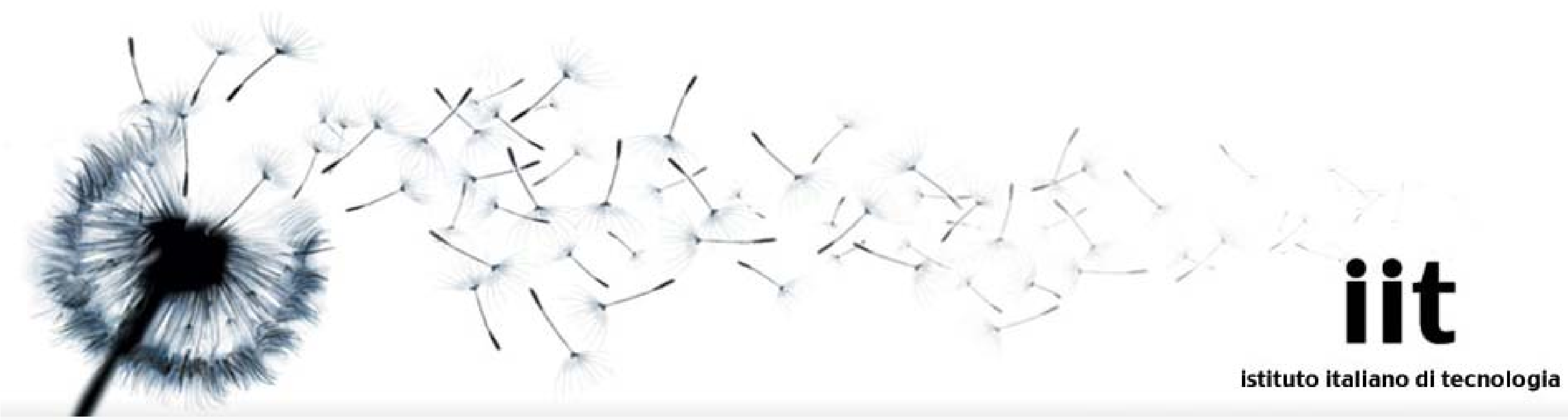
'Cognitive Robots' : From Affordance to Action
& back

Vishwanathan Mohan

Robotics, Brain and Cognitive Sciences Dept

Italian Institute of Technology,

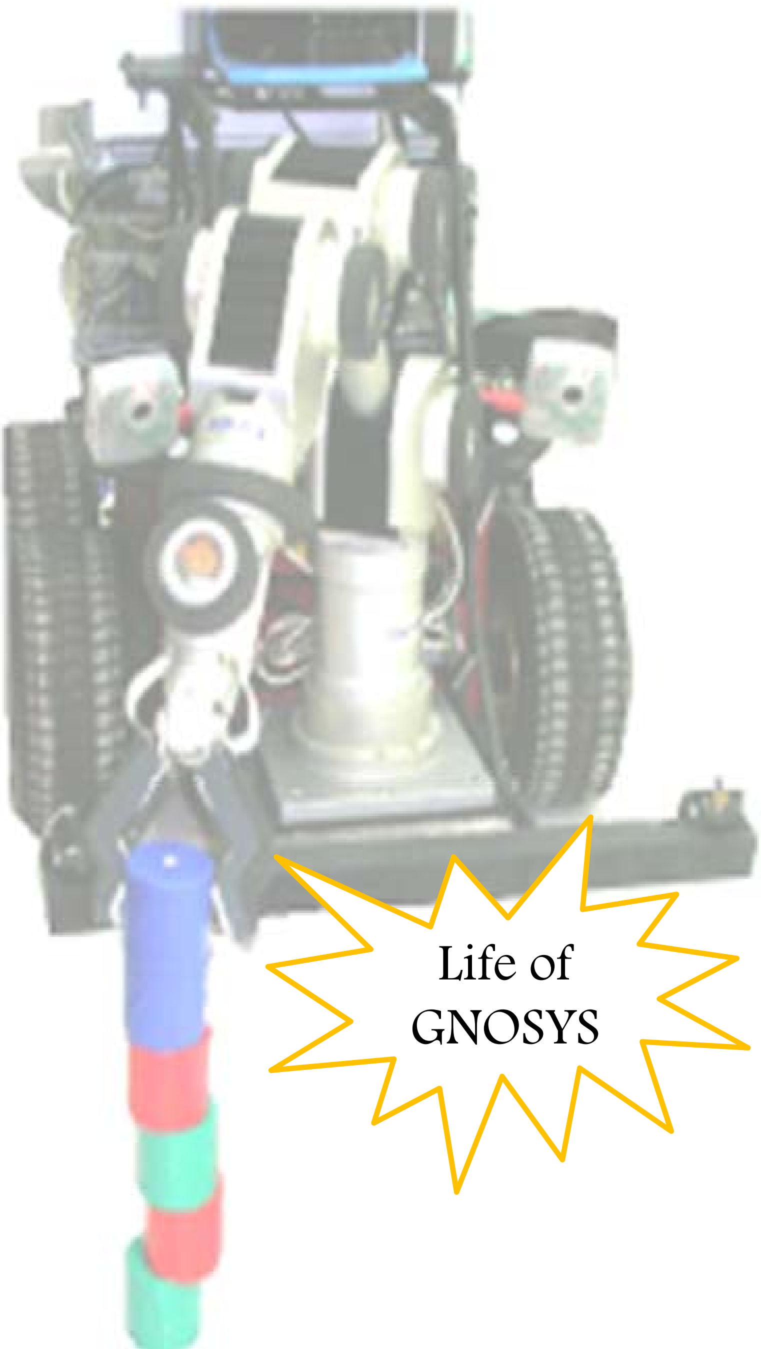
Genova, Italy



Today's Menu

- Objectives (**Overview**)
- Actions (**Local** View)
- 'Reasoning' about 'Actions' (**Global** View)
- Fabric of Reason's and Action's

  (**Objective**/ **Subjective** View)
- Atomic Cognitive Agents (**Future** View)



Today's Menu

Download: Neurolab Webpage/movies

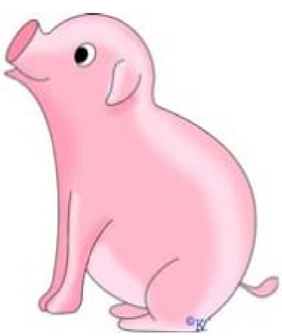
- Conclusions 'Reverse Engineered'
- The Arena of Action
- Actions (Internal Models)
  - Computing With the 'Body'
  - Computing in the 'World'
- Reasoning about Actions
  - Computing in the 'Mind'
- Atomic Cognitive Agents

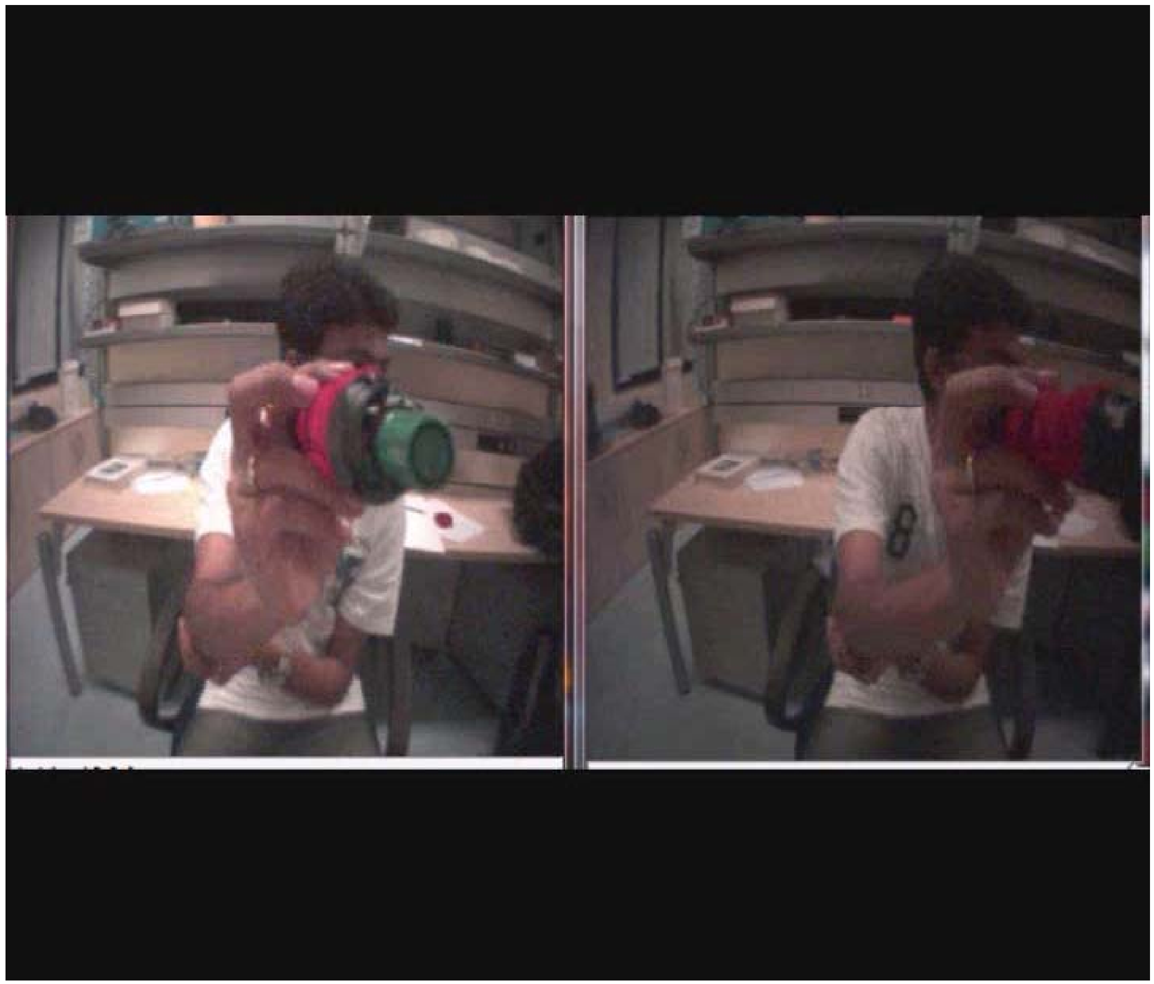
  Abstraction: Atoms everywhere

  Speaking Atoms: A WITS Enabled world

Tomorrow's Menu

Perception and Synthesis of 'Shape'





A B C D

1 + 1= 3

Reaching 'Goals': From Affordance to Action

Reaching 'Goals': From Affordance to Action

Well Connected

system of

body

and

environment

in order to

realize a

goal

Affordances are the seeds of Action
& back

Goals

Actions

Objects

Relationships

Choices

Experiences

What is **Possible** (Environment and Body)

What is **Useful** (in the **Context** of an Active Goal)

Exploit

Structure

Moving in the Mental Space for Acting in the Physical Space

Novel tool making in Caledonian crows

Using 'Thoughts' at the service of 'Action'

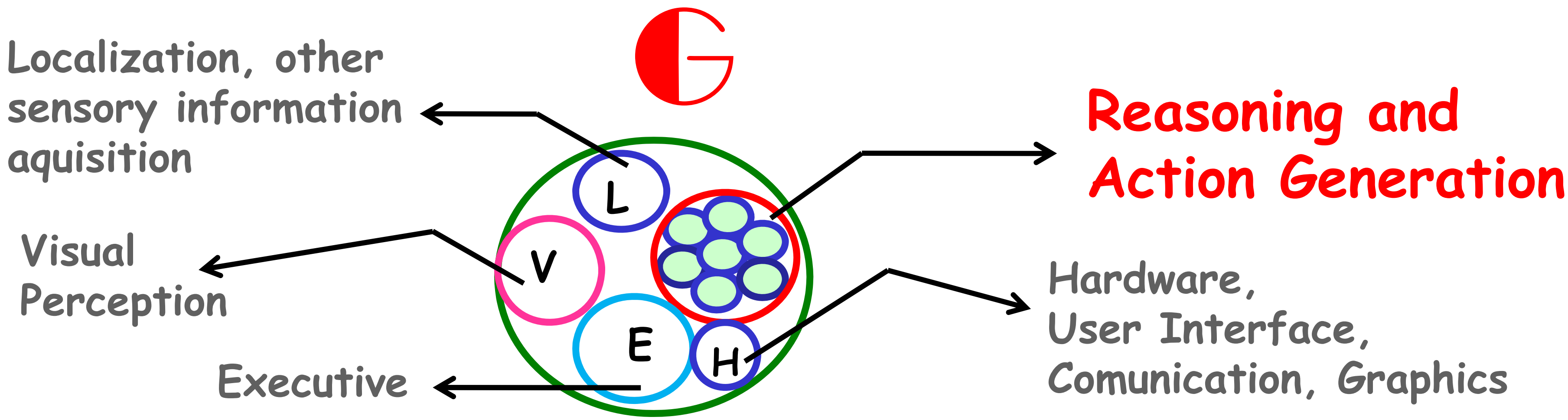
What 'Additonal affordances' can I create in the world ?

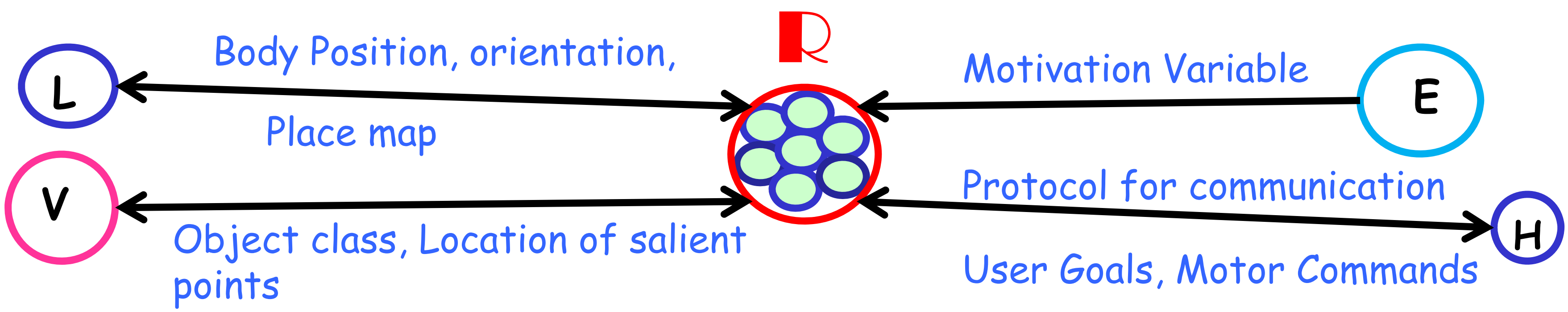
How will the world change as a result of my actions?

Will that be useful in the context of my internal goals?

Decouple behaviour from the direct control of the environment and react to situations that donot exist but could exist as a result of ones actions in the world

Developing a **Computational framework 'G'** which could drive an Artificial agent / **Robot** to exhibit a preliminary level of **cognitive control** over its **perception**, **action** and **imagination** ….





FOUNDATION FOR RESEARCH AND TECHNOLOGY HELLAS
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Playful Physical Interaction is Critical



Prediction

Judgment

Modeling/Causality

Experimentation / Diagnosis

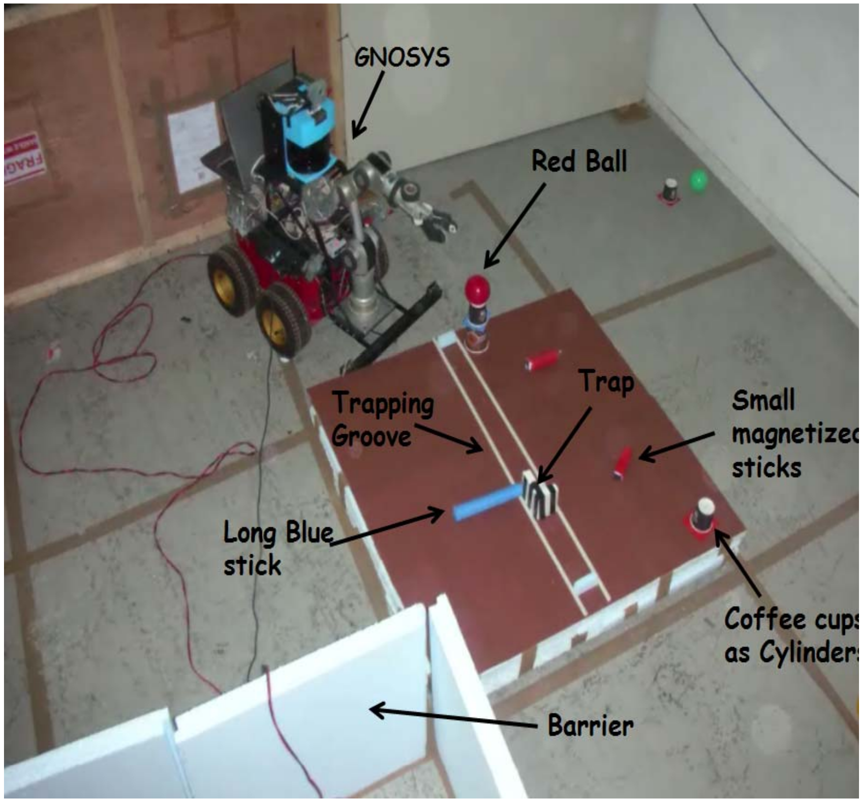
Describing, Negotiating, Team Work

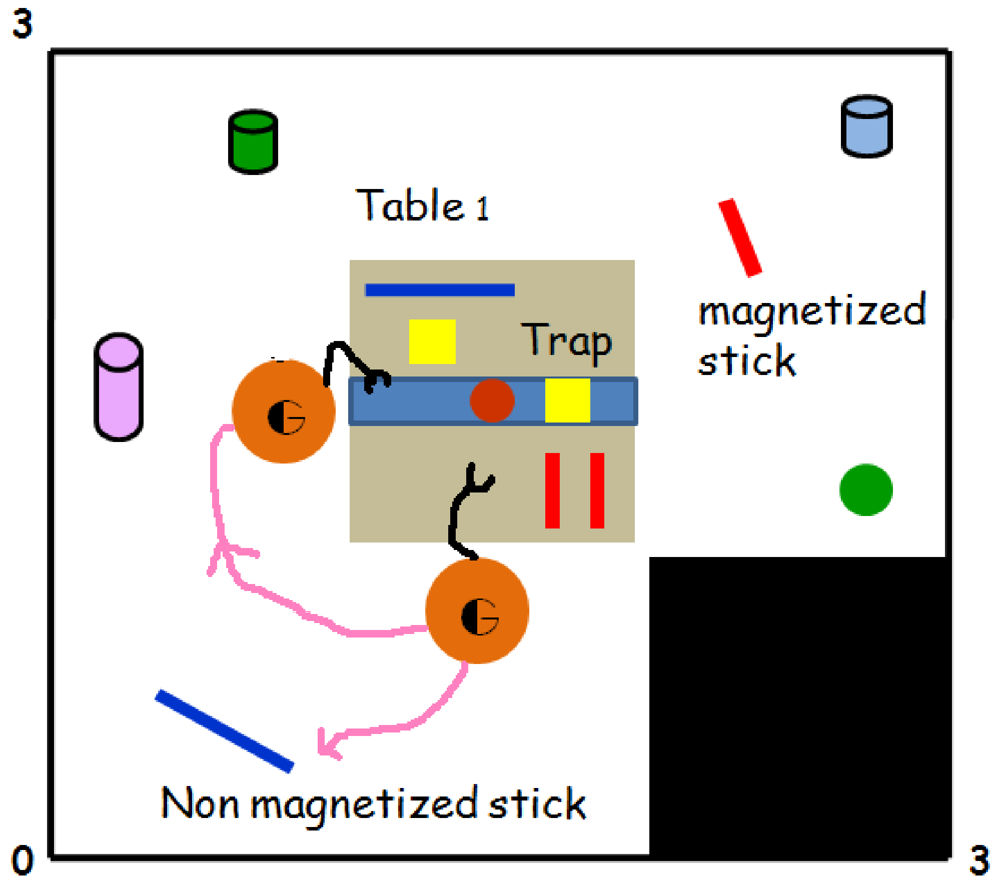
Artistry

Planning

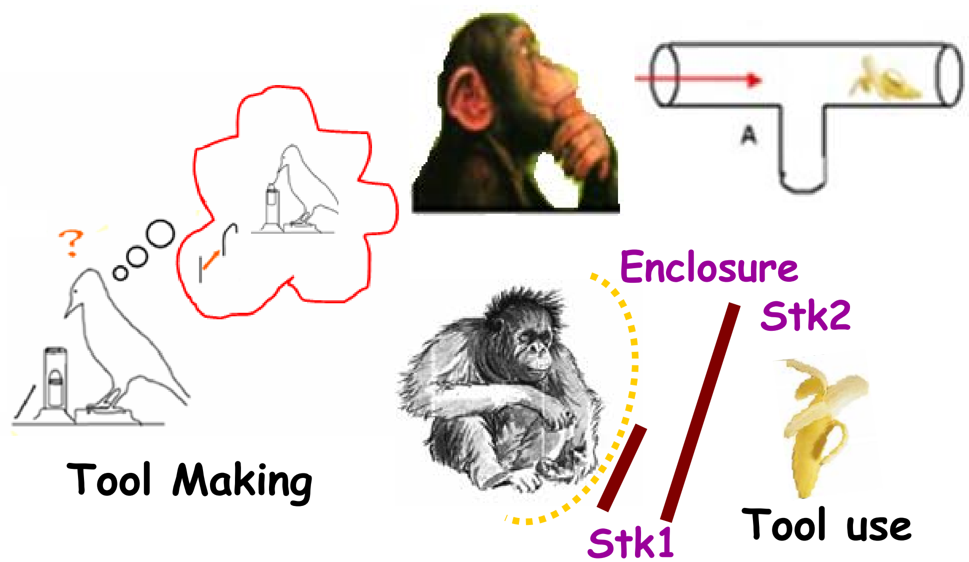
Playing is the most natural thing we do and there is much more to it than just having fun.

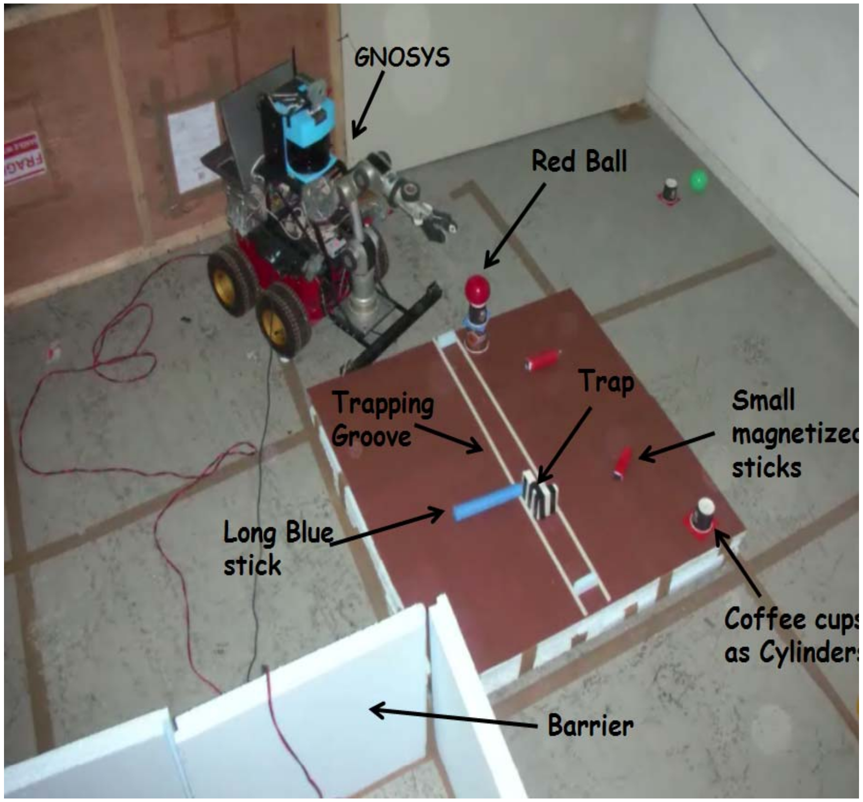
GNOSYS Playground: The Arena of Action

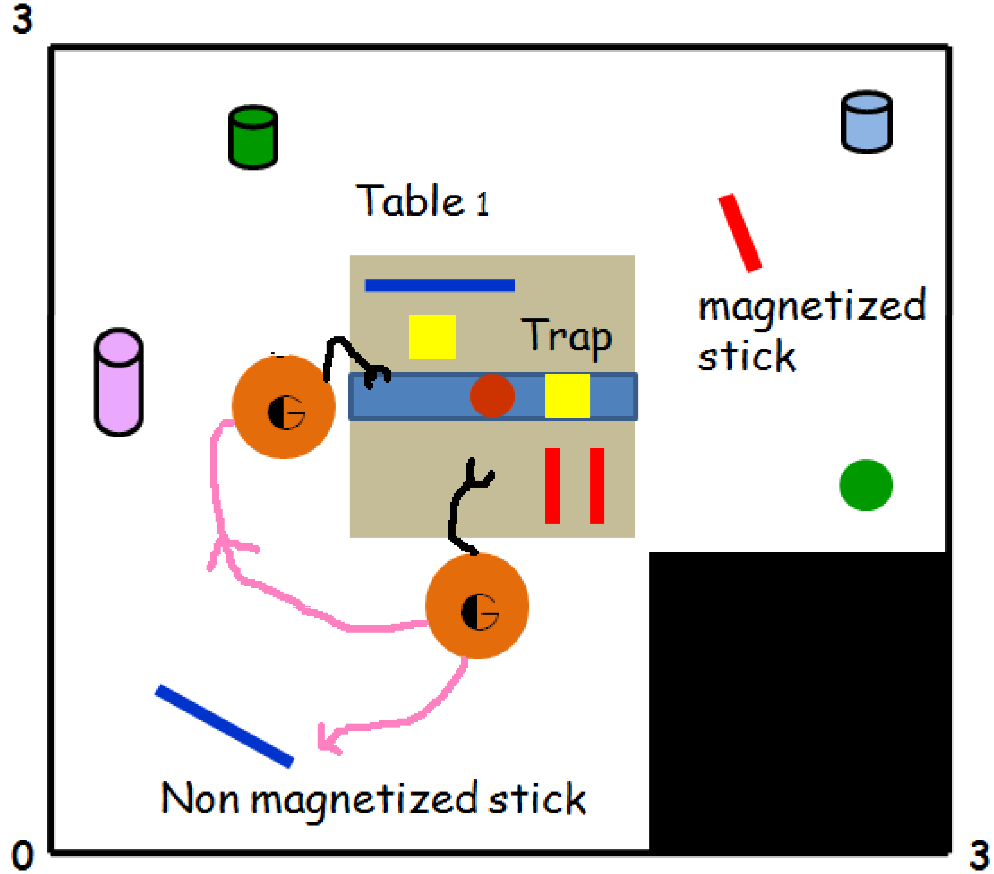




The playground designed for GNOSYS robot implicitly hosts experimental scenarios of tasks related to physical cognition known to be solved by different species of primates, corvids and children below 3 years.







Reach **(Reaching Goals)**

Grasp, Fetch

Push

Stack, Collect

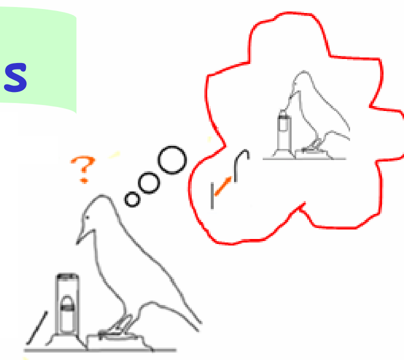
Use Sticks as Tools

Use Magnetized sticks as tools

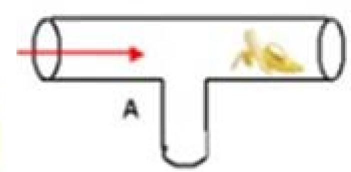
Imagine using objects as tools

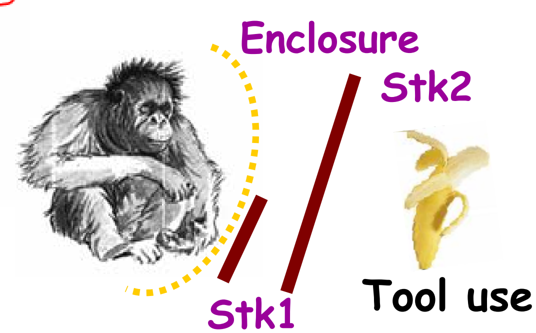
Imagine creating new tools

User Goals

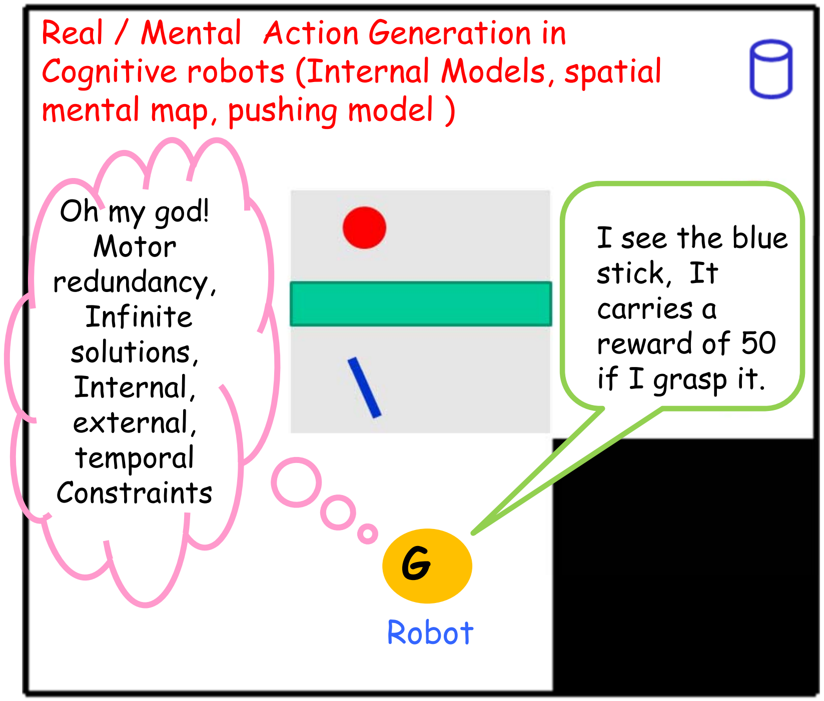


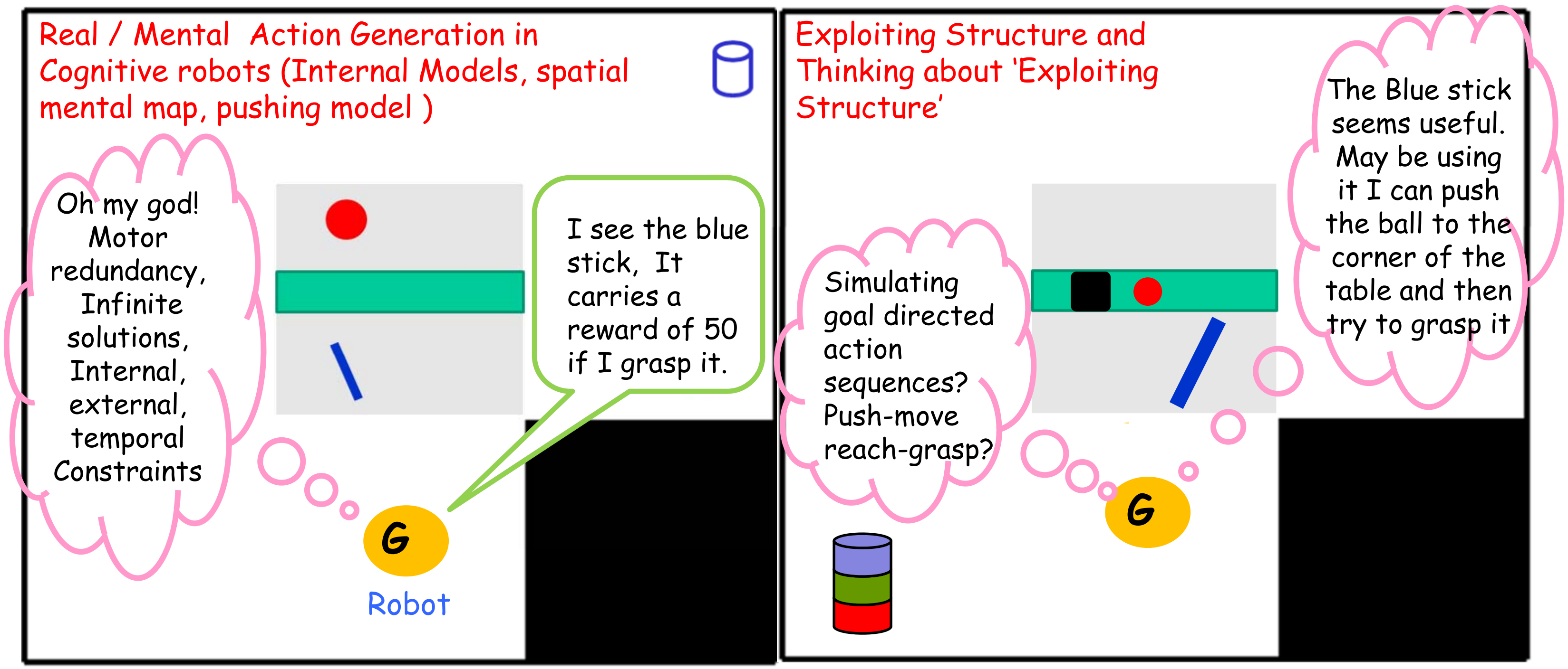
Tool Making



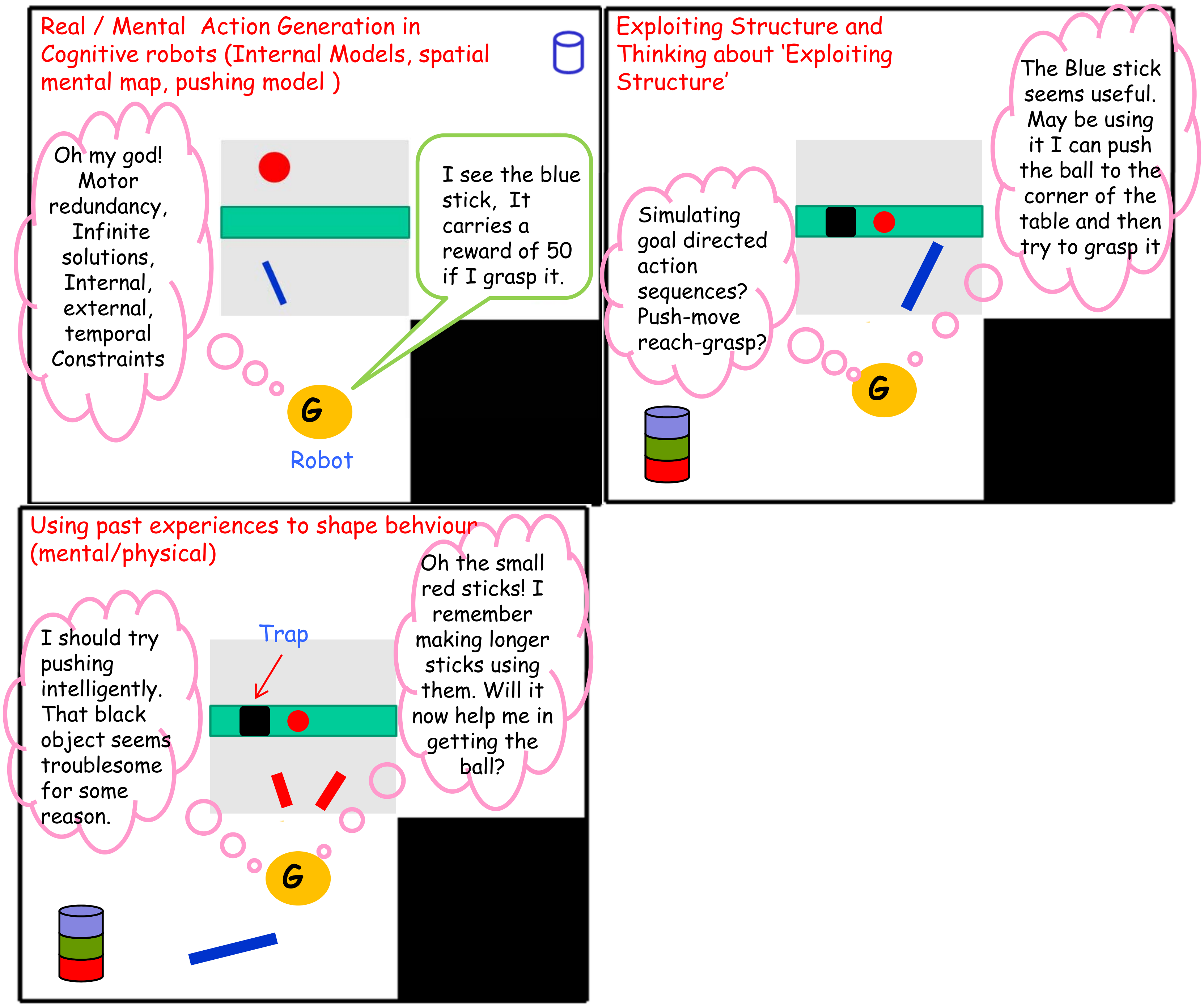


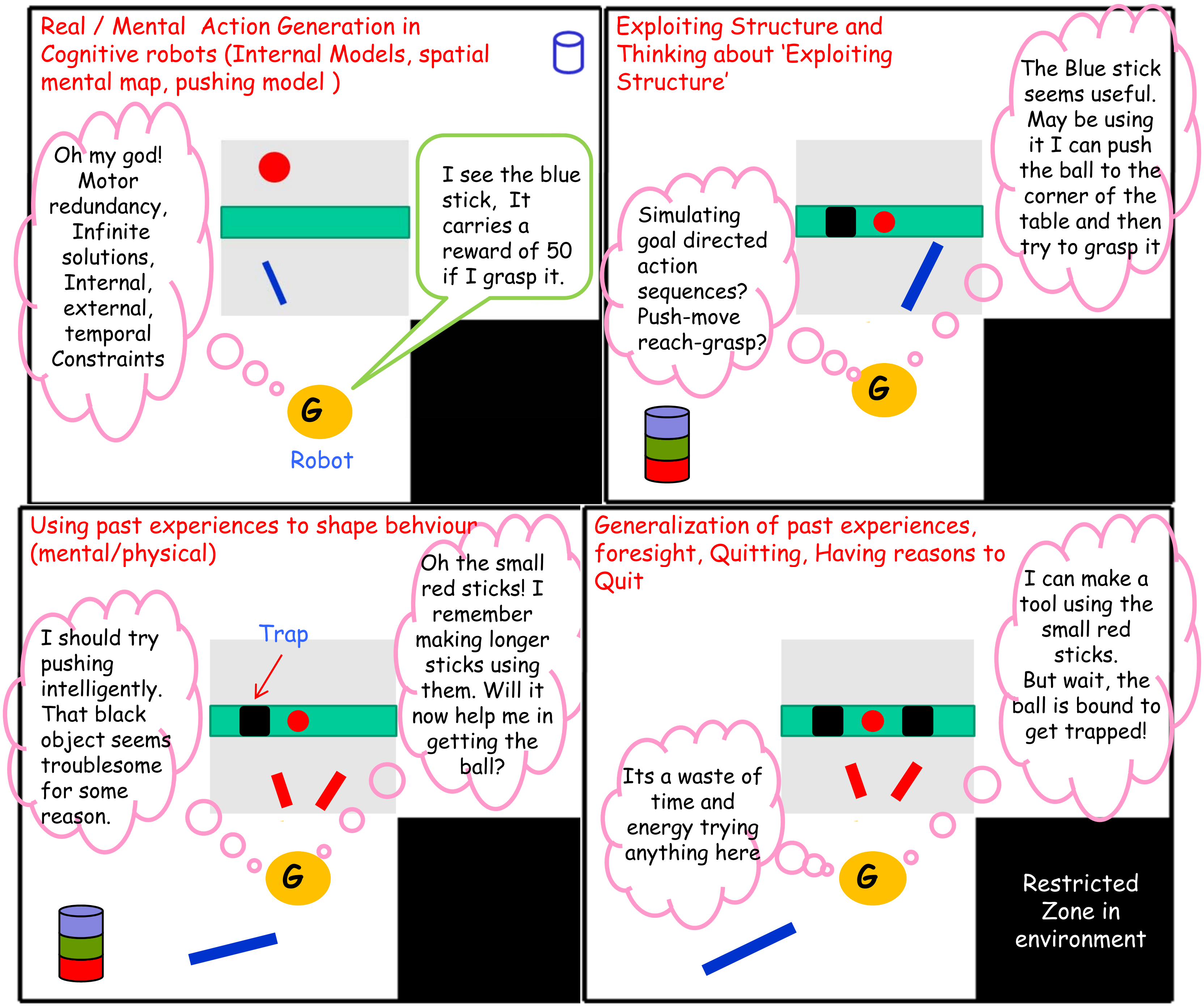
Tool use


Real / Mental Action Generation in Cognitive robots (Internal Models, spatial mental map, pushing model)
Oh my god! Motor redundancy, Infinite solutions, Internal, external, temporal Constraints
I see the blue stick, It carries a reward of 50 if I grasp it.
G
Robot



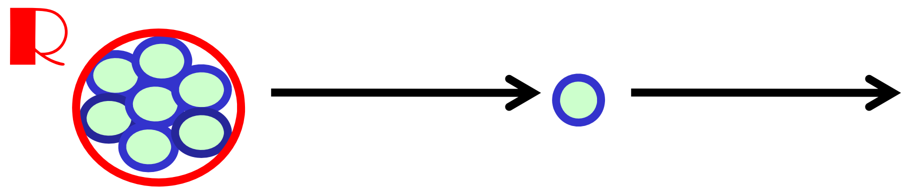
Real / Mental Action Generation in Cognitive robots (Internal Models, spatial mental map, pushing model)
Oh my god! Motor redundancy, Infinite solutions, Internal, external, temporal Constraints
I see the blue stick, It carries a reward of 50 if I grasp it.
G
Robot
Exploiting Structure and Thinking about 'Exploiting Structure'
Simulating goal directed action sequences? Push-move reach-grasp?
The Blue stick seems useful. May be using it I can push the ball to the corner of the table and then try to grasp it
G


Real / Mental Action Generation in Cognitive robots (Internal Models, spatial mental map, pushing model)
Oh my god! Motor redundancy, Infinite solutions, Internal, external, temporal Constraints
I see the blue stick, It carries a reward of 50 if I grasp it.
G
Robot
Exploiting Structure and Thinking about 'Exploiting Structure'
Simulating goal directed action sequences? Push-move reach-grasp?
The Blue stick seems useful. May be using it I can push the ball to the corner of the table and then try to grasp it
G
Using past experiences to shape behviour (mental/physical)
I should try pushing intelligently. That black object seems troublesome for some reason.
Trap
Oh the small red sticks! I remember making longer sticks using them. Will it now help me in getting the ball?
G



Real / Mental Action Generation in Cognitive robots (Internal Models, spatial mental map, pushing model)
Oh my god! Motor redundancy, Infinite solutions, Internal, external, temporal Constraints
I see the blue stick, It carries a reward of 50 if I grasp it.
G
Robot
Exploiting Structure and Thinking about 'Exploiting Structure'
Simulating goal directed action sequences? Push-move reach-grasp?
The Blue stick seems useful. May be using it I can push the ball to the corner of the table and then try to grasp it
G
Using past experiences to shape behviour (mental/physical)
I should try pushing intelligently. That black object seems troublesome for some reason.
Trap
Oh the small red sticks! I remember making longer sticks using them. Will it now help me in getting the ball?
G
Generalization of past experiences, foresight, Quitting, Having reasons to Quit
Its a waste of time and energy trying anything here
I can make a tool using the small red sticks. But wait, the ball is bound to get trapped!
G
Restricted Zone in environment

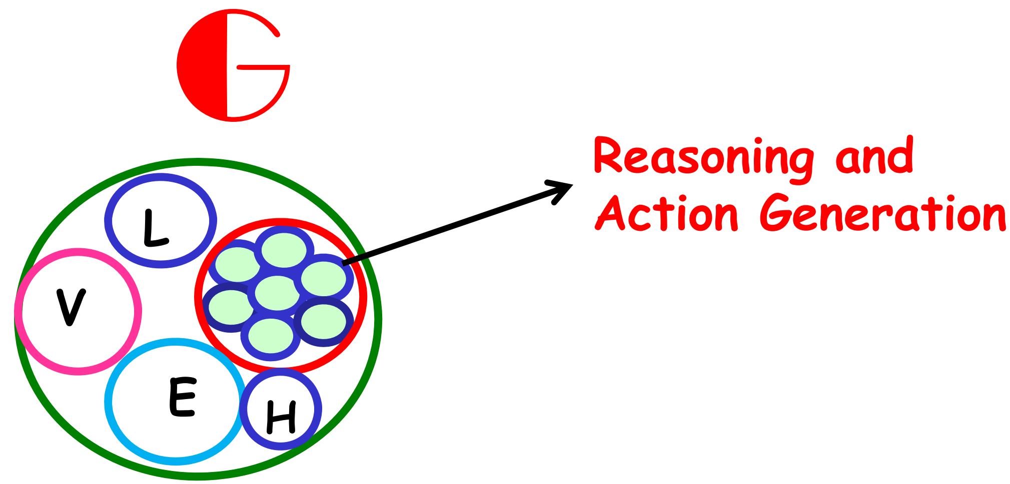
Developing a **Computational framework** which could drive an Artificial agent / **Robot** to exhibit similar levels of **cognitive control** over its **perception**, **action** and **imagination** ?

