Neural Basis Function Control of

Super Micro Autonomous Reconfigurable Technology (SMART) Nano-Systems

Steven A. Curtis
NASA Goddard Space Flight Center, Greenbelt, MD, 20771

Michael L. Rilee
L-3 Communications GSI, Largo, MD, 20774

Walter F. Truszkowski and Cynthia Y. Cheung
NASA Goddard Space Flight Center, Greenbelt, MD, 20771

and

Pamela E. Clark
L-3 Communications GSI, Largo, MD, 20774

First AIAA Intelligent Systems Technical Conference
20-22 September 2004

Nano-Systems Revolution

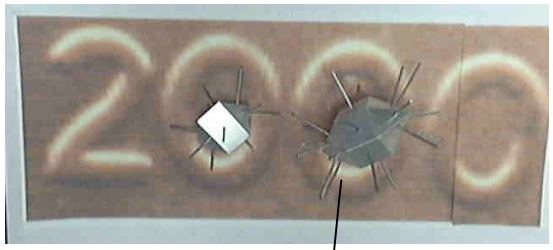
There's plenty of room at the bottom.

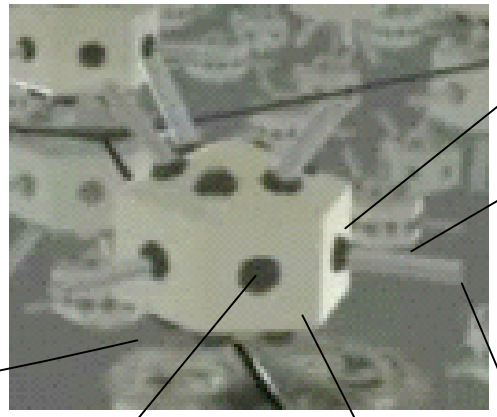
- R. Feynman

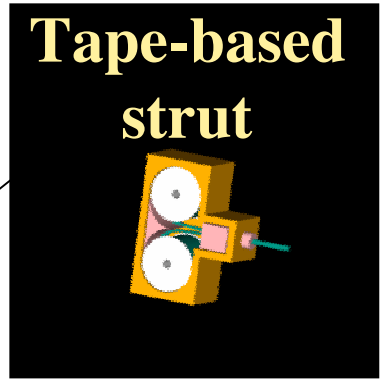
- Numbers of system elements is huge
- Number of possible interactions is extreme
- Time scales suffer all extremes
  - reflexive, long term, "c" - *yes "c"*

Super Miniaturized Autonomous Technology

Nodes & Struts







Extensible/retractable strut

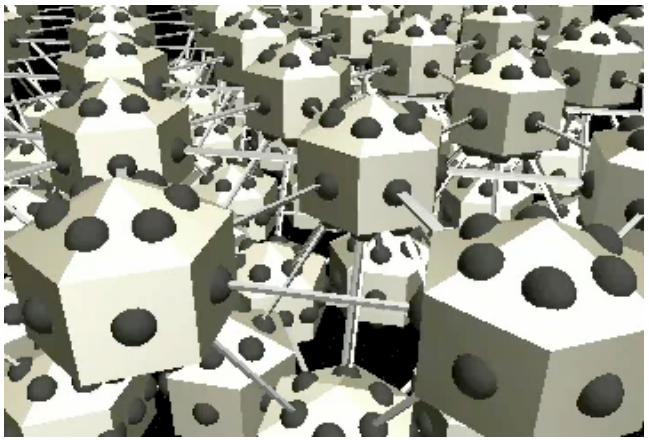
Clamping mechanism on end

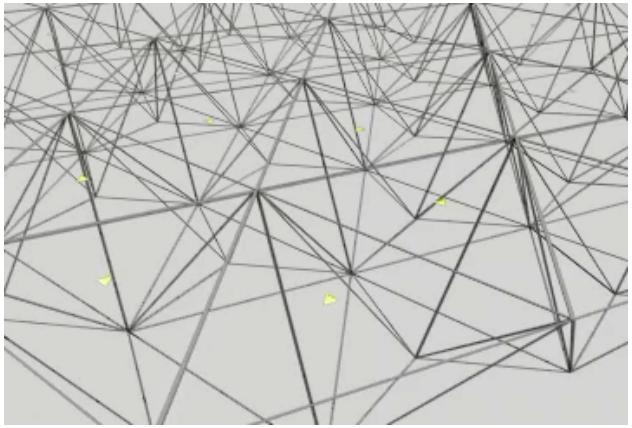
Socket for clamp

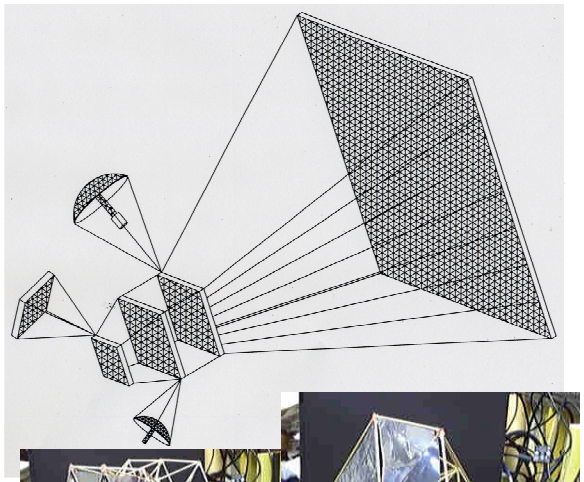
Node

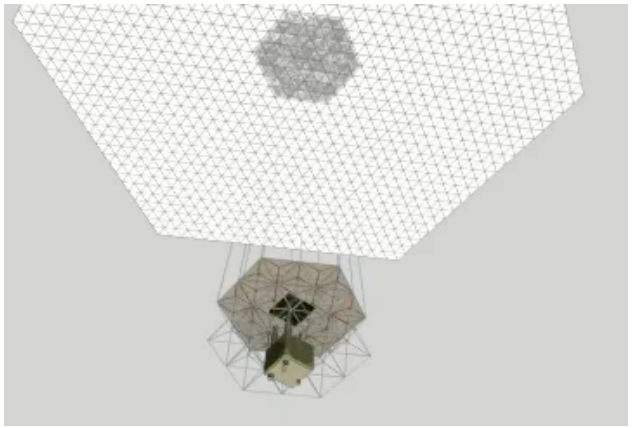
Super Miniaturized Autonomous Technology

Structures









Current Approaches

- Low-level - Behavior based - Reflexive
  - Simple behaviors
  - Fast & efficient
  - Meaningful outcomes are a problem
- High-level - Intelligent - Deliberative
  - Complex sets of actions
  - Careful, slower, expensive
  - Meaning is central- details are the problem
- Hybrid approaches
  - tend towards deliberative with supporting low-level control

Difficulties

- Scaling high-level to complex systems
  - Too complicated at the low-level
- Scaling low-level to purposeful behavior
  - Where and how does this behavior emerge?

What is the aim of the NBF?

- Behaviors
- Combining behaviors
- Conflicting behaviors
- Self similarity -- scale invariance
- The meaning of behaviors, situation
- Software or hardware implementation
- Three-dimensional, concurrent
- Inherently distributed

Neural Basis Function Architecture

- Neural
  - Trainable, evolvable, adaptable
  - Inherently scalable to different geometries
  - Synchrony is not required
- Basis
  - A set of functions that span a space of functions
  - Composition as an interface issue
- Function
  - Drives the system state
  - Might be coupled to actuators or sensors

Autonomous Nano Technology Swarm
Autonomous Worker Neural Software System: *ANTS Worker*
Neural Map for a Neural Basis Function

