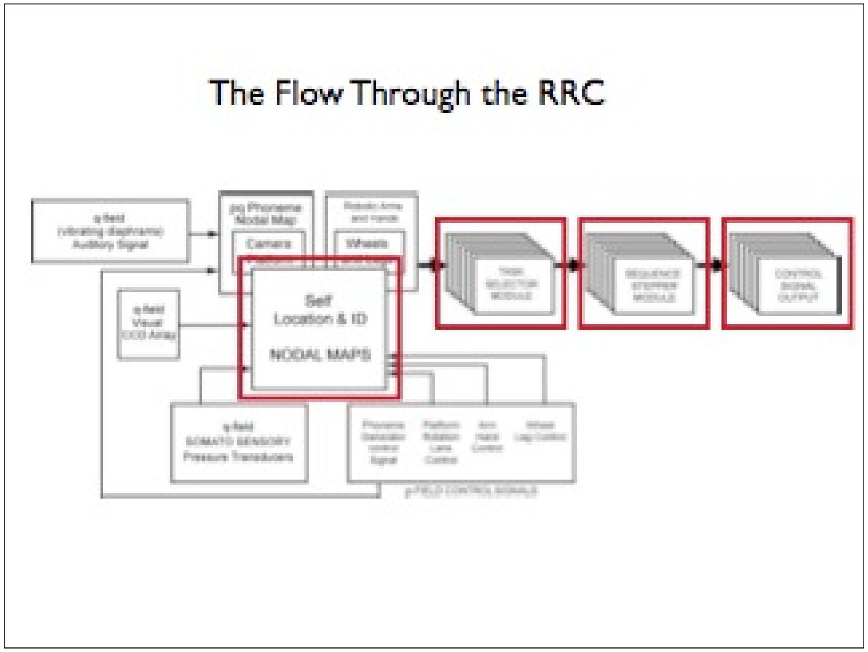


12

Task Selector Module (TSM) activates the motion of a free-endjoint-NMM by recording the goal position q-final on that NMM.

• The activation input signal is called a Task-initiating Trigger (TT) or a q-final TT.

• In general, the TSM generates tactile and visual input sensory q-patterns that are recognized and prioritized by a pattern recognition circuit.



13

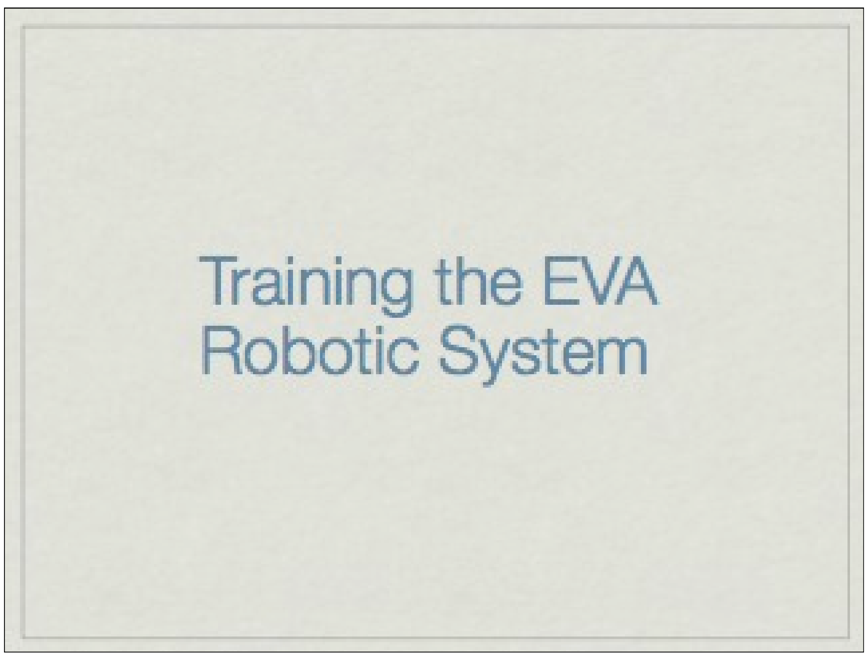
• The TSM generates a high priority TT and applies it to the appropriate NMM-mapping.

• The Sequence Stepper Module (SSM) is activated to scan the region between qi and qf.

• The SSM then generates a pre-planned sequence of control signals, p.

• The Control-signal Output Module (COM) then determines, whether to pause, or to apply the first control signal of the pre-planned sequence to the motor joints of the system.