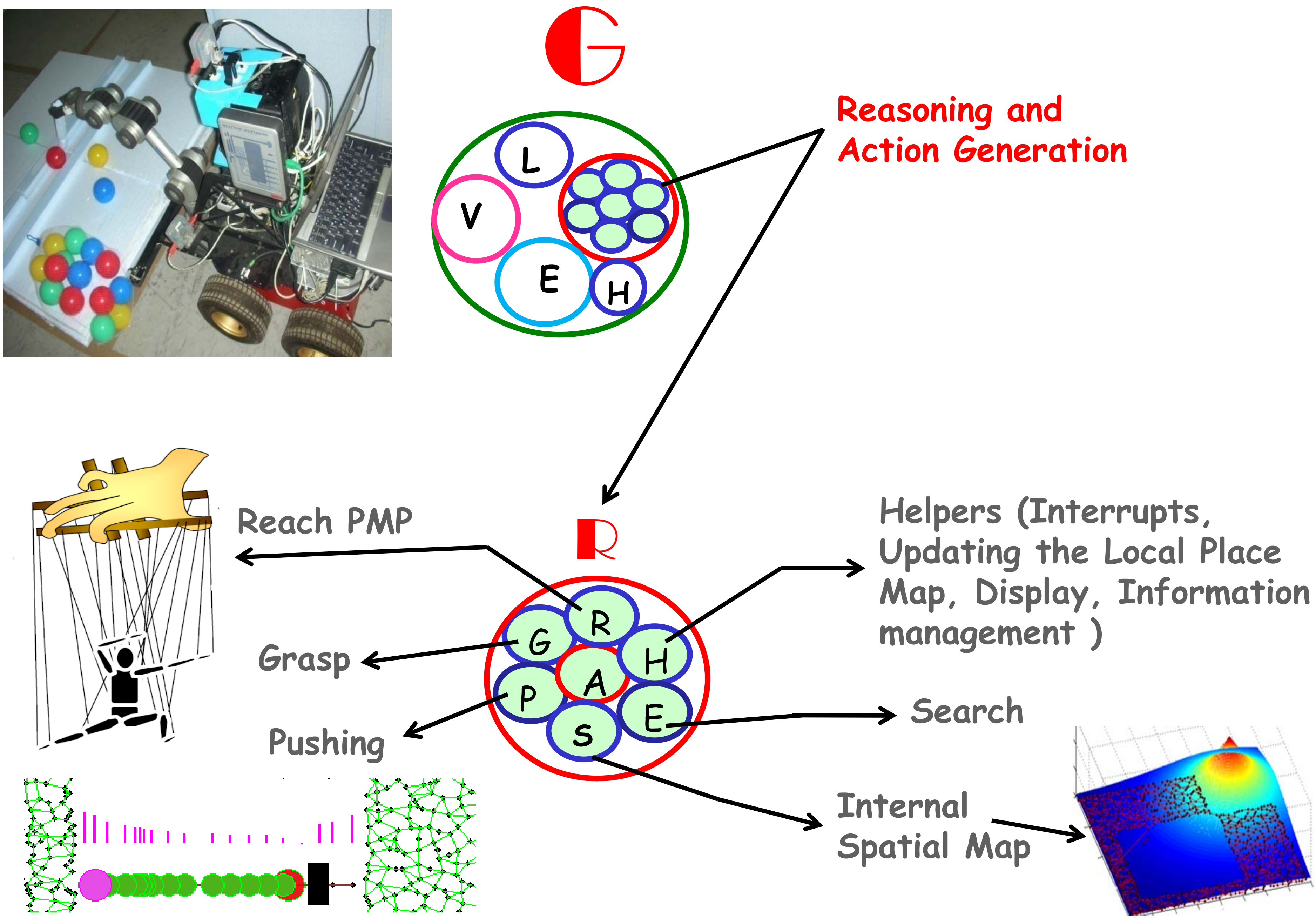


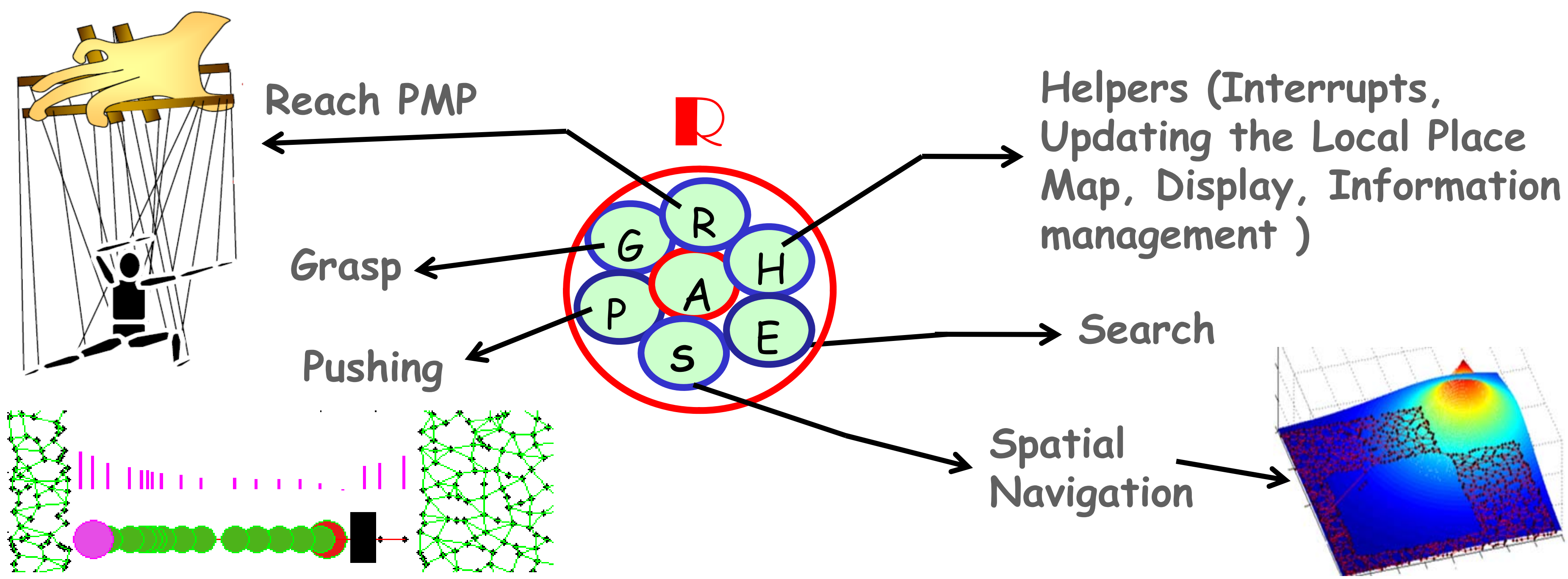
Sub-objectives

- **Internal Models:** Forward/Inverse functions of sensorimotor dependencies
- **Representation:** Goal directed planning, Virtual experiments, Value Fields
- **Learning** : State representations (sensory/motor), Dynamic Changes
- **Redundancy** : Heterogenous optimality cireteria in a task specific way
- **Temporal synchrony:** maintenance of continuity in perception, action and time
- **Integration:** Top Down, Bottom Up , Goal
- **Coherence :** To switch between explorative and normal dynamics to maintain psycho-logical consistency in the sensorimotor world

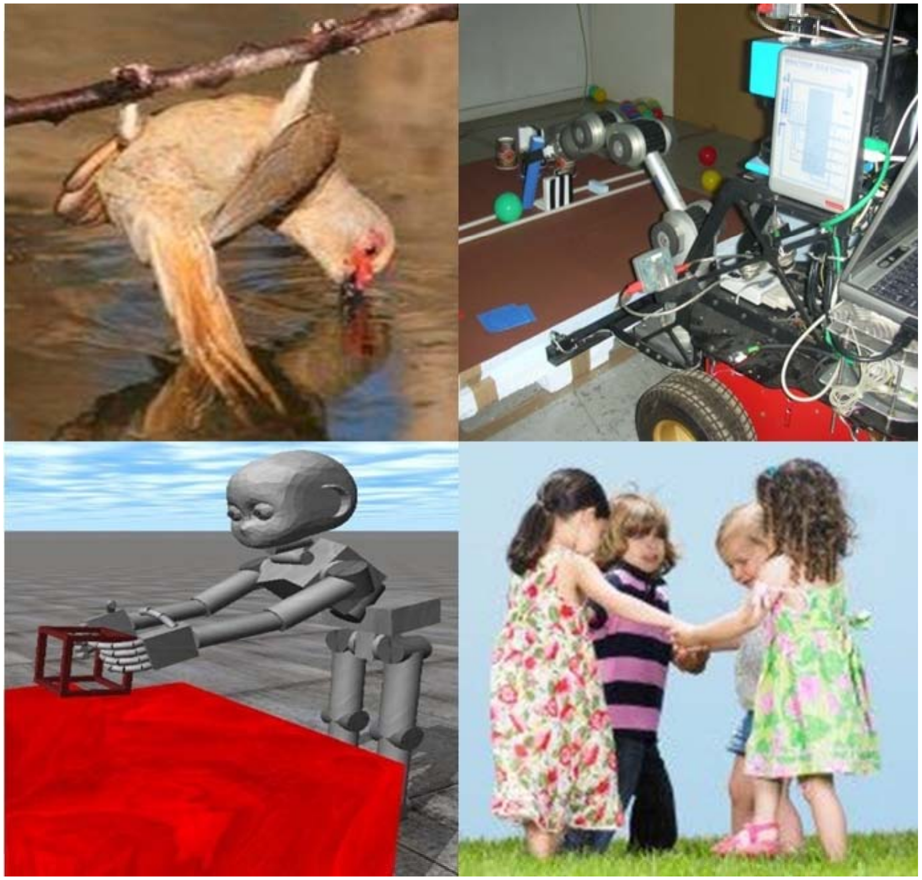
Demonstrate: the effectiveness of the architecture in a **physical instantiation**, acting in **ecologically realistic environments**, and **goals**

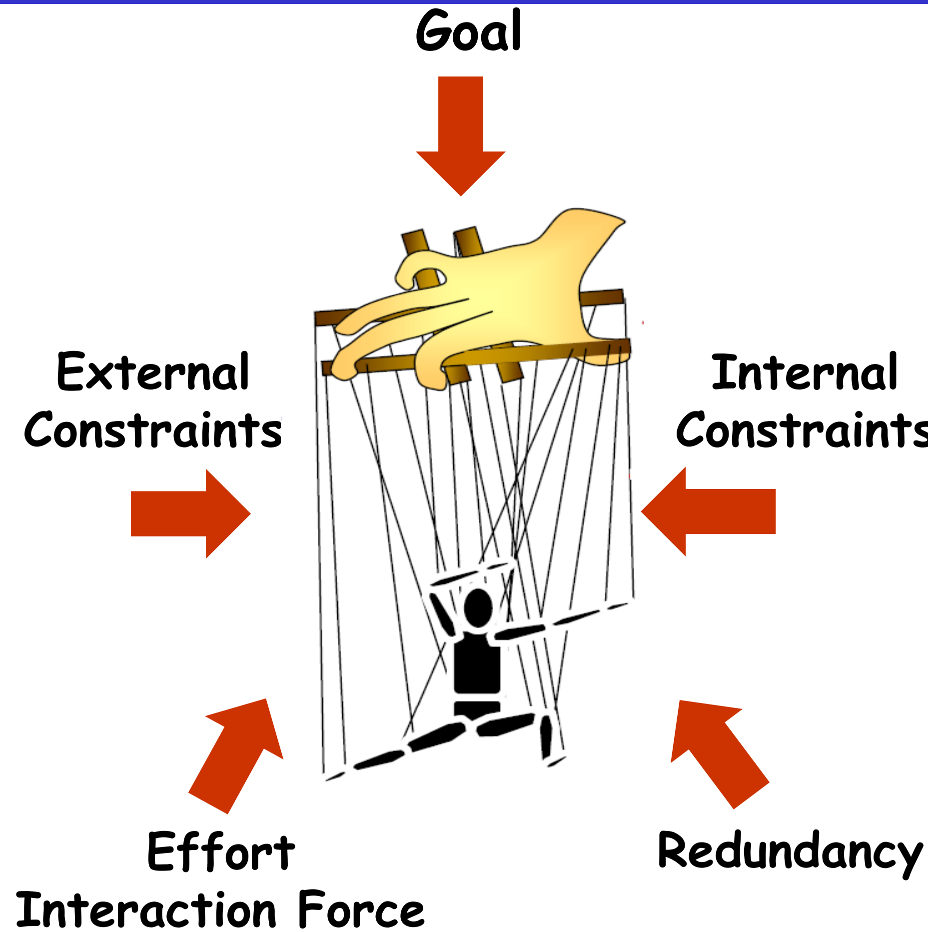

L
V
E
H
Reasoning and
Action Generation


G
L
V
E
H
Reasoning and
Action Generation
R
Reach PMP
R
G
H
A
P
E
S
Grasp
Pushing
Helpers (Interrupts,
Updating the Local Place
Map, Display, Information
management)
Search
Internal
Spatial Map


Reach PMP
R
Grasp
Pushing
G
R
H
A
P
E
S
Helpers (Interrupts, Updating the Local Place Map, Display, Information management)
Search
Spatial Navigation

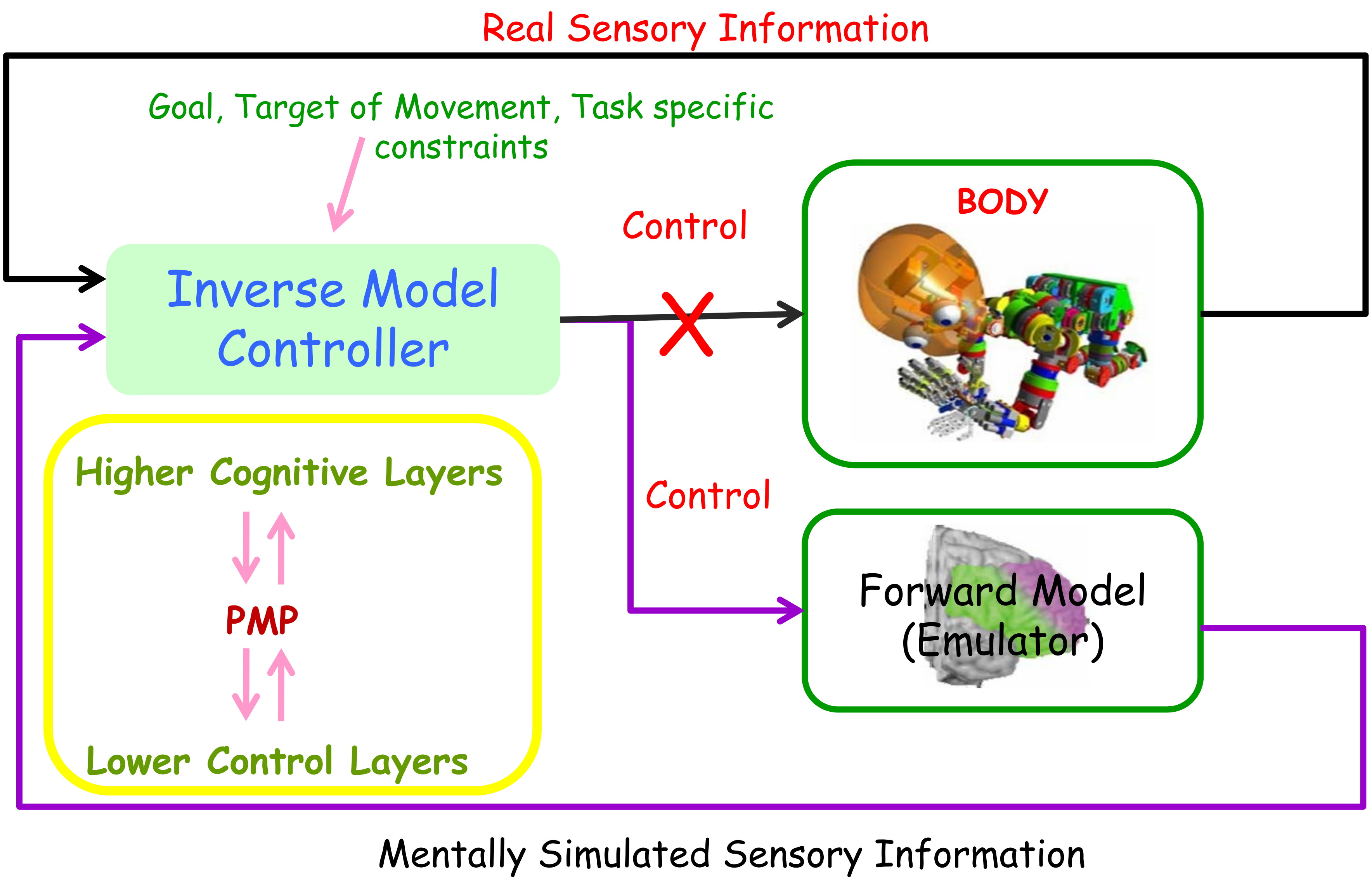
Passive Motion Paradigm (PMP): Real/Mental Action Generation

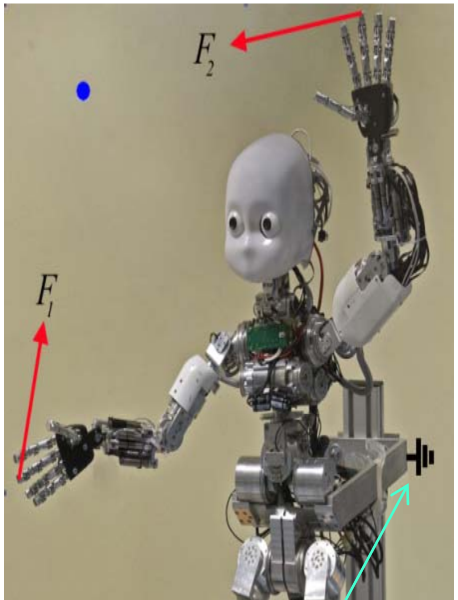




- 'Multireferential' Non Linear Attractor dynamics
- Compositionality
- Local to Global computing
- Forward inverse models

Passive Motion Paradigm: Forward Inverse Model pair

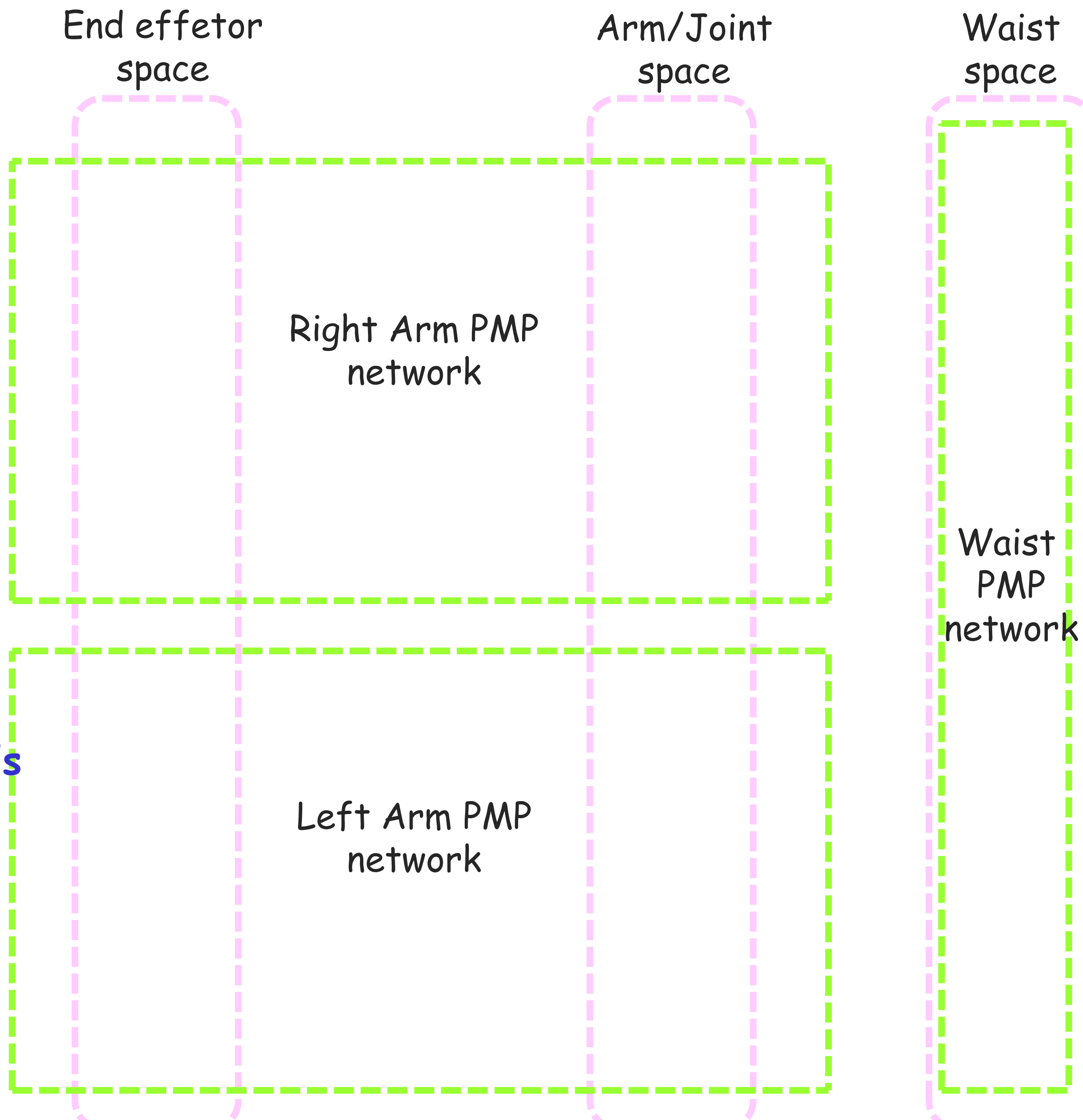




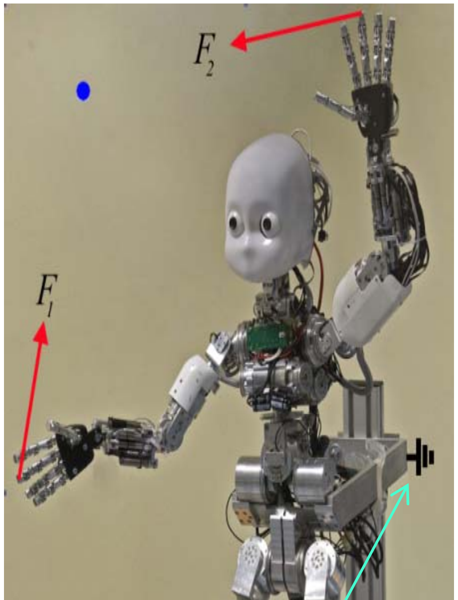
Waist is Grounded

Grounding

Different Motor space's



Will focus on Compositionality in PMP (iCub Centric)