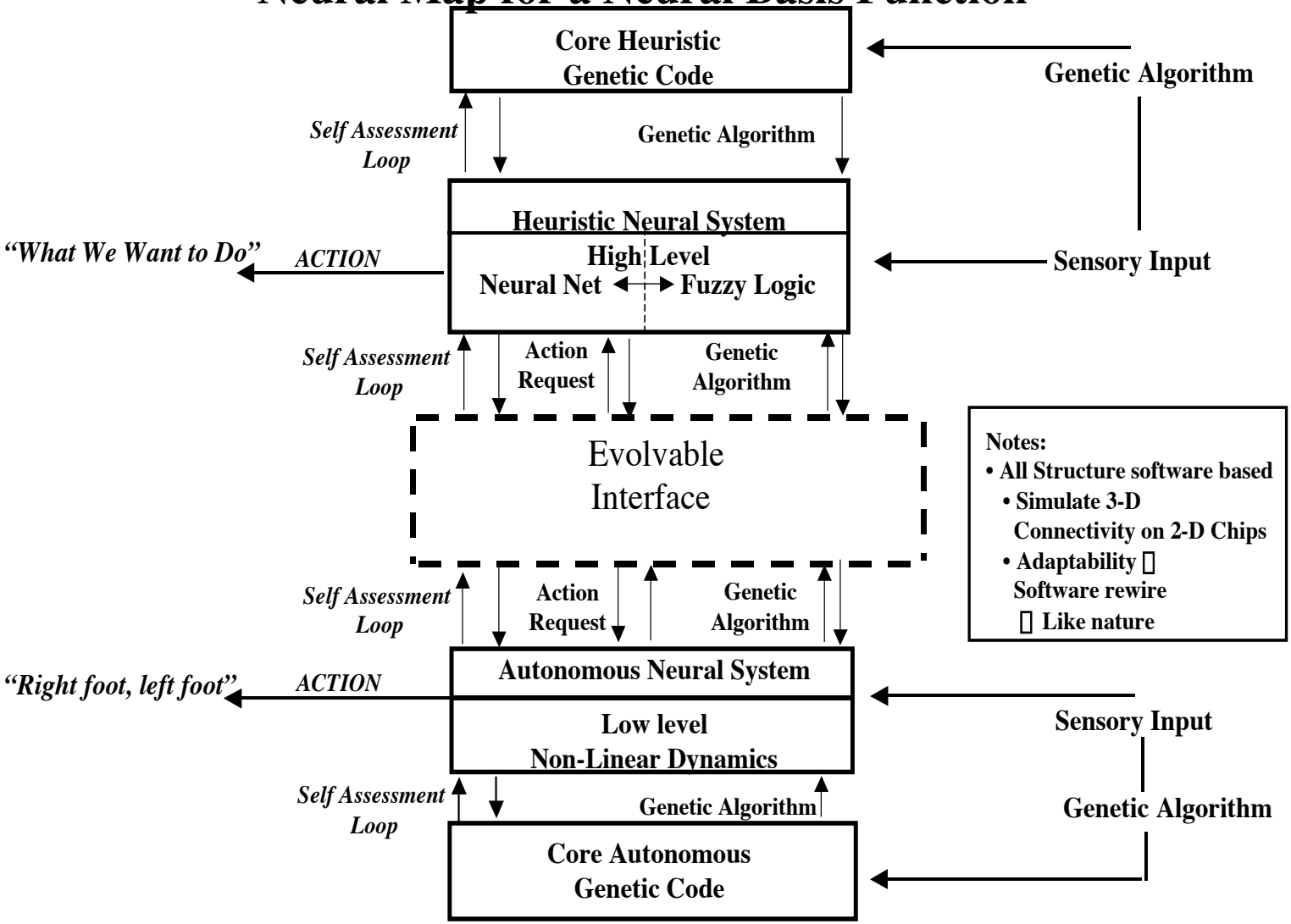
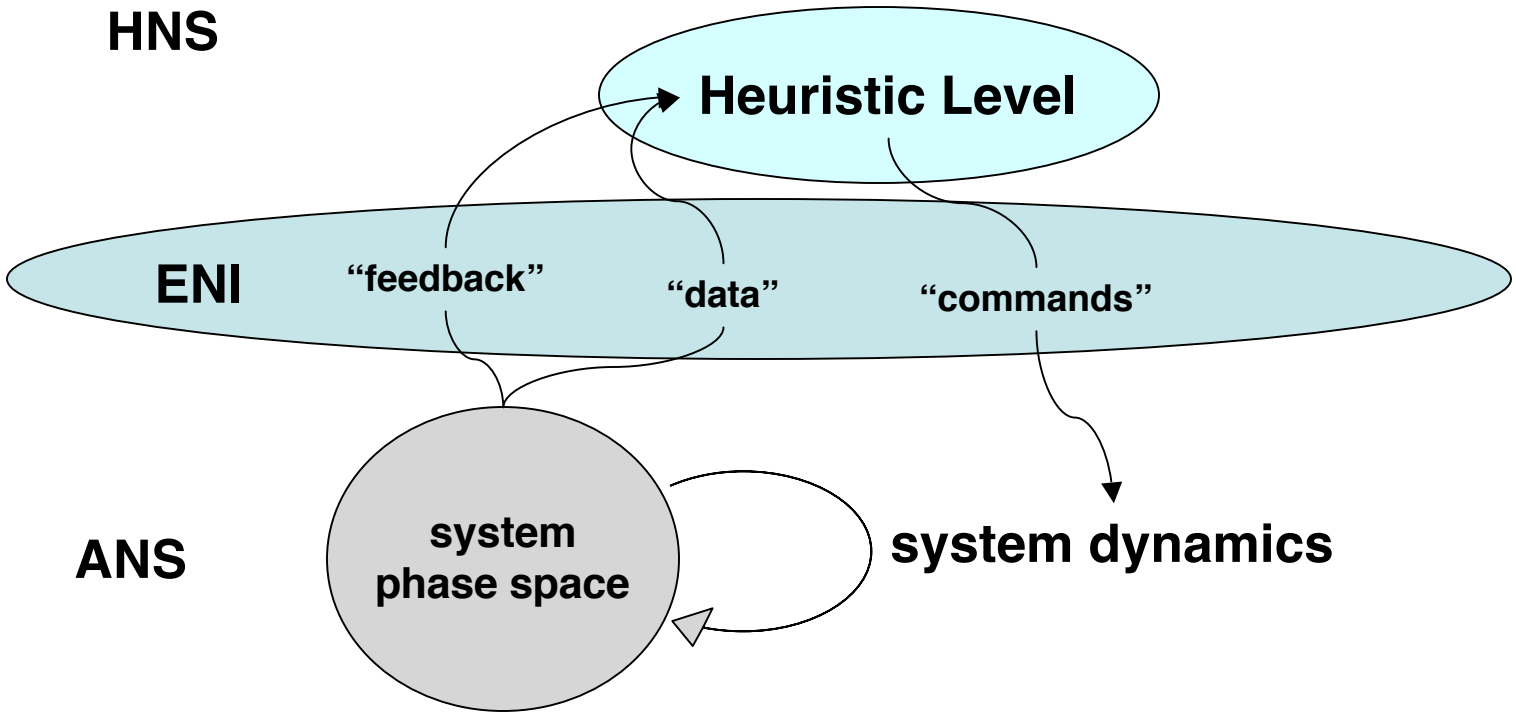
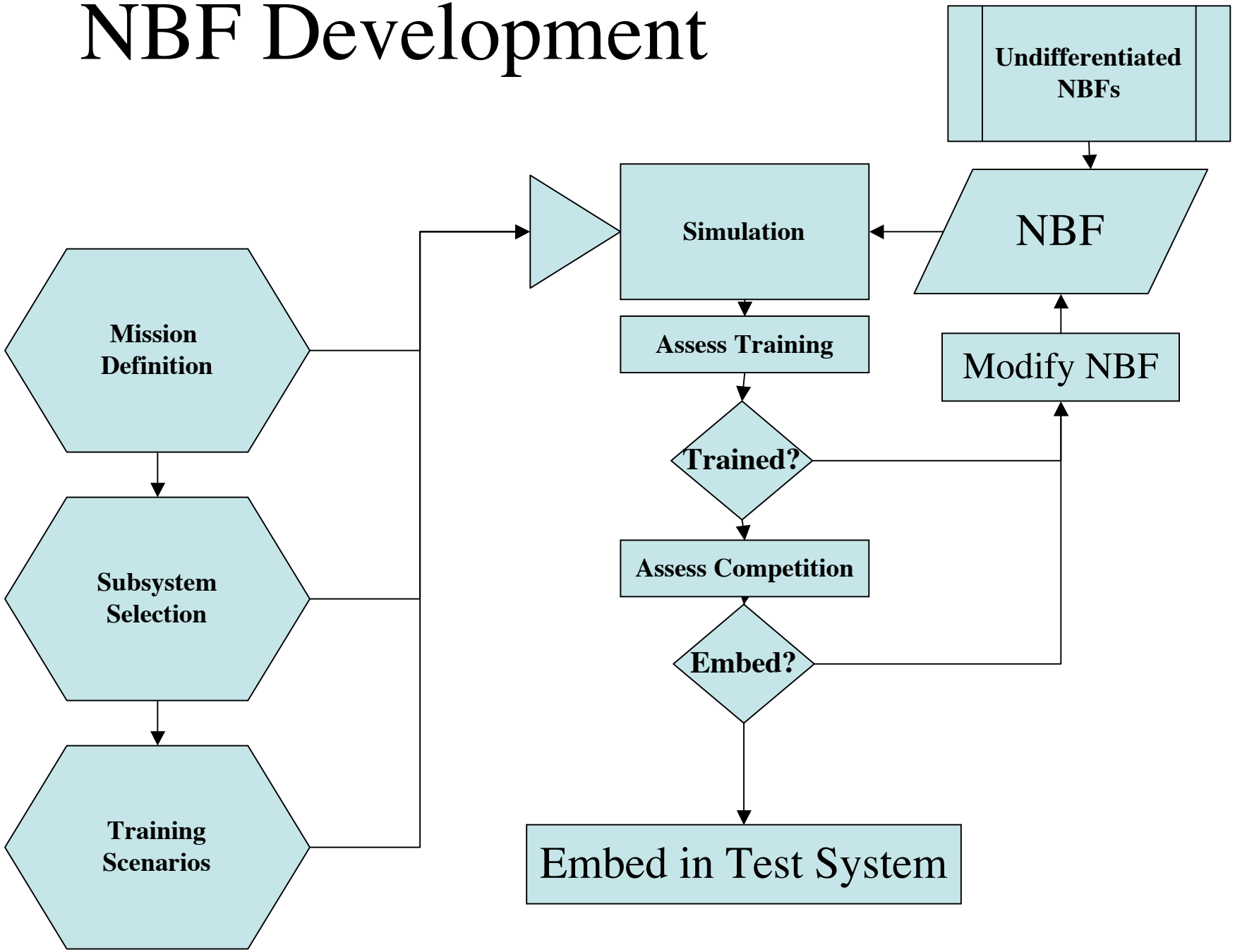