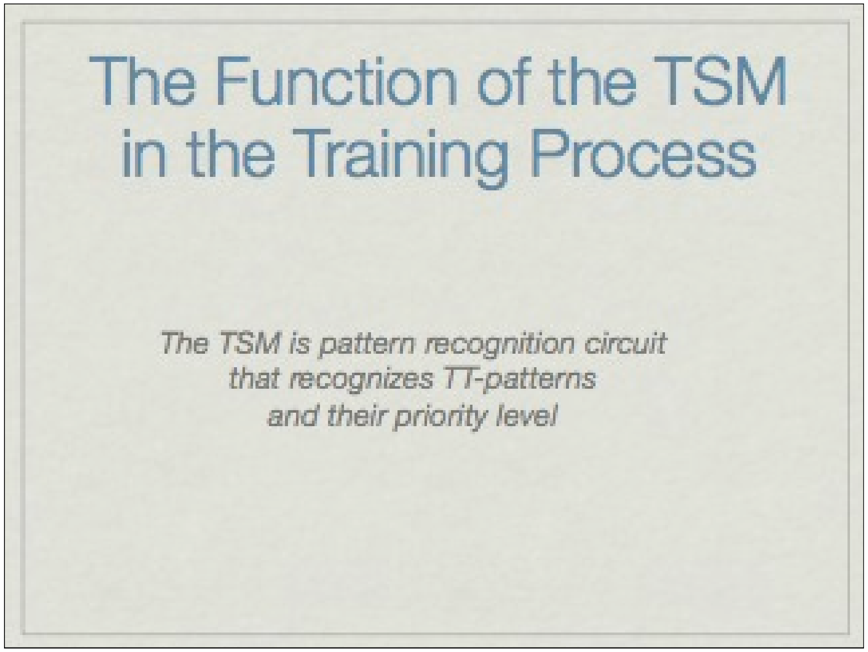
• During that same frame period, all secondary NNMs are trained to utilize the default inverse kinematics position at all secondary NMMs.



14

The description of trial and error programming is the main part of this presentation.

It’s important to note that final training must be performed on a fully operational system.

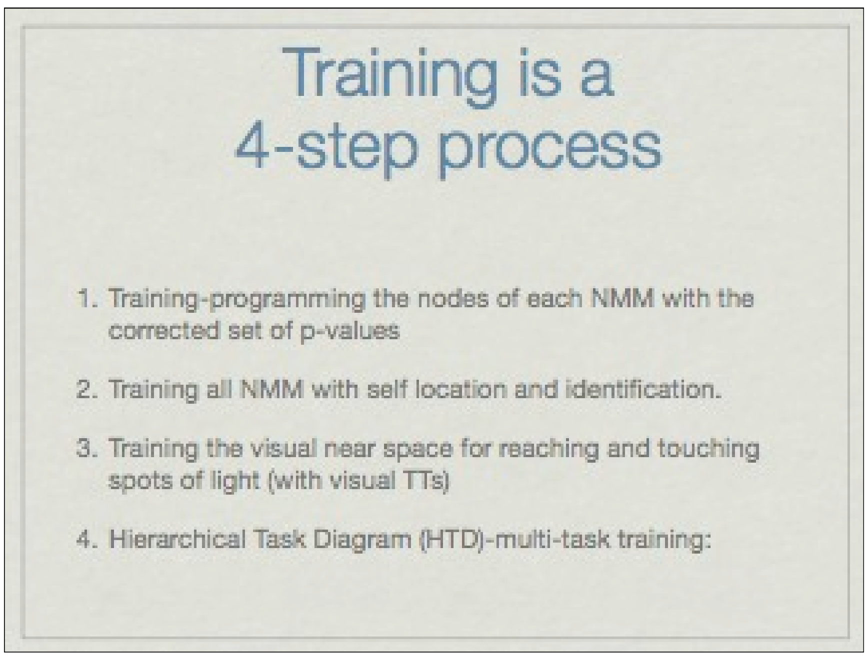


15

The TSM must first be trained. The TSM is a pattern recognition circuit that recognizes TT-patterns and their priority level.

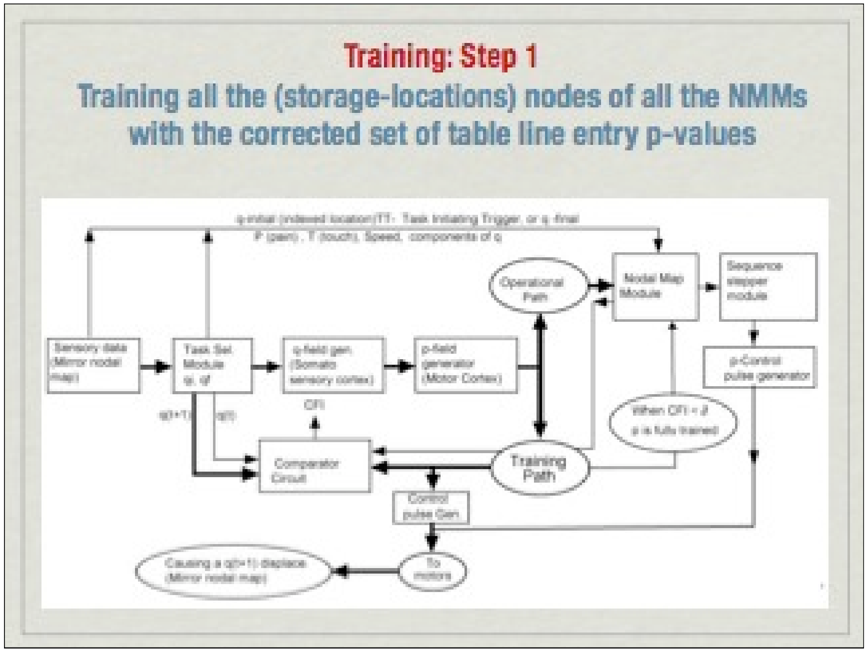
• During normal operations of a robot, the tactile and visual q-field data is applied to the TSM, that recognizes and generates the TT, that activates the SSM, that initiates a sequence of actions that navigate the robot from a q-initial to a q-final position.