Waist is Grounded

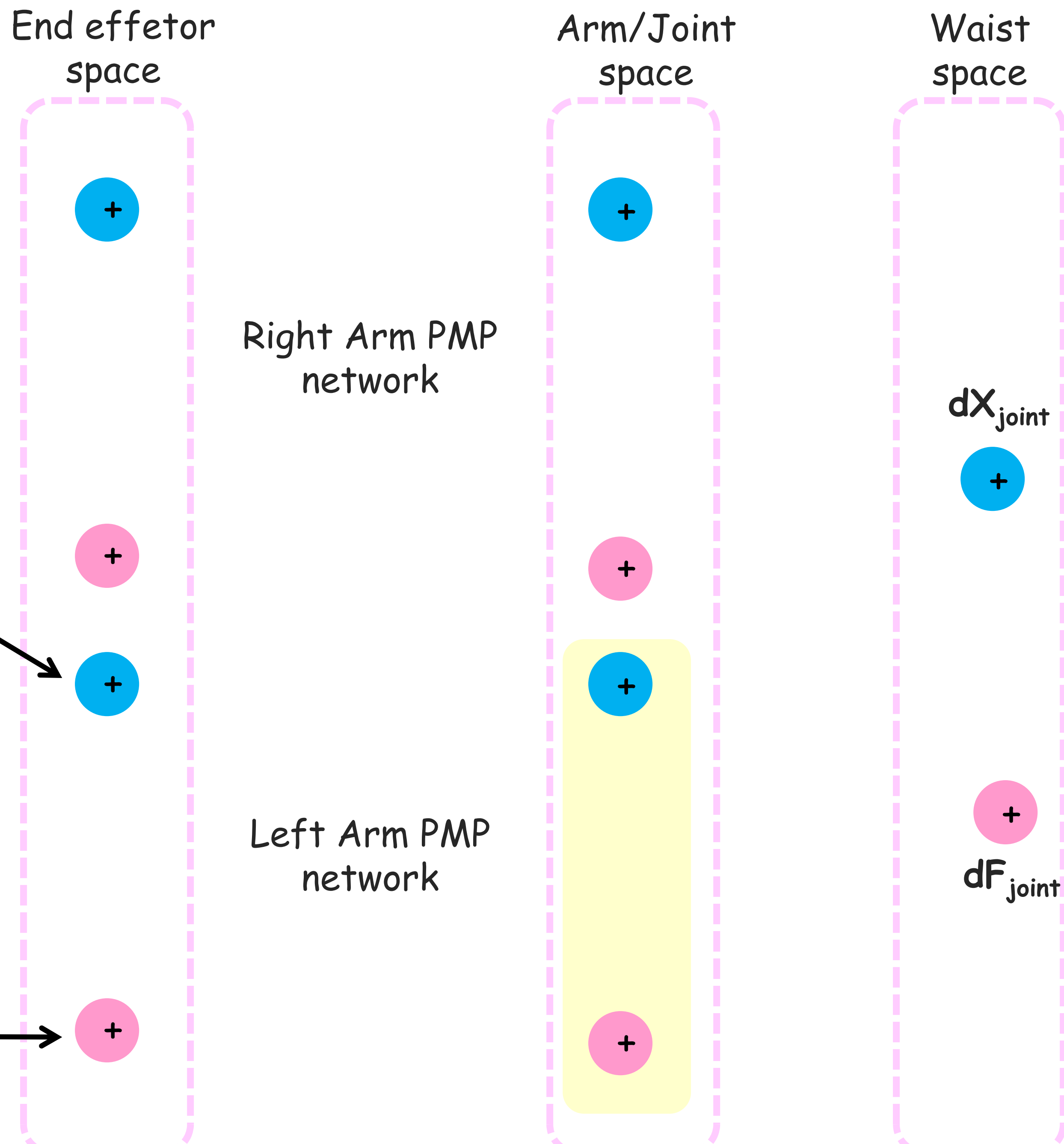
Grounding

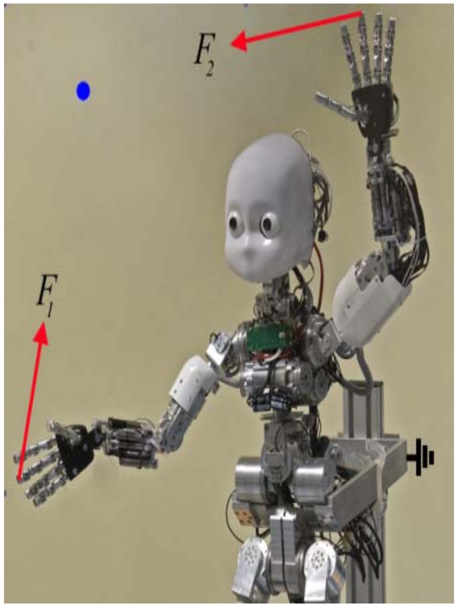
Different Motor space's

Work Units

Generalized Displacement node

Generalized Force node



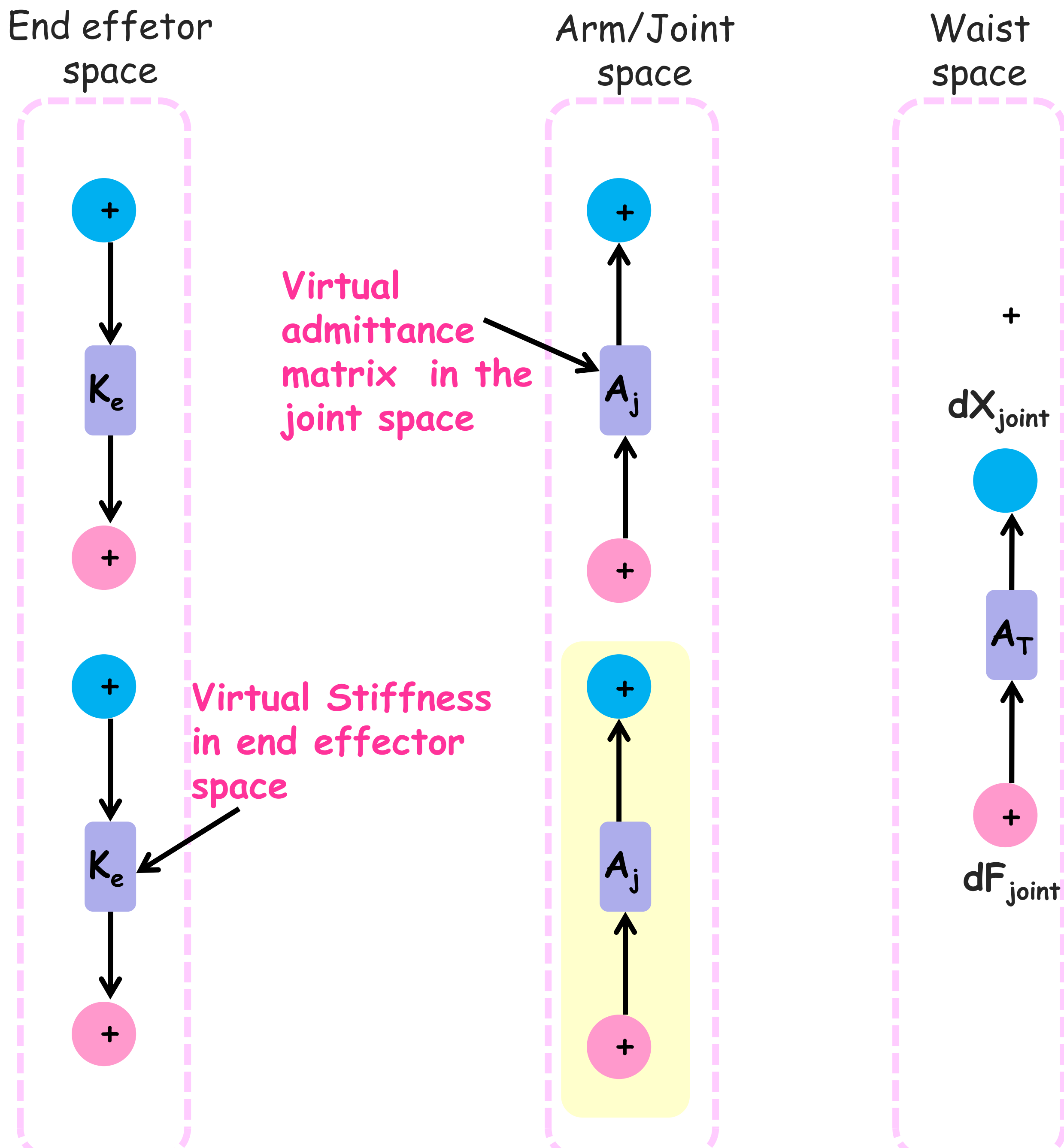


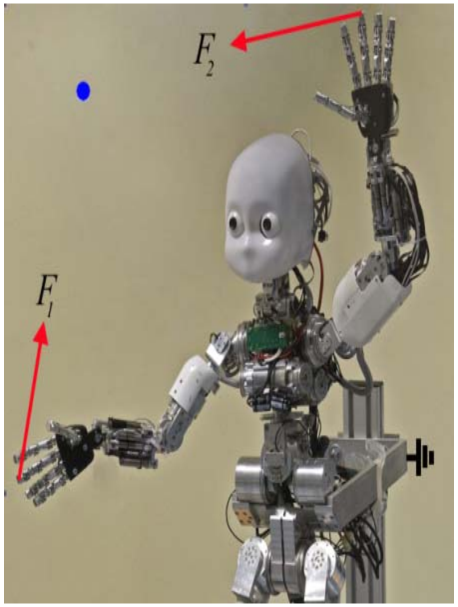
Grounding

Different Motor space's

Work Units

Elastic Transformation





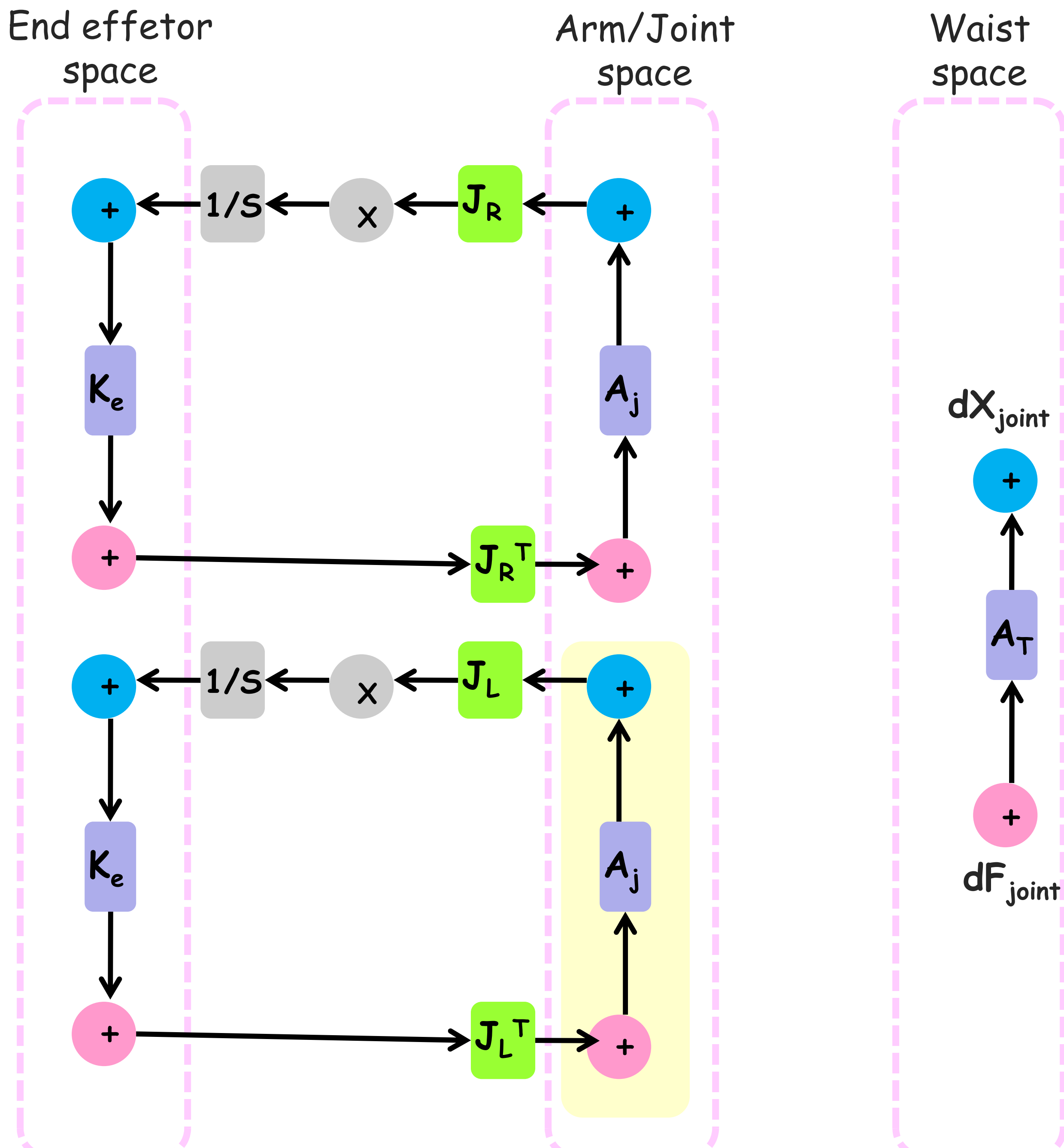
Grounding

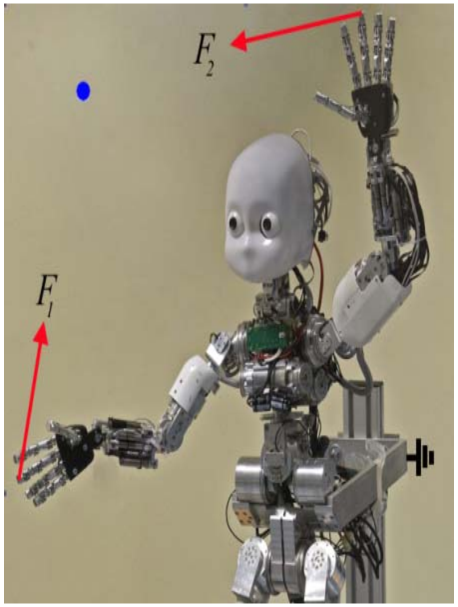
Different Motor space's

Work Units

Elastic Transformation

Geometric Transformation





Grounding

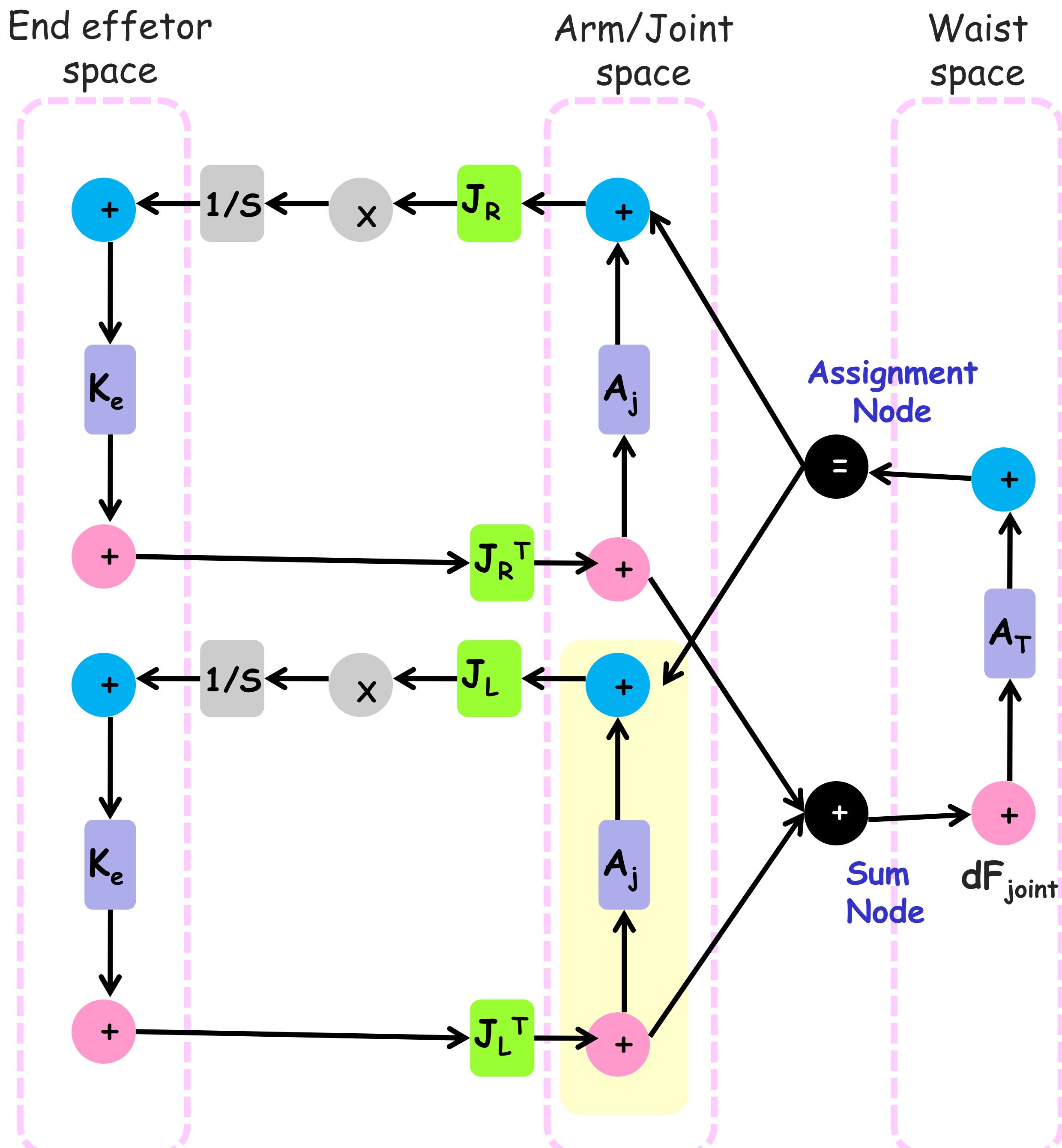
Different Motor space's

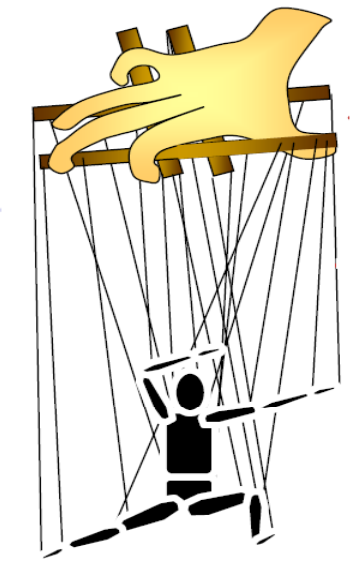
Work Units

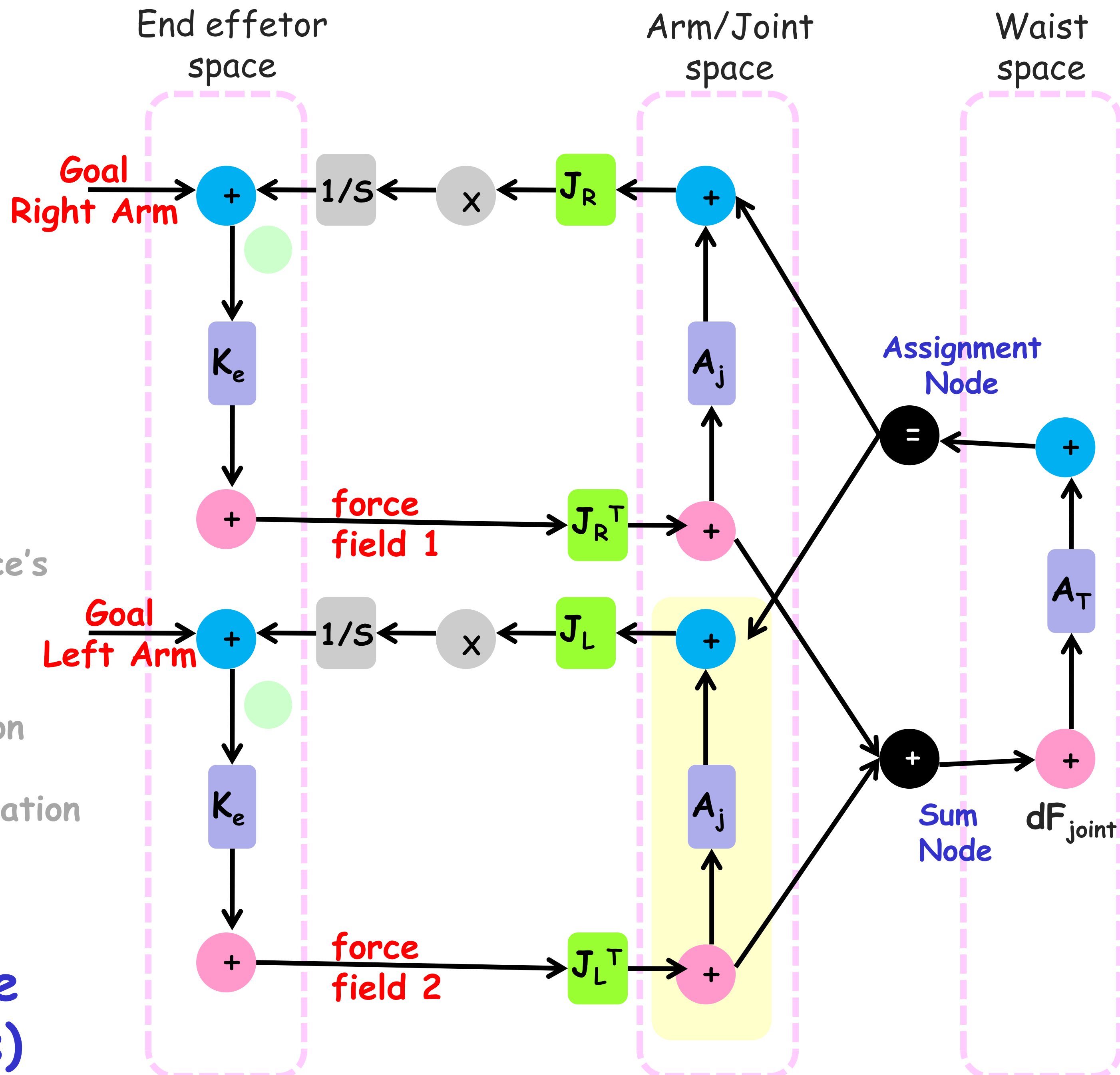
Elastic Transformation

Geometric Transformation

Branching Nodes







Grounding

Different Motor space's

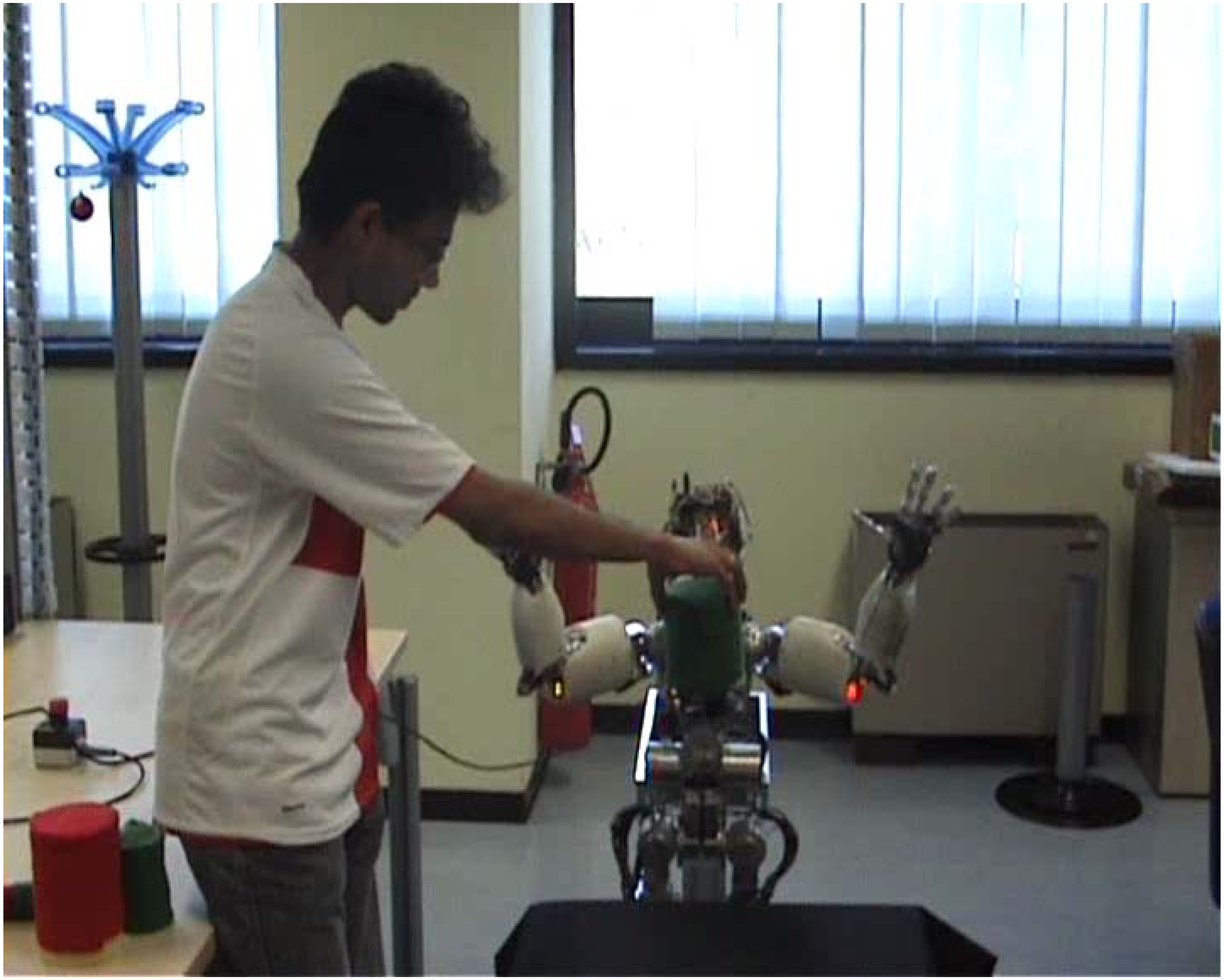
Work Units

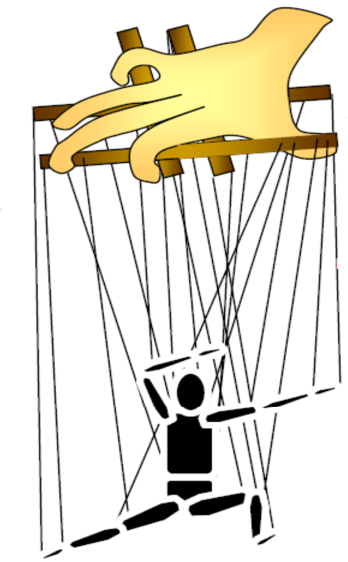
Elastic Transformation

Geometric Transformation

Branching Nodes

A set of Force fields (Strings)





Grounding

Different Motor space's

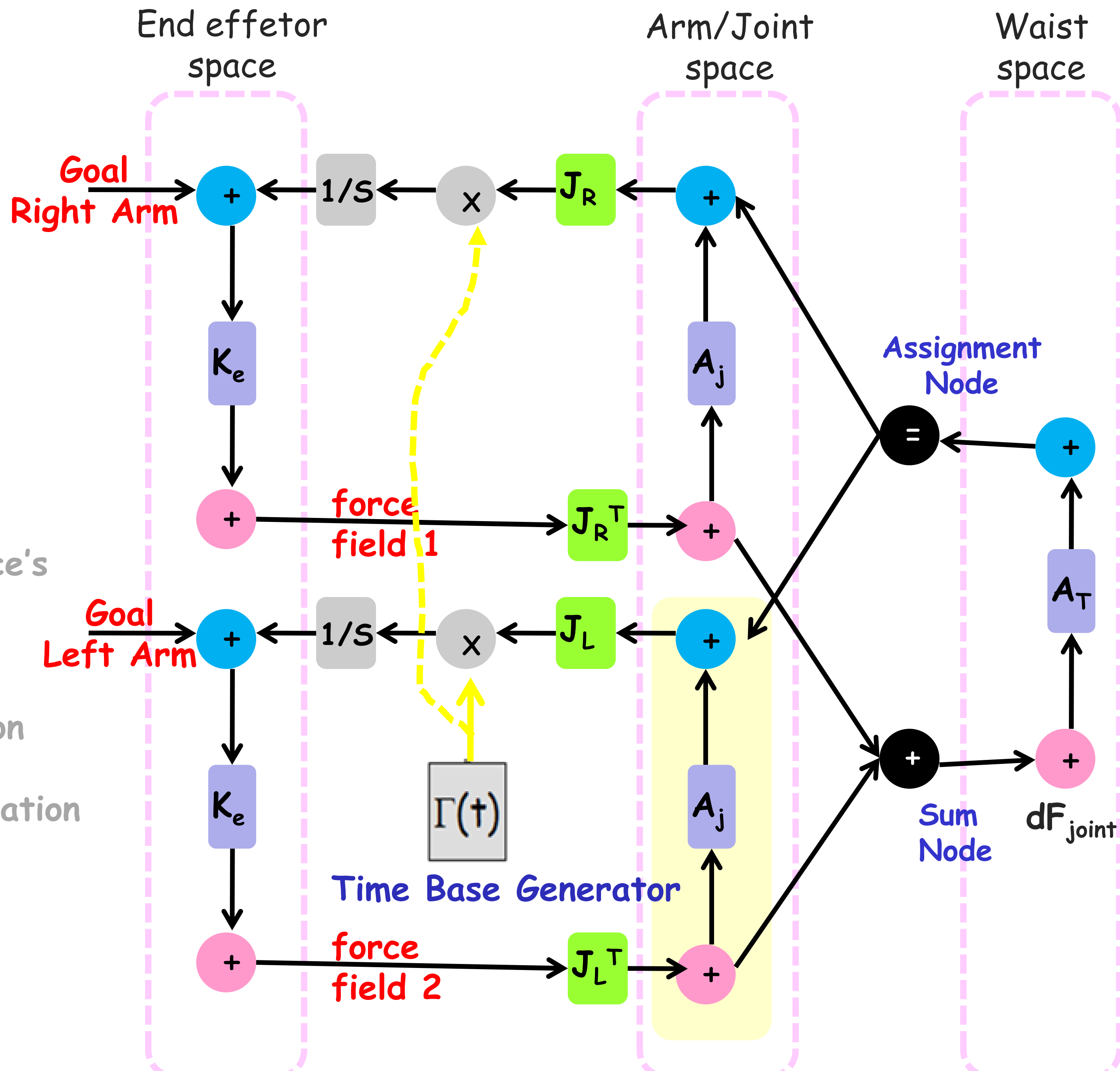
Work Units

Elastic Transformation

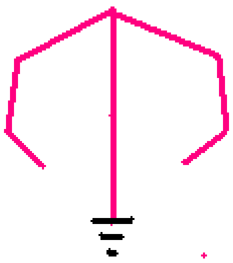
Geometric Transformation

Branching Nodes

Timing

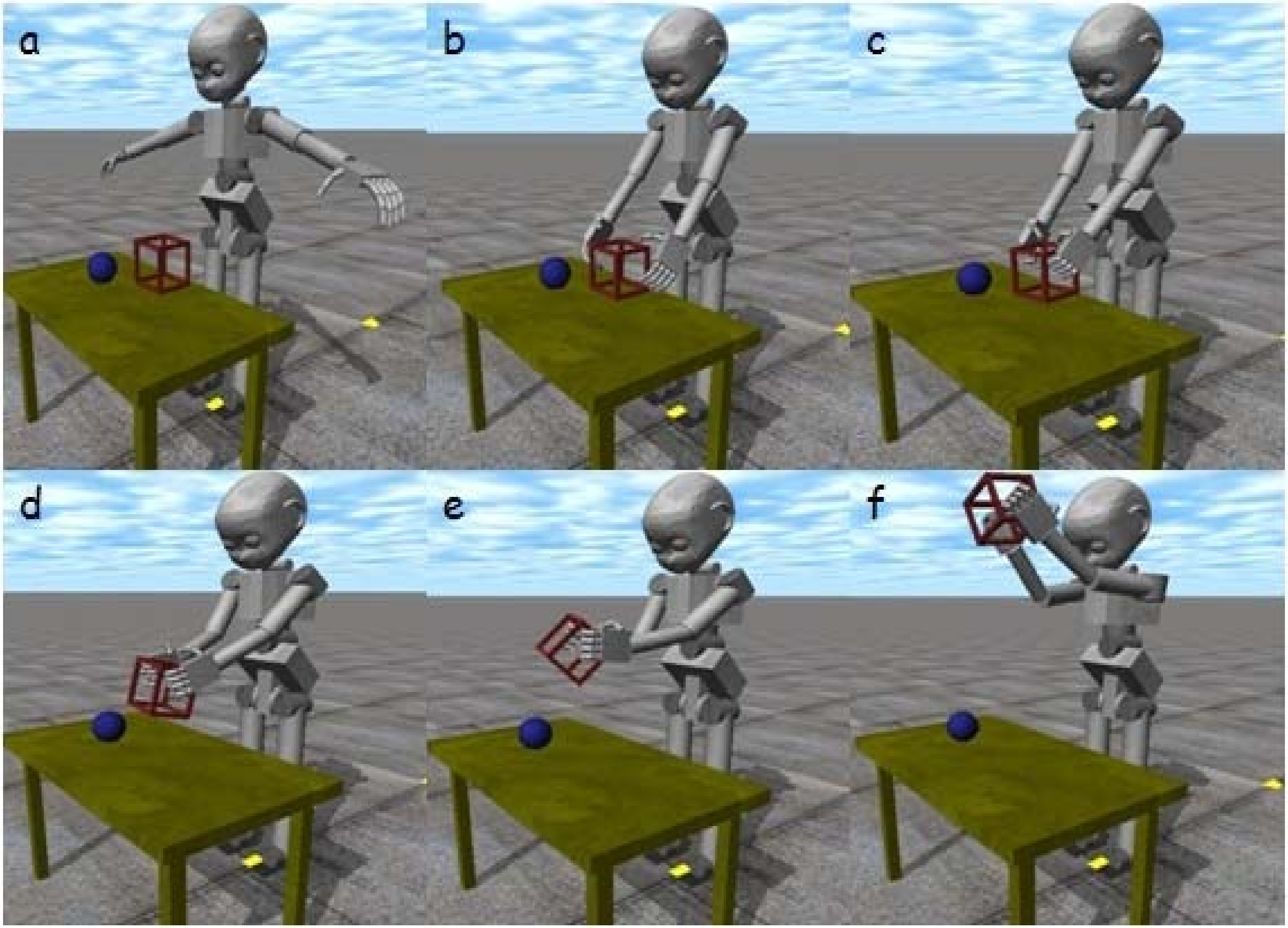


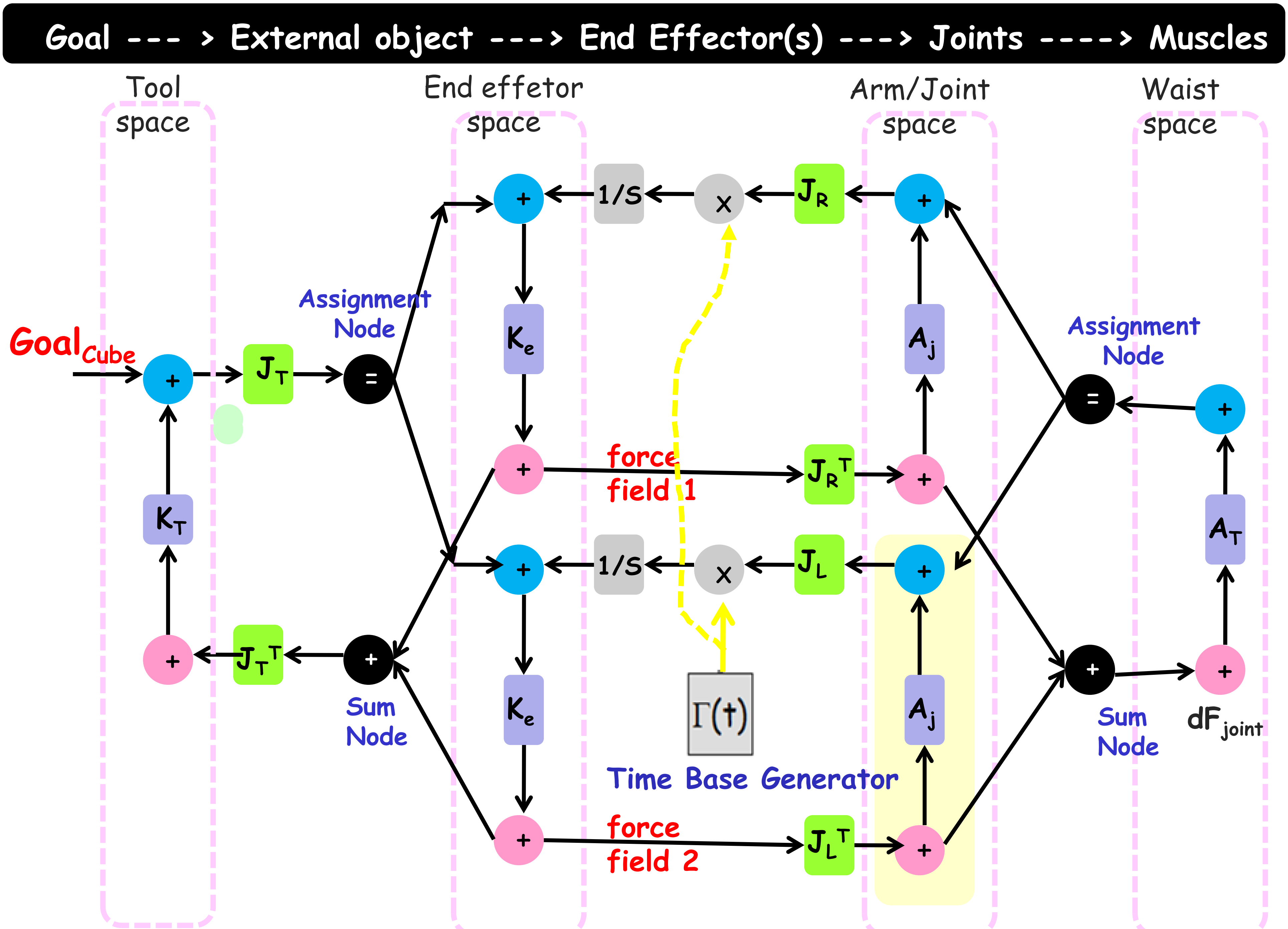
Bimanual Coordination Task
Right Arm- Waist-Left Arm Network



If the trunk is very stiff, only the DoFs of the arms contribute to the final solution reached by the system: this is equivalent to "ground" both shoulders.

When External objects are coupled to the body

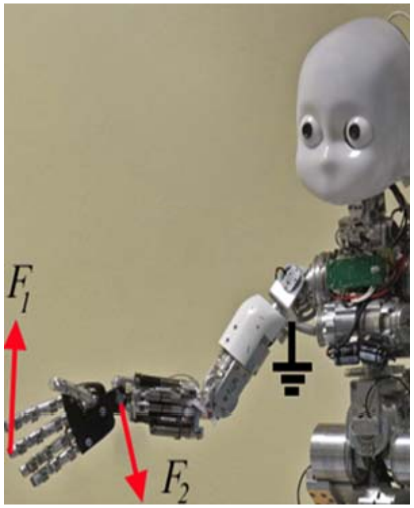




Goal --- > External object ---> End Effector(s) ---> Joints ----> Muscles
Tool space
End effetor space
Arm/Joint space
Waist space
Goal$_{Cube}$
J_T
Assignment Node
K_T
J_T^T
Sum Node
1/S
x
J_R
K_e
A_j
force field 1
J_R^T
Assignment Node
A_T
J_L
$\Gamma(t)$
Time Base Generator
force field 2
J_L^T
Sum Node
dF$_{joint}$

iCub Bimanual Coordination III: Bimanual Transportation Task

Compositionality: Inside the PMP for a single arm



Grounding

Different Motor space's

Work Units

Elastic Transformation

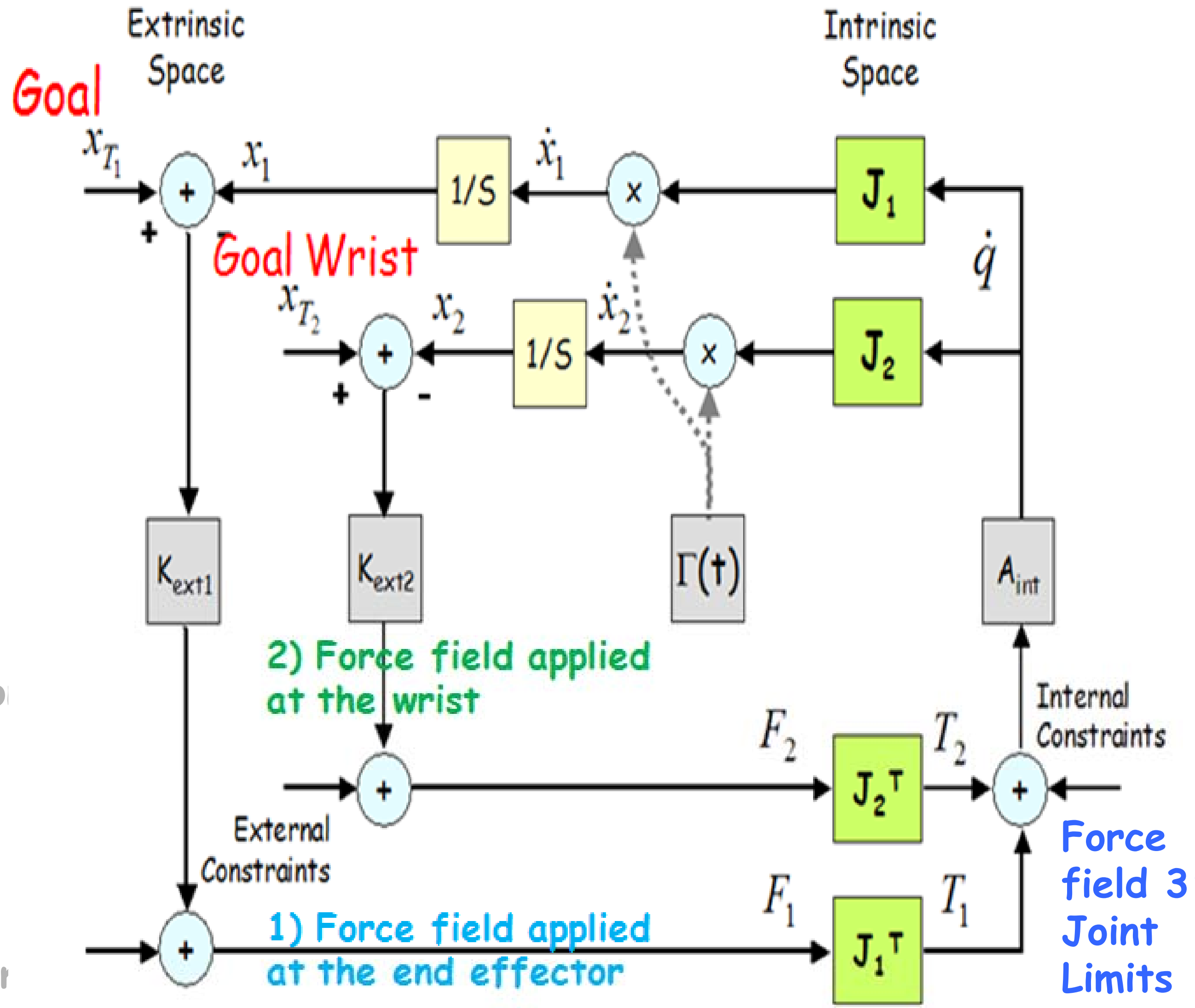
Geometric Transformatio

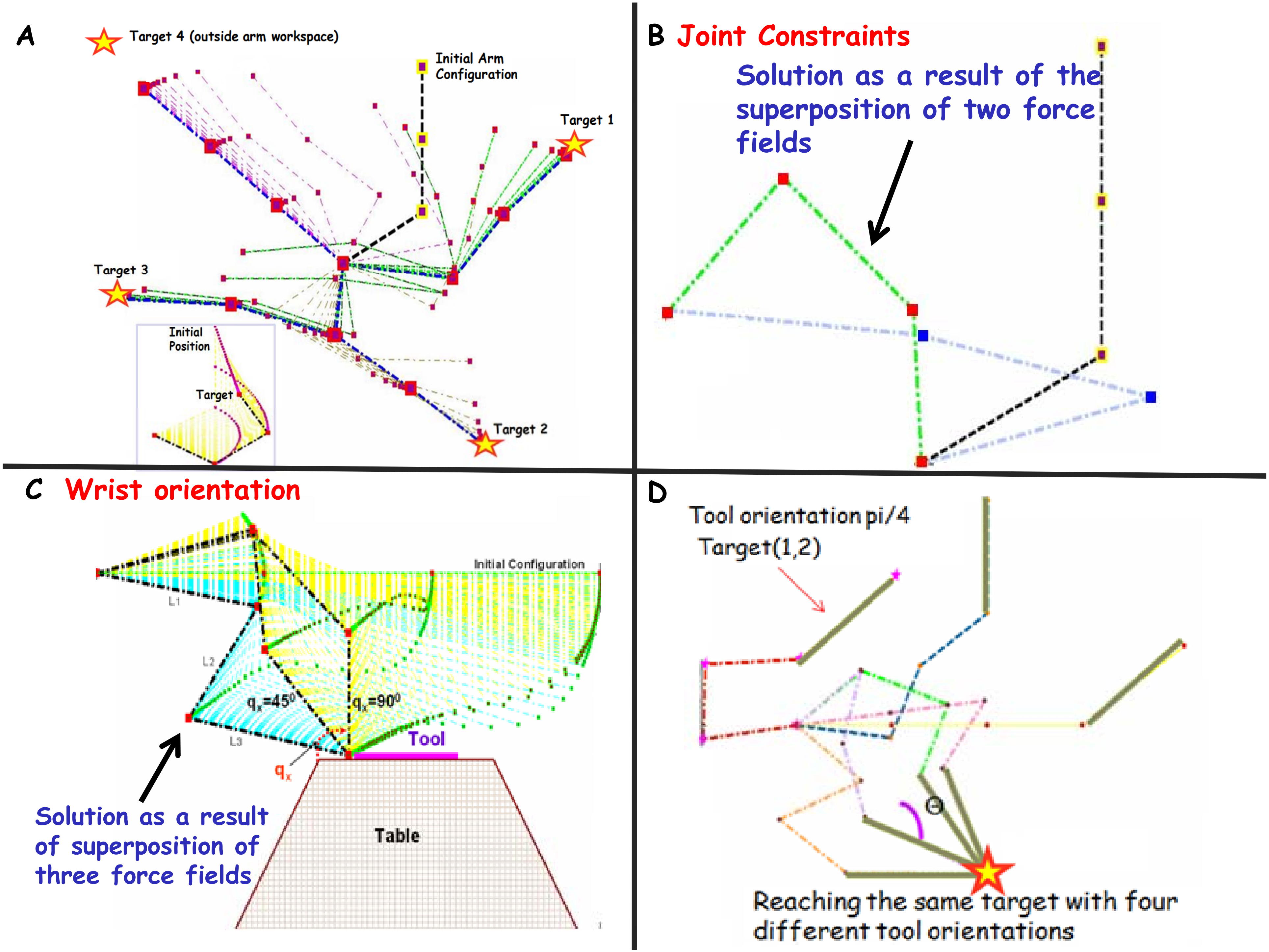
Branching Nodes

Force field to Target

Terminar attractor dynai

Task Specific constraints

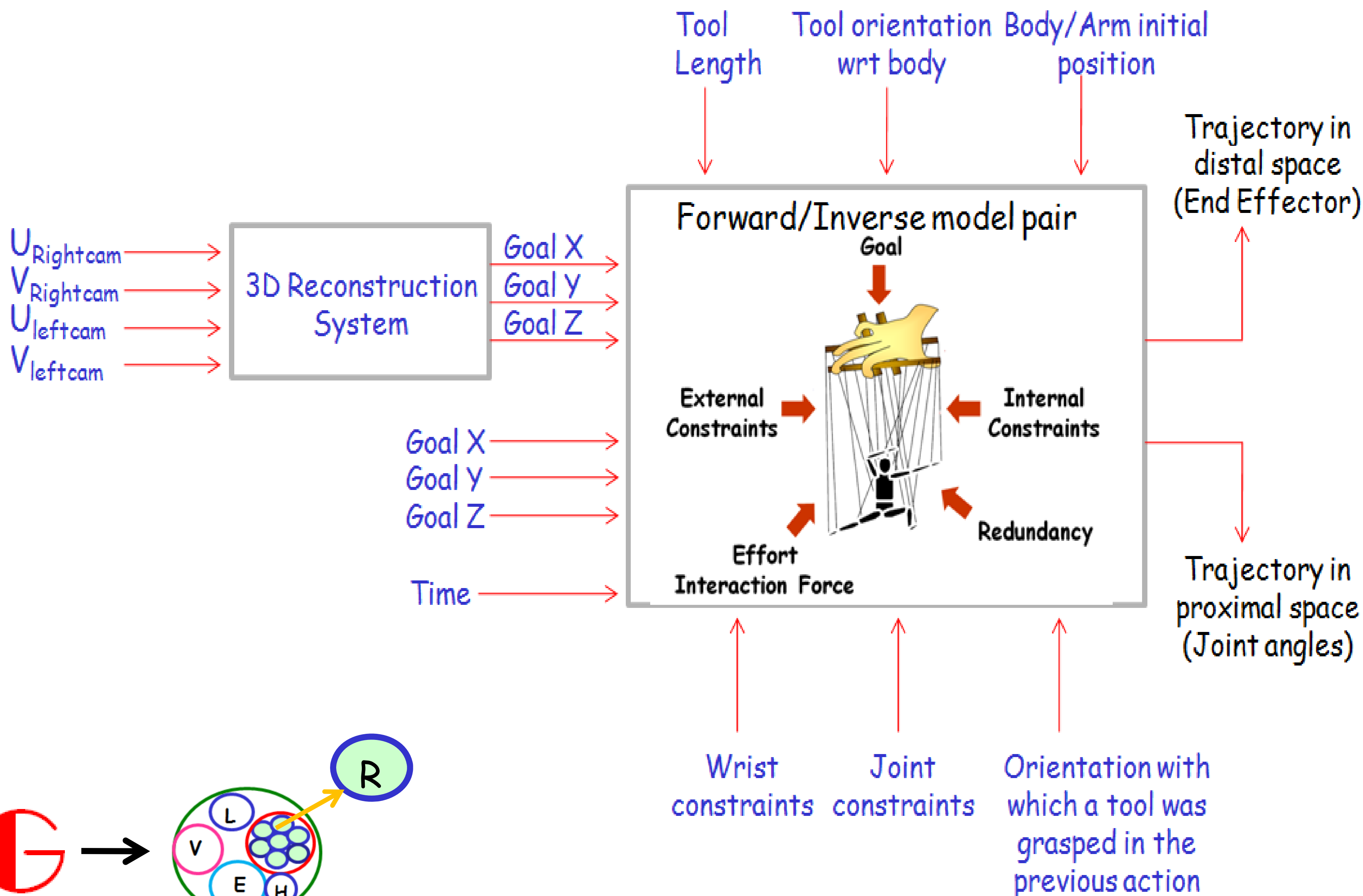




A
Target 4 (outside arm workspace)
Initial Arm Configuration
Target 1
Target 3
Initial Position
Target
Target 2
B
Joint Constraints
Solution as a result of the superposition of two force fields
C
Wrist orientation
Initial Configuration
L1
L2
L3
q_x=45^0
q_x=90^0
Tool
q_x
Table
Solution as a result of superposition of three force fields
D
Tool orientation pi/4
Target(1,2)
Θ
Reaching the same target with four different tool orientations