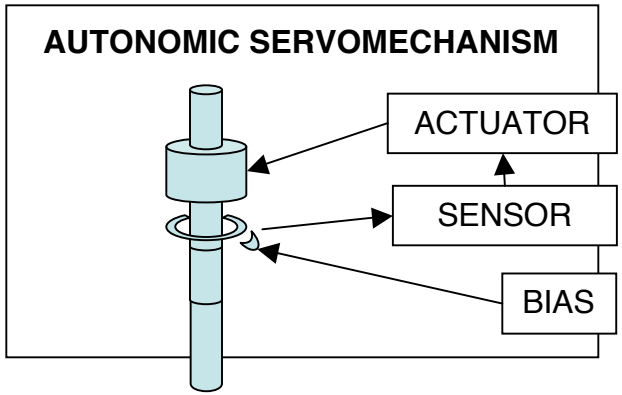
Figure 2 Conceptual diagram of the neural basis function


HNS
Heuristic Level
ENI
“feedback”
“data”
“commands”
ANS
system
phase space
system dynamics

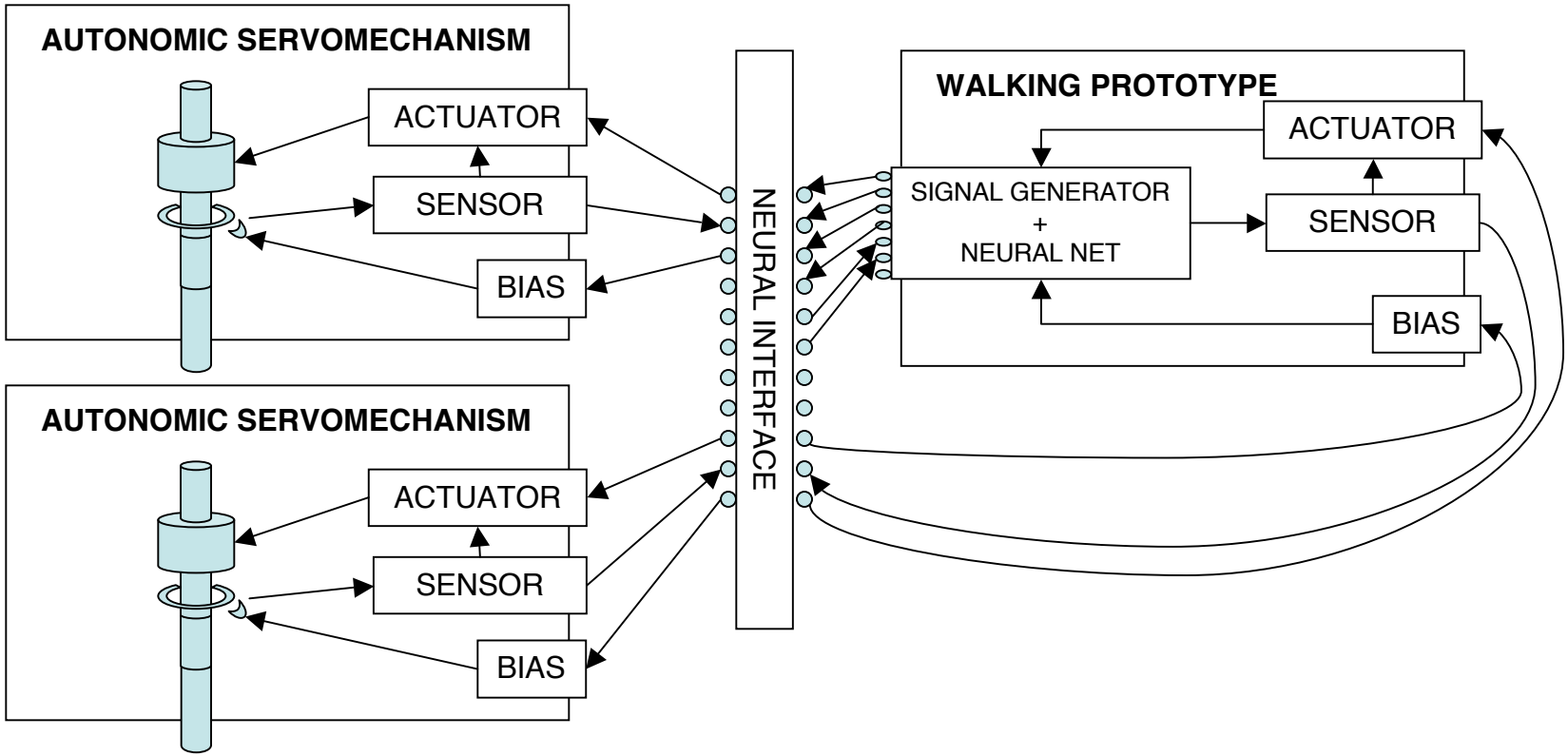
NBF Development



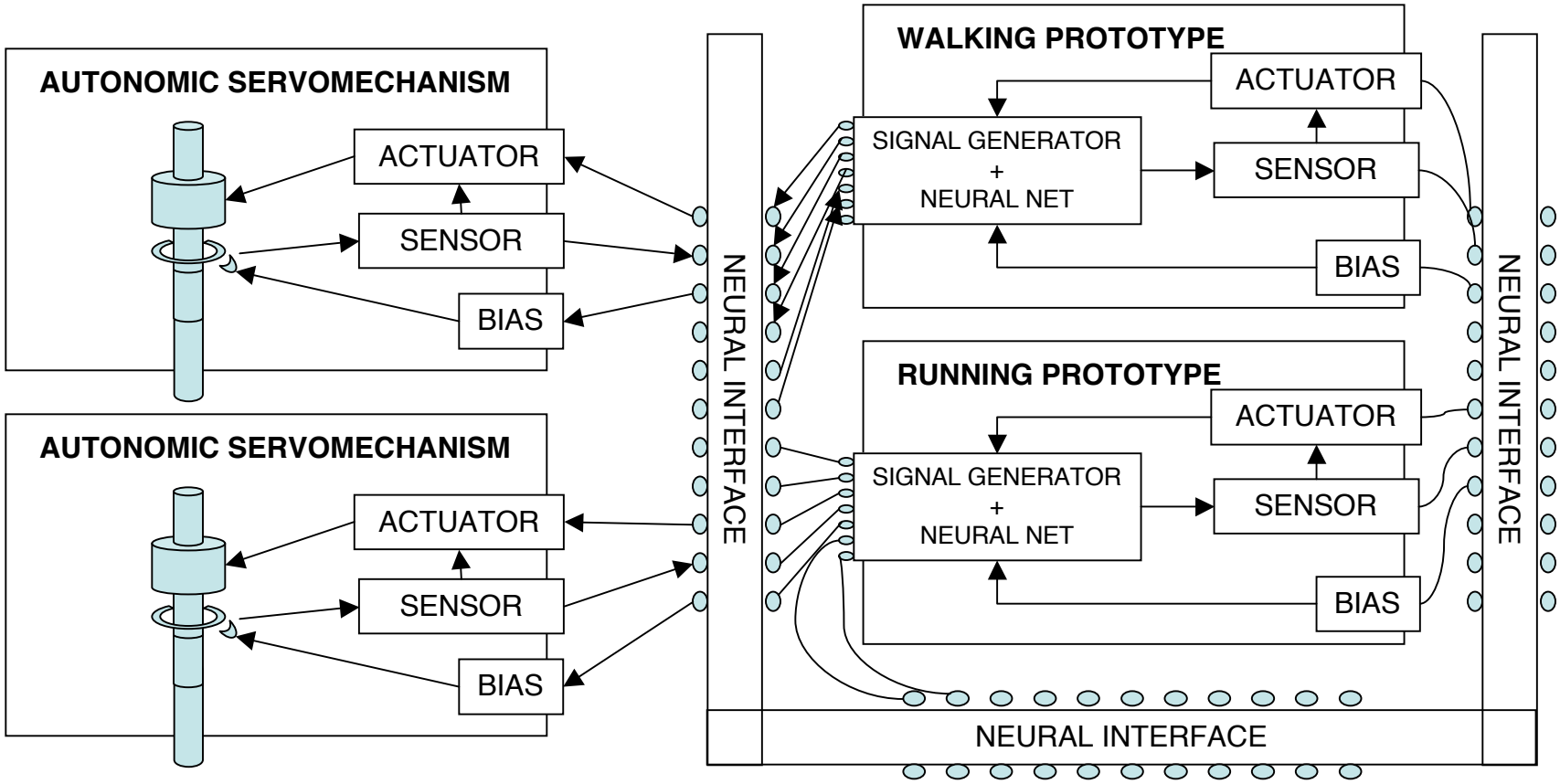
Tilden-like approach to Behavior-based Control