• For training purposes, all TTs may be artificially generated by the designer and applied to the appropriate NMM.



16

Training is a 4-step process



17

The first training step: is training all the (storage-locations) nodes of all the NMMs with the correct set of table line entry p-values.
• The correct p-value is that p-signal that causes an exact motor displacement of a robotic part, to an adjacent node.

• Training may be activated autonomously whenever the displacement error is high.

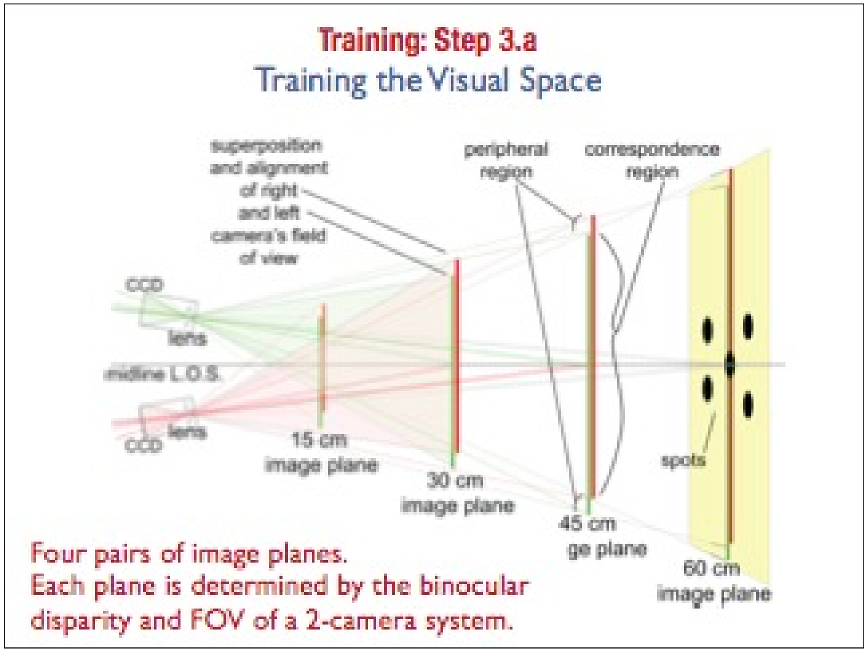
• If the displacement error is high, The Correction Factor Increment > ∂, (CFI> ∂), The table line p-value is corrected to (p+∂p) and repeatedly sent through the system until CFI< ∂.

• When CFI < ∂ the p-value is recorded as the correct table line entry at that node.

18

The second step is training all NMMs with self location and identification:
It consists of training the robot to scratch all itch points.
• For each nodal map, and for each node defined in that map, the training proceeds with the q-final itch locations placed at progressively larger nodal distances from q-initial, until all nodes are covered.

• The robot will have "learned" by this programming methodology, how to move every robotic limb towards any and every part of the robotic body.

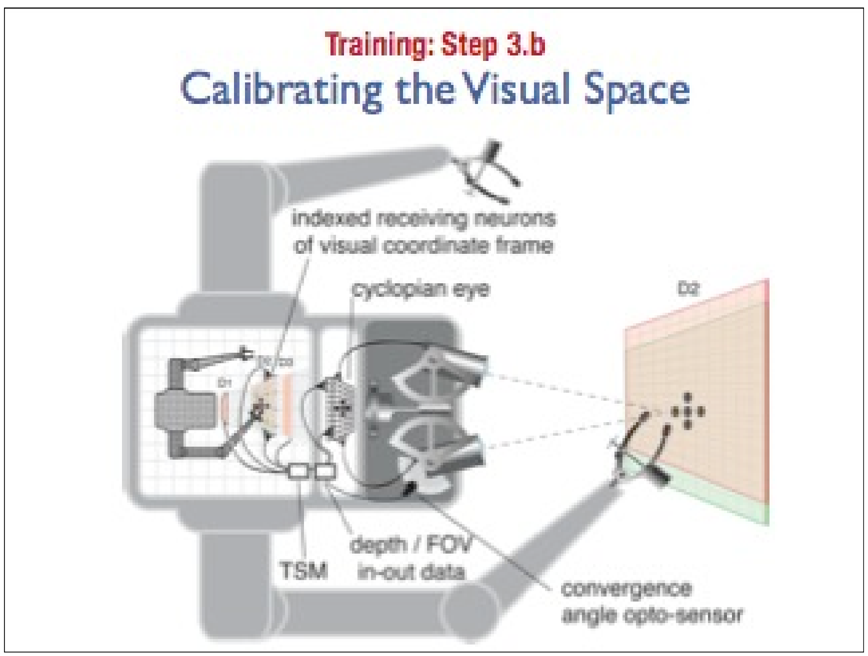


19

The 3rd step is training the visual space: How to calibrate the 3D-visual coordinate frame with the 3D-tactile coordinates.