Fine Tuning the Motor System of GNOSYS

Input/output interfaces to the forward inverse model pair in the GNOSYS robot

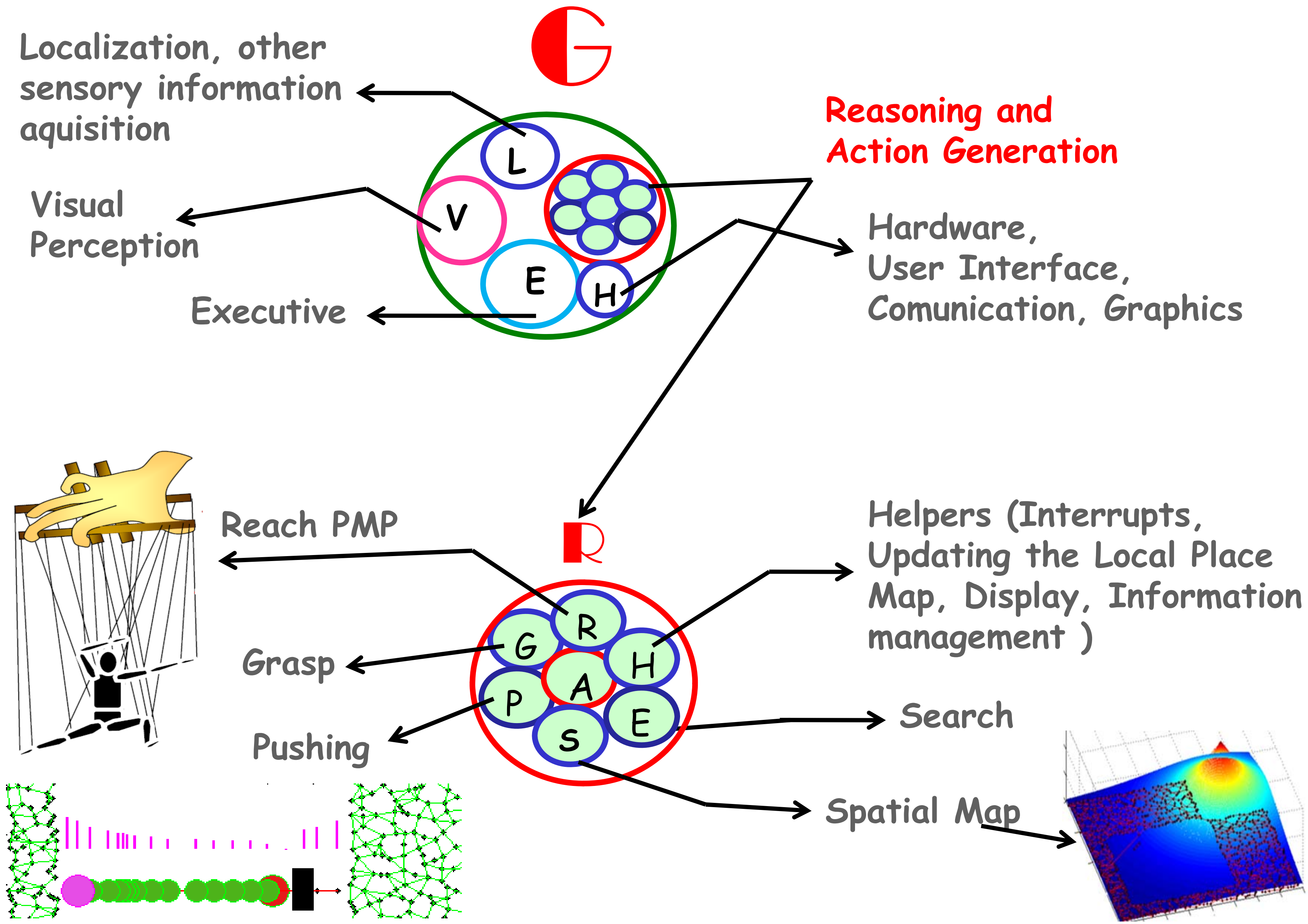


ACTION GENERATION SYSTEM in “action” on GNOSYS

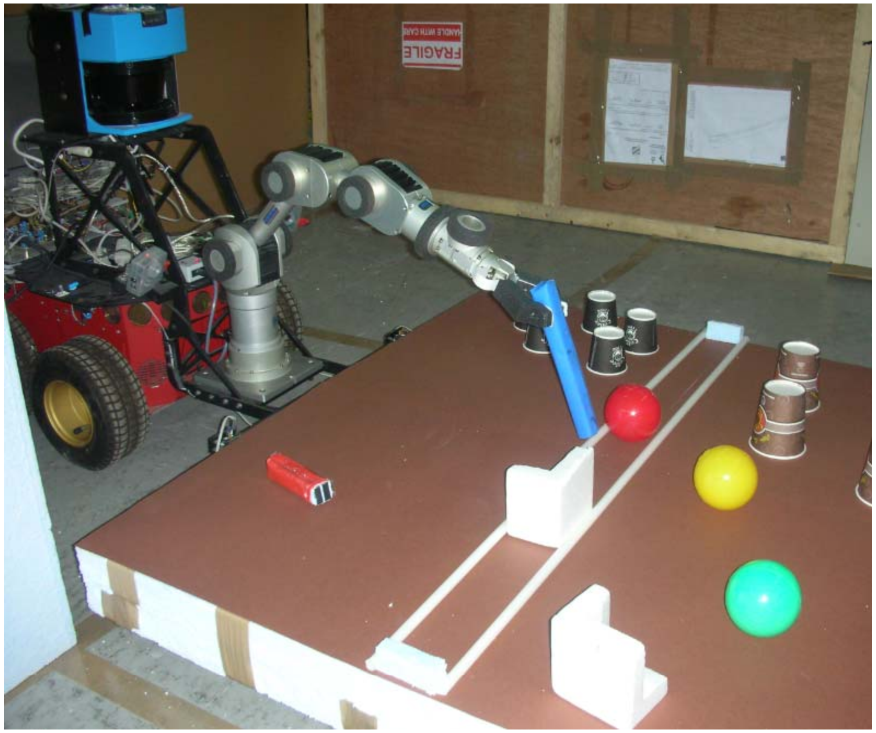
Outline

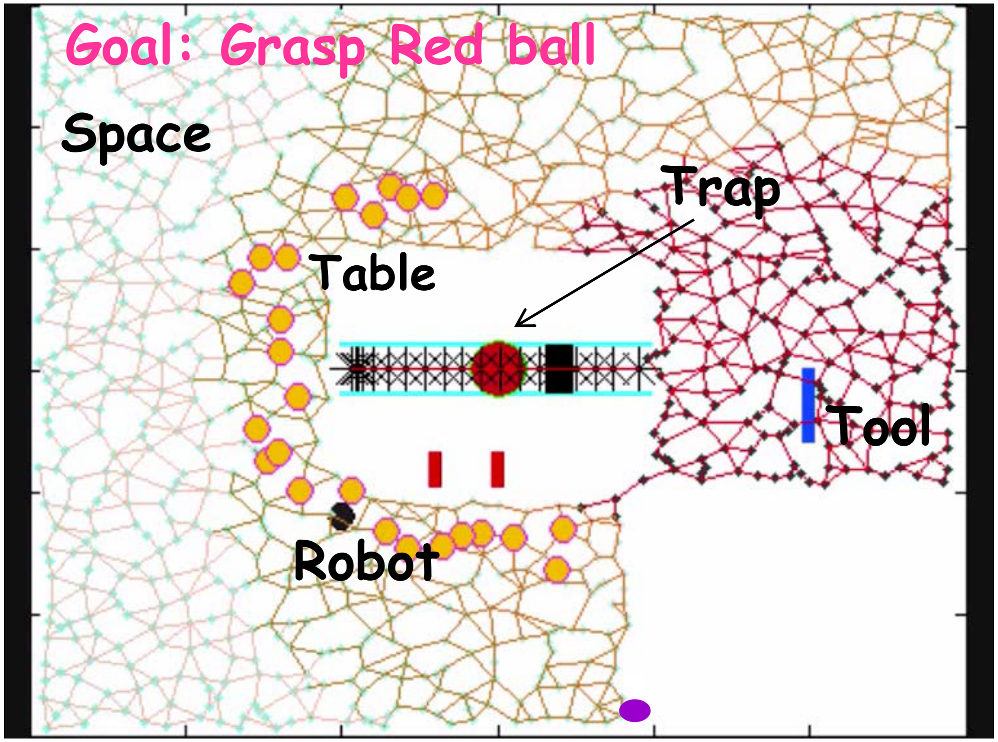
- Objectives (**Overview**)
- Actions (**Local** View)
  - > Arena of Action
  - > Computing With the **'Body'**
  - > Computing in the **'World'**
- Reasoning about Actions (**Global** View)
  - > Computing in the **'Mind'**
- Fabric of Reason's and Action's

  (**Objective/ Subjective** View)
- Atomic Cognitive Agents (**Future** View)


Localization, other sensory information aquisition
Visual Perception
Executive
L
V
E
H
Reasoning and Action Generation
Hardware, User Interface, Comunication, Graphics
Reach PMP
Grasp
Pushing
R
G
H
A
P
E
S
Helpers (Interrupts, Updating the Local Place Map, Display, Information management)
Search
Spatial Map

Central Principles (Invariants)





1) Sensorimotor Exploration
2) Self organization
3) Field Computing
4) Value Dependent learning
5) Abstraction
6) Recursivity

1) A set of **sensorimotor variables** (State space)
2) A set of **connectivity structures** (Interactions)
3) A set of **value fields** (Goal-directedness)
4) A set of **trajectories** (output)

T.Kohonen (Self organization), M.Toussaint(Sensorimotor Maps), Hopfield (AM), Barto, Sutton and Watkins (RL), Amari (Neural Fields)

1) A set of sensory-motor variables

> **Forward/Inverse model for Reaching**

Motor: Array of Joint angles (DoF)

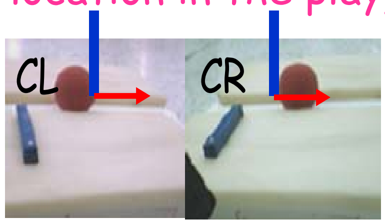
Sensory: End Effector Positions

> **Internal Spatial Map**

Motor : Input array (Of Translation, Rotation vectors)

Sensory : Global location in the playground

> **Pushing**



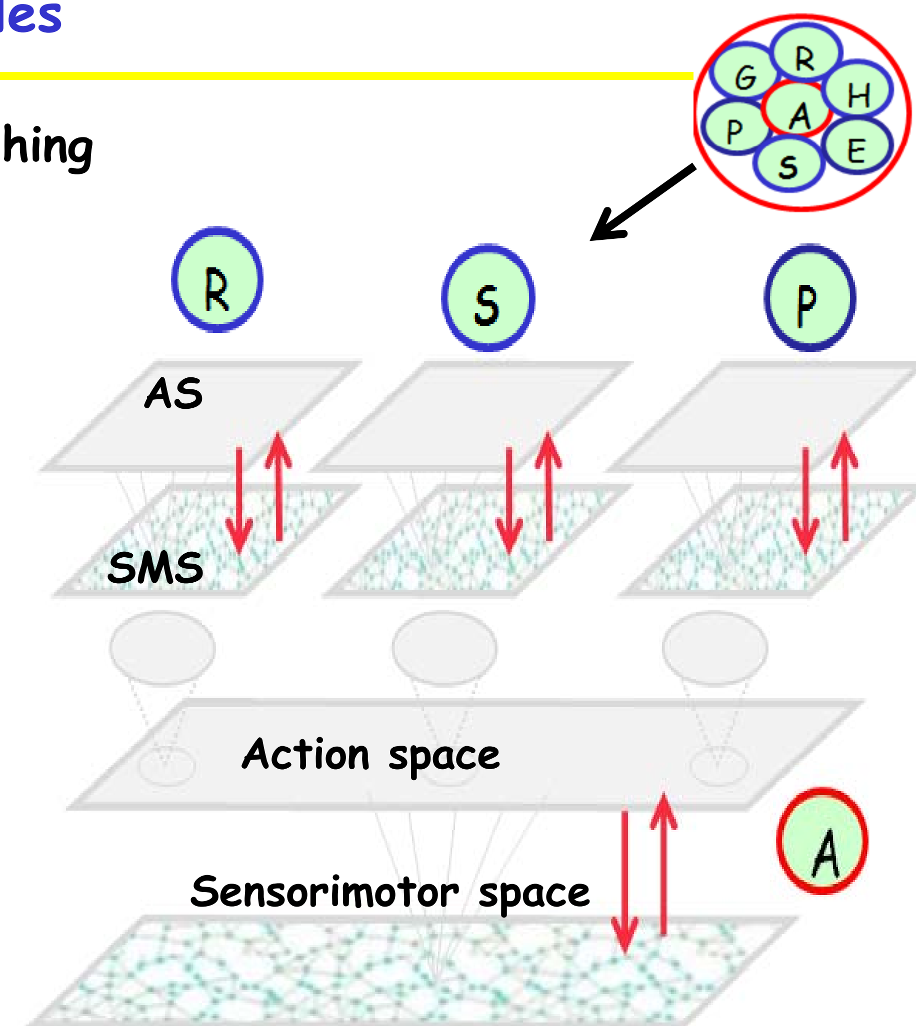
Motor : Direction of Pushing,
Location of tool wrt object

Sensory : Position of object after push

> **Abstract Reasoning**

Motor : Array of Actions {Reaching, Pushing, S.navigation, Visual Exploration, Grasp, H}

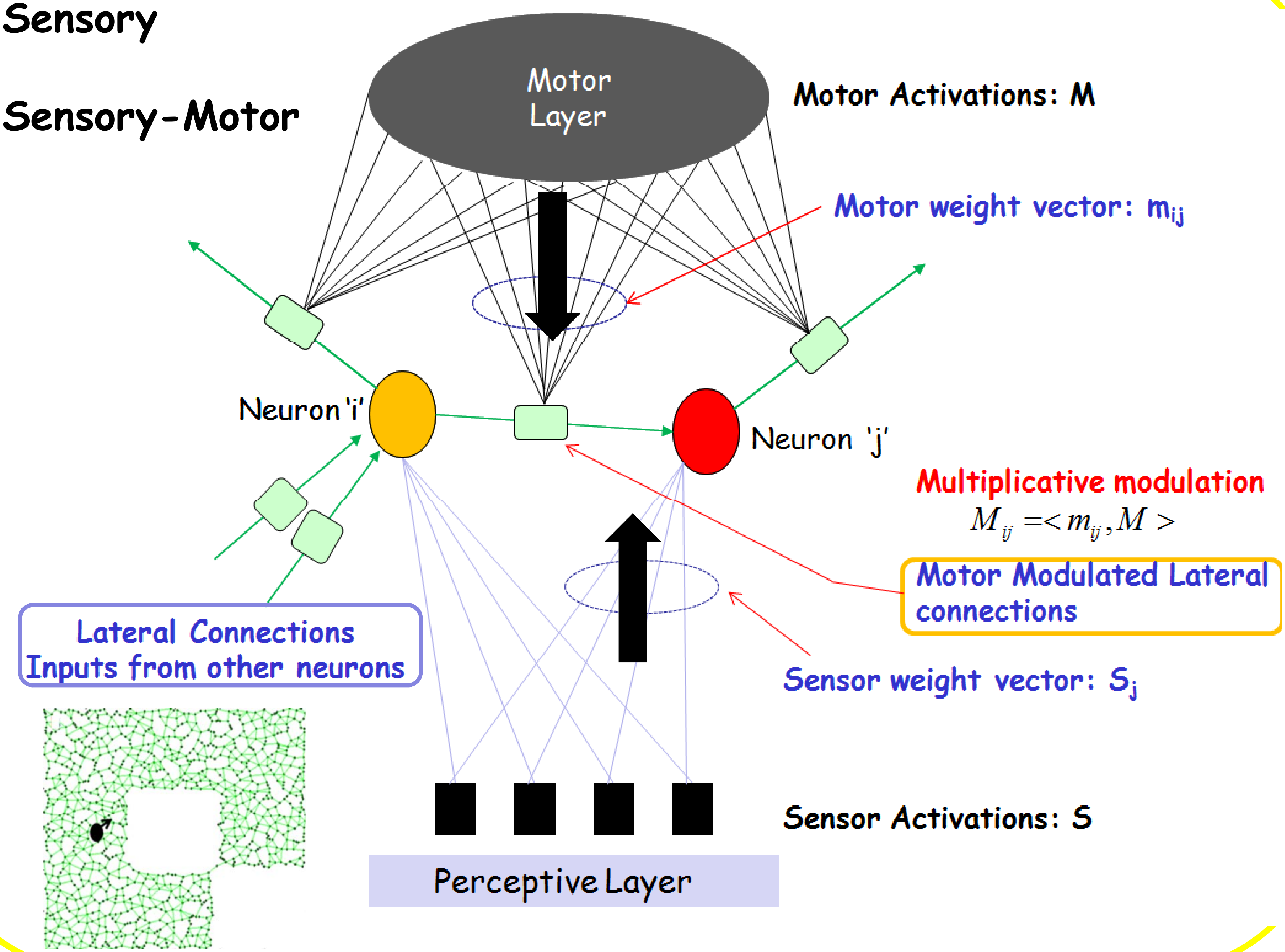
Sensory : Composite Agent State Body : (F/I models, Gripper, Vision, Motivation, Activity flag) Environment: (object descriptors: Goal and tool if any)



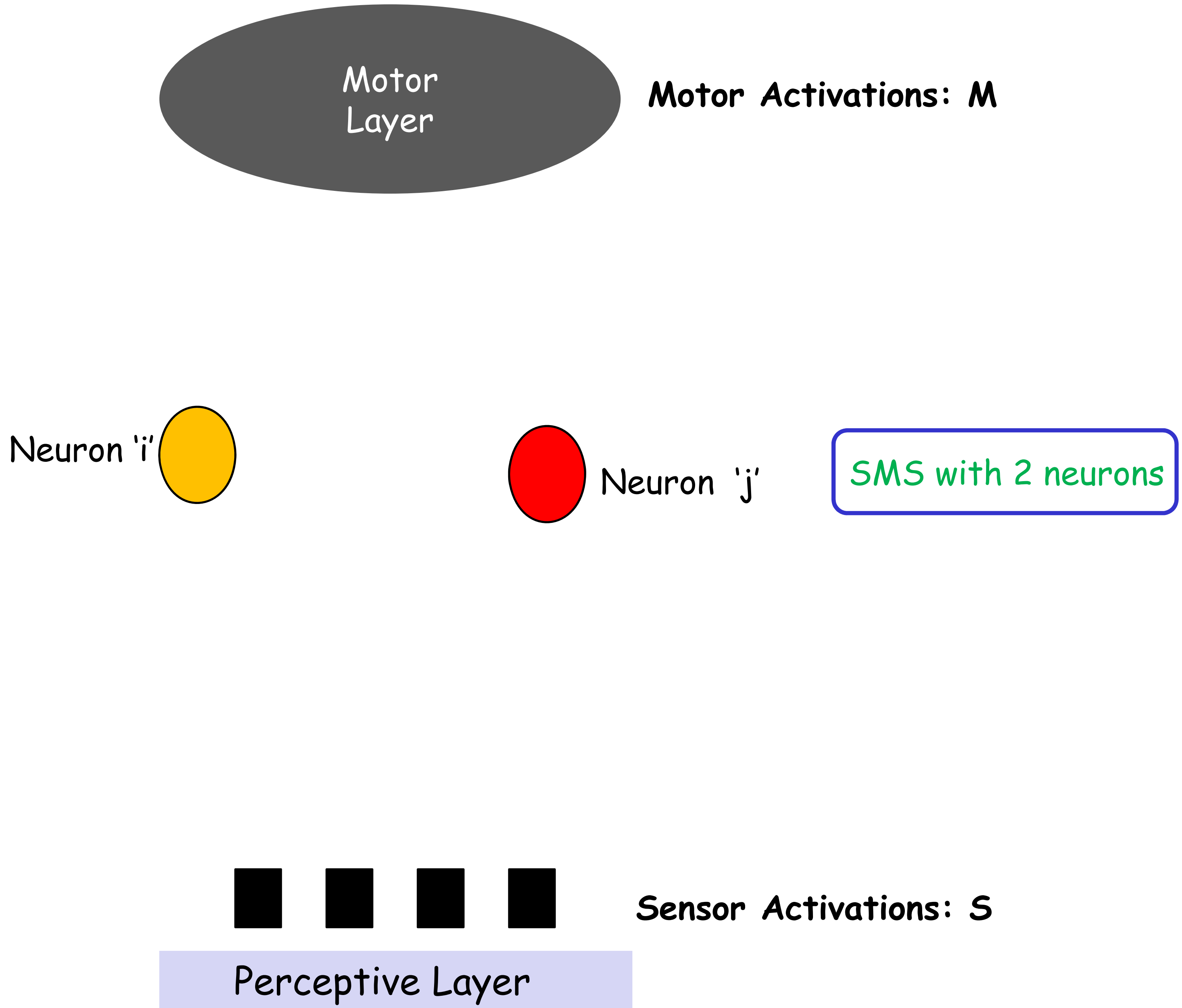
Computational Substrate (A two neuron zoomed view)

L1. Sensory

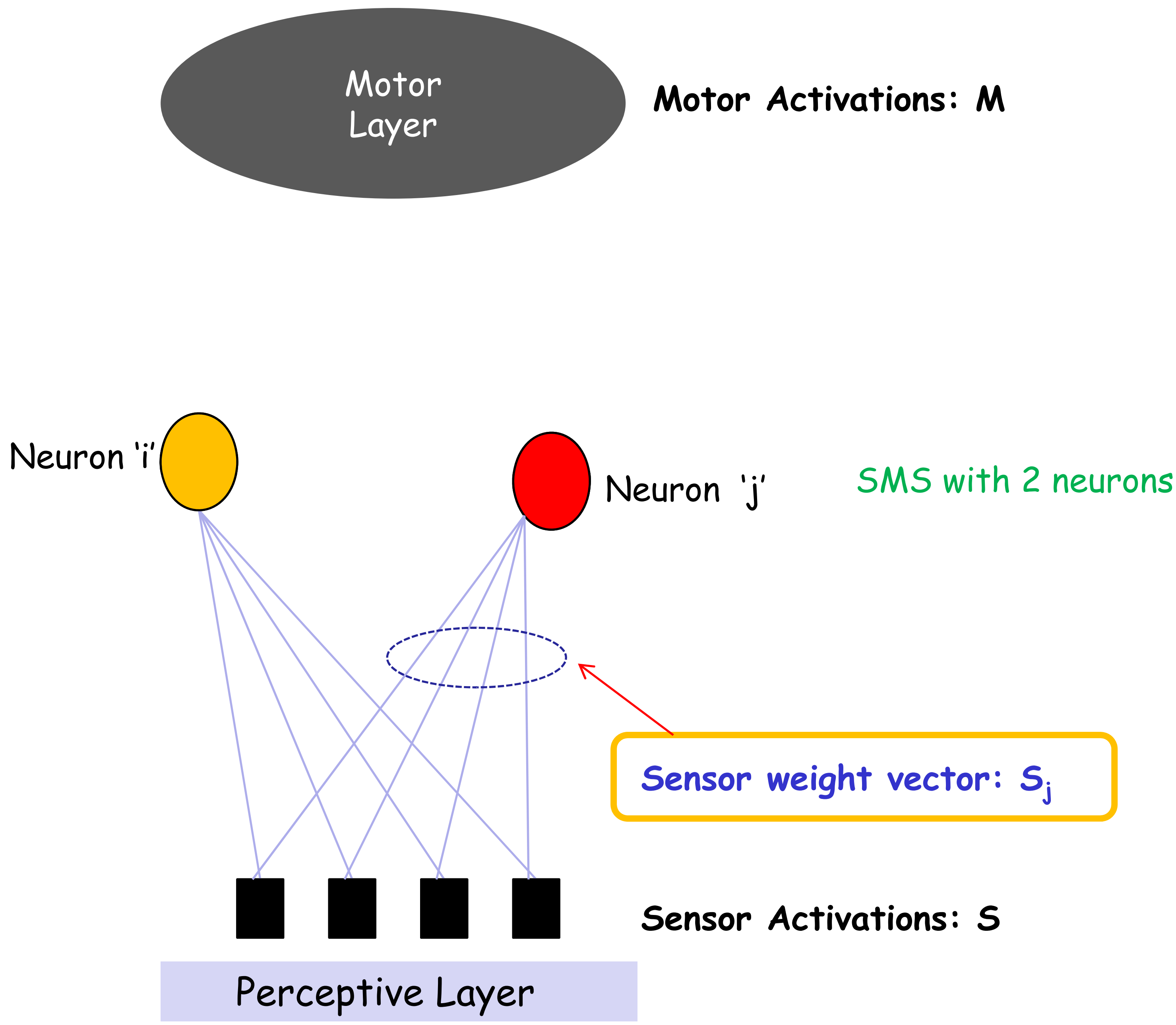
L2. Sensory-Motor



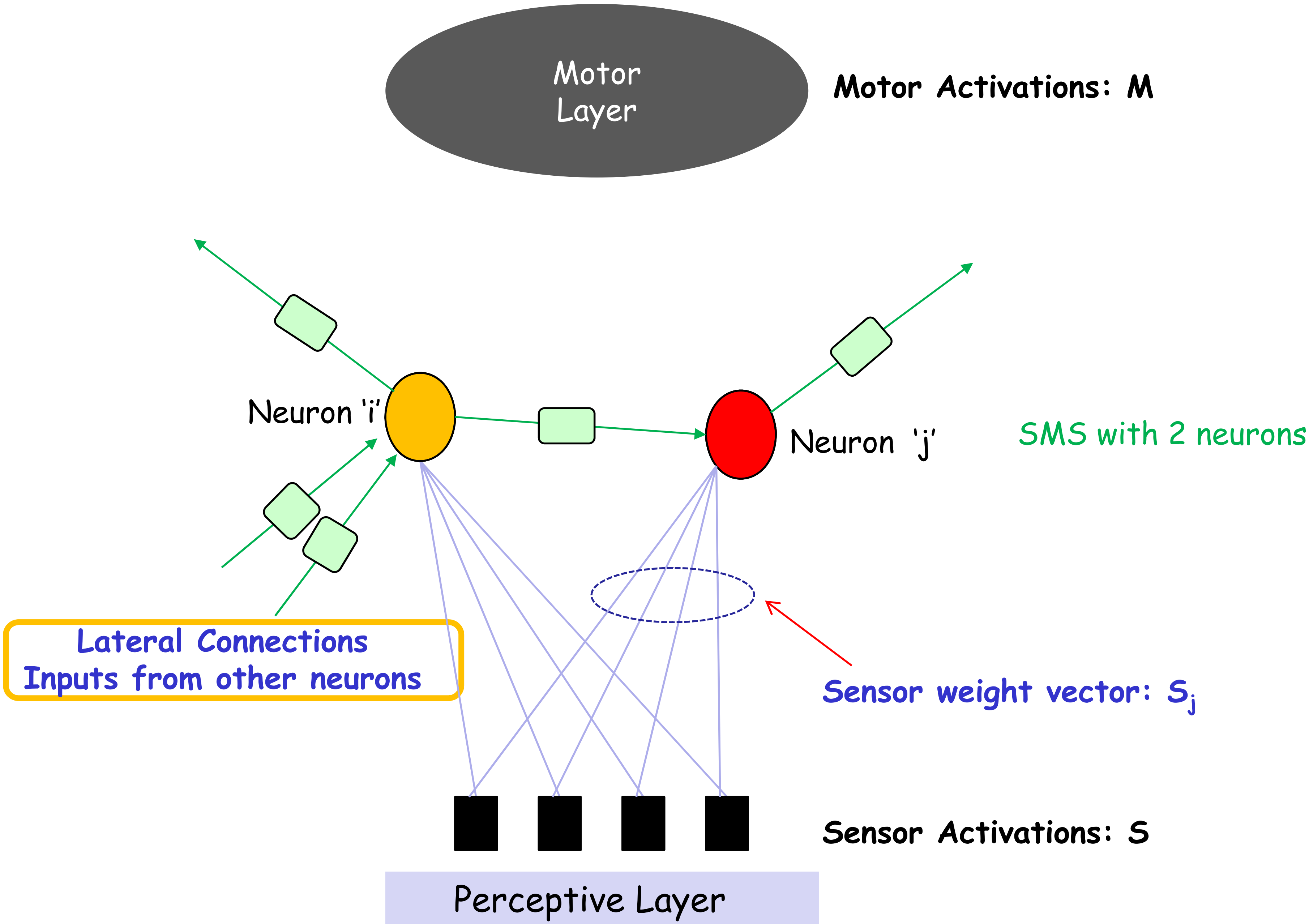
Connectivity in the Sensorimotor space (SMS)



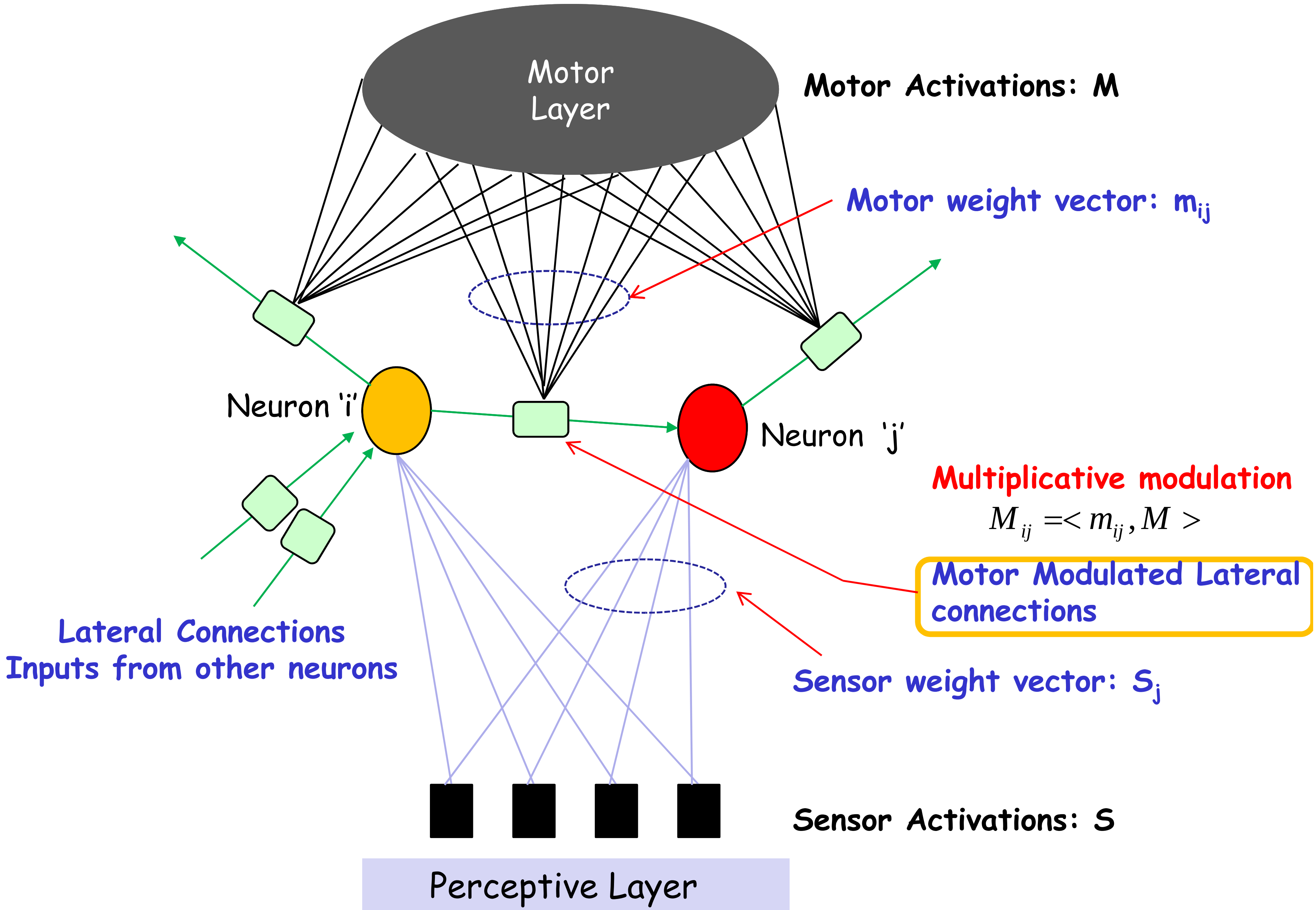
Connectivity in the Sensorimotor space (SMS)



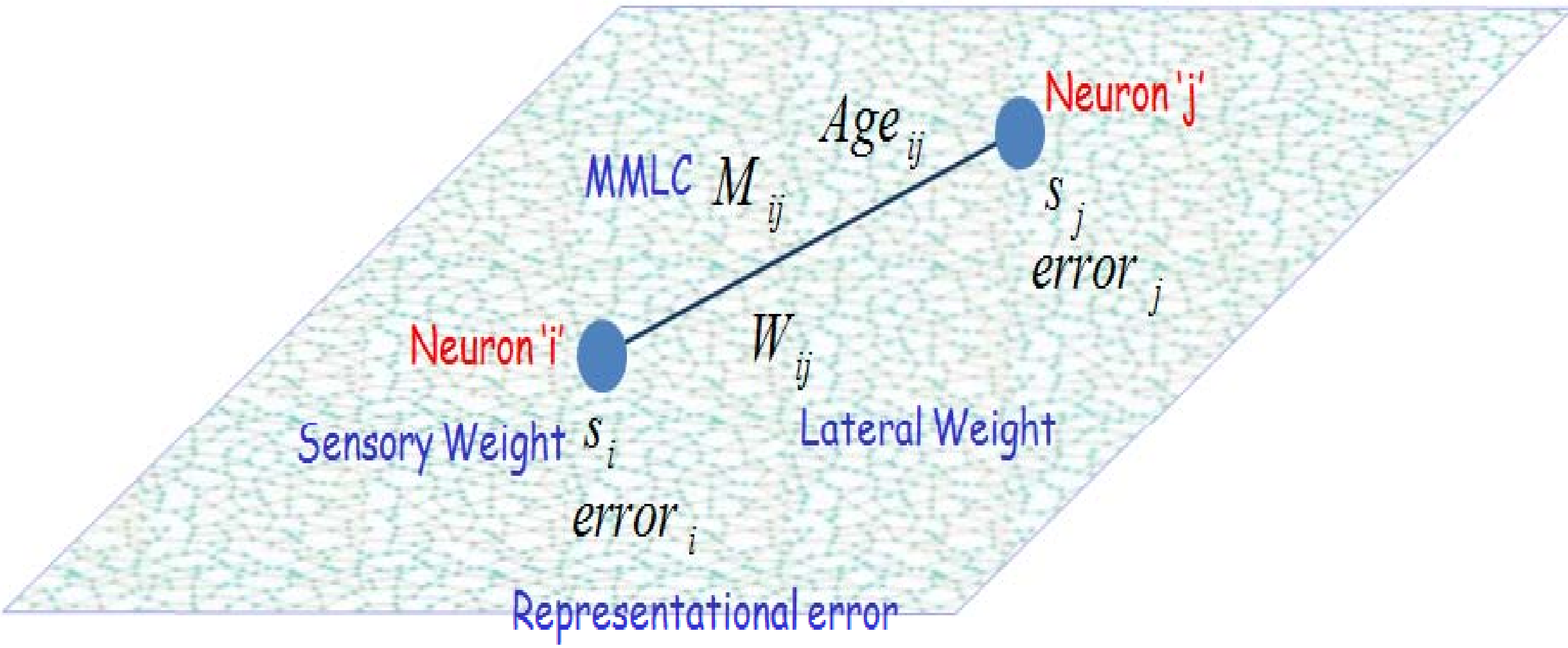
Connectivity in the Sensorimotor space (SMS)



Connectivity in the Sensorimotor space (SMS)



Free variables that need to be learnt in this phase of self organization.



N: No. of neurons in the sensorimotor space (N);

S: Sensory weights for each neuron ($N \times D_{Sensor}$), these are randomly initialized;

$error_i$: local estimate of representational error (useful information for growing) (N)

W_{ij}: Lateral weights ($N \times N$)

Age_{ij} : Age of lateral connection ($N \times N$).

M_{ij}: Motor Modulated lateral connections ($N \times N \times D_{motor}$)

1) Start with **one neuron** with randomly initialized sensory weights

2) Generate a **random motor activation M^t** and **observe** the incoming sensory information S^t

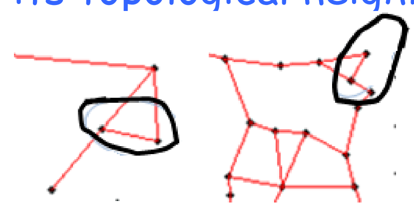
3) Find the neuron 'i' that shows maximum activity for the observed sensory stimulus S^t at time t. **(Winner)**

4) **Grow** based on local representational error

5) Adapt the **sensory weights** of the Winner and its topological neighbors

$$s_i \longleftarrow s_i + e_w(\overline{S} - \overline{s_i})$$

$$s_n \longleftarrow s_n + e_n(\overline{S} - \overline{s_n}), \forall n \in Neighbours\ (i)$$



6) Adapt the **motor weights** and **lateral weights** for all the neurons

$$m_{ij}^T = \frac{1}{\sum_{t'=1}^{T} \alpha_{ij}^{t'}} \sum_{t=1}^{T} \alpha_{ij}^t M^t$$

w_{ij}=1 (if motor action M^t from 'j' resulted in 'i')

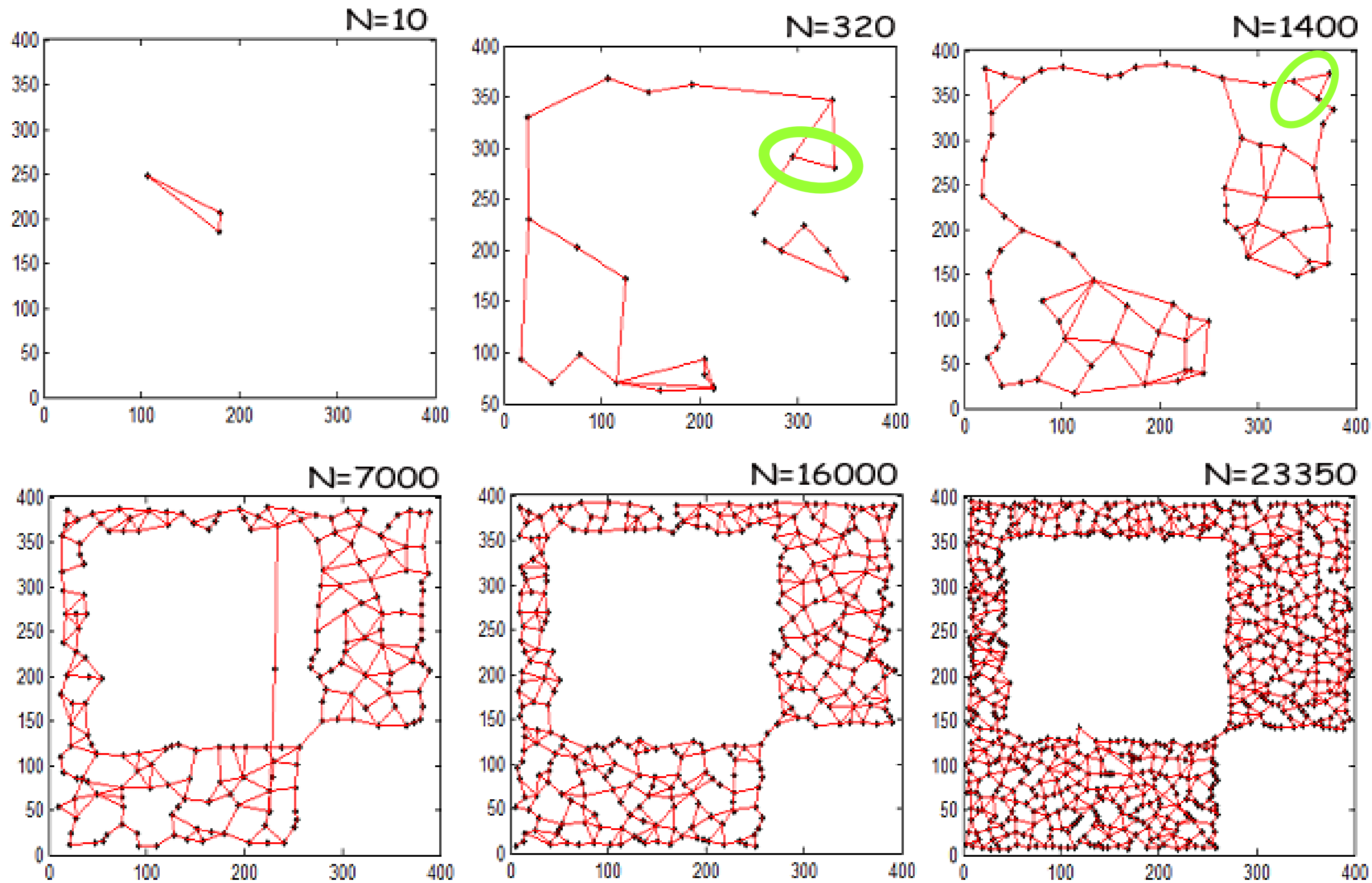
7) Make the **age** of the 'i-j' lateral connection zero, increase the age of all other lateral connections. Eliminate lateral connections who's Age > Age_{max}. Eliminate deal neurons.