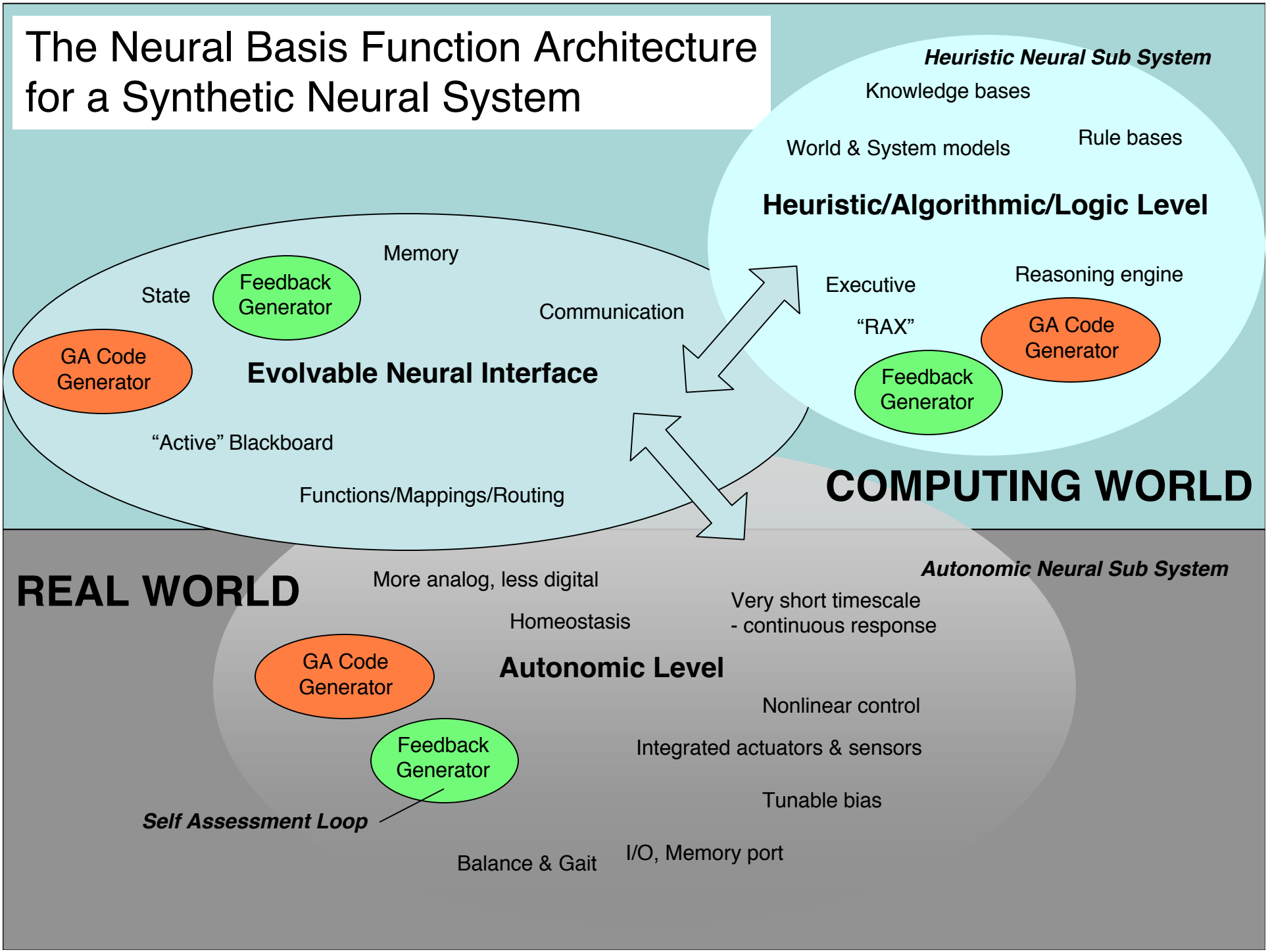
SENSING IS ACTUATION

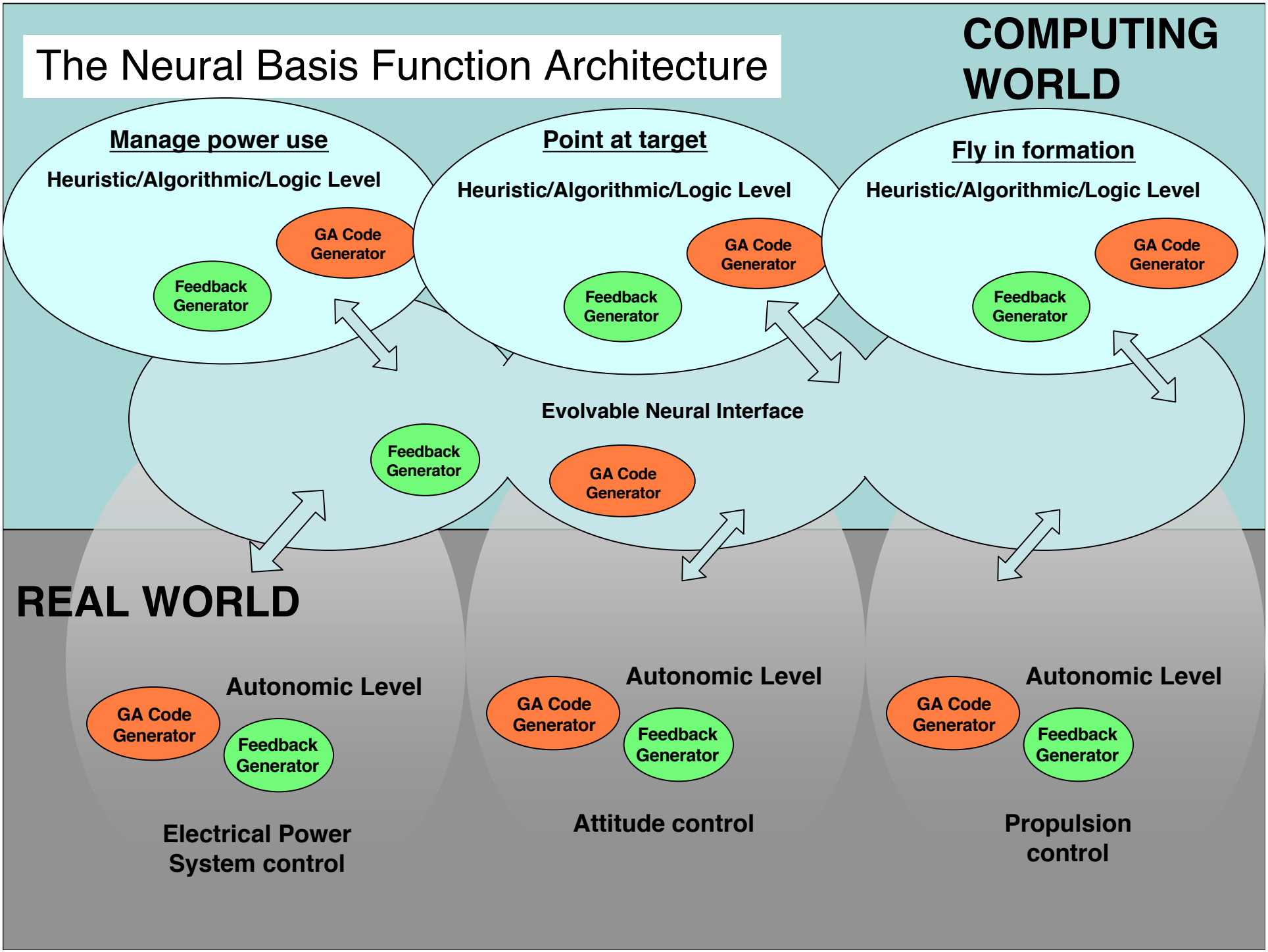


SENSING IS ACTUATION

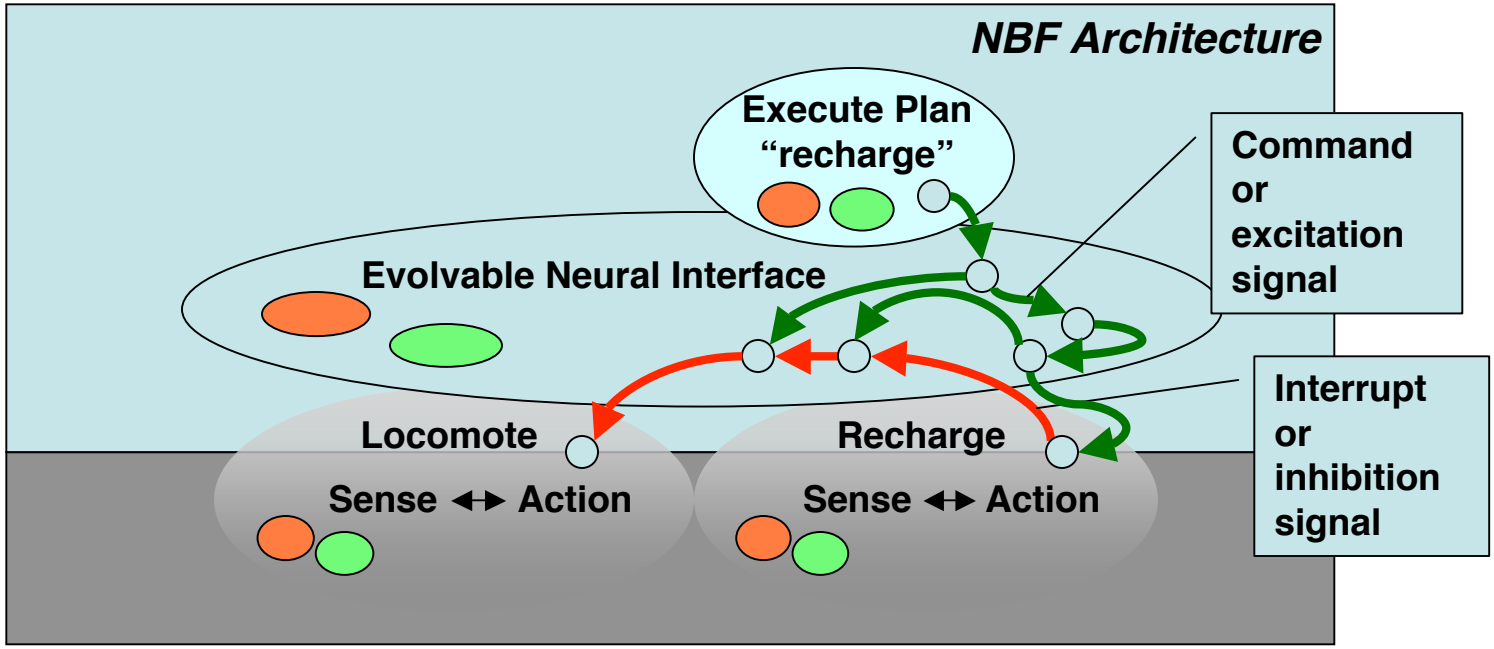


SENSING IS ACTUATION


The Neural Basis Function Architecture for a Synthetic Neural System
Heuristic Neural Sub System
Knowledge bases
World & System models
Rule bases
Heuristic/Algorithmic/Logic Level
Memory
State
Feedback Generator
Communication
Executive
Reasoning engine
"RAX"
GA Code Generator
GA Code Generator
Evolvable Neural Interface
Feedback Generator
"Active" Blackboard
Functions/Mappings/Routing
COMPUTING WORLD
REAL WORLD
More analog, less digital
Autonomic Neural Sub System
Very short timescale
- continuous response
Homeostasis
GA Code Generator
Autonomic Level
Nonlinear control
Feedback Generator
Integrated actuators & sensors
Tunable bias
Self Assessment Loop
I/O, Memory port
Balance & Gait


The Neural Basis Function Architecture
COMPUTING WORLD
Manage power use
Heuristic/Algorithmic/Logic Level
GA Code Generator
Feedback Generator
Point at target
Heuristic/Algorithmic/Logic Level
GA Code Generator
Feedback Generator
Fly in formation
Heuristic/Algorithmic/Logic Level