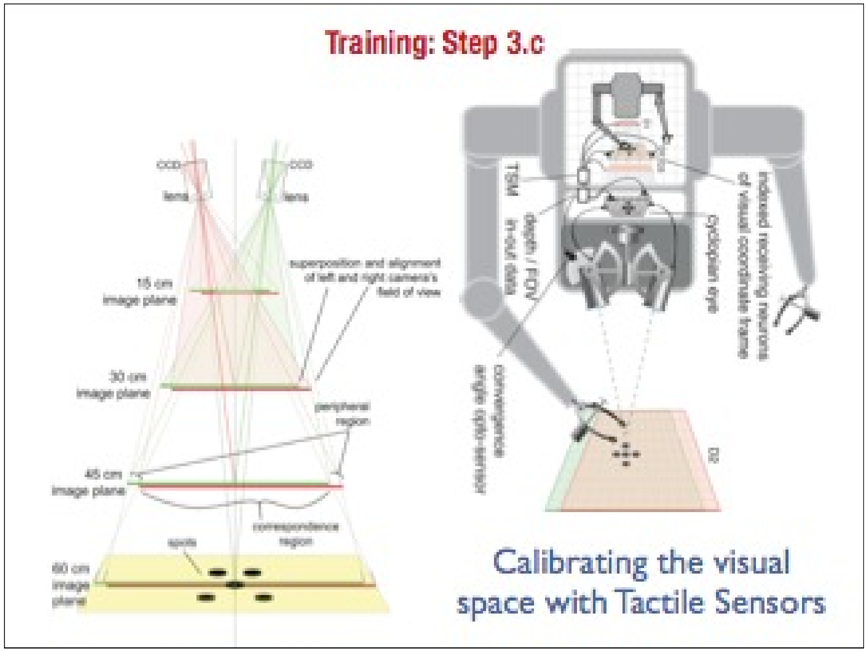
The total FOV-coordinate space may be defined by image planes located at various depths along the midline LOS.

Four pairs of image planes are shown in the Figure.



20

A visual 2D FOV-image plane coordinate frame is placed within the controller at the fixation depth of the central spot of light shown as D2 in the illustration.



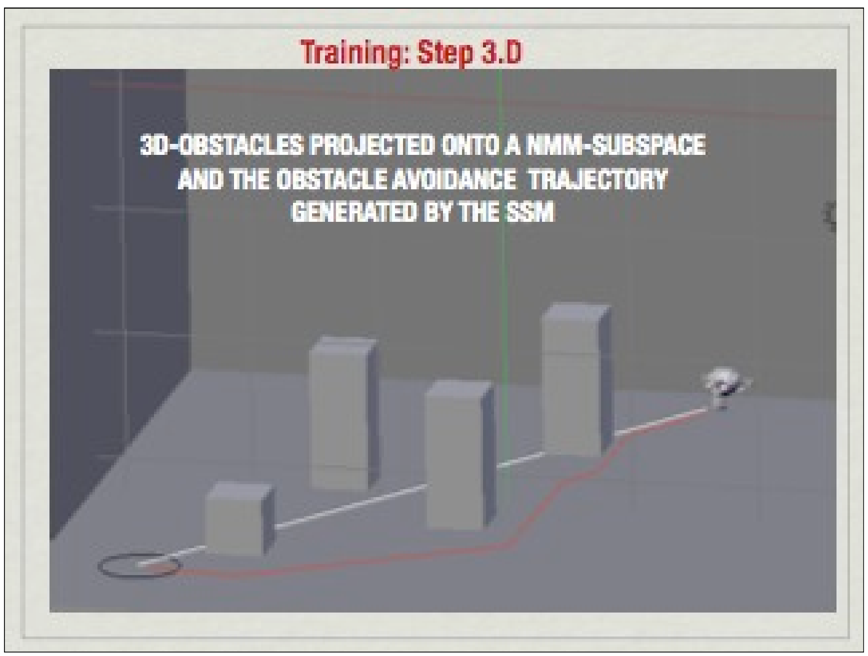
21

The depths of spots of light that are offset from the fixation point must be located by forming an internal retinotopic depth collective within the controller.

• The 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.

• This newly formed retinotopic depth collective may now be indexed to the correct corresponding 3D-position of the tactile coordinates.

to reiterate:
The photometric perceptual images, both depths and locations, are now correlated and calibrated with the actual physical locations of the external objects (spots)

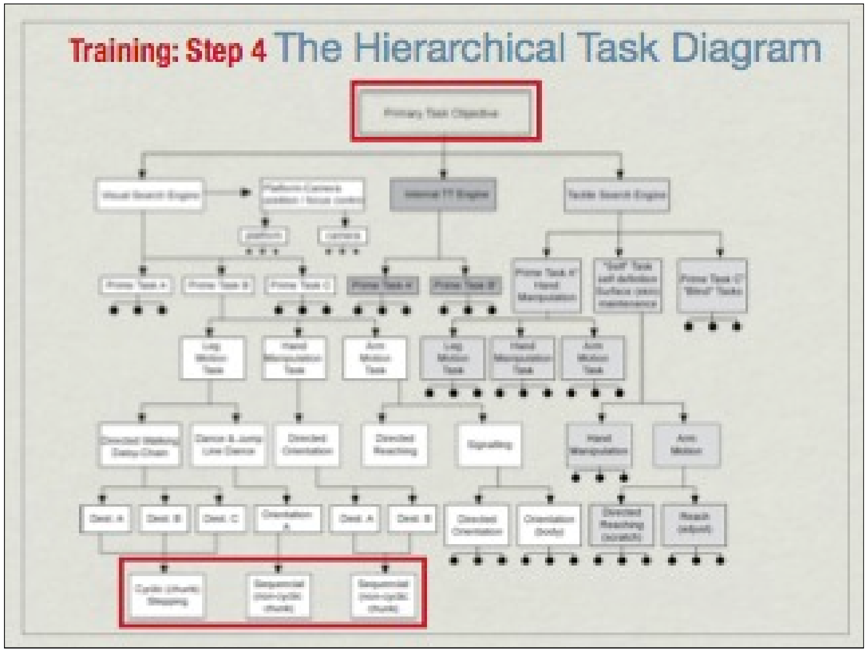


22

This Figure shows 3D-obstacles (a depth collective) projected onto a NMM-subspace.