8) Move to **next step of random motor action generation**, observation and self organization

growing.avi

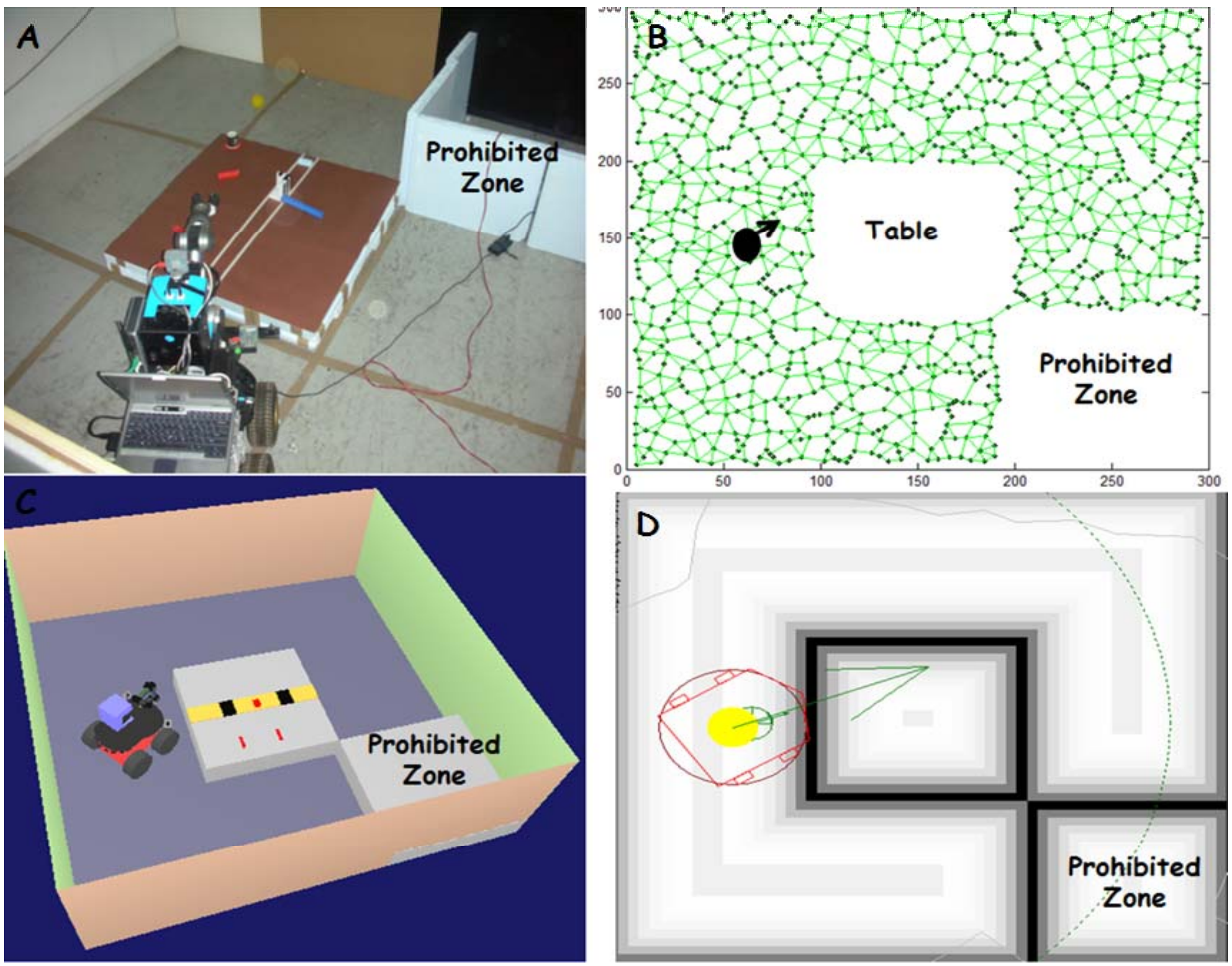
1) A set of sensory-motor variables

2) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)

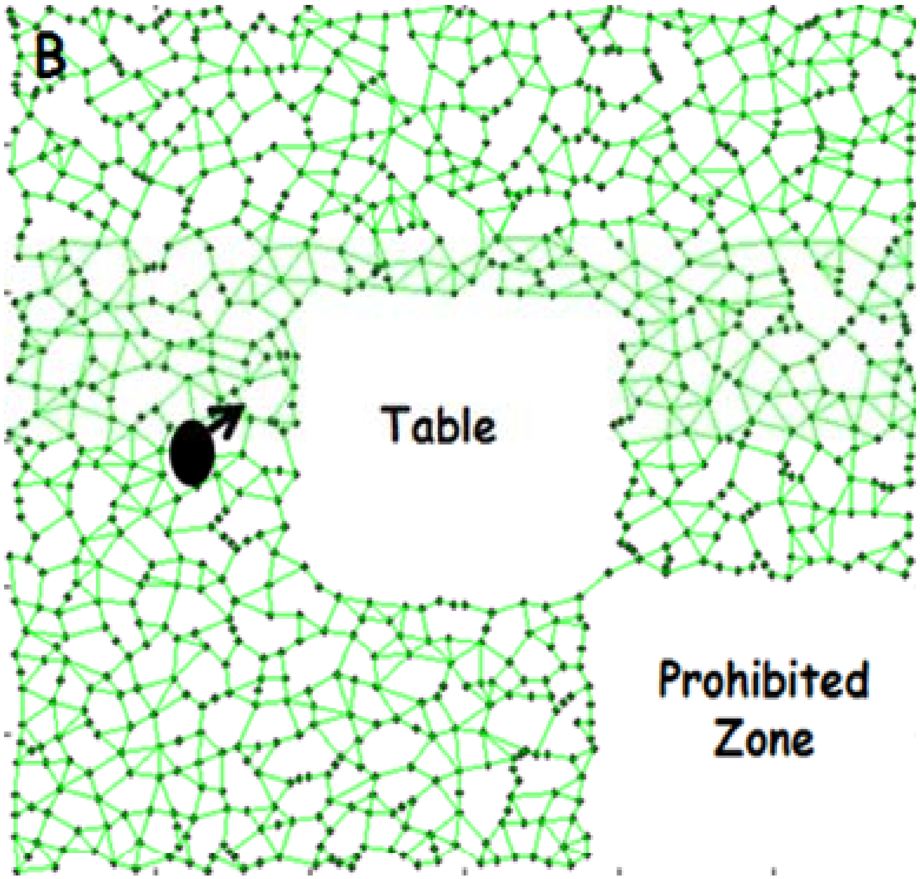


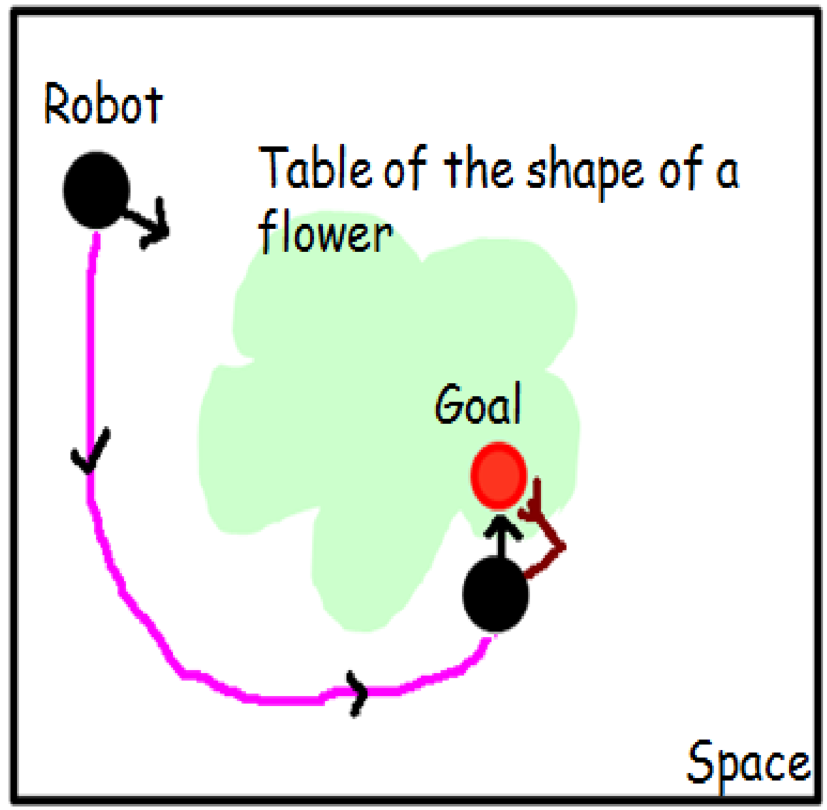
1) A set of sensory-motor variables

2) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)



Obstacles are implicity represented in the SMS





What is learnt through exploration is the free space where motion is possible and whatever remains independent of its geometry is an obstacle in the playground.

1) A set of sensory-motor variables

2) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)

3) Dynamics of the SMS

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Exponential relaxation to the dynamics

Feed Forward Input

Bifurcation

Lateral and Top Down interaction

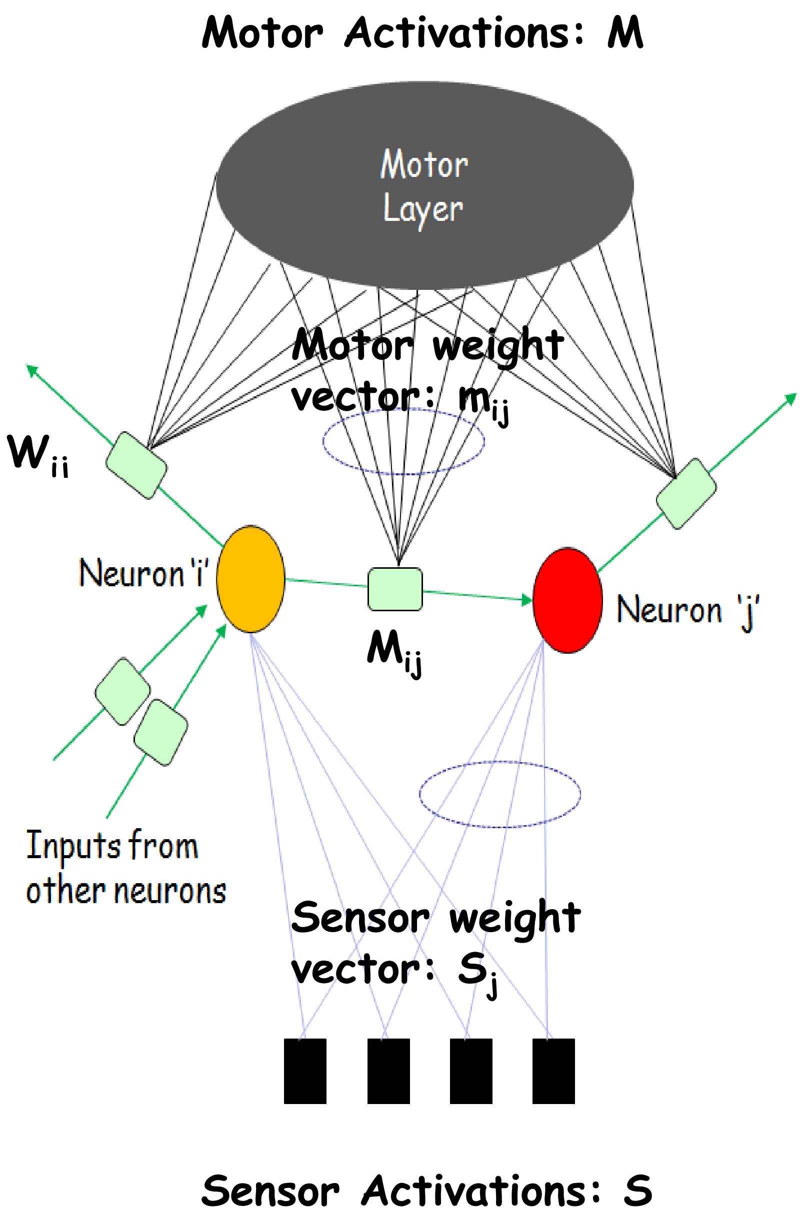
$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}}\right) + \zeta(\bar{r})$$

Normal Computation of incremental motor action based on values

Shift to random exploration

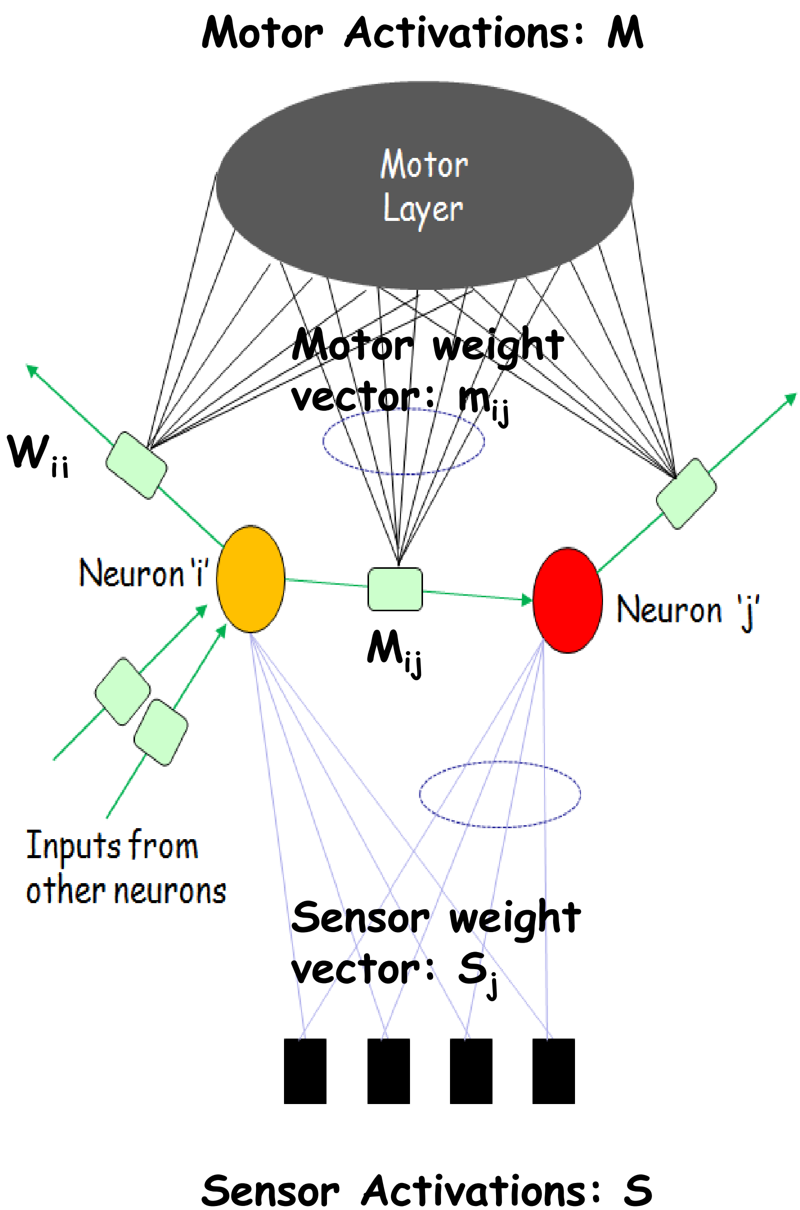
How activity moves **Bidirectionally** between sensory and motor units

Dynamics: On moving in the sensorimotor space



$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

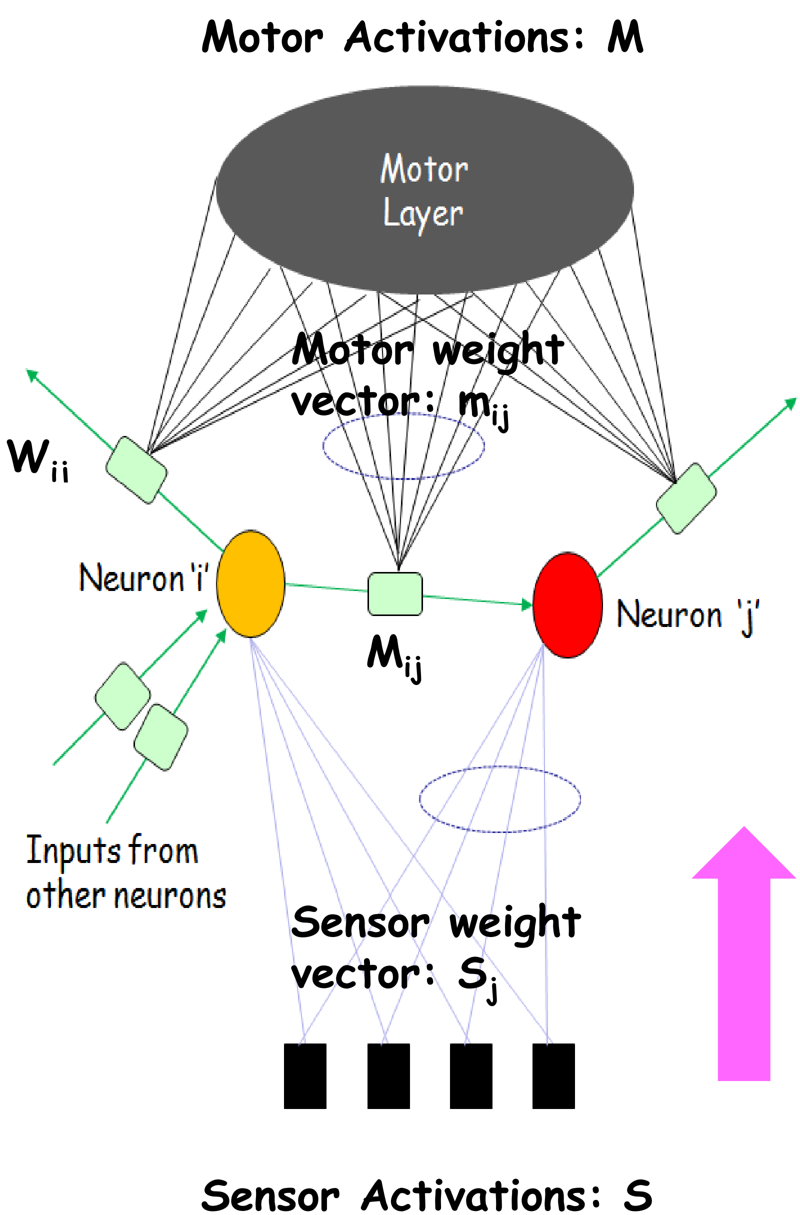
Dynamics: On moving in the sensorimotor space



$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Exponential relaxation to the dynamics

Dynamics: On moving in the sensorimotor space



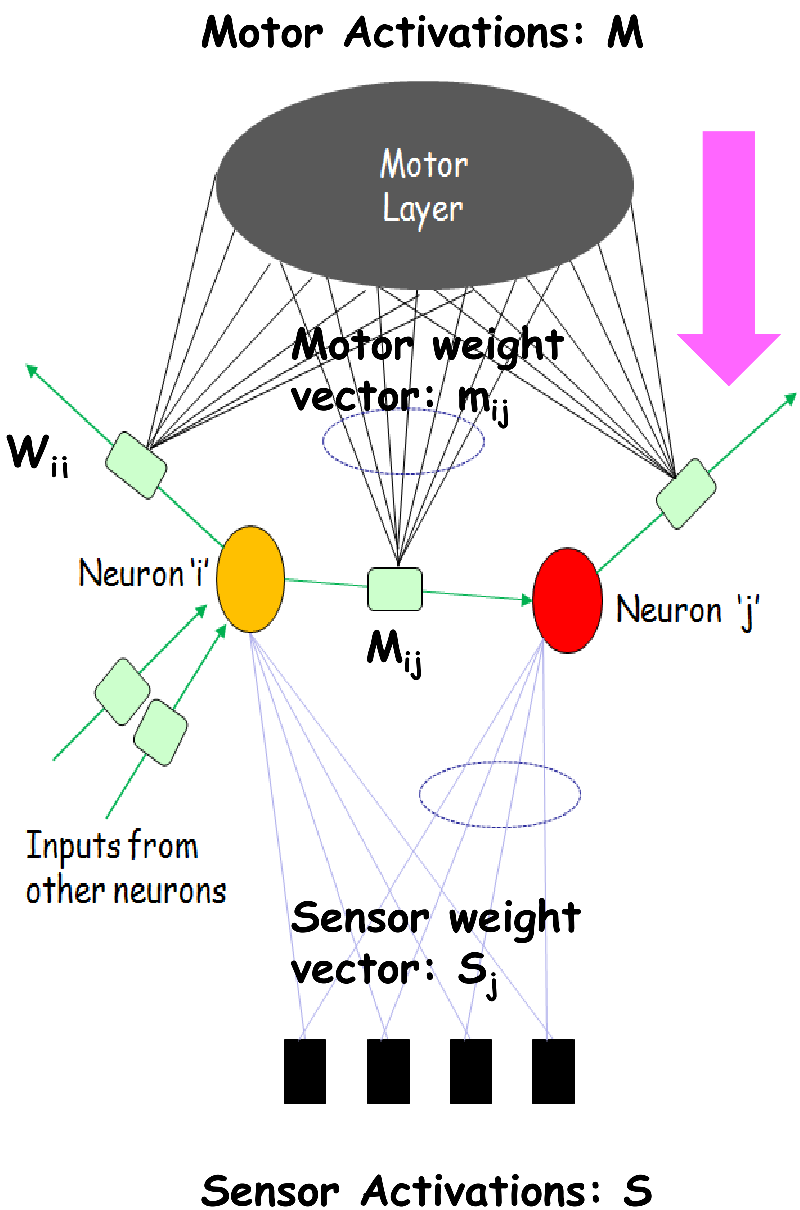
$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Feed Forward Input (bottom up)

$$S_i = \frac{1}{\sqrt{2\pi\sigma_s}} e^{\frac{-(s_i - s)^2}{2\sigma_s^2}}$$

The Gaussian kernel compares the sensory weight s_i of neuron i with current sensor activations S^t.

Dynamics: On moving in the sensorimotor space



$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Lateral and Top Down Input

$$M_{ij} = < m_{ij}, M >$$

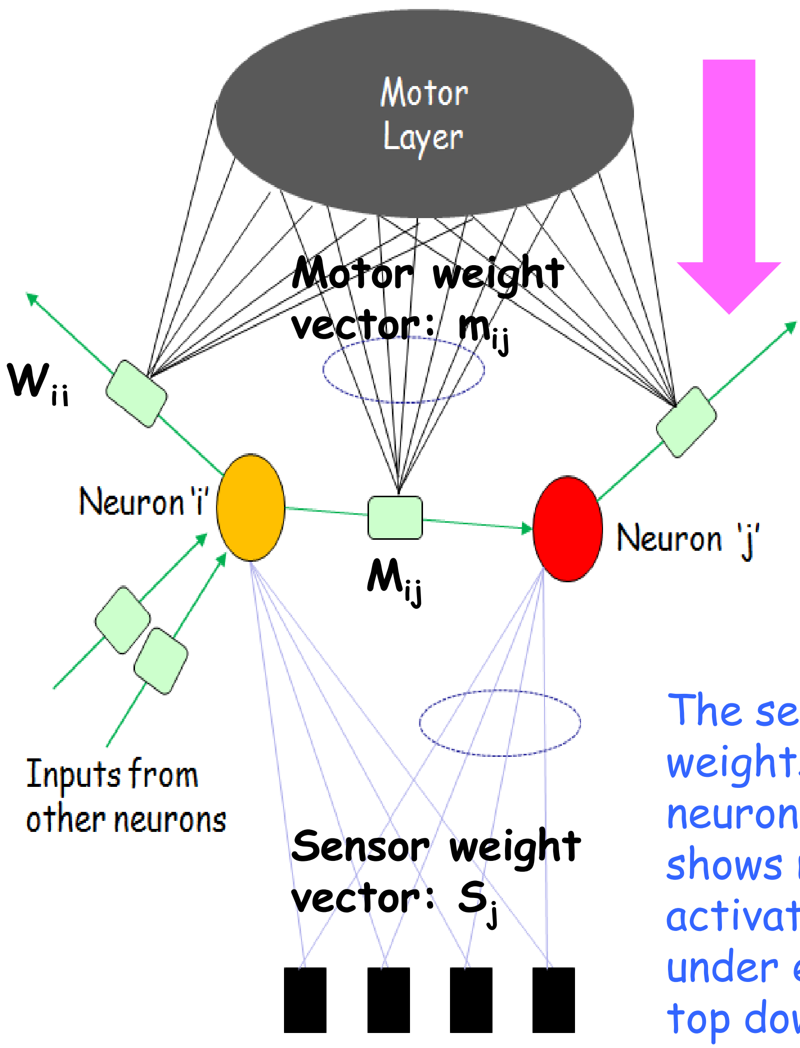
The instantaneous value M_{ij} i.e. the scalar product of motor weight vector m_{ij} with the ongoing motor activations M keeps changing with the activity in the action space.

Due to this multiplicative coupling, a lateral connection contributes to lateral interaction between two neurons only when the current motor activity correlates with the motor weight vector of this connection.

Situation-Action-Consequence loop

Dynamics: On moving in the sensorimotor space

Motor Activations: M



Sensor Activations: S

The sensory weights of the neuron k that shows maximum activation the under effect of top down modulation.

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Bifurcation Parameter

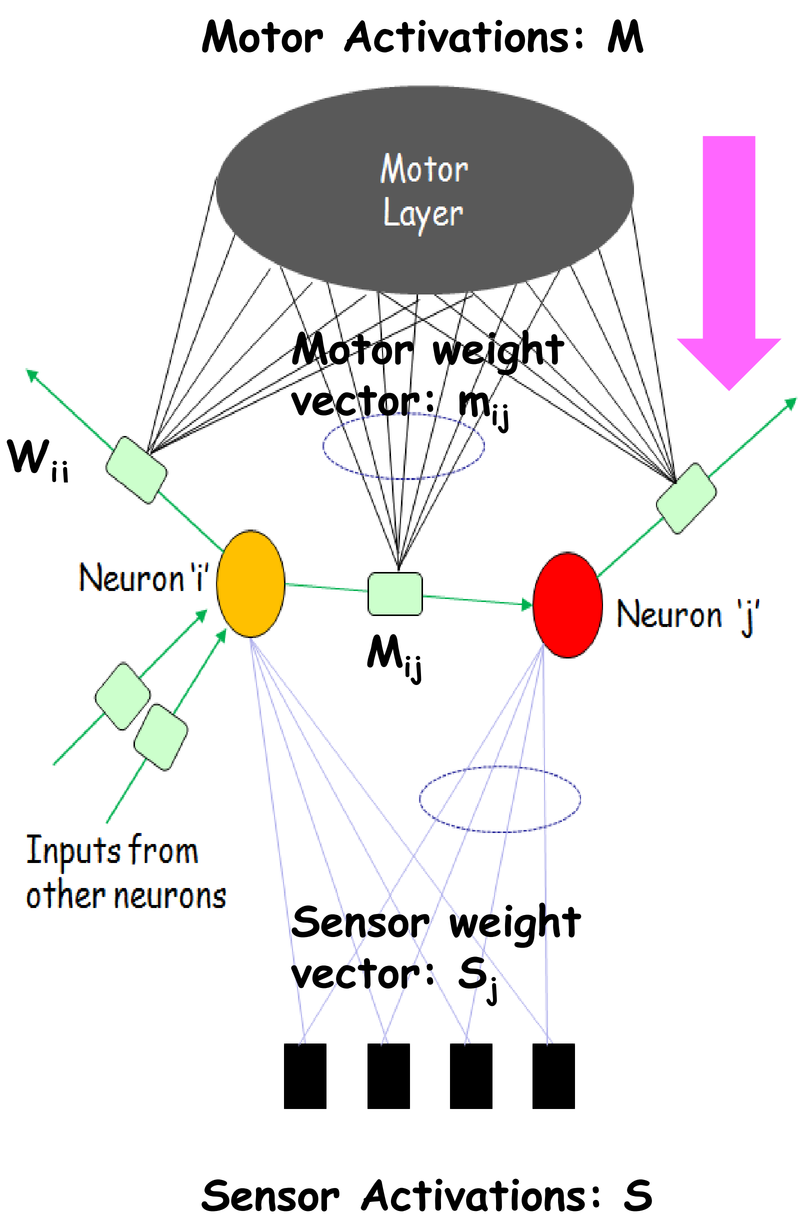
$$\beta_{if} = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(S_{Anticip} - S)^2}{2\sigma^2}}$$

How closely the top down prediction correlates with the real sensory information

$$\beta_{if} \longrightarrow 0$$

Implies that the internal model is locally inaccurate or there is a dynamic change in the real world i.e. 'the world is working differently in comparison to the way the robot thinks the world should be working'.

Dynamics: On moving in the sensorimotor space



$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Motor Layer Dynamics

$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}} \right) + \zeta(\bar{r})$$

Small random motor signal

$$\beta_{if} \longrightarrow 0$$

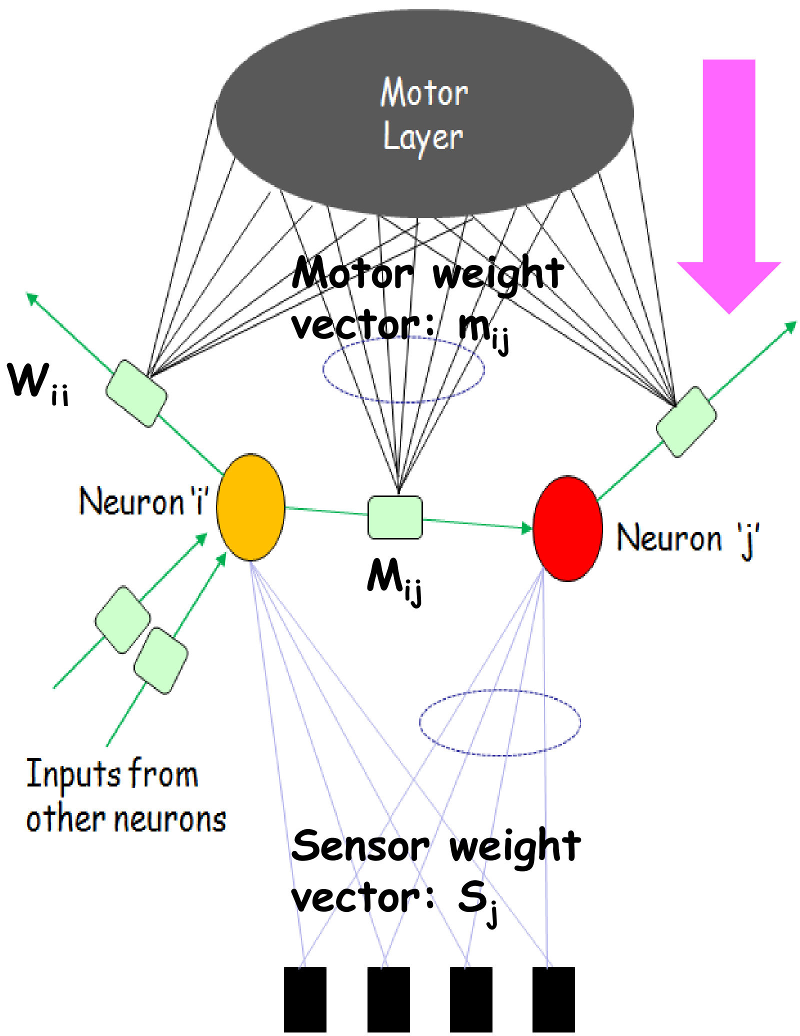
Dynamics is a function of only real sensory information (feed forward)
Contribution of first term in motor dynamics is zero

$$\zeta \longrightarrow 1$$

System dynamics switches to random exploration (like learning the SMS)

Dynamics: On moving in the sensorimotor space

Motor Activations: M



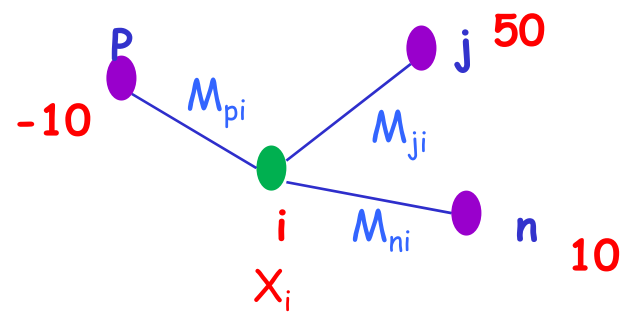
Sensor Activations: S

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Motor Layer Dynamics

$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}} \right) + \zeta(\bar{r}\ )$$

Activation average of all motor weight vectors coded in the MMLC Where K_i is the most valuable neighbor to the i^{th} neuron



$$k_i = \mathrm{argmax}_j (w_{ij} V_j)$$

1) A set of sensory-motor variables

2) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)

3) Dynamics of the SMS

How activity moves Bidirectionally between sensory and motor units

4) Value Field Dynamics

(in a goal directed fashion)

1) A set of sensory-motor variables

2) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)

3) Dynamics of the SMS

How activity moves Bidirectionally between sensory and motor units

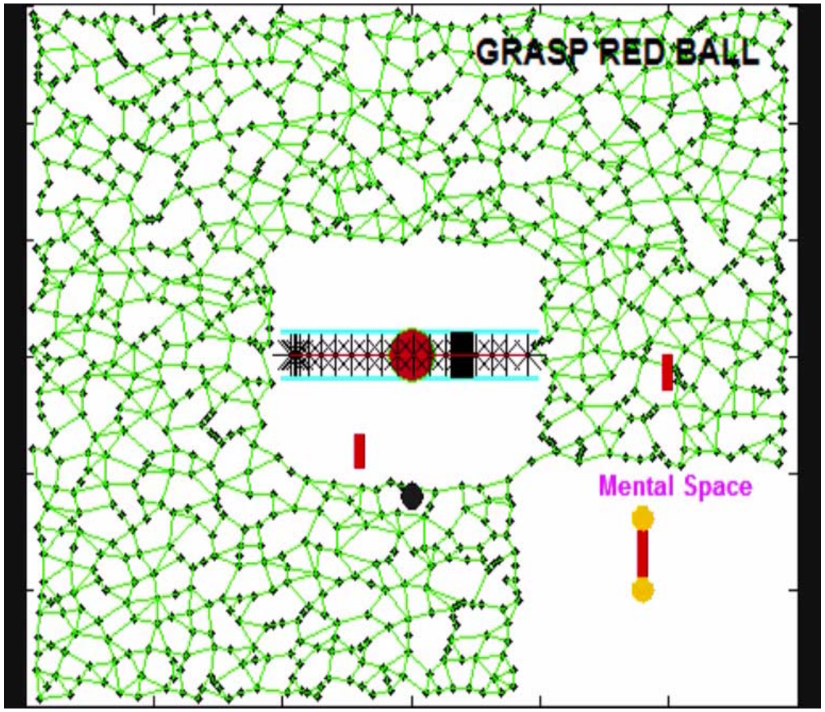
4) Value Field Dynamics (in a goal directed fashion)

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

Instantaneous reward

Future Reward

Future reward



Value fields are Quaistationary and change with goal

Cause Goal directed shifts in activity in the sensorimotor space

1) A set of sensory-motor variables

2) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)

3) Dynamics of the SMS

How activity moves Bidirectionally between sensory and motor units

4) Value Field Dynamics

(in a goal directed fashion)

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + Q$$

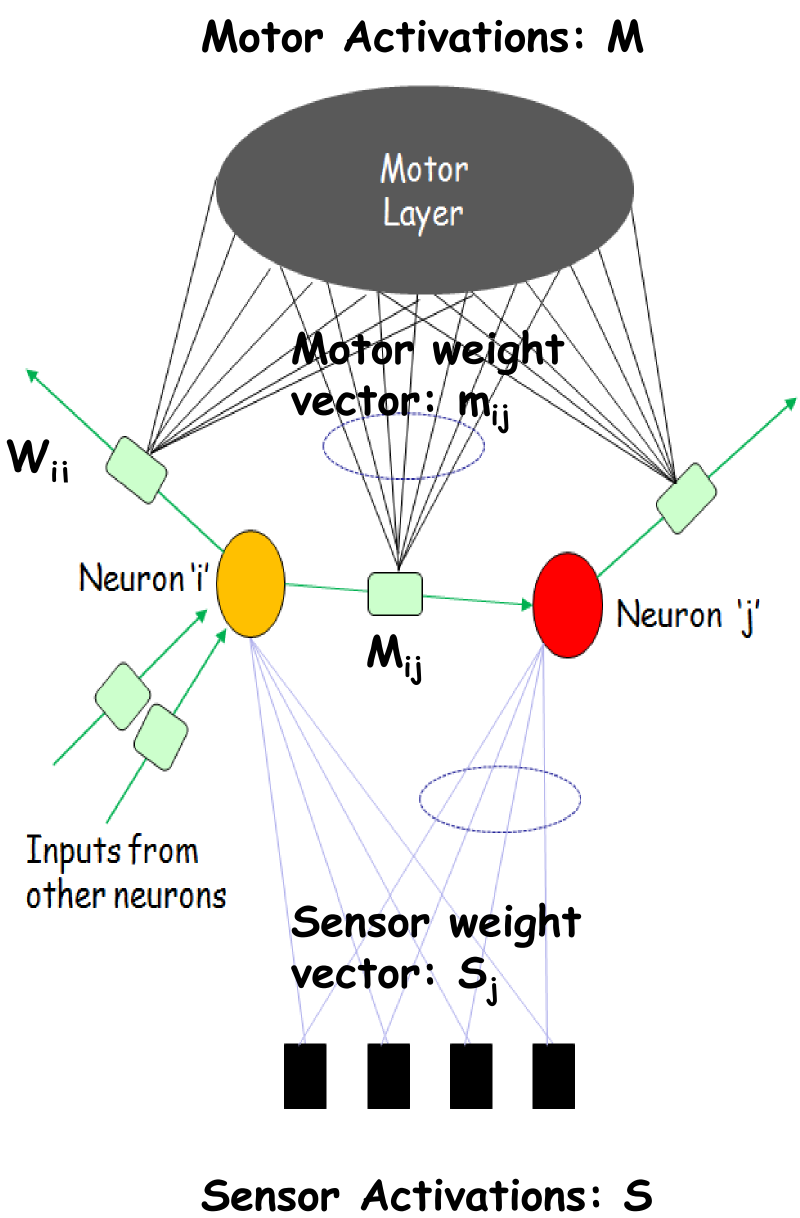
Default Plan (if any) + 'weighted Superposition' of new learnt reward fields in SMS

$$Q = q_1 + q_2 + ... + q_n$$

$$q_i = \varphi_n . U_i . \frac{1}{\sqrt{2\pi\sigma_G}} e^{\frac{-(G-Gi)^2}{2\sigma_G{}^2}}$$

How much value a good/bad experience encountered in the past while performing a goal G_i holds in relation to the currently active goal G.