GA Code Generator
Feedback Generator
Evolvable Neural Interface
Feedback Generator
GA Code Generator
REAL WORLD
Autonomic Level
GA Code Generator
Feedback Generator
Electrical Power System control
Autonomic Level
GA Code Generator
Feedback Generator
Attitude control
Autonomic Level
GA Code Generator
Feedback Generator
Propulsion control

Example of HNS directed activity



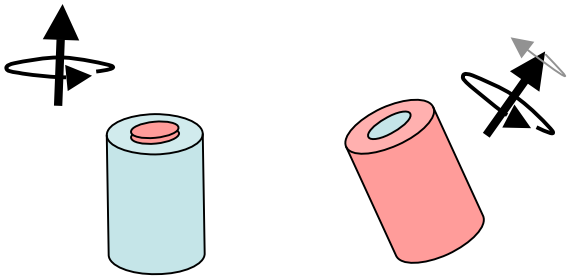
Conflicts and interrupts are processed and communicated by the ENI.

Communication paths and signal transforms are trained to optimize system performance metrics during simulation and initialization.

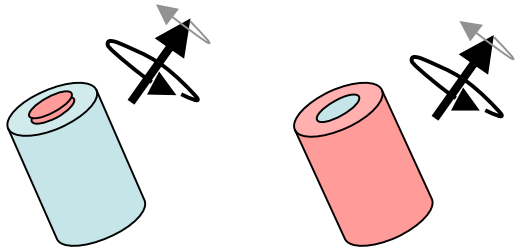
Hubble Space Telescope Autonomous Rendezvous & Capture

- A test of the NBF concept
- Chaotically Tumbling Target (CTT)
  - Internal degrees of freedom
  - Driven by environmental couplings
  - To be enhanced with HST dynamics model
- Autonomous Capture Vehicle (ACV)
  - Astrodynamics simulation
  - Reactive nonlinear control (a la Tilden)
  - High-level command executive
  - Evolvable Neural Interface (ENI)

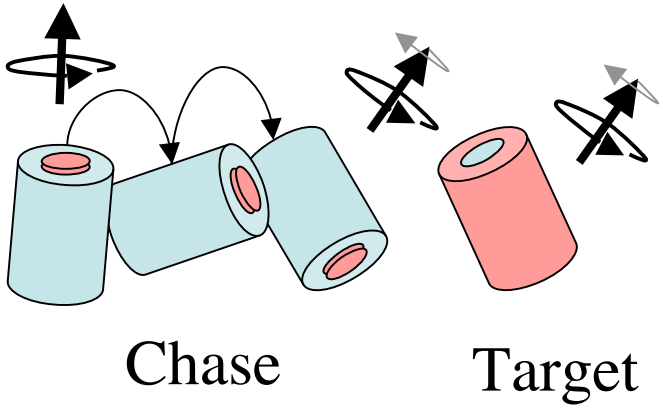
Low-level NBF Control Demo
Rendezvous & Capture of a Chaotic Tumbler