Note that if the retinotopic depth collective is properly formed
• The photometric positioning of the obstacles in a 3D-nodal coordinate frame, And the calibration of this image with the tactile coordinates,

avoids the pitfalls of attempting to convert a 2D image into a 3D image and trying to calibrate that image with the 3D coordinate frame in which the robot is operating.



23

Training step 4:
The HTD is the top level programming and training specification for a multi-tasking robot.

• Every task listed on the HTD must have a q-final-TT and a priority level assigned to it by the system designer (supervised programming)

• 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.

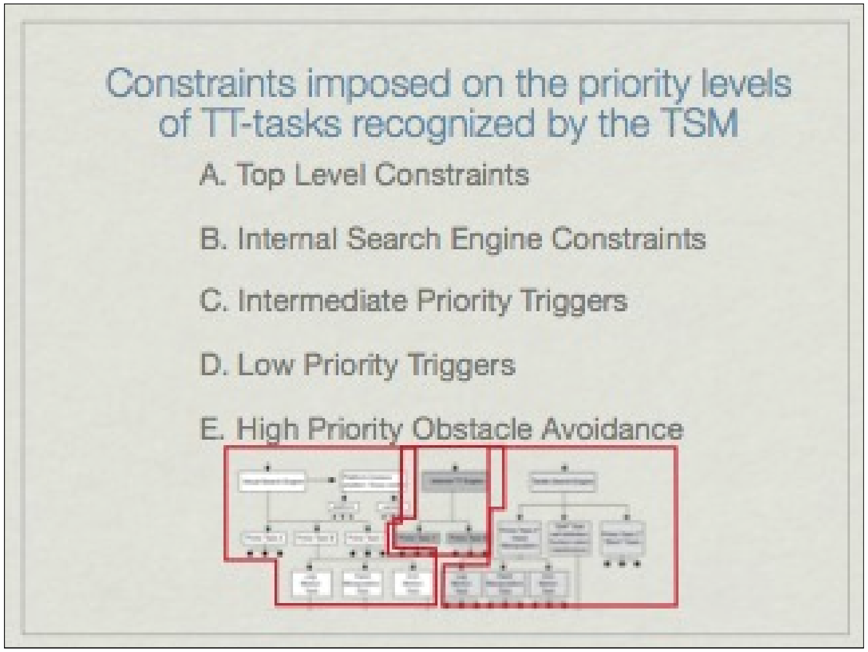
A Daisy Chain Representation of Directed Walking

Chunks	TT=...
Lift leg #1 and leg #3	Reaction force on Legs #2 and #4
Bringing both legs forward 50 cm	zero reaction force on leg #1 and #3
Setting down and shifting the body weight to leg #1 and #3	reaction force on leg #1 and #3

Each sequential set of chunks are initiated by force feedback TTs.

24

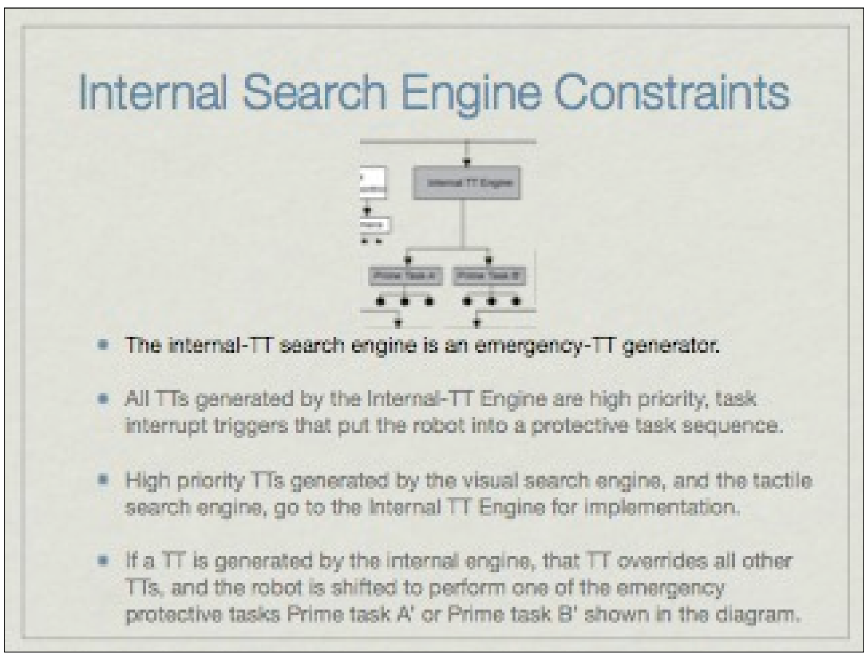
The daisy chain representation of directed walking is controlled by feedback reaction forces on the foot pads.