Dynamics: On moving in the sensorimotor space in a goal directed way



Sensorimotor space dynamics

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Motor space dynamics

$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}} \right) + \zeta(\bar{r} \;)$$

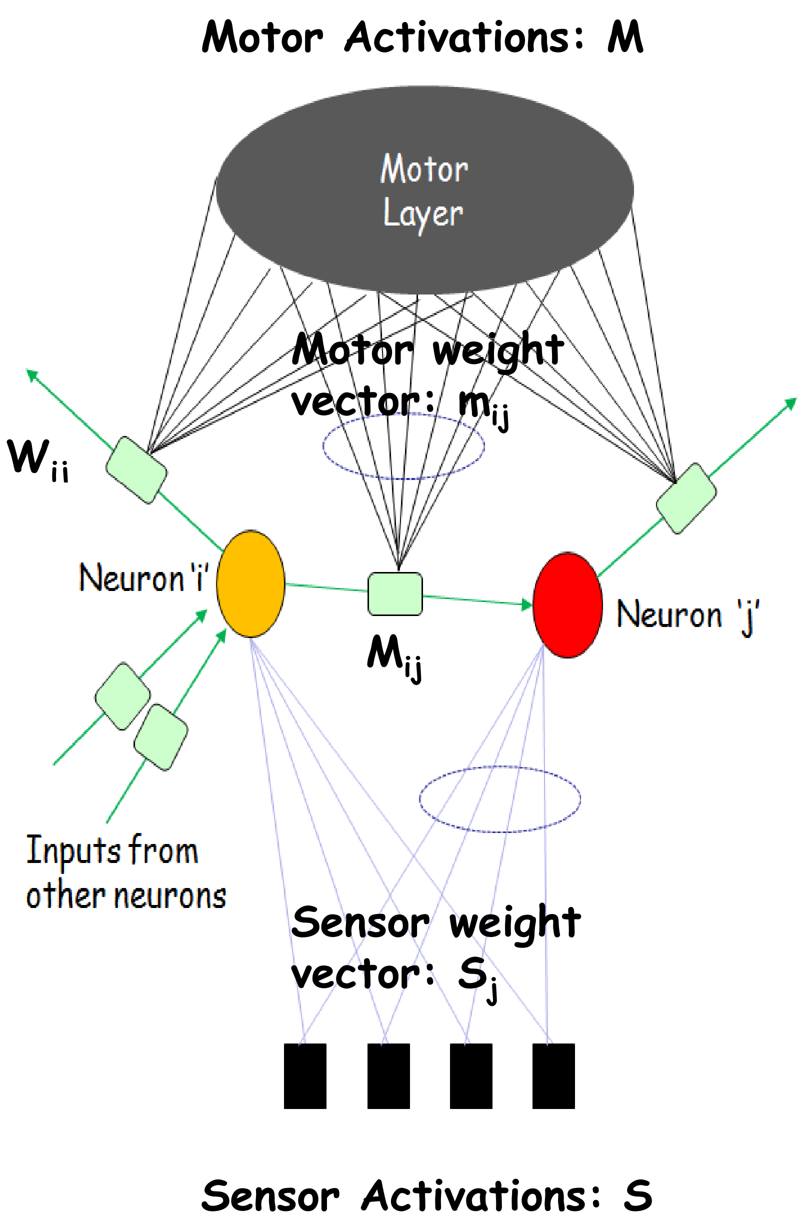
Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$\downarrow$$

$$R_i = DP$$

Dynamics: On moving in the sensorimotor space in a goal directed way



Sensorimotor space dynamics

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Motor space dynamics

$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}} \right) + \zeta(\bar{r}\ )$$

Value Field dynamics

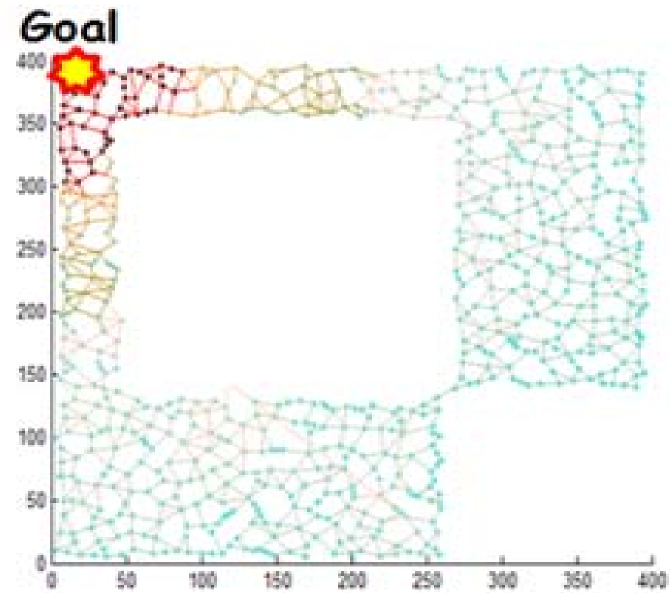
$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

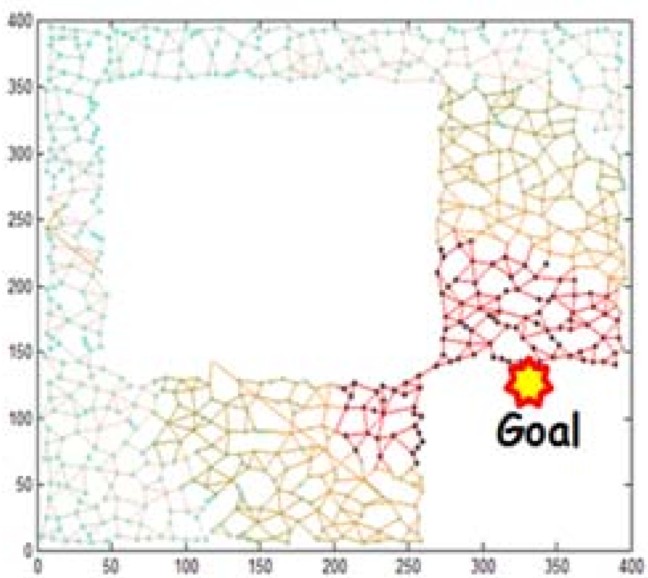
$$R_i = DP$$

$$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R^2}}$$

Simply, if we want to reach a goal = 5, neuron representing a state '4' will fetch greater reward than neuron represneting a state '-1'

Coupling between the value field and the dynamics of the SMS

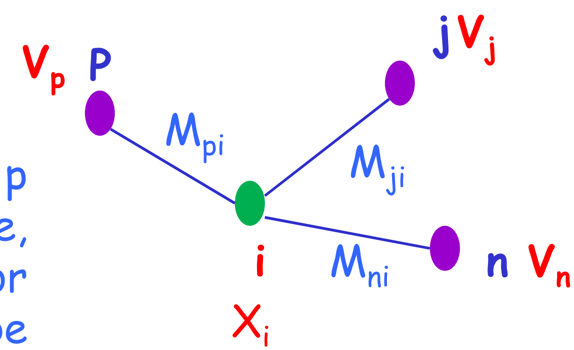




Goal $\longrightarrow$ $$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R^2}}$$

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$v_i^* = R_i + \gamma (W_{ij} v_j)_{\max}$$

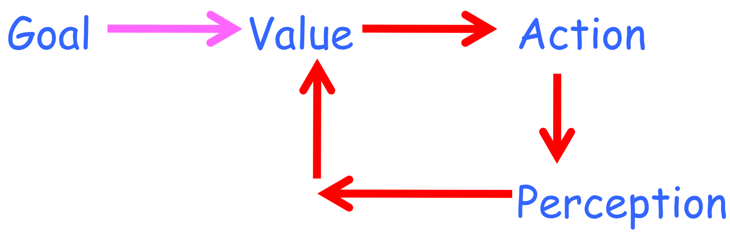


If wrt neuron i, neuron p holds maximum value, next incremental motor action that needs to be generated is M_{pi}

$$\overline{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}} \right) + \zeta (\overline{r}\ \ )$$

$k_i = \mathrm{argmax}_j (w_{ij} V_j)$

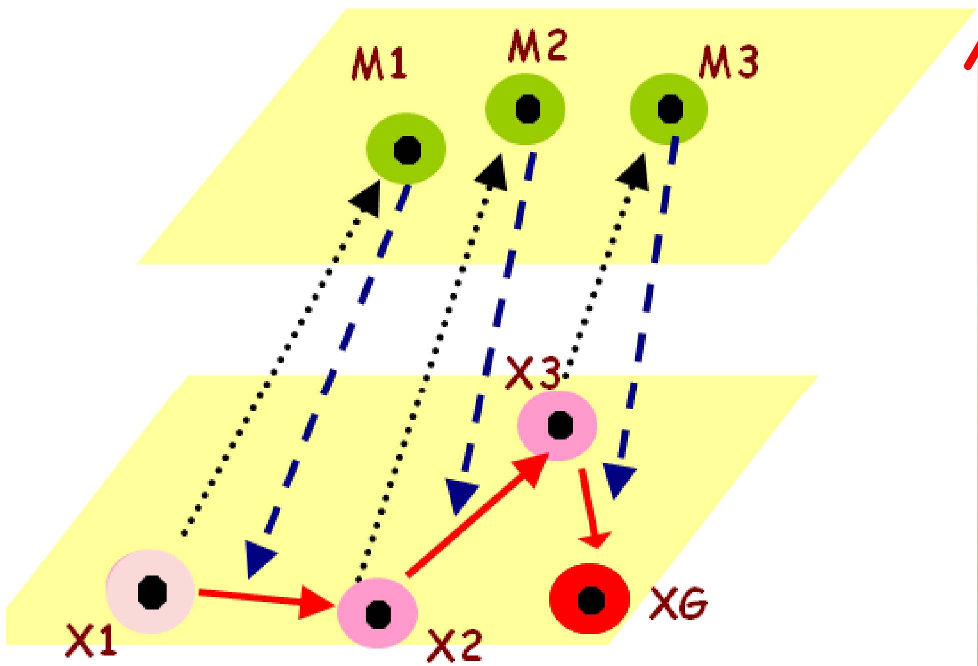
Value field influences the motor activity by determining the neighboring neuron that holds maximum value with respect to the currently active goal.



Goal

Value Field on SMS (Quasistationary)

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$



Value to action

$$k_i = \mathrm{argmax}_j (w_{ij} V_j)$$

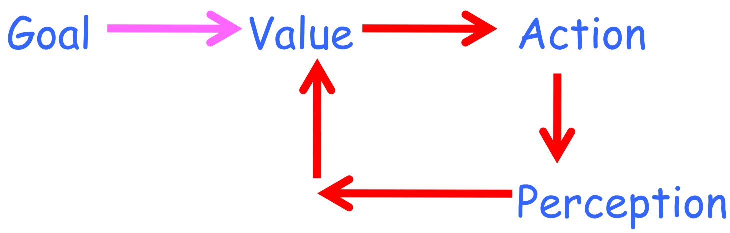
$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}}\right) + \zeta(\bar{r})$$

Motor Activity modulates lateral connections in SMS

$$M_{ij} = \langle m_{ij}, M \rangle$$

Activity shift in SMS

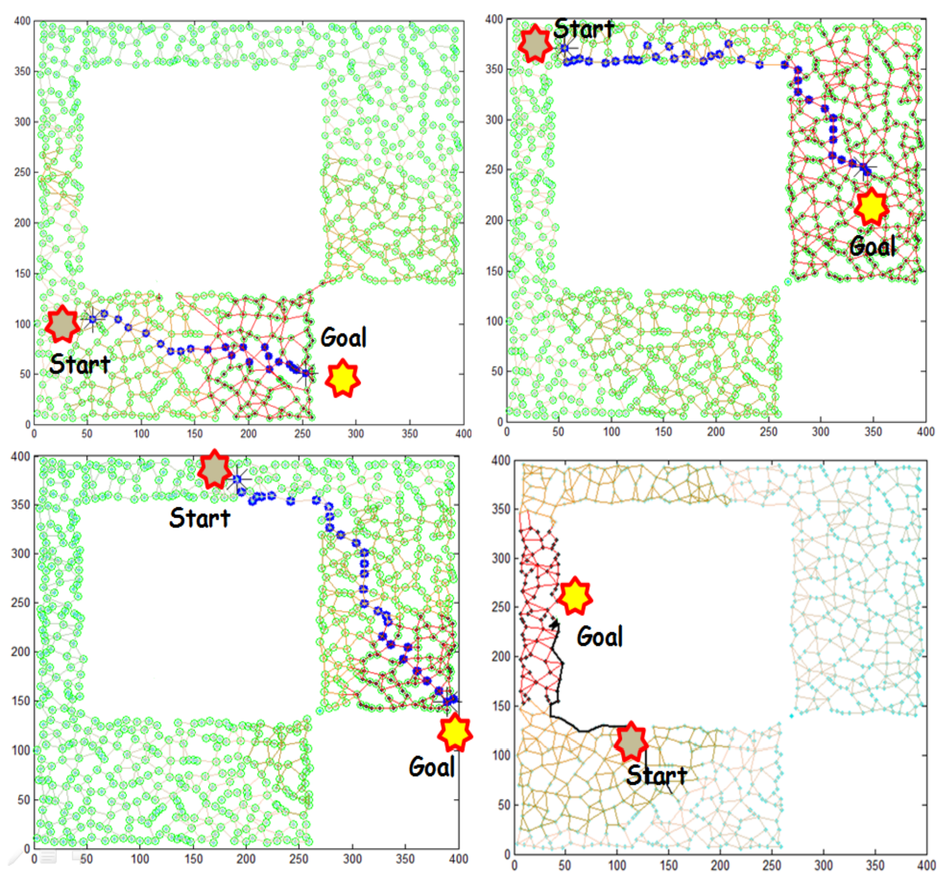
$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$



Goal

Value Field on SMS (Quasistationary)

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$



Value to action

$$k_i = \mathrm{argmax}_j (w_{ij} V_j)$$

$$\bar{a} = \beta_{if} \left(\sum_{i=1}^{N} x_i \overline{m_{k_i i}} \right) + \zeta(\bar{r})$$

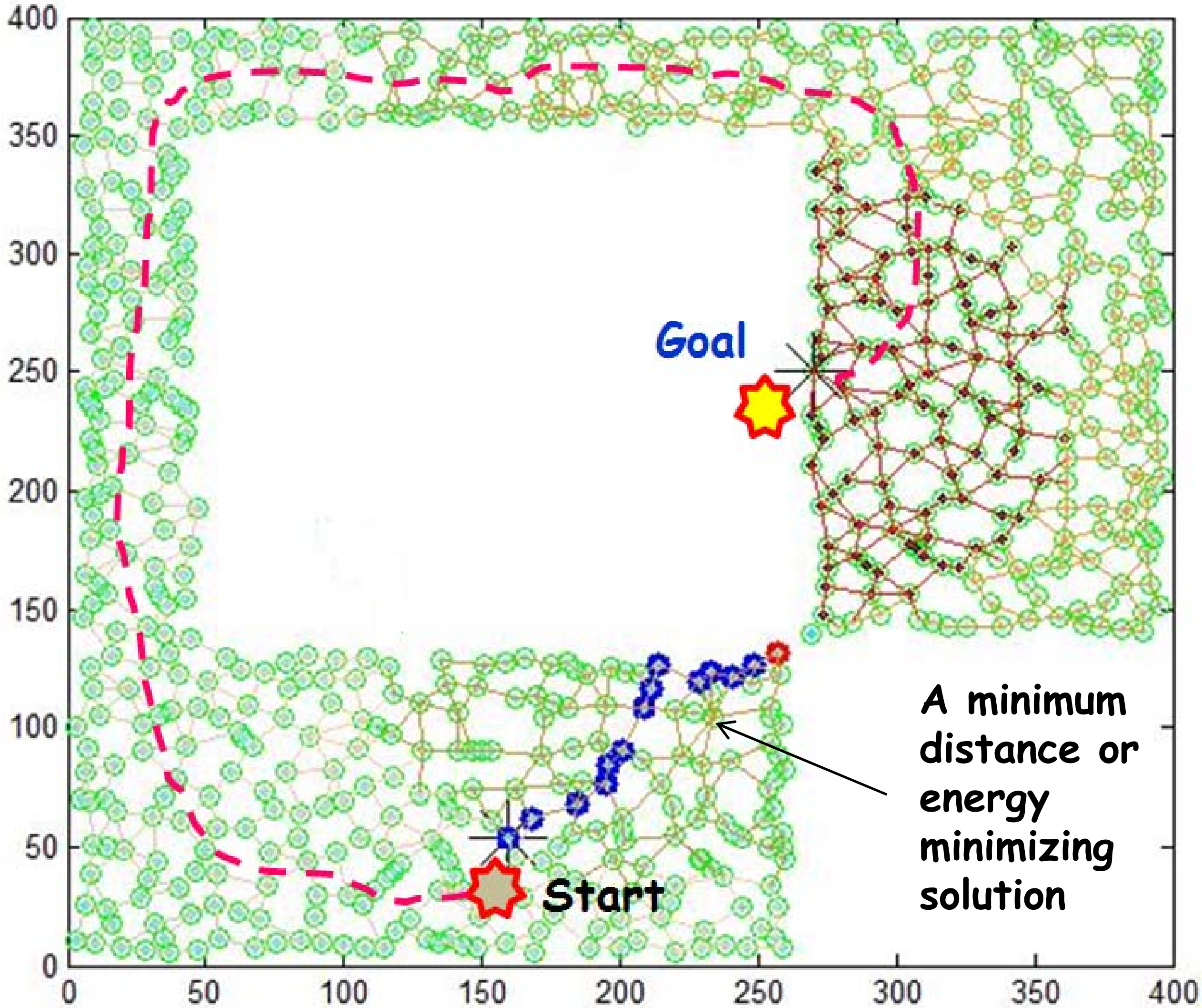
Motor Activity modulates lateral connections in SMS

$$M_{ij} =< m_{ij}, M >$$

Activity shift in SMS

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

Constraints: Learning 'when' to optimize 'what'



Value Field dynamics

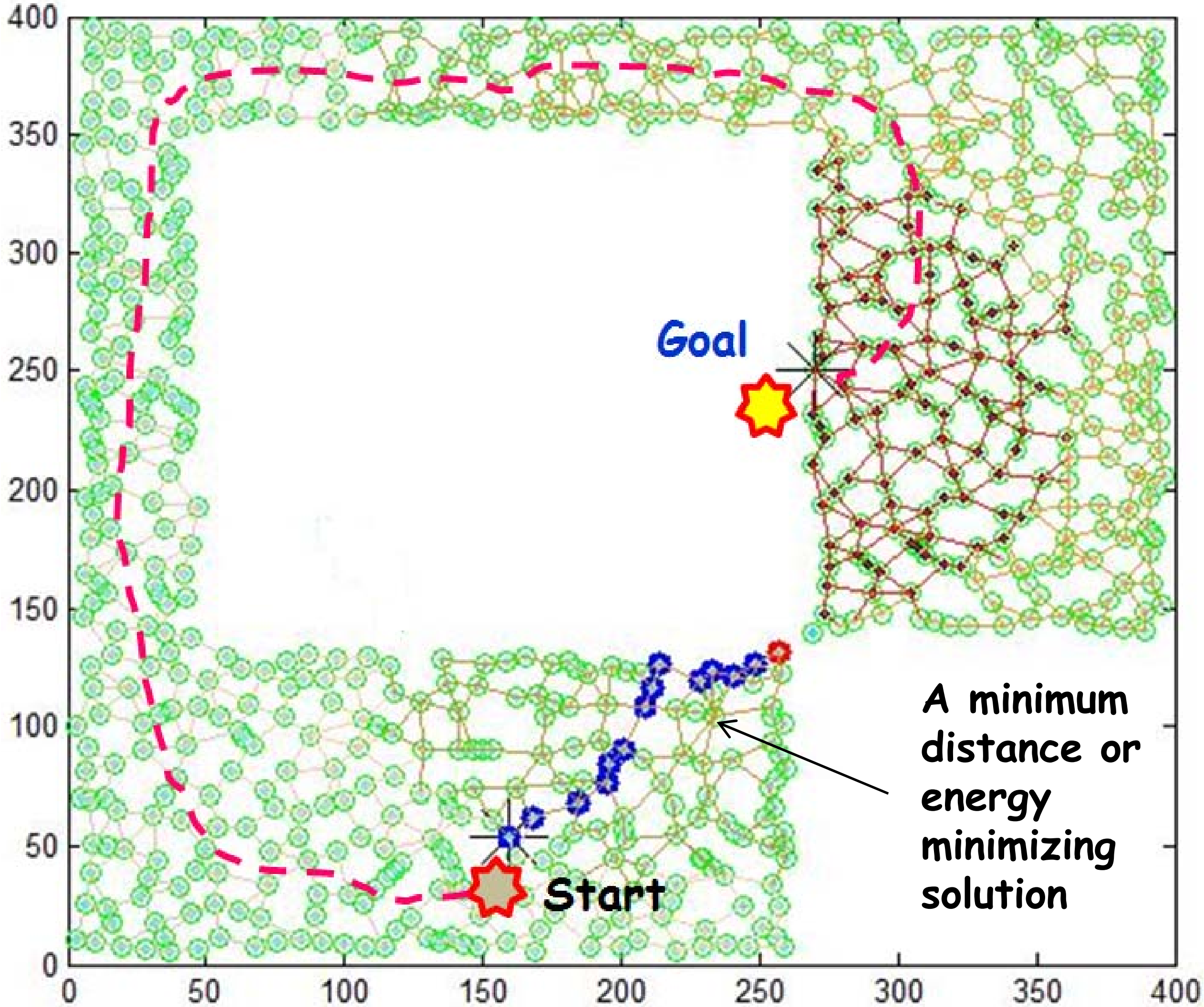
$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + Q$$

Default Plan

$$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R{}^2}}$$

Constraints: Learning 'when' to optimize 'what'



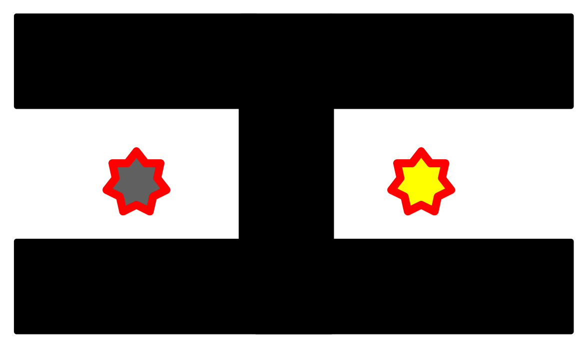
Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + Q$$

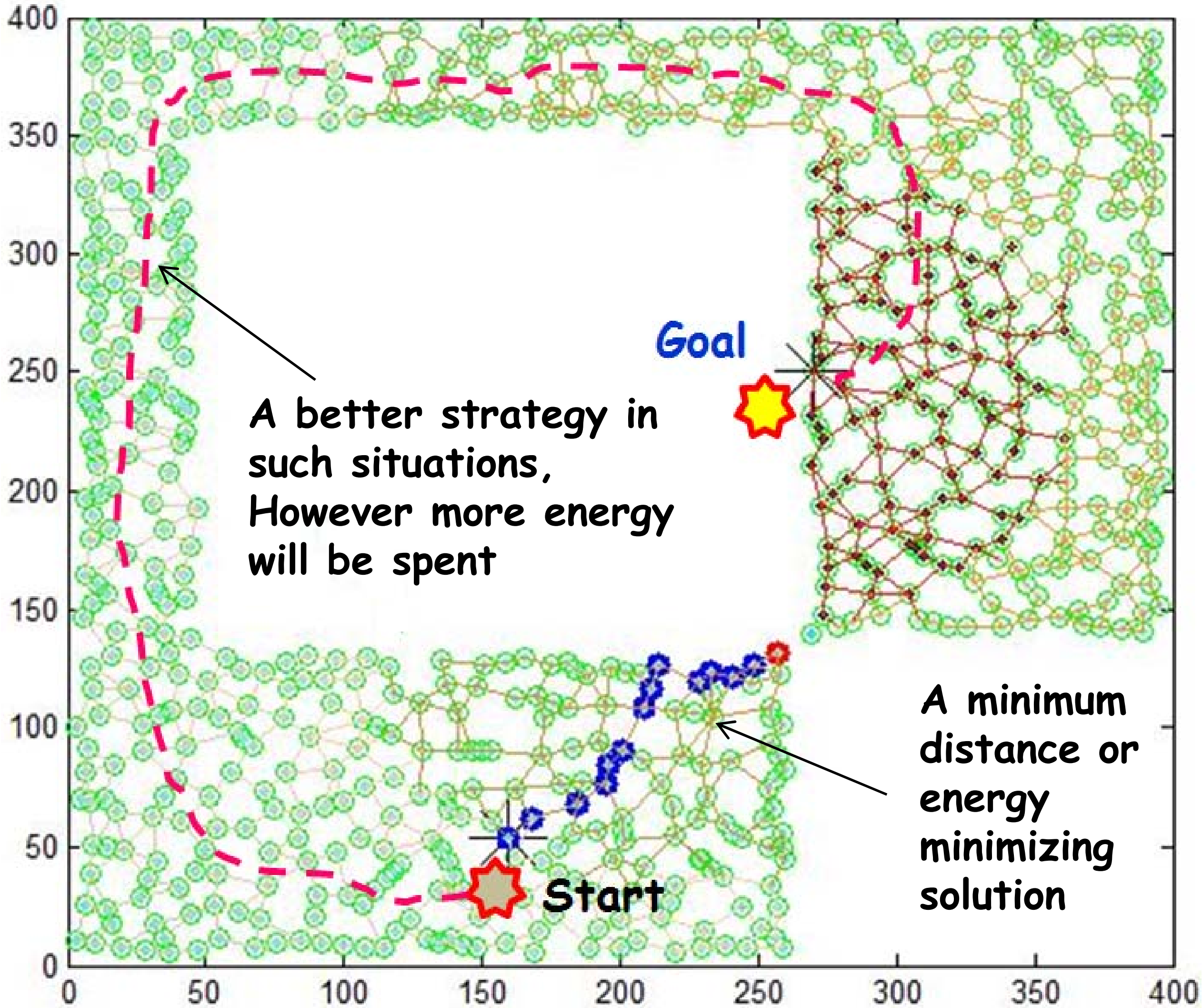
Default Plan

$$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R^2}}$$



Maze navigation problem

Constraints: Learning 'when' to optimize 'what'



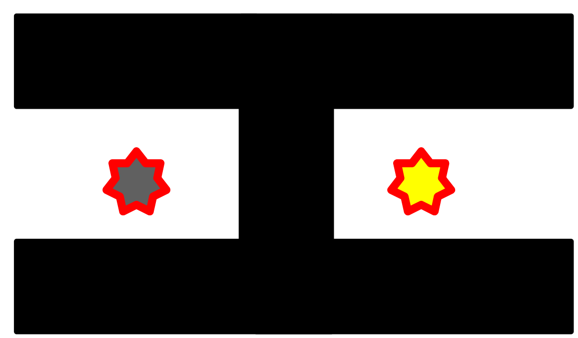
Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + Q$$

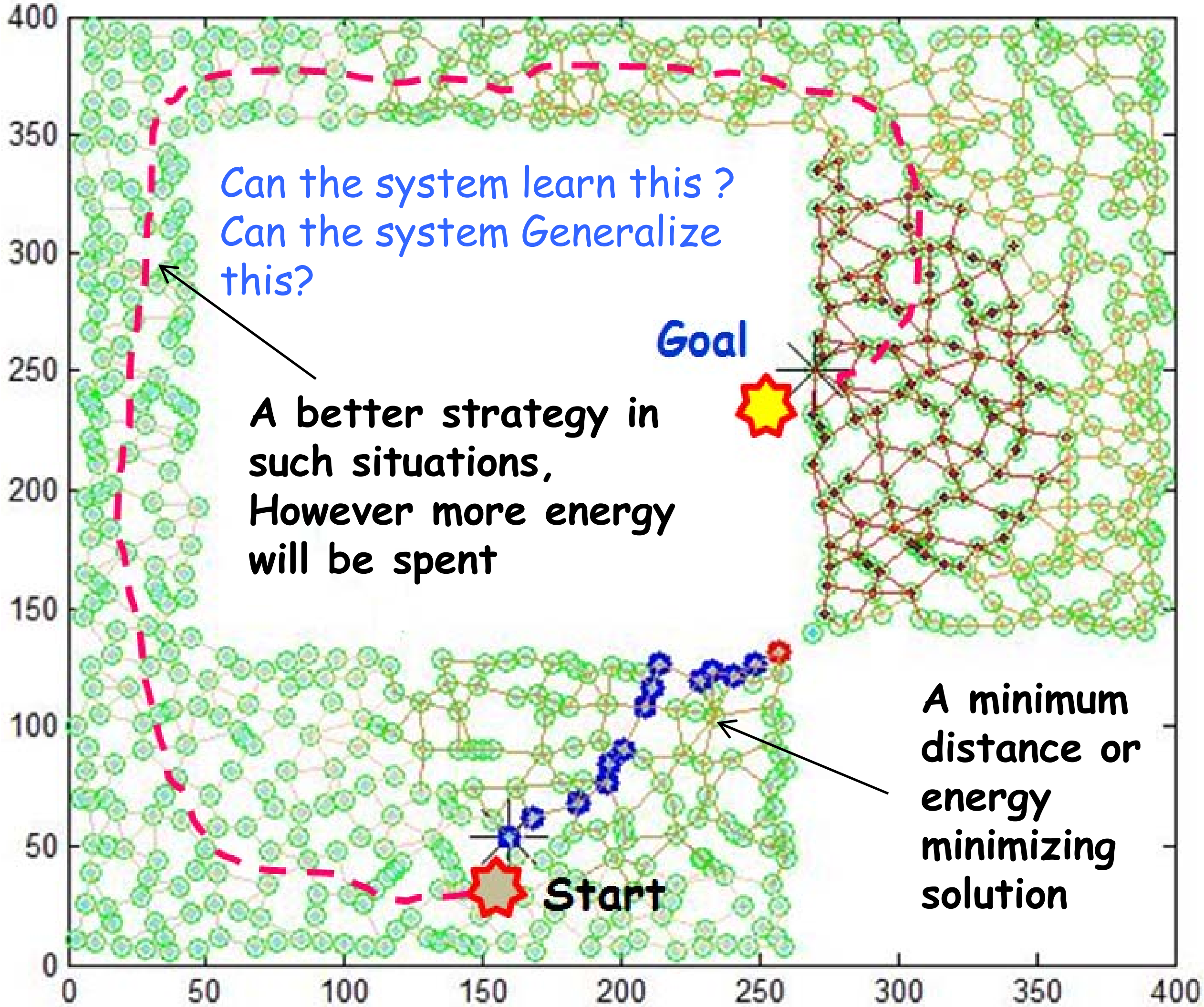
Default Plan

$$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R^2}}$$



Maze navigation problem

Constraints: Learning 'when' to optimize 'what'



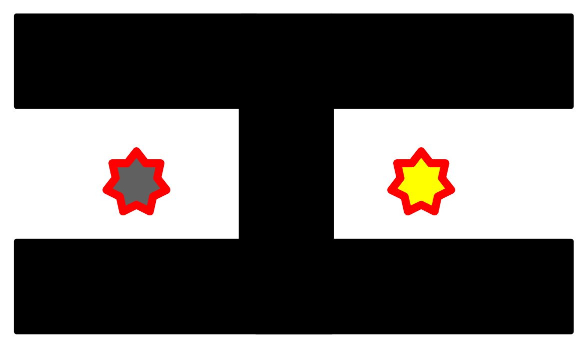
Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + \boxed{Q}$$

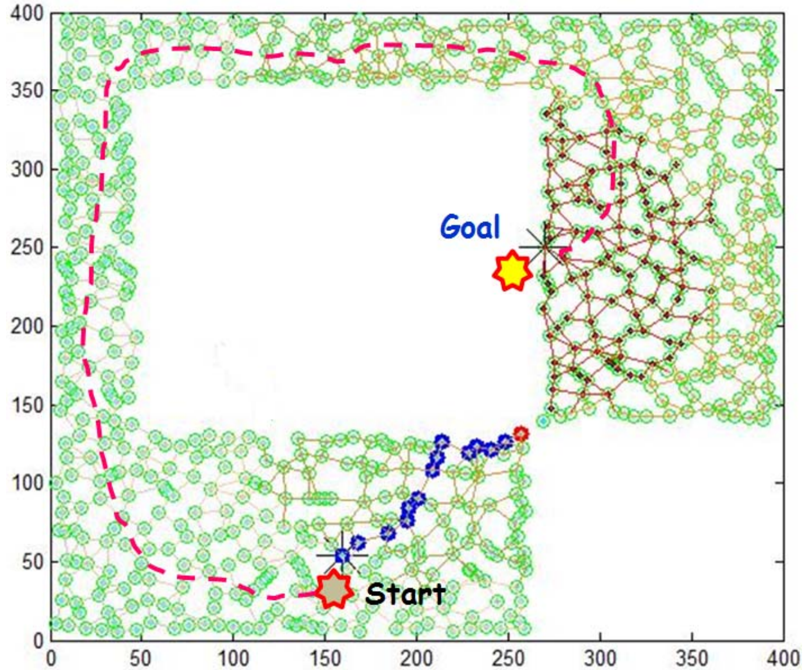
Default Plan

$$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R^2}}$$



Maze navigation problem

Constraints: Learning 'when' to optimize 'what'



Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

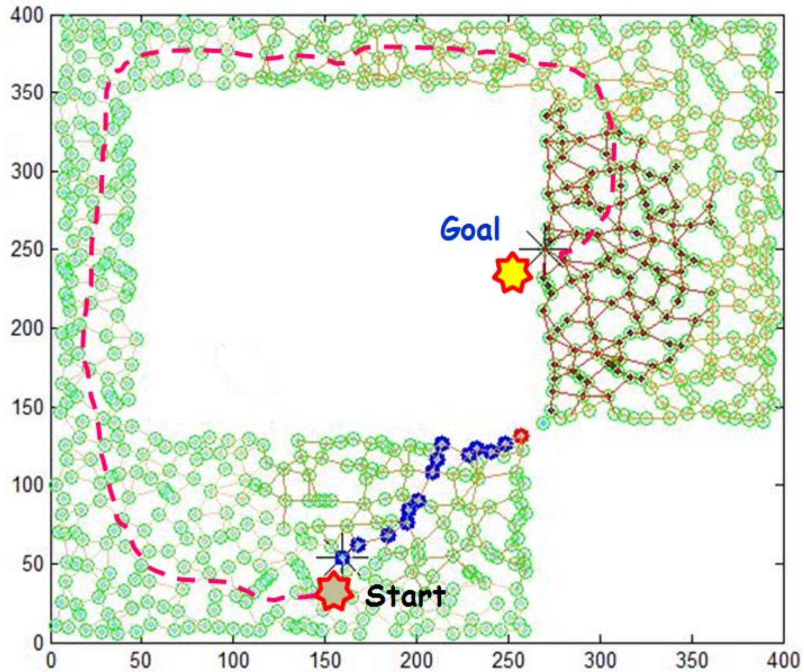
$$R_i = DP + \boxed{Q}$$

$$Q = q_1 + q_2 + \dots + q_n$$

Q is a superposition of a set of learnt 'Experience ' fields q1, q2… qn

Every individual component 'q_i' has a scalar value on every neuron in the SMS.

Constraints: Learning ‘when’ to optimize ‘what’



Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + \boxed{Q}$$

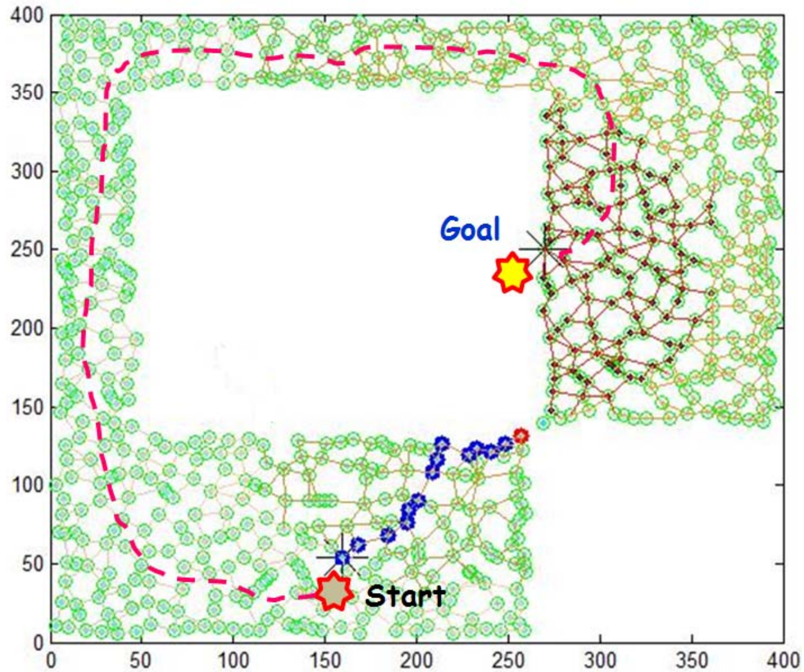
$$Q = q_1 + q_2 + \ldots + q_n$$

$$q_i = \varphi_n . U_i . \frac{1}{\sqrt{2\pi}\sigma_G} e^{\frac{-(G-Gi)^2}{2\sigma_G^2}}$$

U_i is the i^{th} interactive or self penalization/reward given to the system.

For example, a penalty of -5 is given to a bad solution (during i^{th} trial)

Constraints: Learning 'when' to optimize 'what'



Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + \boxed{Q}$$

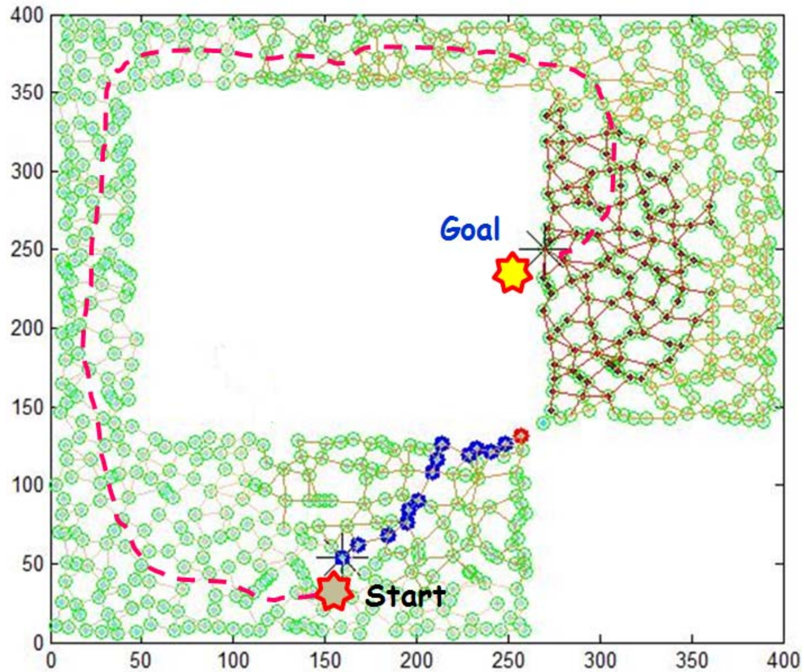
$$Q = q_1 + q_2 + \dots + q_n$$

$$q_i = \varphi_n . U_i . \frac{1}{\sqrt{2\pi}\sigma_G} e^{\frac{-(G-Gi)^2}{2\sigma_G^2}}$$

Scaling term for the n^th neuron

(Penalty U_i is given only at the end, and we need to distribute the punishments and rewards to other neurons in some way)

Constraints: Learning 'when' to optimize 'what'



Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + \boxed{Q}$$

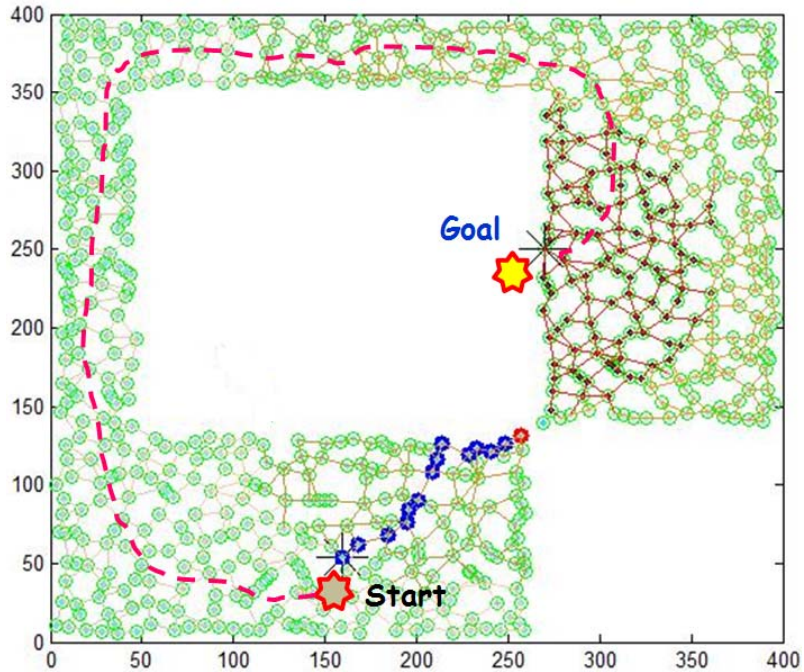
$$Q = q_1 + q_2 + \dots + q_n$$

$$q_i = \varphi_n . U_i . \frac{1}{\sqrt{2\pi}\sigma_G} e^{\frac{-(G-Gi)^2}{2\sigma_G^2}}$$

The final term encodes the dependency of the goal, the solution for which the agent was rewarded/penalized.

This term allows the system to generalize the presence/absence of value field q_i for other goals.

Constraints: Learning 'when' to optimize 'what'



Value Field dynamics

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

$$R_i = DP + \boxed{Q}$$

$$Q = q_1 + q_2 + \dots + q_n$$

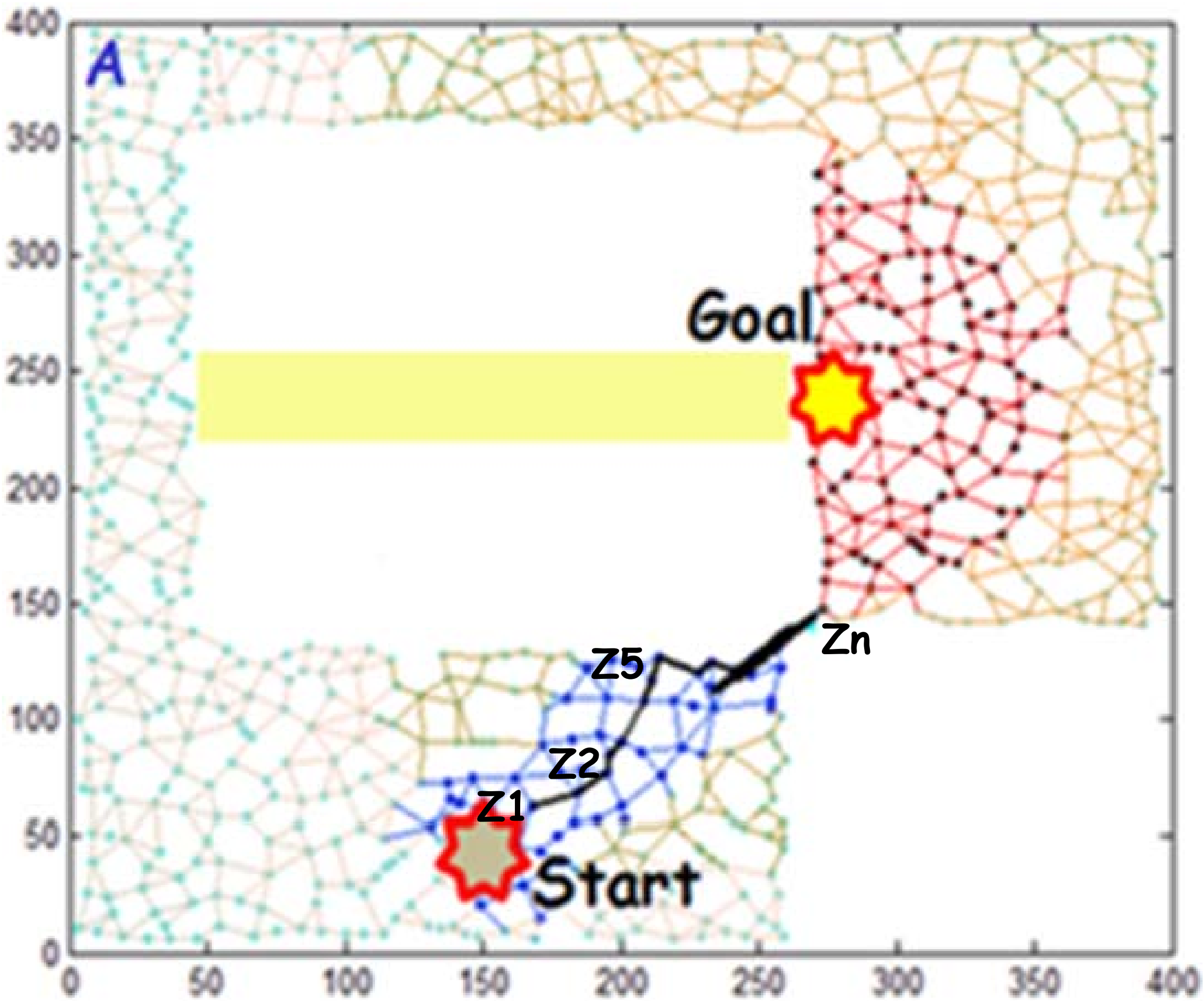
$$q_i = \varphi_n . U_i . \frac{1}{\sqrt{2\pi}\sigma_G} e^{\frac{-(G-Gi)^2}{2\sigma_G{}^2}}$$

This term evaluates how much relevance a good/bad experience encountered in the past while performing a goal G_i (for which the additional field q_i was learnt) holds in relation to the currently active goal G.