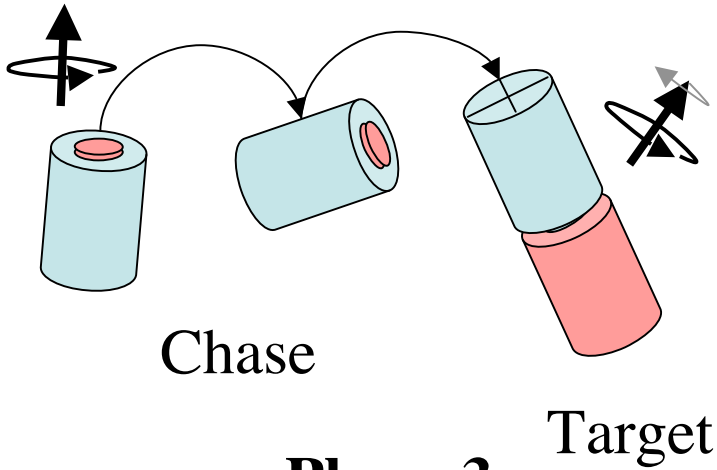
Initial State



Phase 1. Control State

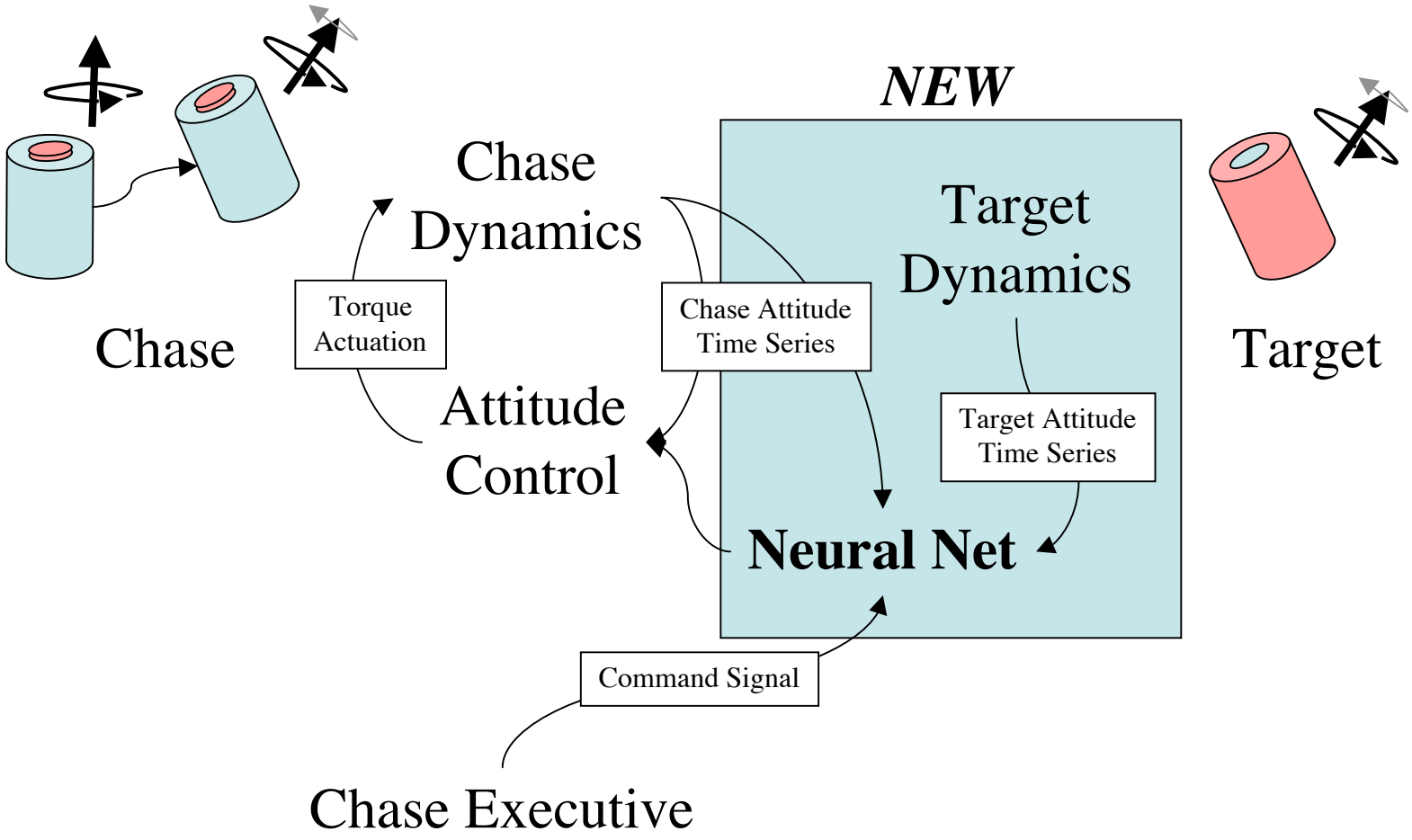


Phase 2.
Detailed Control Chase State



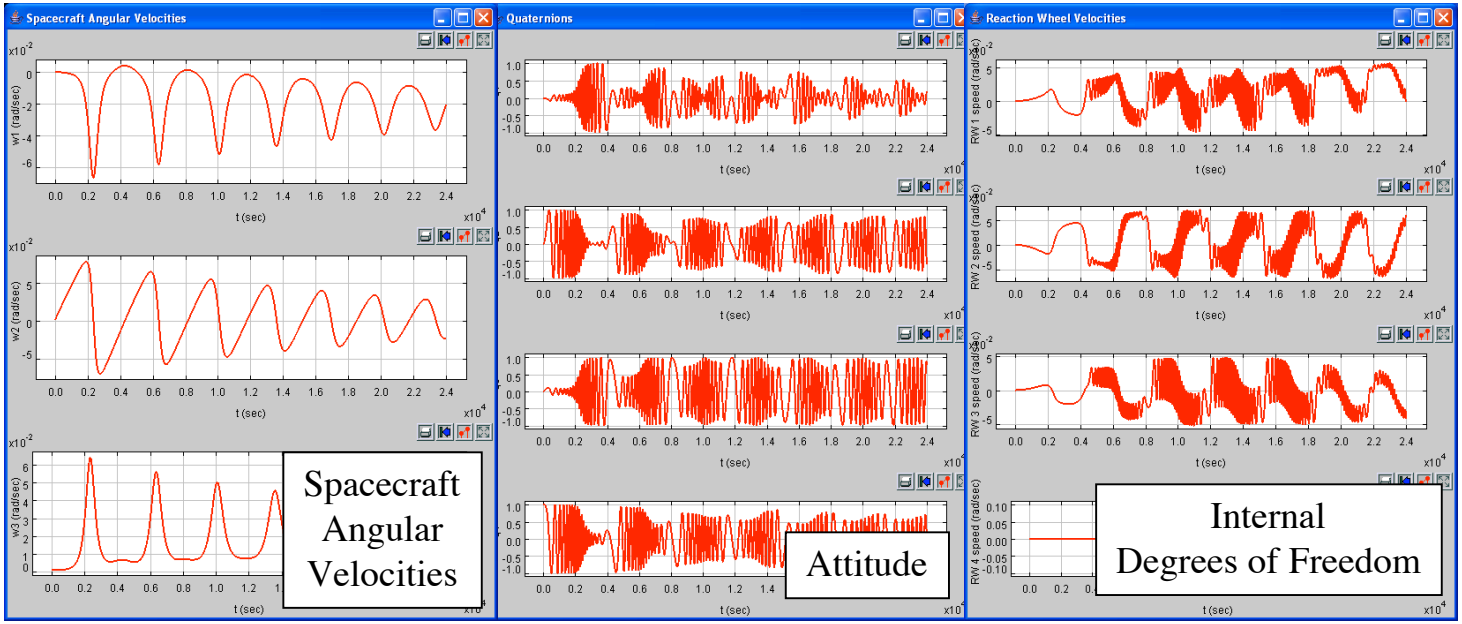
Phase 3.
Rendezvous & Capture

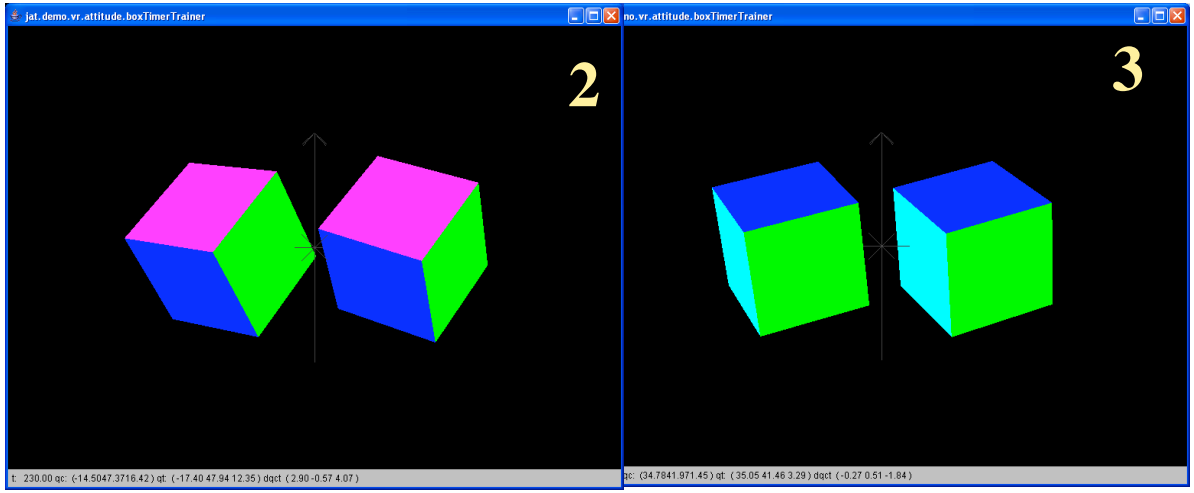
Low-level NBF Control Demo



Phase 1. Control State Test

HST Scenario - NBF Initial Test Results





Results of HST/ARC Testing

- High- and Low-level fixed for experiment
- Mid-level experiment -- ENI
  - Can transmit commands & data between levels
  - Can “short circuit” control pathways
  - Can perform some calculations/memory

Future HST/ARC Testing

- Extend high- and low-level capability
  - More interesting high-level control & goals
  - More low-level components
- Implementing and composing multiple behaviors
  - Approach
  - Avoid
  - Capture
- Training scenarios
- Evolutionary scenarios

Why Neural Basis Function Architecture?

- Scalable
- Autonomic behavior
- Intelligent behavior
- Autonomous behavior
- Specification
- Representation
- Situation
- Undifferentiated
- Inherently distributed

Conclusion

Neural Basis Functions