25

Here is an outline of the constraints imposed on the priority levels of TT-tasks recognized by the TSM.
At the top level, 3-TTs may be generated simultaneously, one by each of the engines shown at the top of the figure.

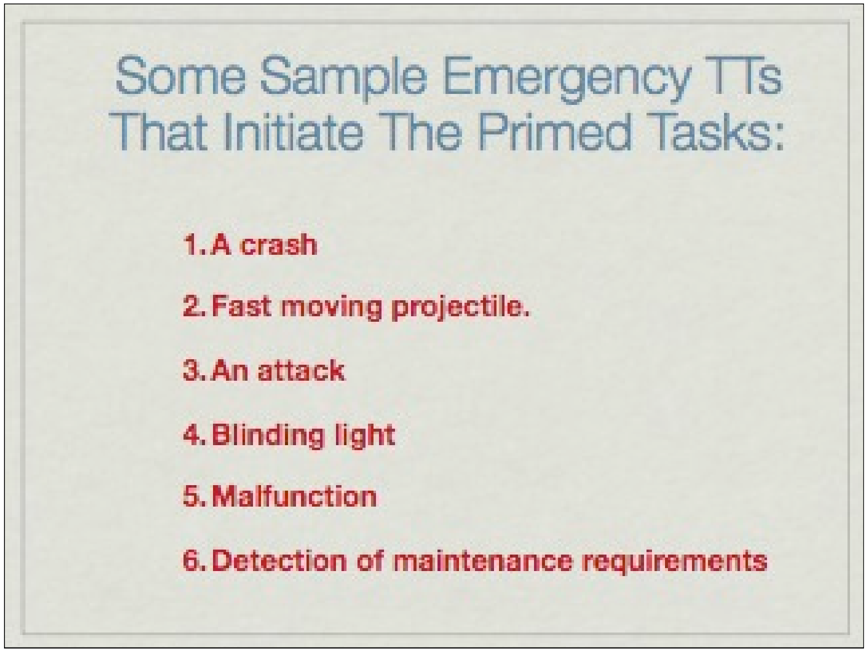
Note that all high level tasks consist of long sequences of simple actions on the bottom levels of the HTD



26

For example, at the top level, The internal search engine is an Emergency TT generator

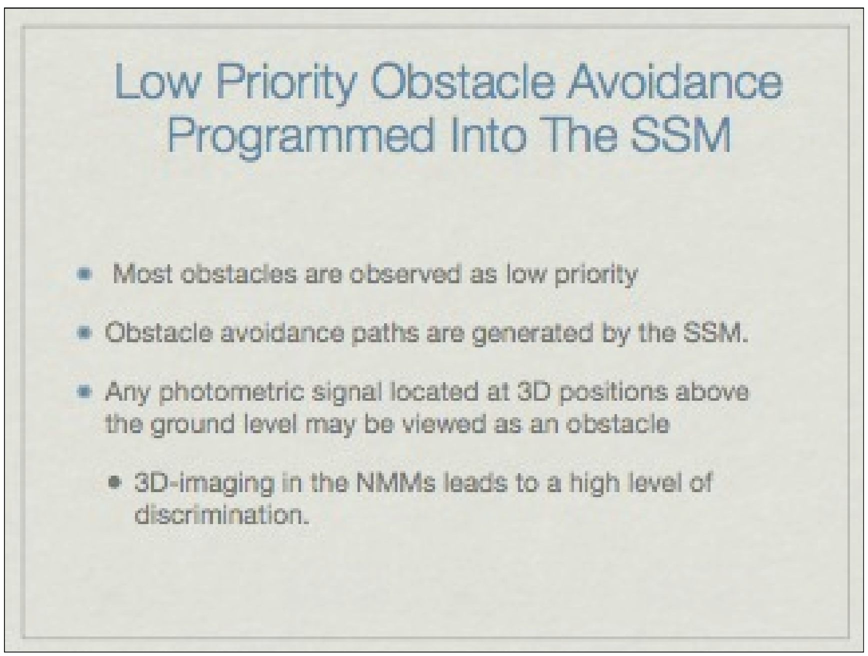
If a TT is generated by the internal engine, it overrides all other TTs, and the robot is shifted to perform one of the emergency protective tasks Prime task A’ or Prime task B’



27

Some sample emergency TT's that initiate the primed tasks are:

1) A crash, or any damaging collision or body blow to the robotic body.
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 Time delayed emergency -TTs.