The final term encodes the dependency of the goal, the solution for which the agent was rewarded/penalized.

This term allows the system to generalize the presence/absence of value field q_i for other goals.

The first solution (using just the DP component of the reward)



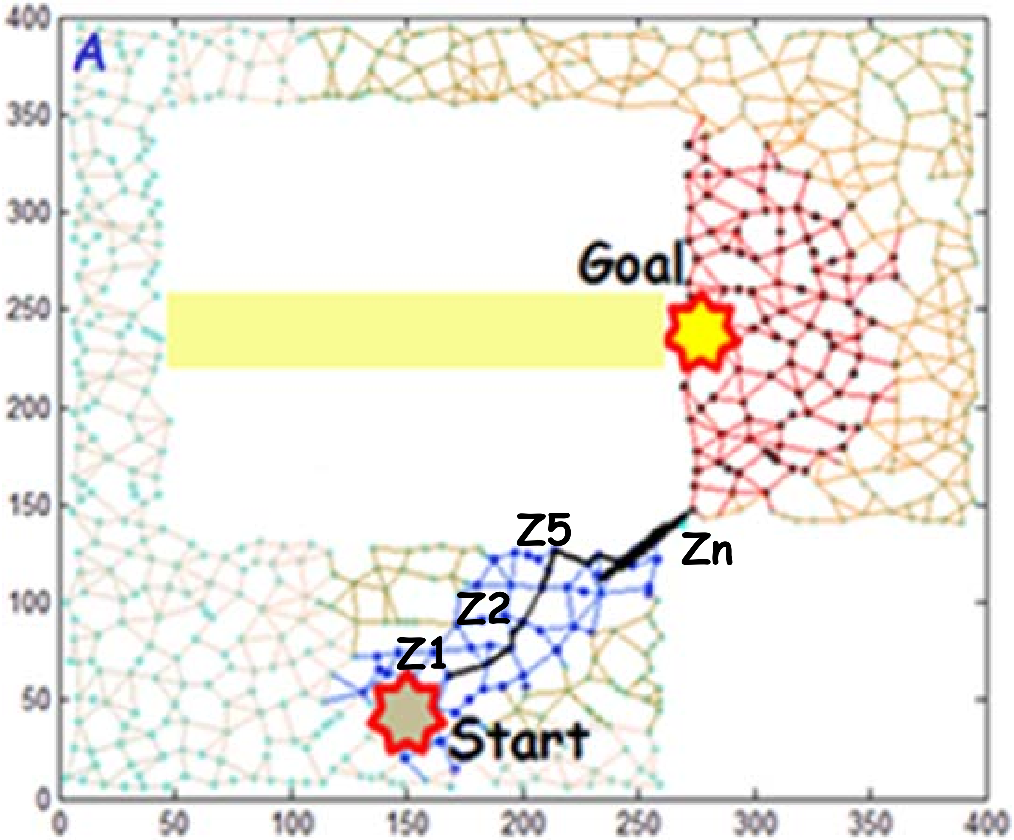
High dimentional State space

Need to find simple way to distribute rewards and penalties

It is always possible to keep track of the temporal sequence of neruons that fired in the SMS during the performance of behaviour.

We can also track their approximate neighbourhood
(if a lateral connection is present)

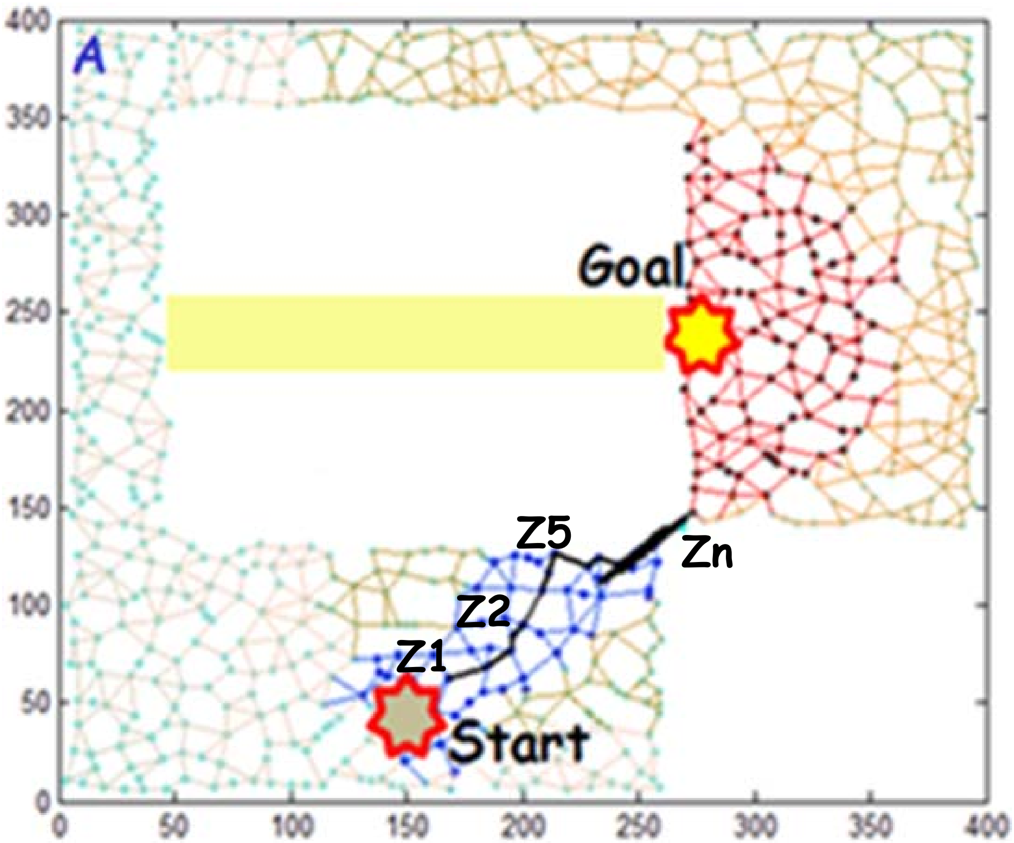
How to distribute rewards/penalties



1) In case of a penalization, the most proximal neuron z_1 receives the maximum penalty U_i, and all other neurons (z2-zn) receive scaled versions $\varphi_n . U_i$ of the penalty.

2)

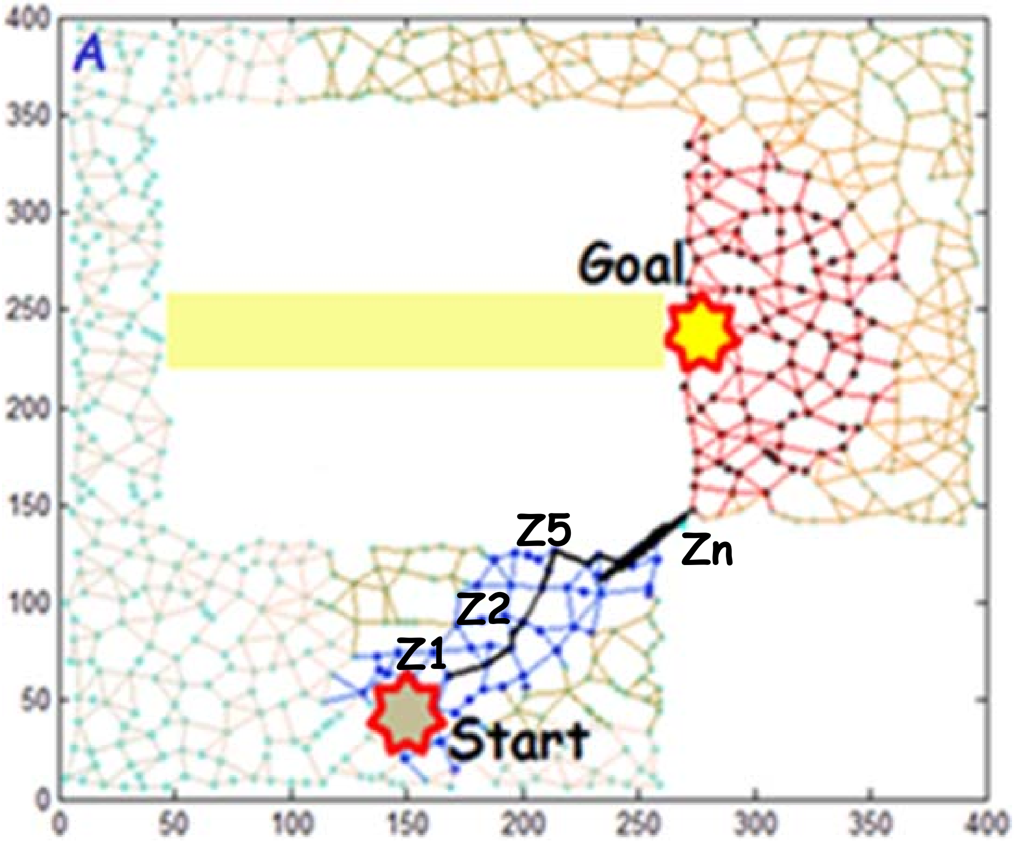
How to distribute rewards/penalties



1) In case of a penalization, the most proximal neuron z_1 receives the maximum penalty U_i, and all other neurons (z2-zn) receive scaled versions $\varphi_n.U_i$ of the penalty.

2) In case of a reward, the most distal neuron zn recives maximum reward and all others receive scaled versions

How to distribute rewards/penalties

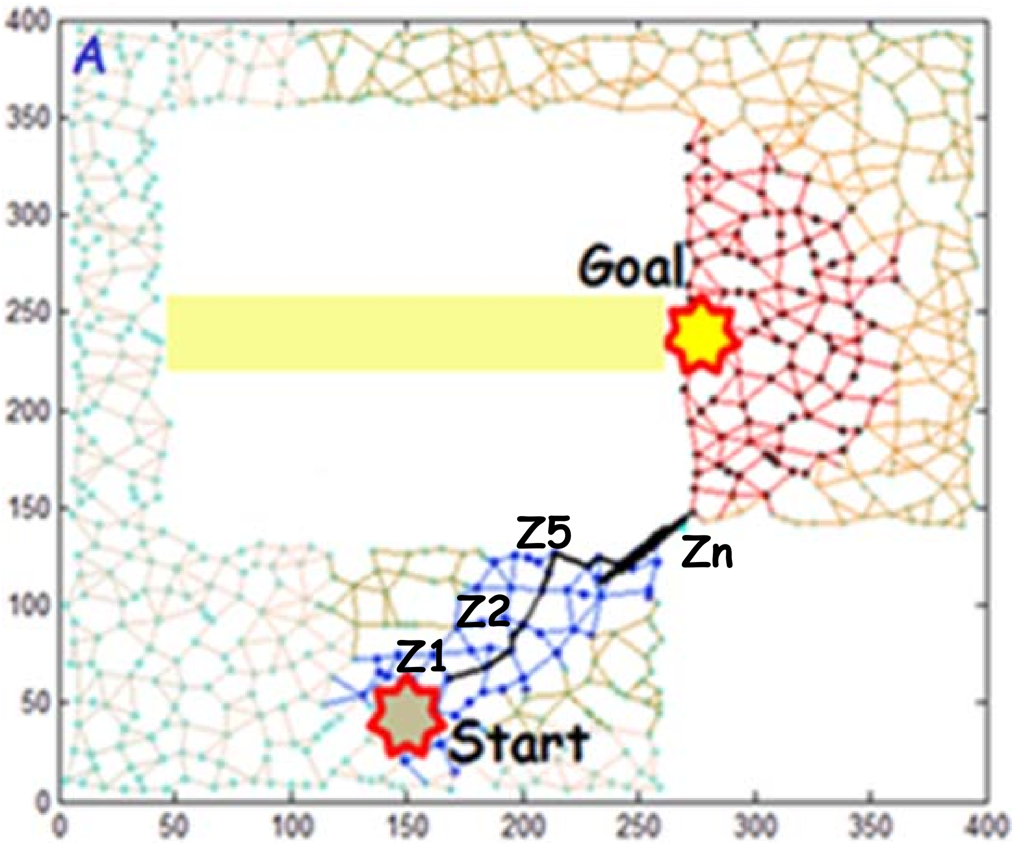


1) In case of a penalization, the most proximal neuron z_1 receives the maximum penalty U_i, and all other neurons (z2-zn) receive scaled versions $\varphi_n . U_i$ of the penalty.
2) In case of a reward, the most distal neuron zn recives maximum reward and all others receive scaled versions

Simple logic: In case of a problem or bad performance, the root is attacked

Incase of success all the contributing elements get the rewards in ways such that elements higher up in the hierarchy earn more benefits than those at the bottom.

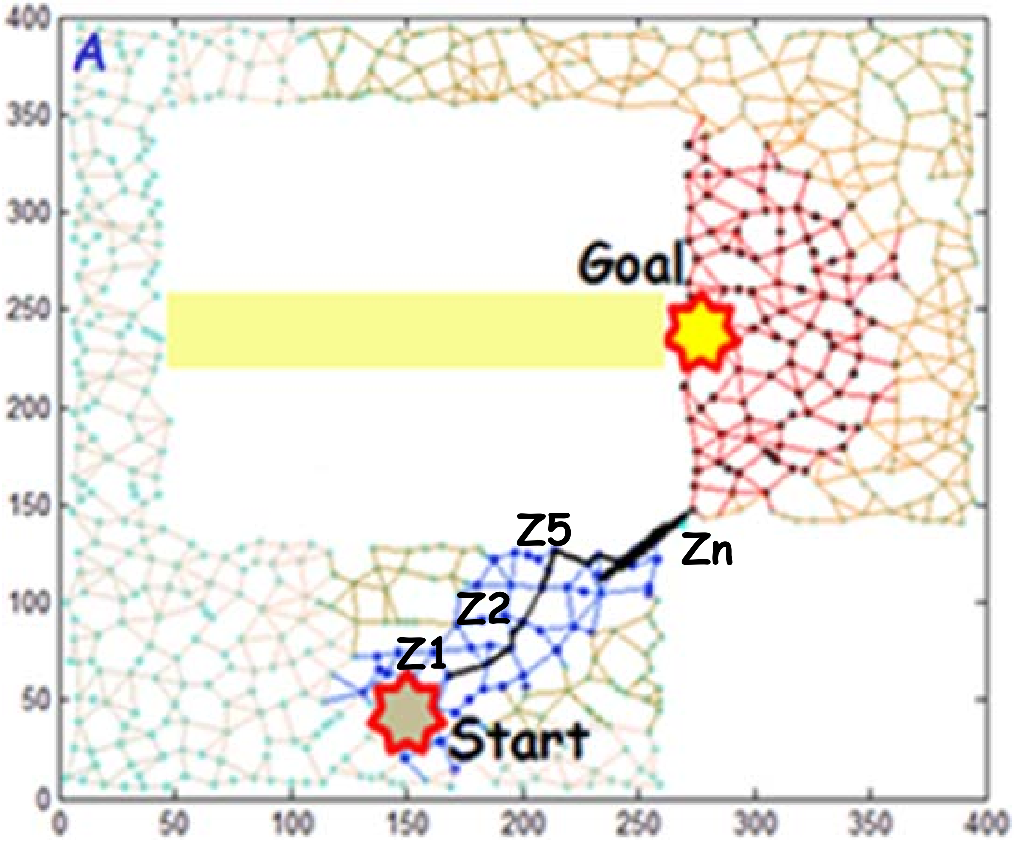
How to distribute rewards/penalties



1) In case of a penalization, the most proximal neuron z_1 receives the maximum penalty U_i, and all other neurons (z2-zn) receive scaled versions $\varphi_n . U_i$ of the penalty.
2) In case of a reward, the most distal neuron zn recives maximum reward and all others receive scaled versions

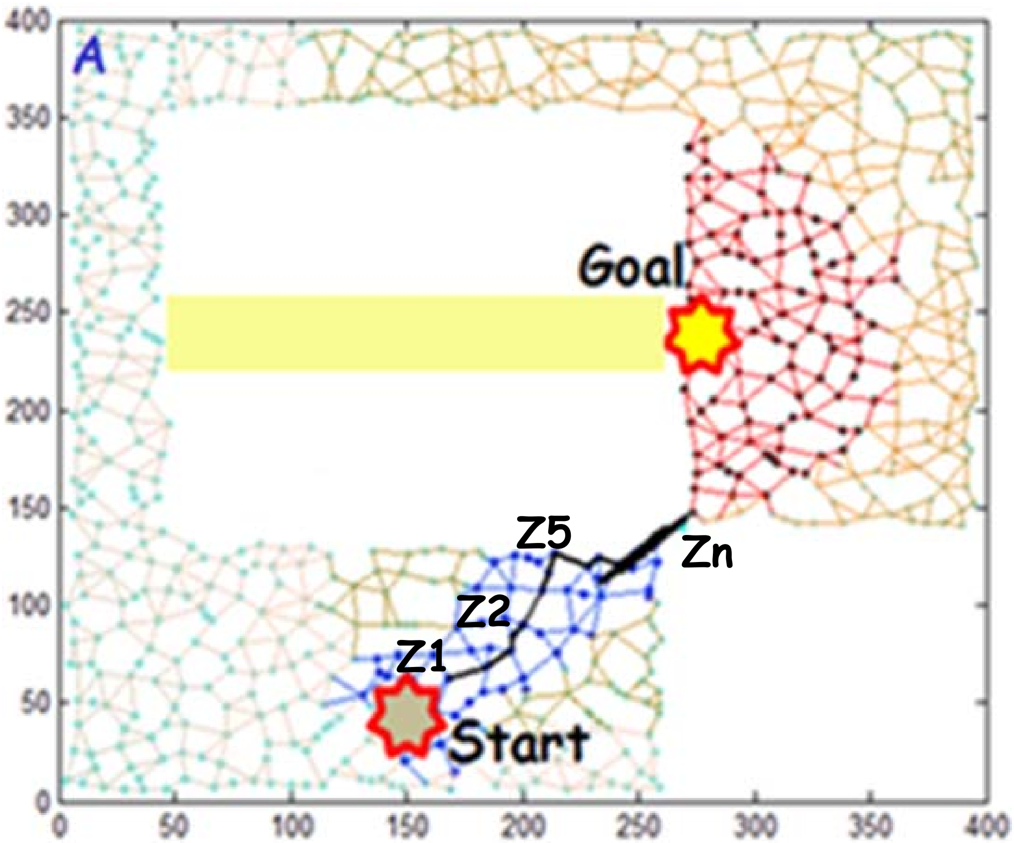
This simple heuristic of distribution of rewards underlies basic human nature of attribution of credits to any collective goal directed behavior. In case of problems, elements at the bottom of the hierarchy face maximum damage and in case of success elements at the top of the hierarchy reap maximum profits!

How to distribute rewards/penalties



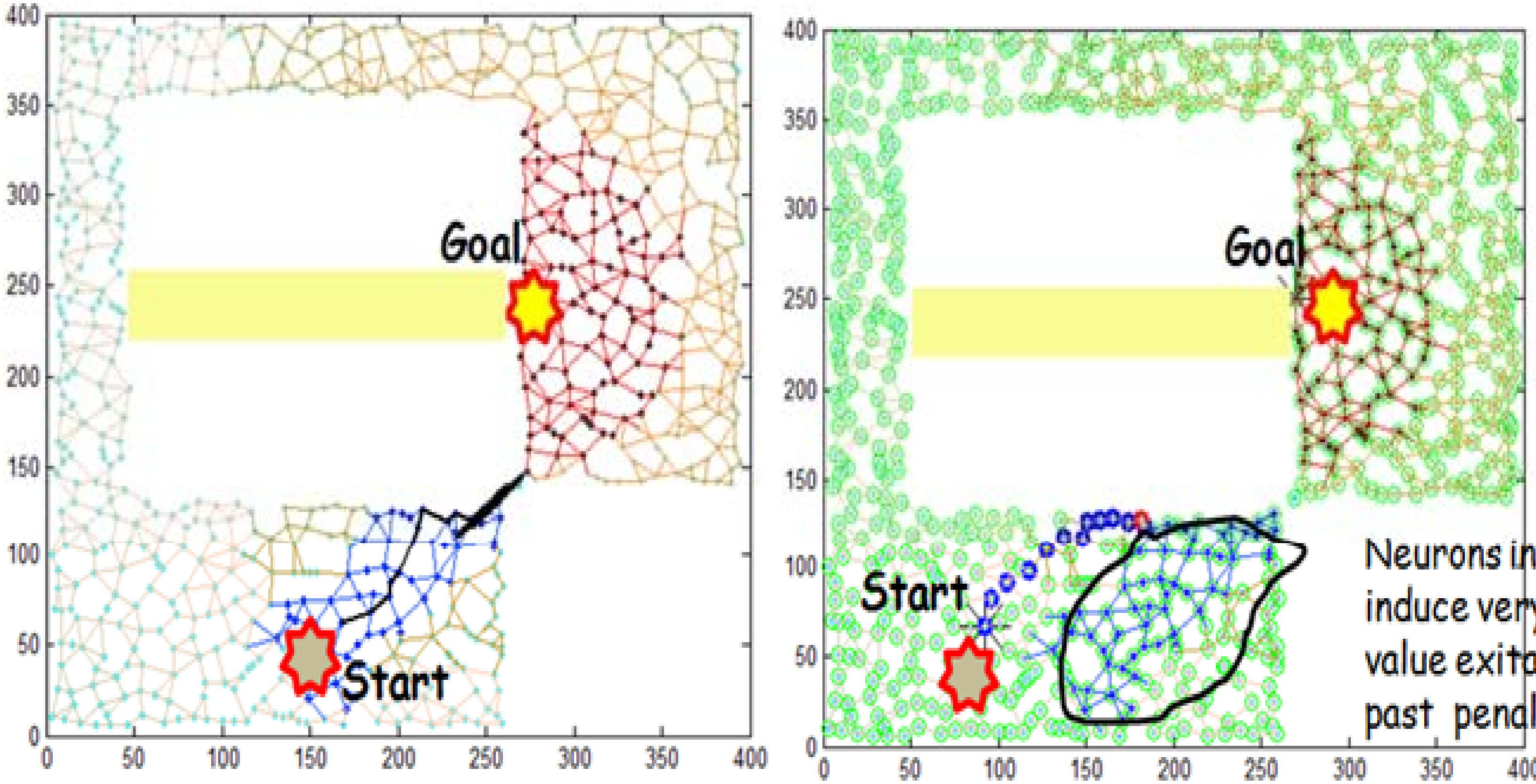
1) In case of a penalization, the most proximal neuron z_1 receives the maximum penalty U_i, and all other neurons (z2-zn) receive scaled versions $\varphi_n . U_i$ of the penalty.
2) In case of a reward, the most distal neuron zn recives maximum reward and all others receive scaled versions
3) Neighbours get scaled versions of rewards/penalty that the master got

How to distribute rewards/penalties



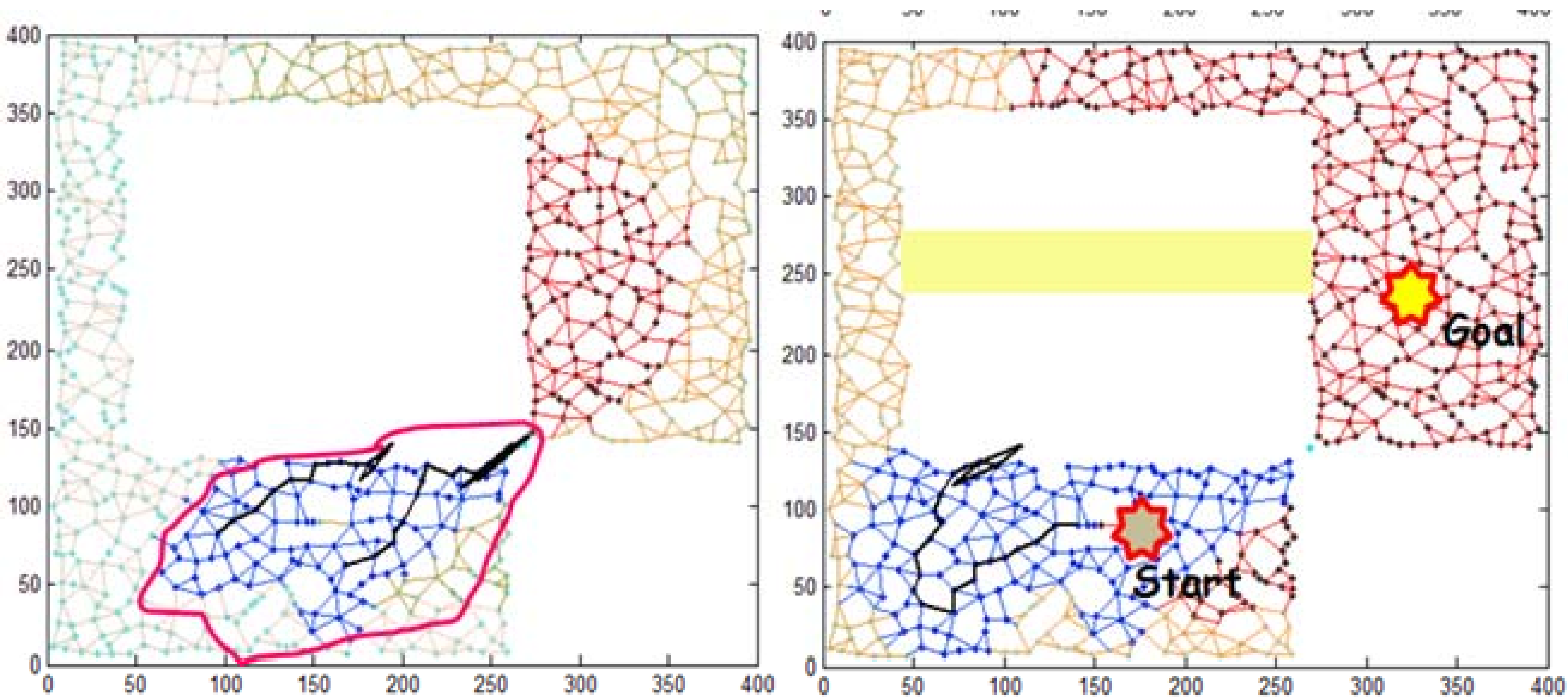
1) In case of a penalization, the most proximal neuron z_1 receives the maximum penalty U_i, and all other neurons (z2-zn) receive scaled versions $\varphi_n .U_i$ of the penalty.
2) In case of a reward, the most distal neuron zn recives maximum reward and all others receive scaled versions
3) Neighbours get scaled versions of rewards/penalty that the master got
4) Penalize circular solutions

Second solution (after first penalization)



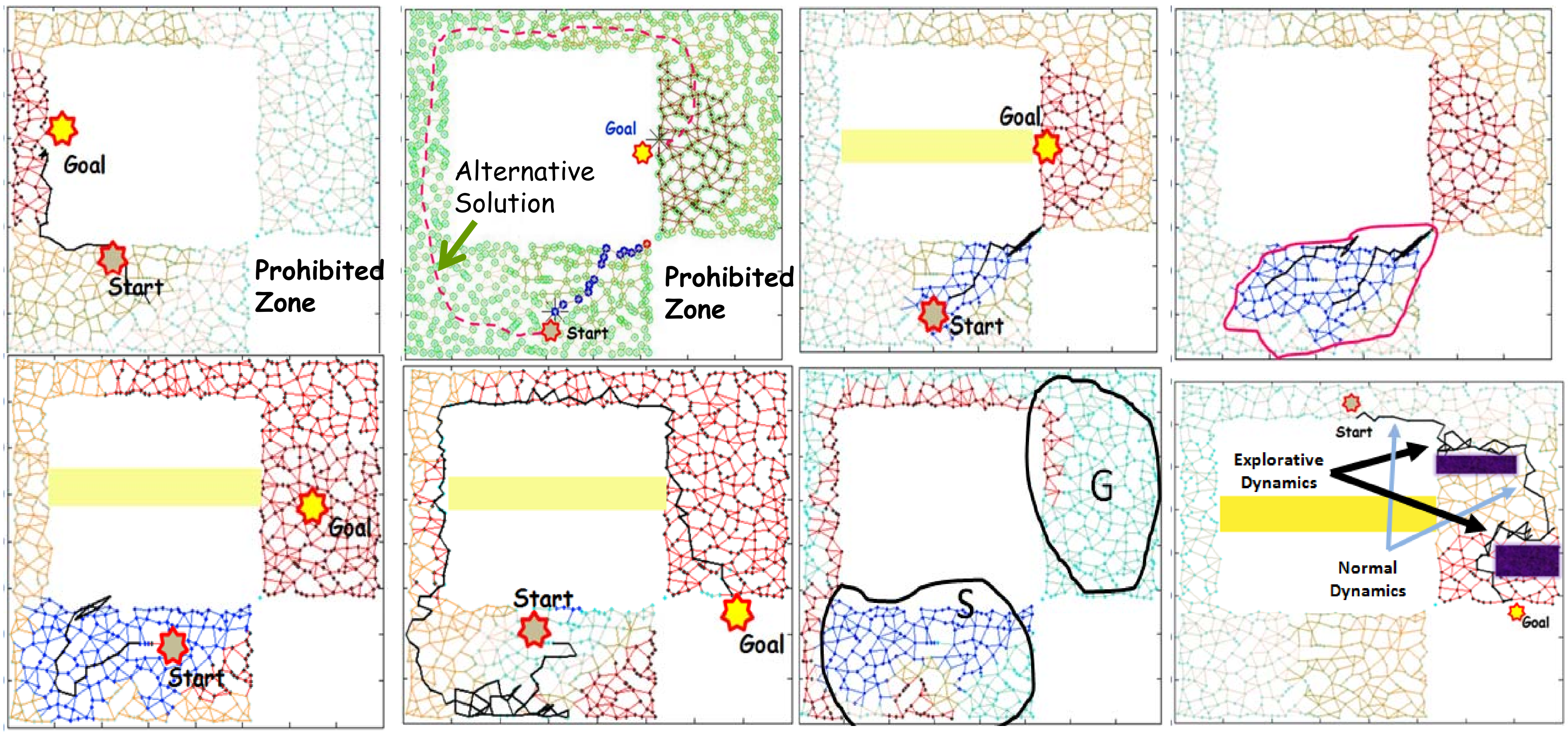
Neurons in blue region induce very less reward value exitations due to past penalization

Solutions in newer field structures



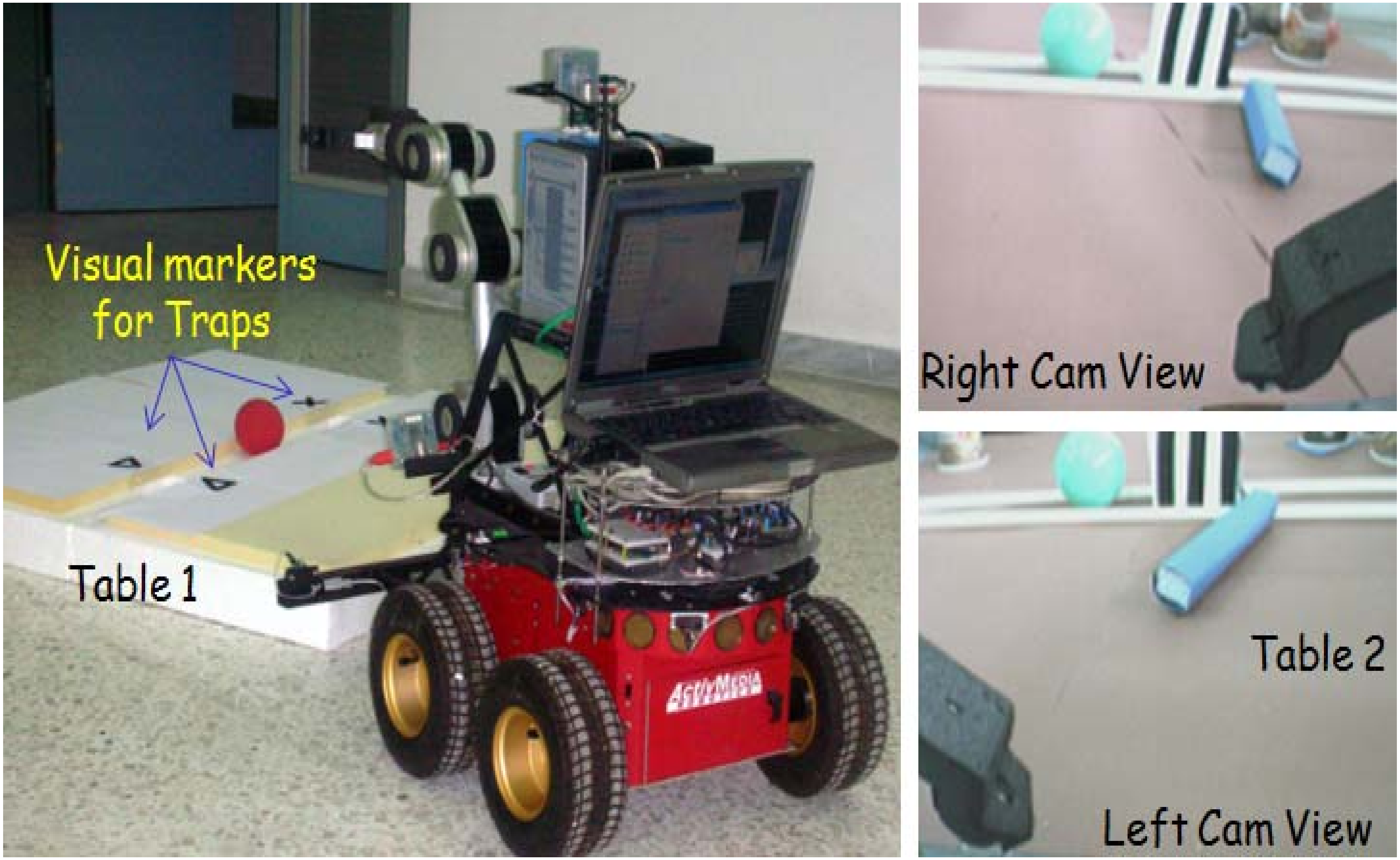
Emitting Goal directed motor sequences

A set of 'weighted' Value fields: Learnt, Adapted

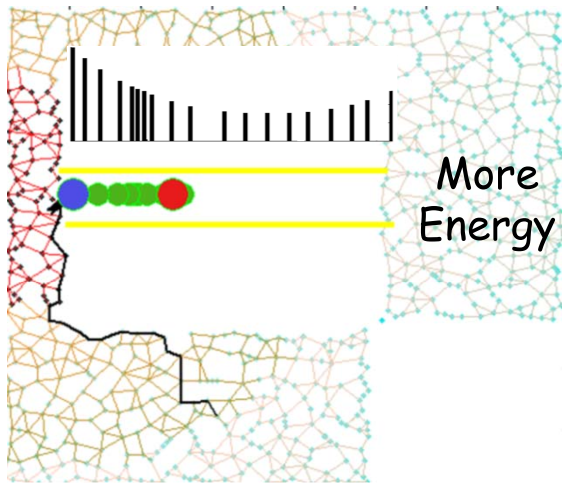


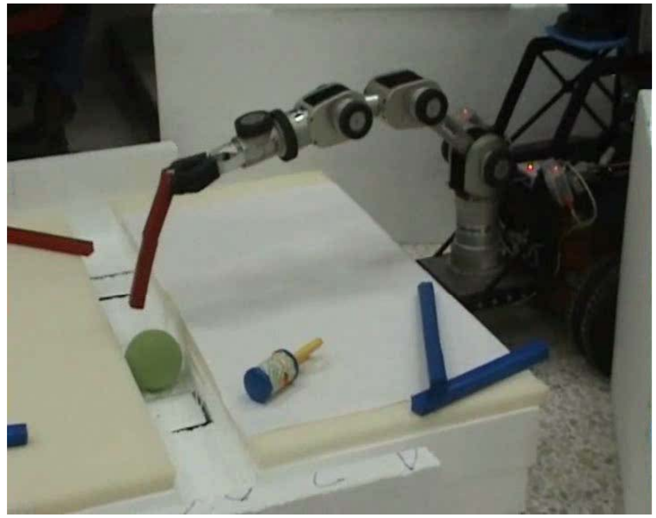
How relevant a past experience is in the context of current goal

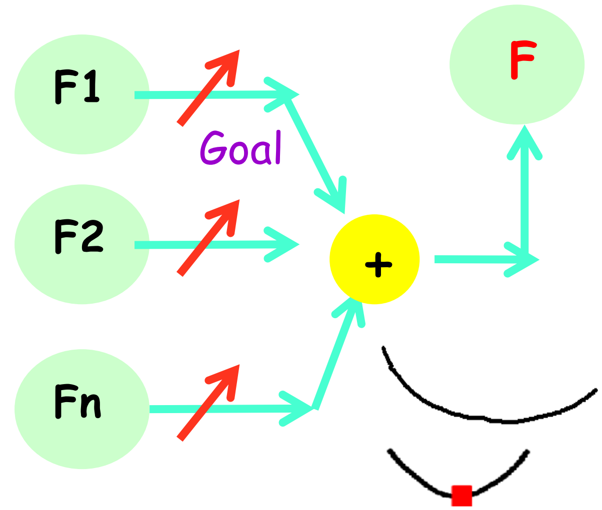
New Experiences:Learning to Avoid Traps



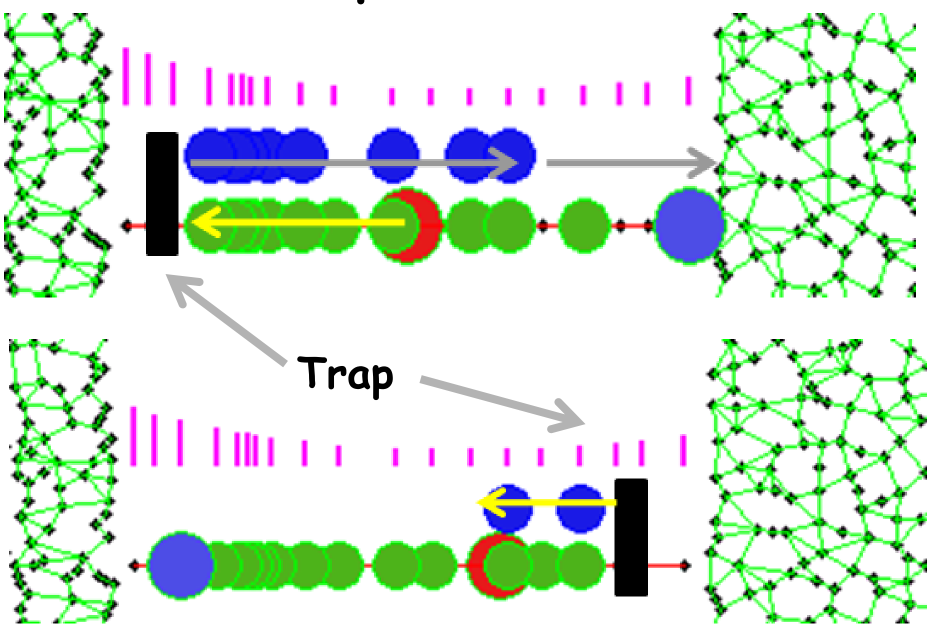
PUSHING : Learning new value fields by exploration