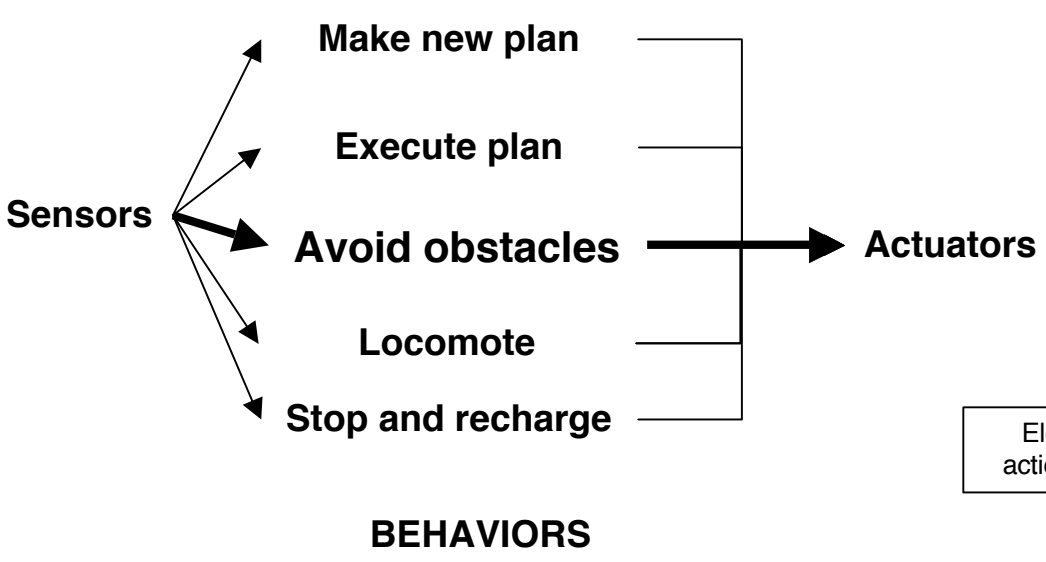
- A unique, new approach to full autonomy
- Synthesizes conventional approaches
- Solves specification problem
- Offers an approach to verification and validation
- Undifferentiated and self specializing
- Self similar, distributed, 3D
- No miracles, no new physics…
- Currently working on HST-motivated scenario
- Plan to start on a new rover technology

How is an NBF implemented?

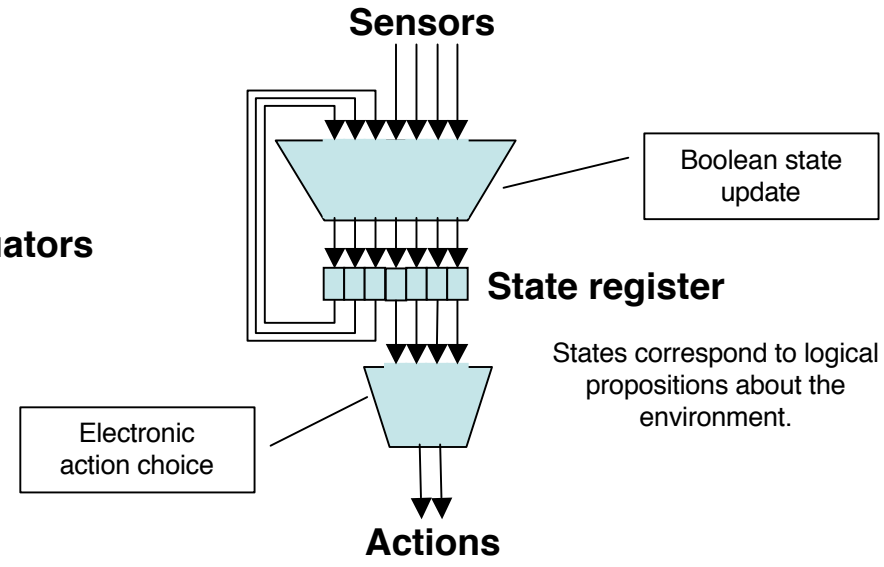
- Phases of NBF development
  - Design, framework, training stage, optimization stage
- Modules or nodes
  - Instantiate behaviors
  - Specialized functions
  - (1) Purely “software”
  - (2) Tightly coupled with real world
- Nodes communicate through a Neural Interface
- An “ontological” mapping or conciliation occurs during training
- Bad mappings or solutions are culled during optimization

What's special about NBFs?

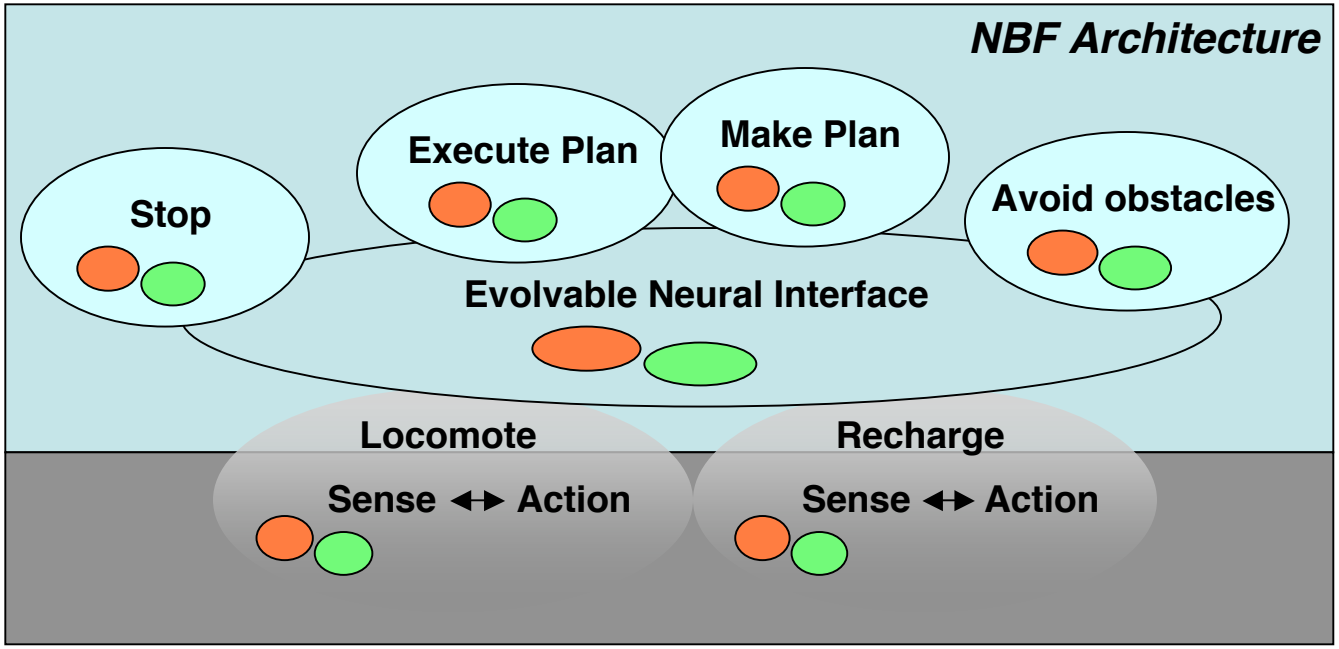
- Strong coupling
  - E.g. low-level control of resources
- State hiding
  - -- (recall object orientation)
  - "evolutionary approach to obj./agent development"
- State-change hiding