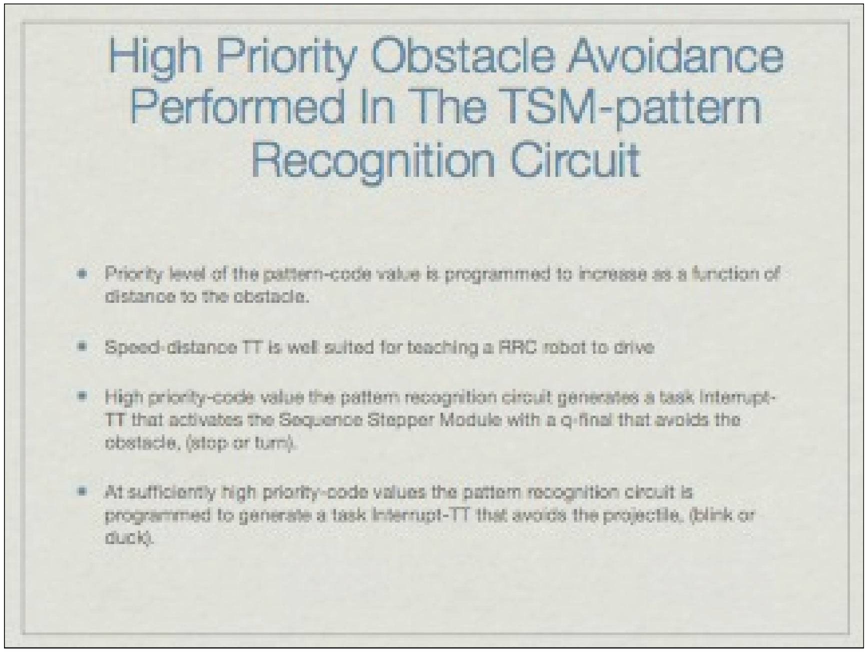


28

Low priority Obstacle Avoidance is programmed into the Sequence Stepper Module (SSM)
• Most obstacles are easily observed as low priority photometric-patterns occurring as photometric sub-images falling on the nodes 0f the 3D-nodal coordinates.

• Any photometric signal (above the noise level located at 3D positions above the ground level), occupying any 3D node may be viewed as an obstacle by the SSM, when scanning for an obstacle free path between qi and qf.

The 3D-imaging in the NMMs lead to a high level of discrimination between full and empty nodes.

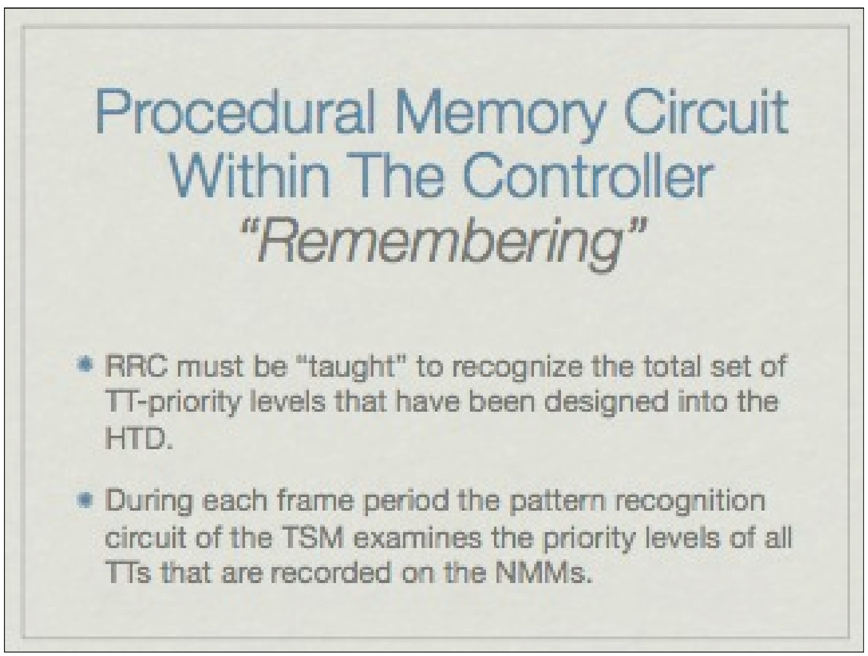


29

High priority obstacle avoidance is performed in the TSM.

Questionable obstacles such as stairs, gates, doors, etc, and moving obstacles (people, balls, projectiles) are best handled by the TSM-pattern recognition circuit.

For example, for all moving obstacles, the priority level assigned to the visual pattern of the obstacle must be a function of distance and speed.



30

The programmed TSM-pattern recognition circuit is a procedural memory system for the controller.

The TSM-pattern recognition circuit and the SSM-scanner, operating on the NMM, are analogous to the procedural memory system in the brain.

The robot remembers and may perform a complex sequence of tasks all aimed at fulfilling the prime tasks shown on the HTD .

For example, once you teach a robot how to tie its shoe laces, it never forgets.
(yes-with the RRC, human-like dextrous manipulation is within our robotic grasp)

EVA Robotic System

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

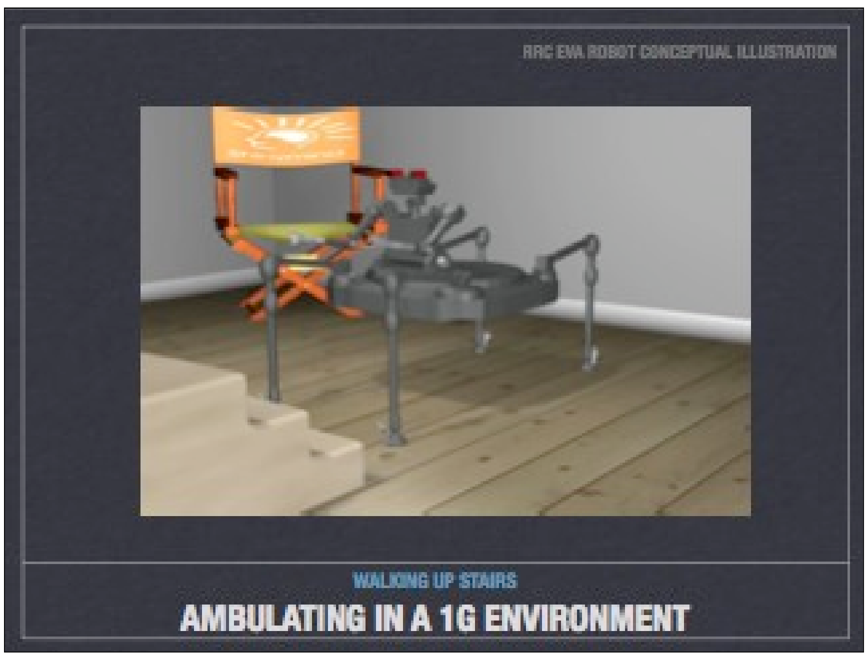
Operating in a 1g environment:

- Inspecting thermal blankets and thermal tiles
- Repair and replacement of parts in 1g

Operating in a zero-g environment:

- Ambulating in a zero g environment
- EVA-Inspection of the exterior surface of a spacecraft for micrometeorite hits

31

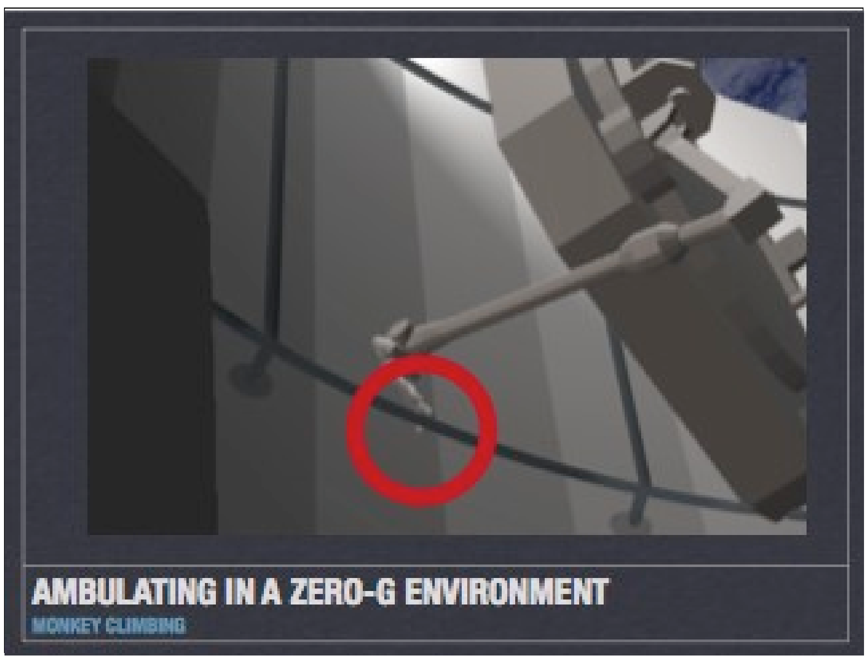


32

Ambulating in a 1g environment
This is a conceptual animation of an ambulating EVA-robot negotiating some steps and arriving at a destination to perform a surface inspection.

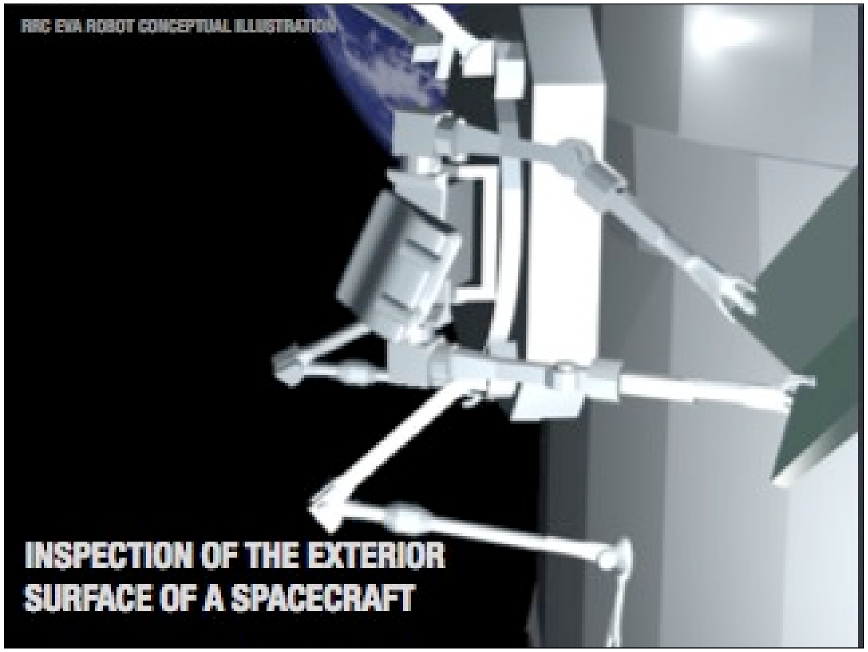
33

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.



34

Ambulating in a zero-g environment
The figure 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.



35

Inspection of the exterior surface of a spacecraft.
This figure 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 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.

36