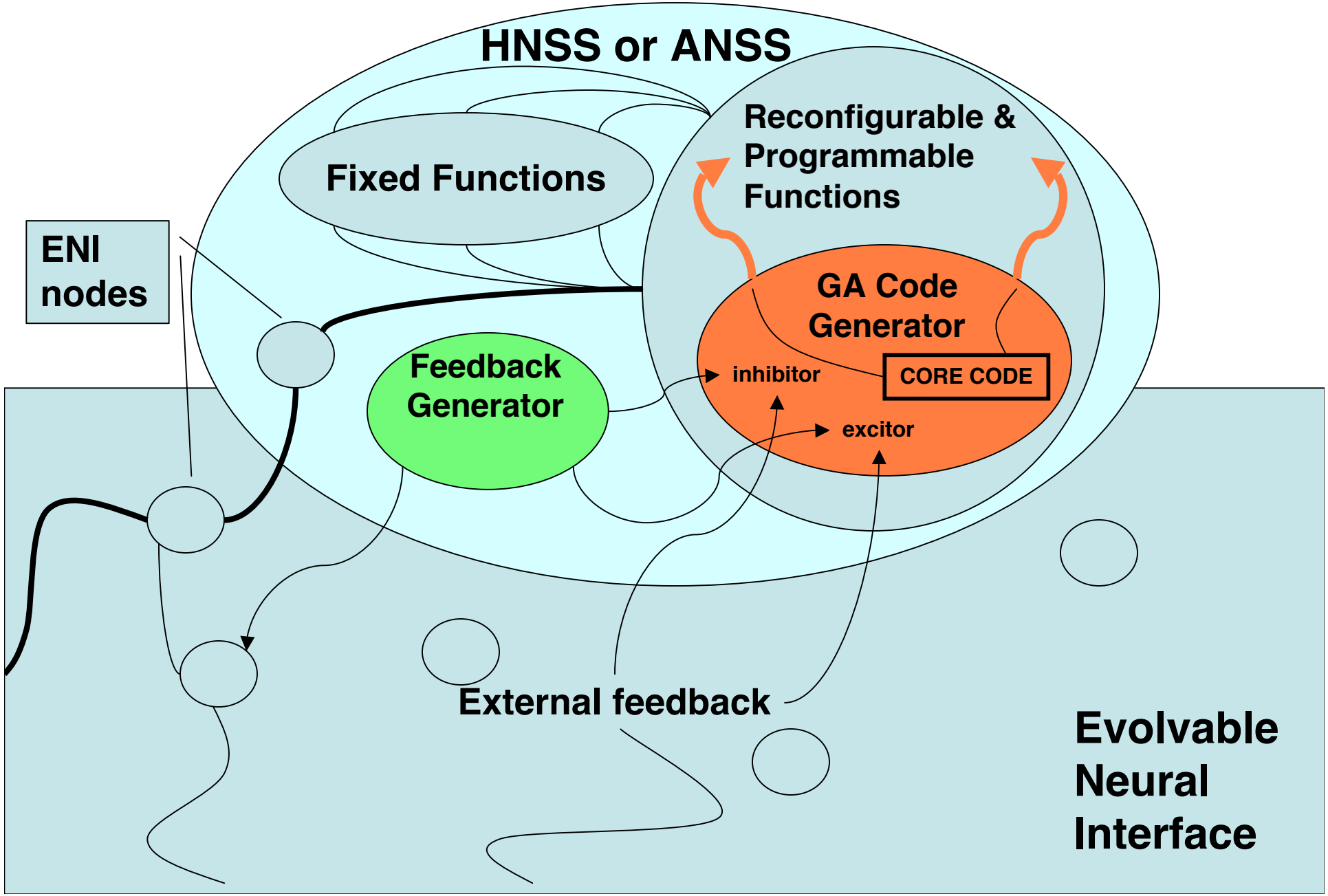
Brooks' Subsumption Architecture



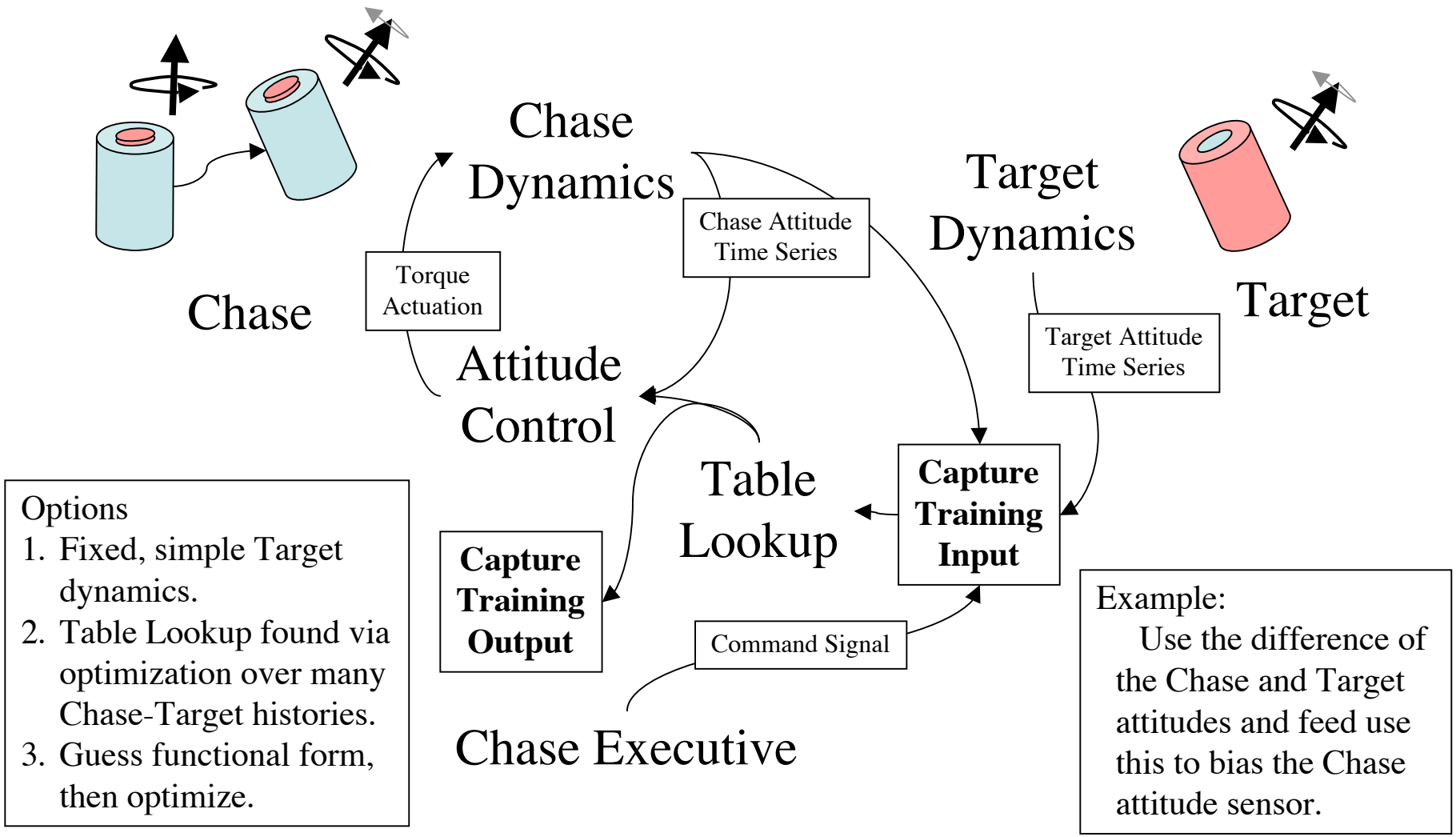
Rosenschein's Situated Automaton



Code Generator uses Genetic Algorithms & Programming to adapt core code to maximize positive feedback during training and initialization.



Low-level NBF Control Demo



Phase 1. Control State: Training Scenario