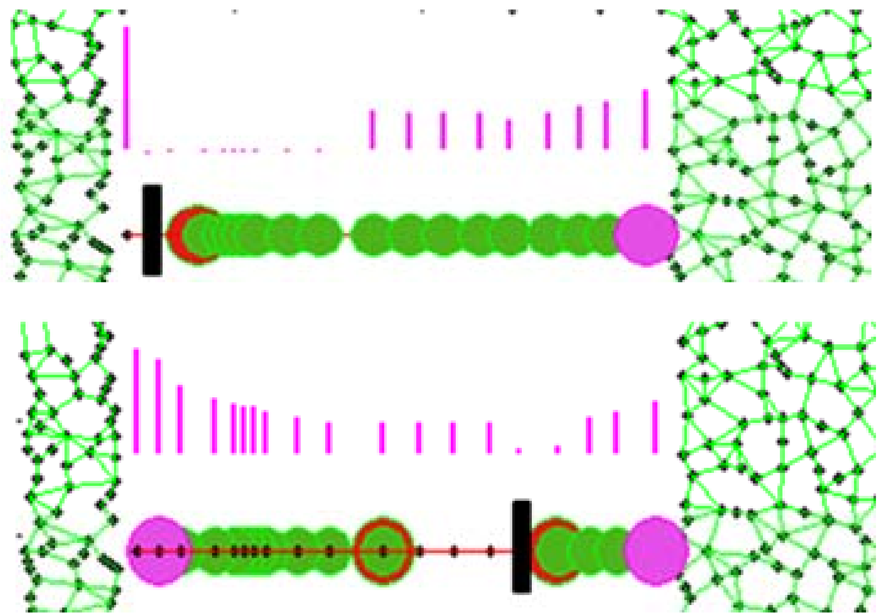


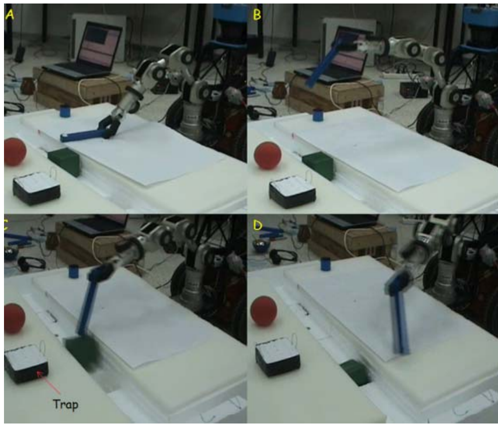


Experience

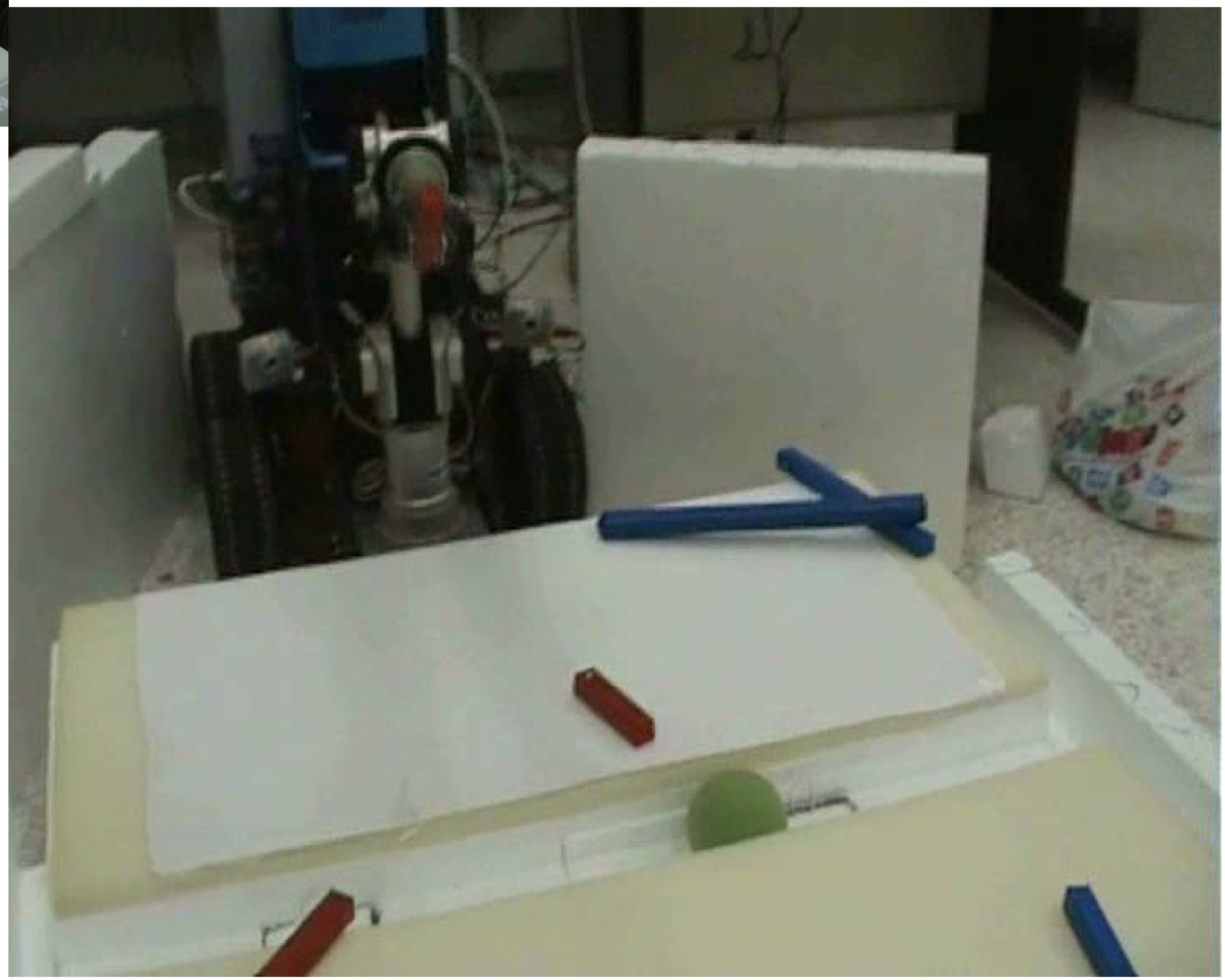


Trap Dependent field change

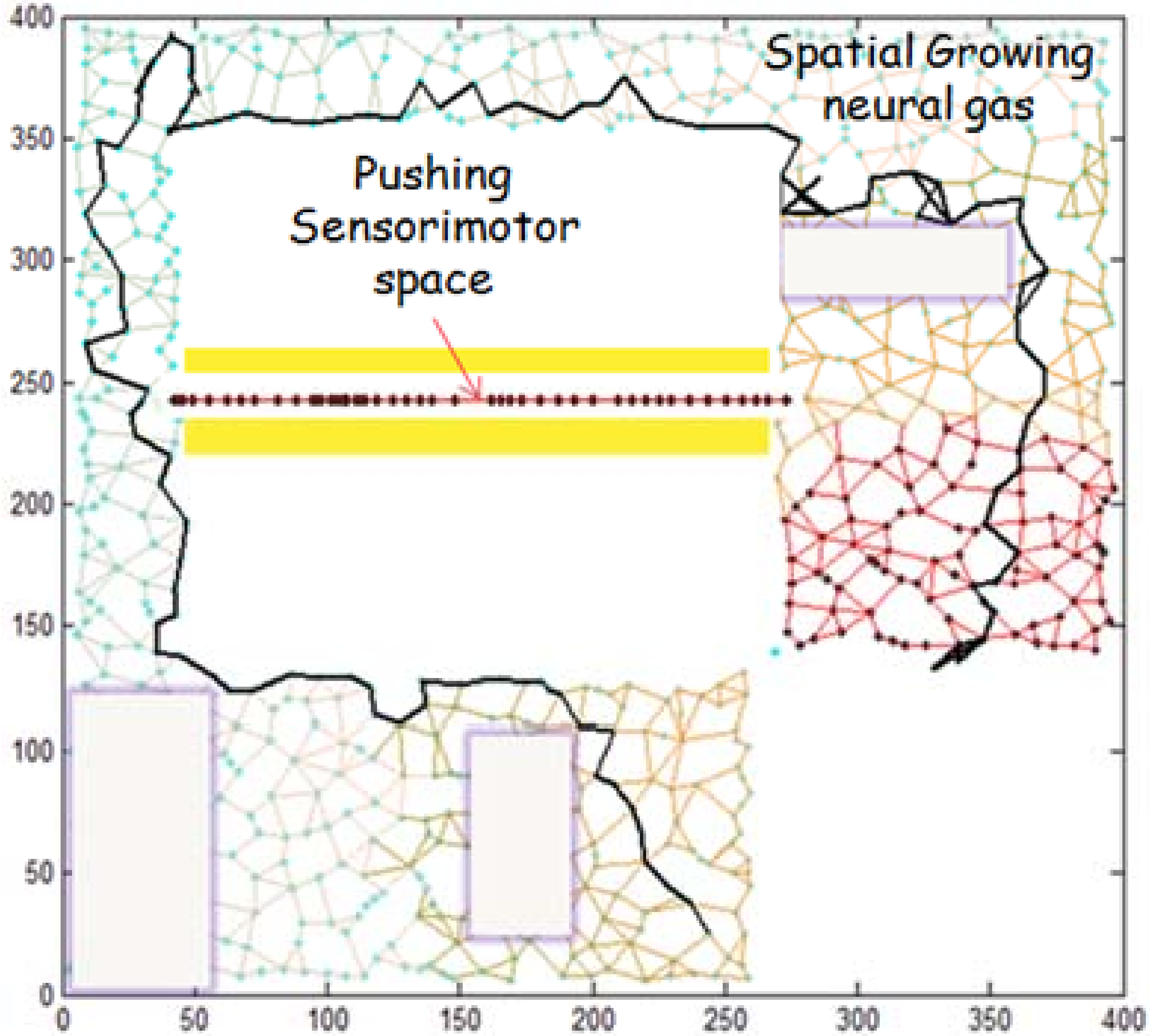




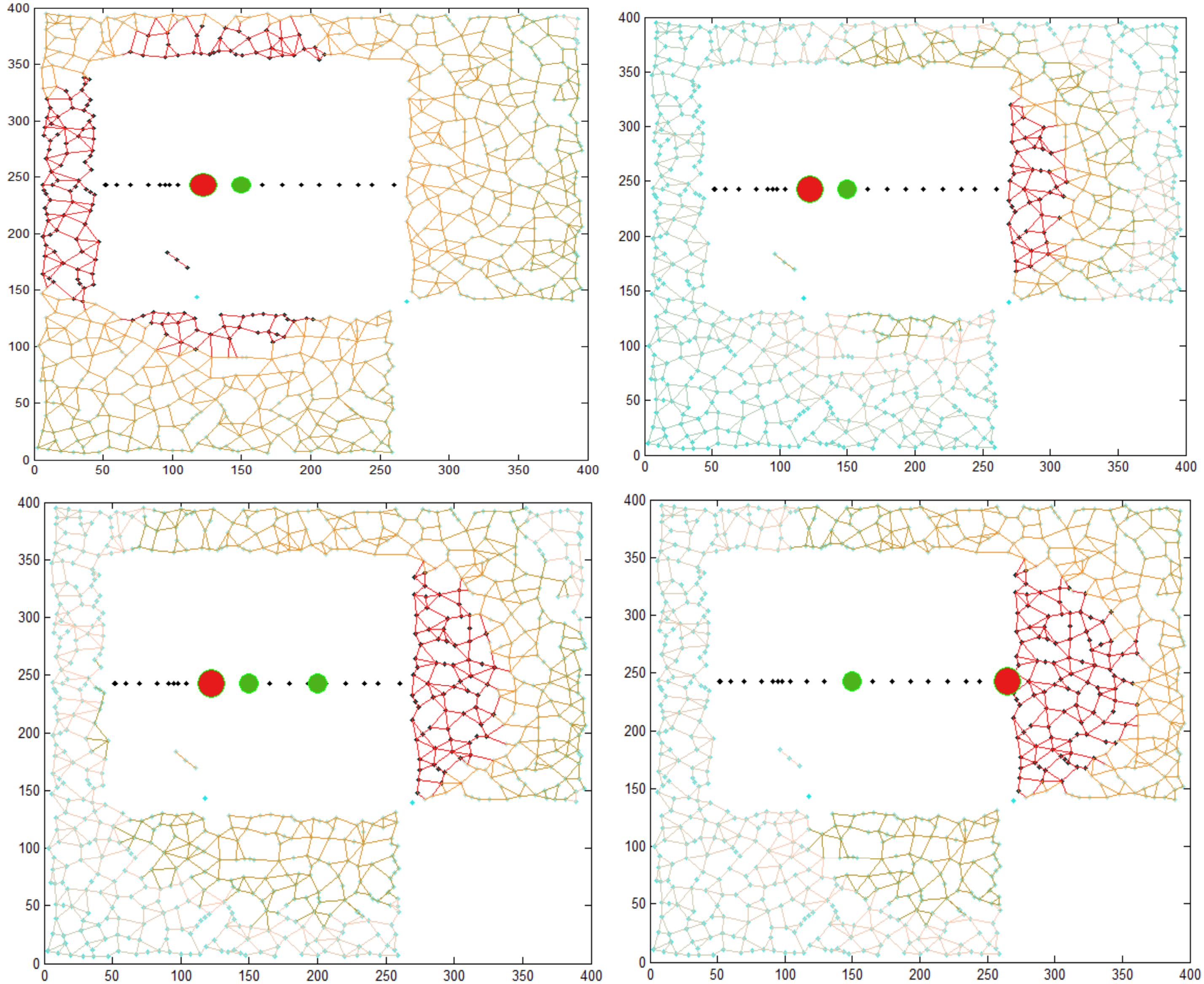
Self organizing the Pushing SMS



Pushing Sensorimotor space



Movement of the Goal due to Pushing can induce reward exitations on the internal spatial map

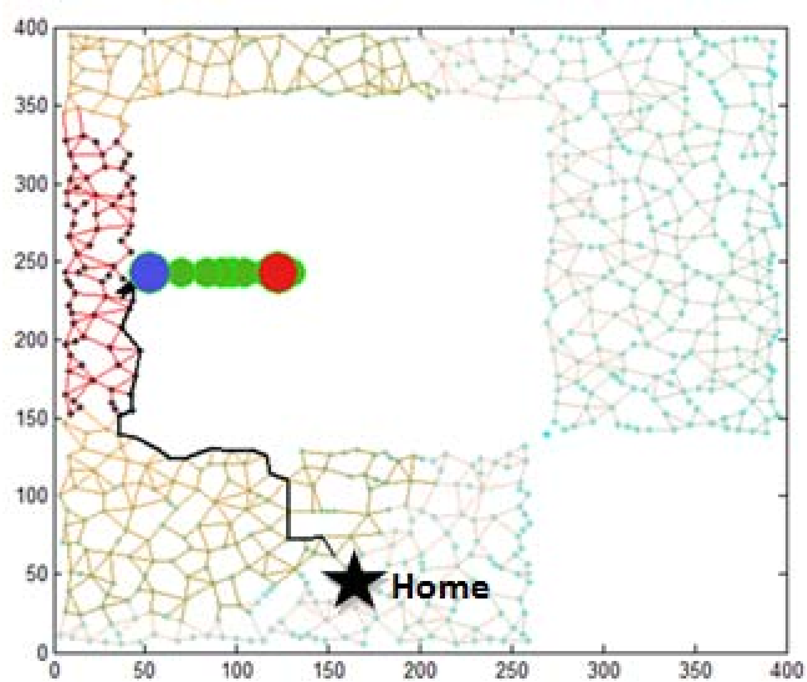


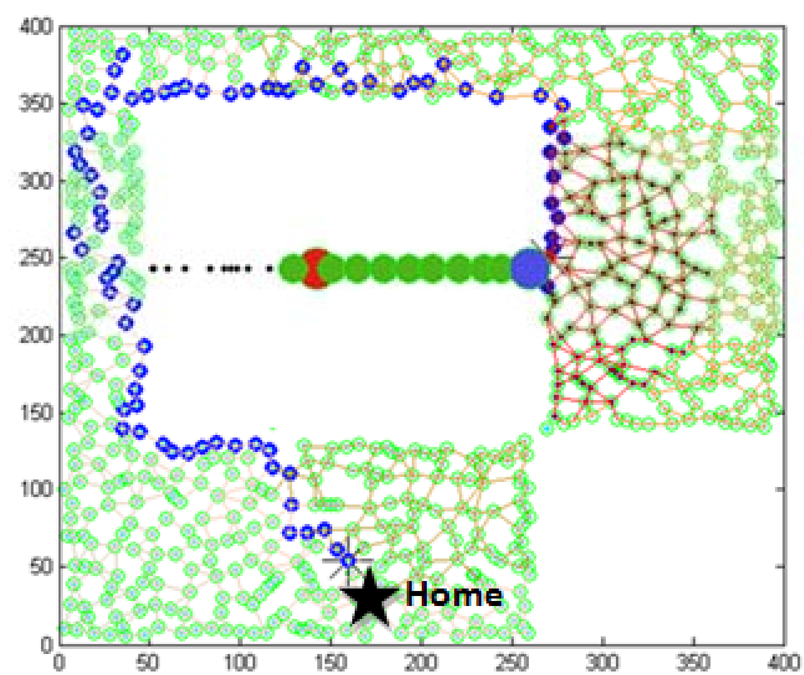
Pushing in ways that are rewarding: Learning the Pushing reward structure

$$R_i = DP + Q$$

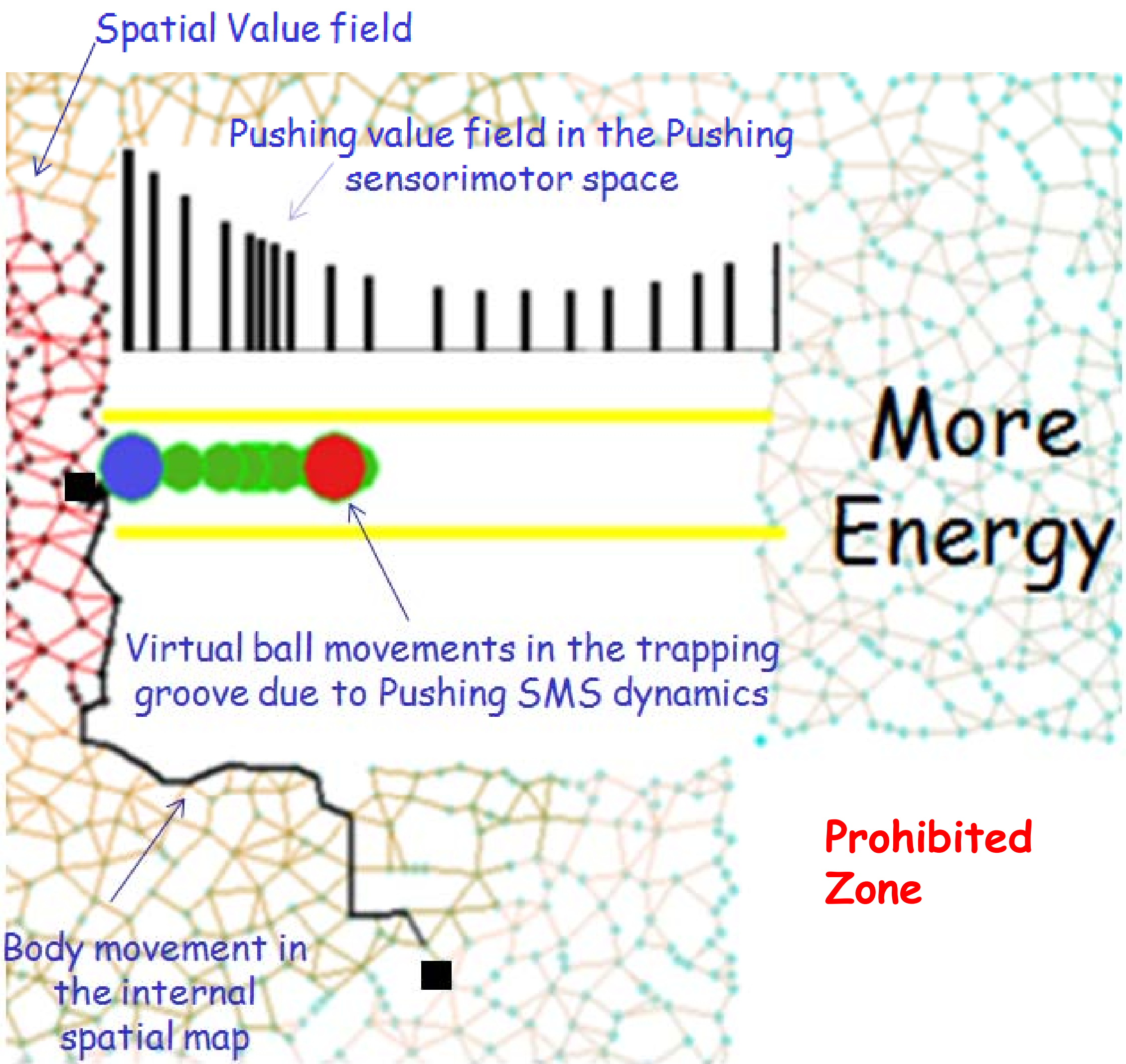
There is no need to have a default plan

DP can be learnt by repeated trials of random explorative pushing of the goal in different directions along the groove, followed by an attempt to grasp the goal (by moving and pushing).





Energy related constraints can also be embedded in the reward structure



$$R_T = R_{net} \quad if \quad Dist_{iter} < \delta$$

$$R_T = R_{nst} e^{-(Dist_{iter}/125)} \quad if \quad Dist_{iter} < \delta$$

$$\delta = \frac{Goal - Initpos}{1.5}.$$

R_T is the actual reward received in the end of the T^{th} trial in case of success

R_{net} is the net reward promised (= 50)

$Dist_{iter}$ is an approximate calculation of the distance navigated by the robot to get the goal

Goal → Value → Action

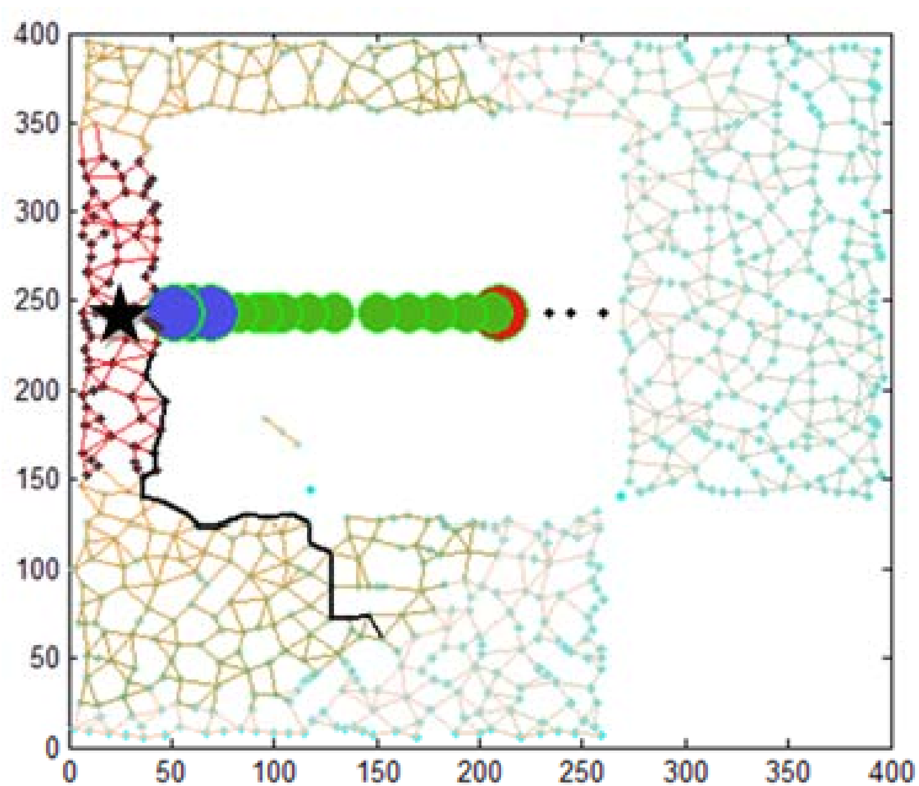
Perception

Goal

↓

Value Field on SMS (Quasistationary)

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$



Value to action

$$k_i = \mathrm{argmax}_j (w_{ij} V_j)$$

$$\bar{a} = \beta_{if} \Big(\sum_{i=1}^{N} x_i \overline{m_{k_i i}}\Big) + \zeta(\bar{r}\ )$$

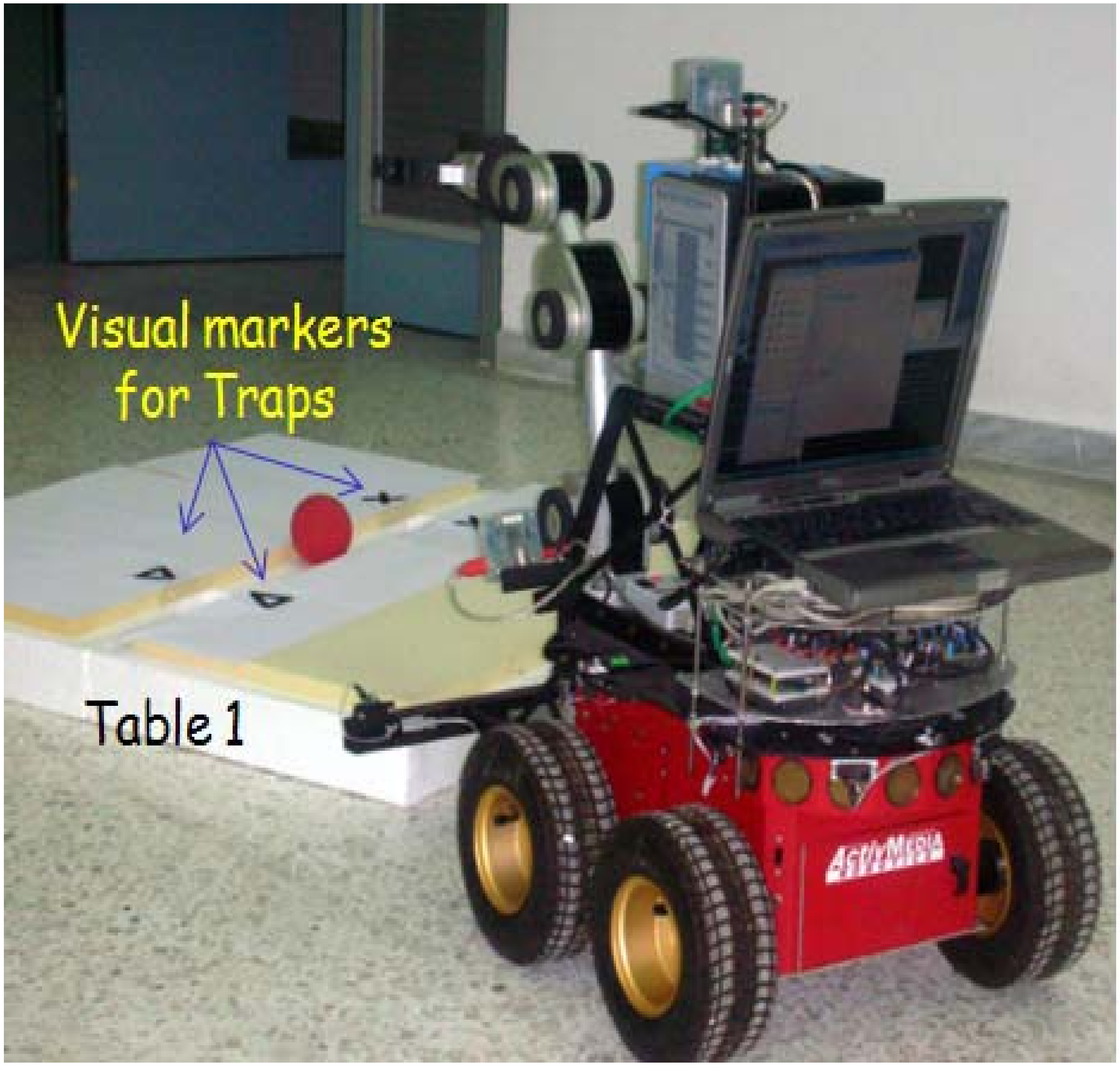
Motor Activity modulates lateral connections in SMS

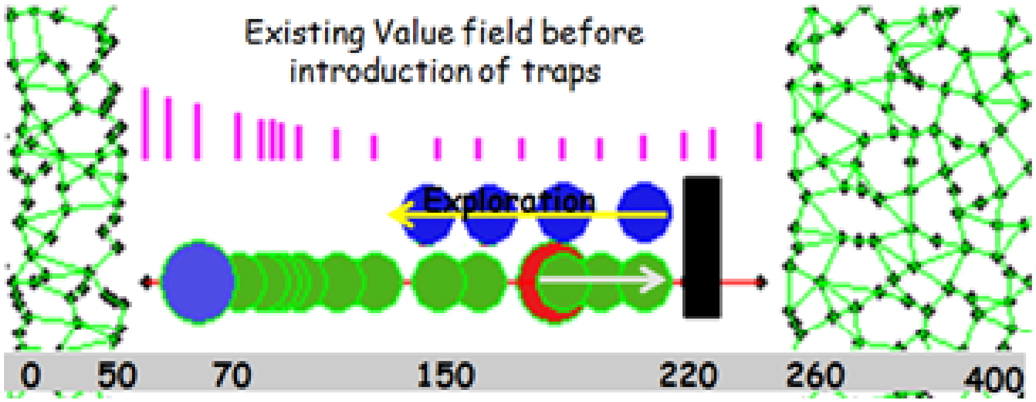
$$M_{ij} =< m_{ij}, M >$$

Activity shift in SMS

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{if} \sum_{i,j} (M_{ij} W_{ij}) x_j$$

New Experiences: Learning to Avoid Traps



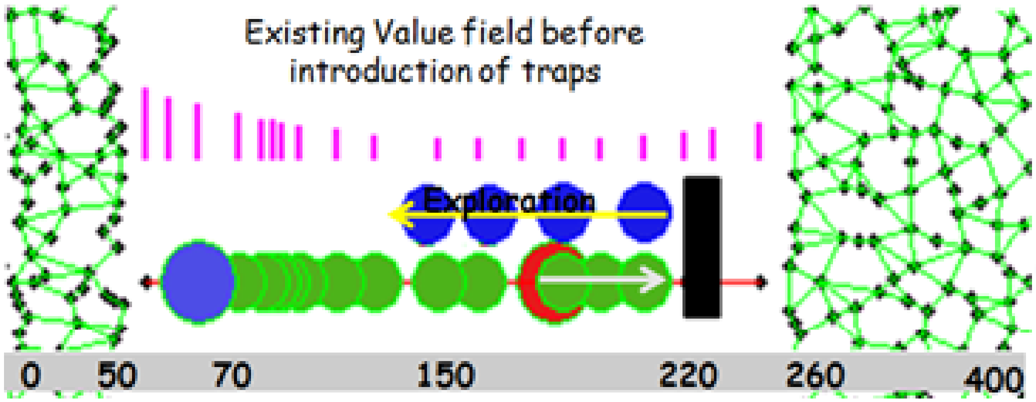


$$\beta_{if} \longrightarrow 0$$

$$\zeta \rightarrow 1$$

In every trial the robot has

an experience of contradiction because of the trap,



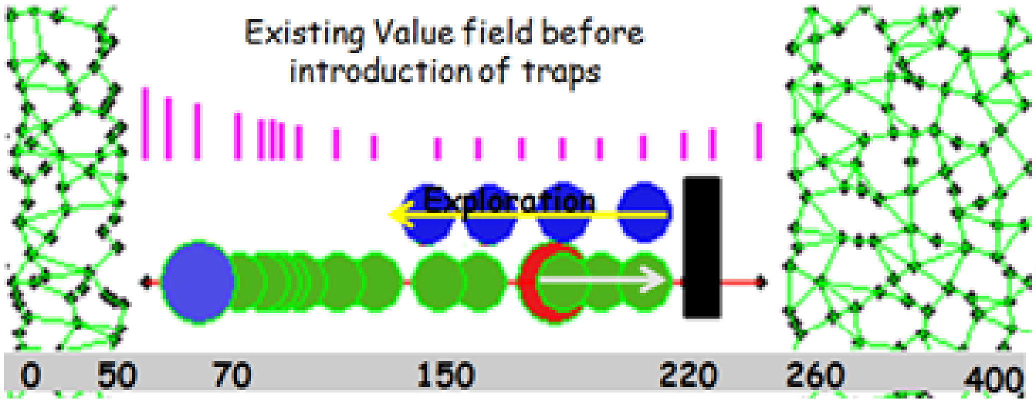
$$\beta_{if} \longrightarrow 0$$

$$\zeta \rightarrow 1$$

In every trial the robot has

an experience of contradiction because of the trap,

an experience of exploration which characterizes its attempt to nullify the effect of the trap so as to realize the goal



$$\beta_{if} \longrightarrow 0$$

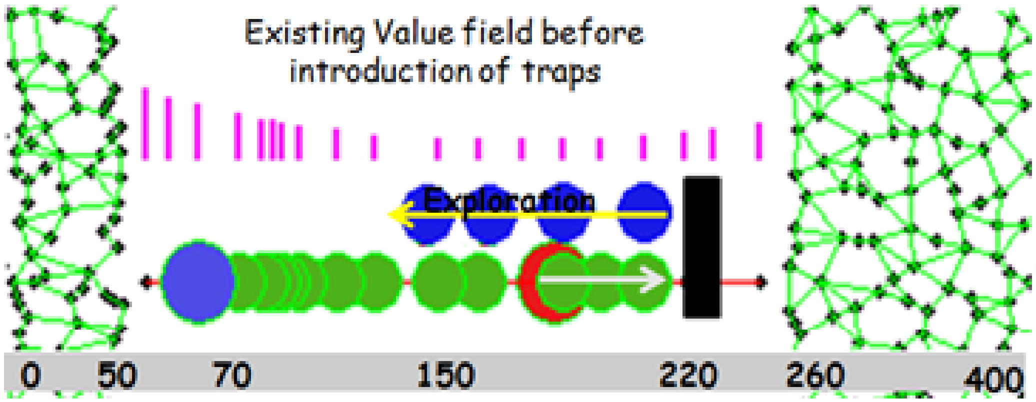
$$\zeta \rightarrow 1$$

In every trial the robot has

an experience of contradiction because of the trap,

an experience of exploration which characterizes its attempt to nullify the effect of the trap so as to realize the goal

an experience of being rewarded by the user/self in case of success.



$$\beta_{if} \longrightarrow 0$$

$$\zeta \rightarrow 1$$

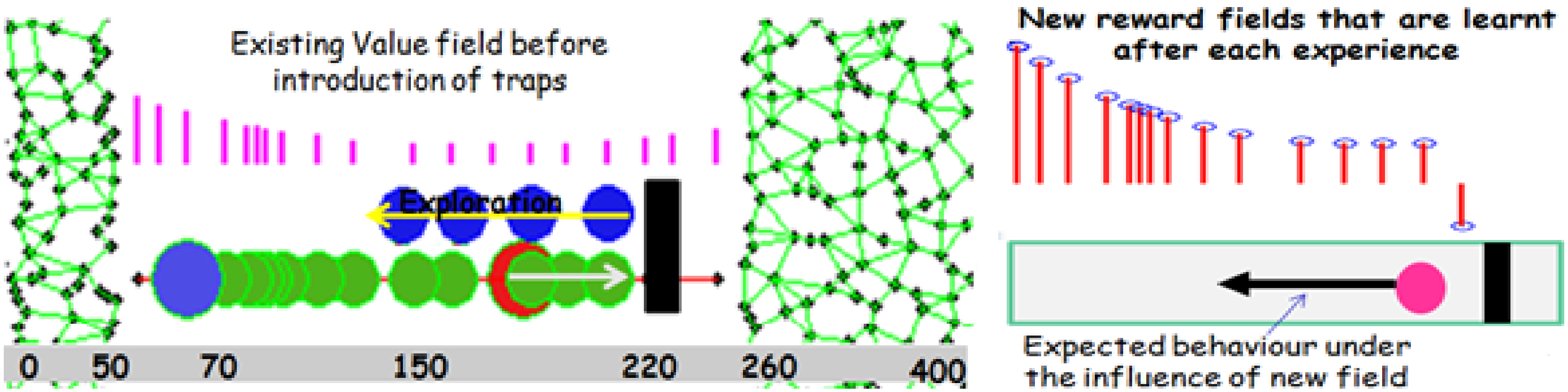
In every trial the robot has

an experience of contradiction because of the trap,

an experience of exploration which characterizes its attempt to nullify the effect of the trap so as to realize the goal

an experience of being rewarded by the user in case of success

Rewards are distributed using the simple 3 point heuristics



In every trial the robot has

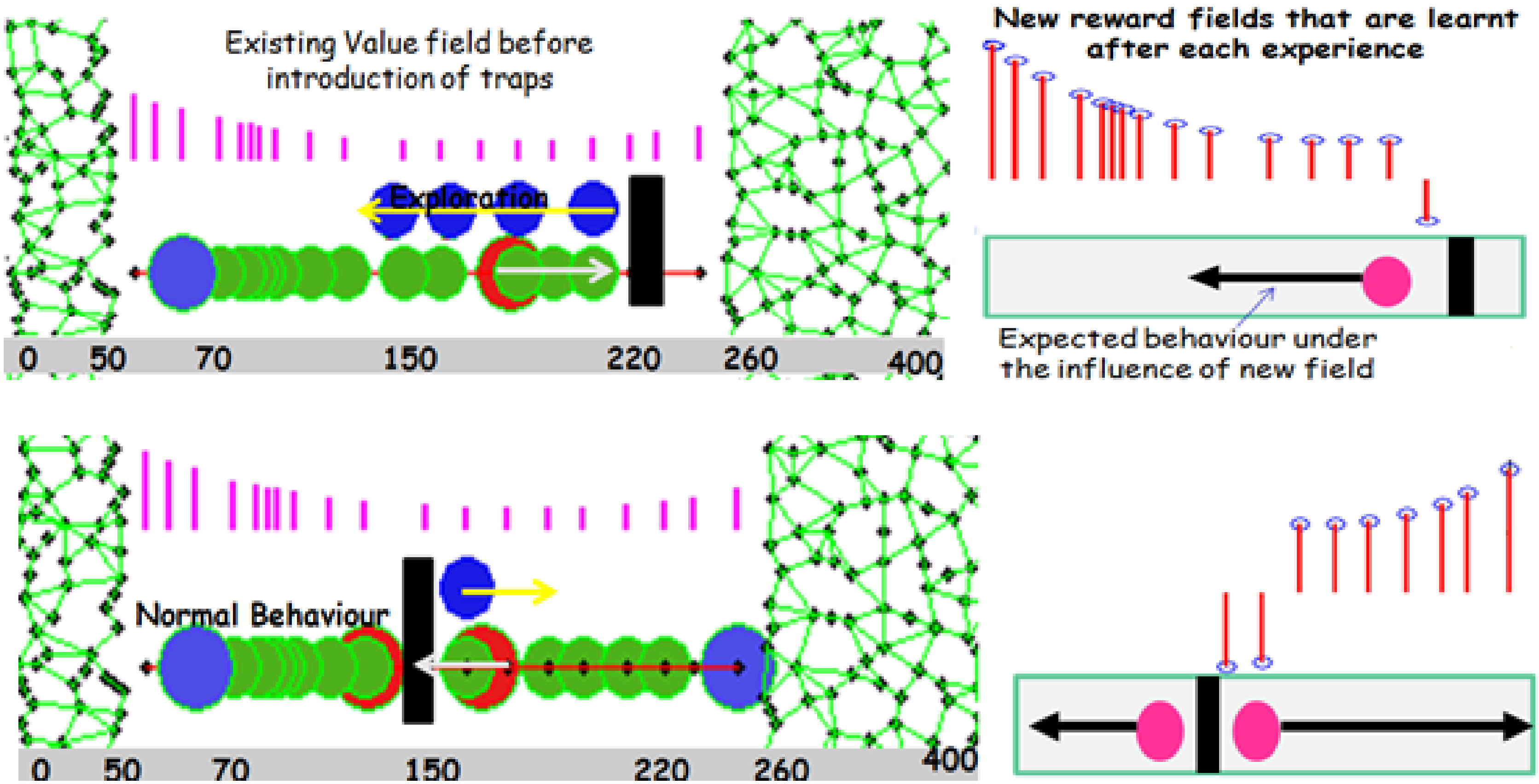
an experience of contradiction because of the trap,

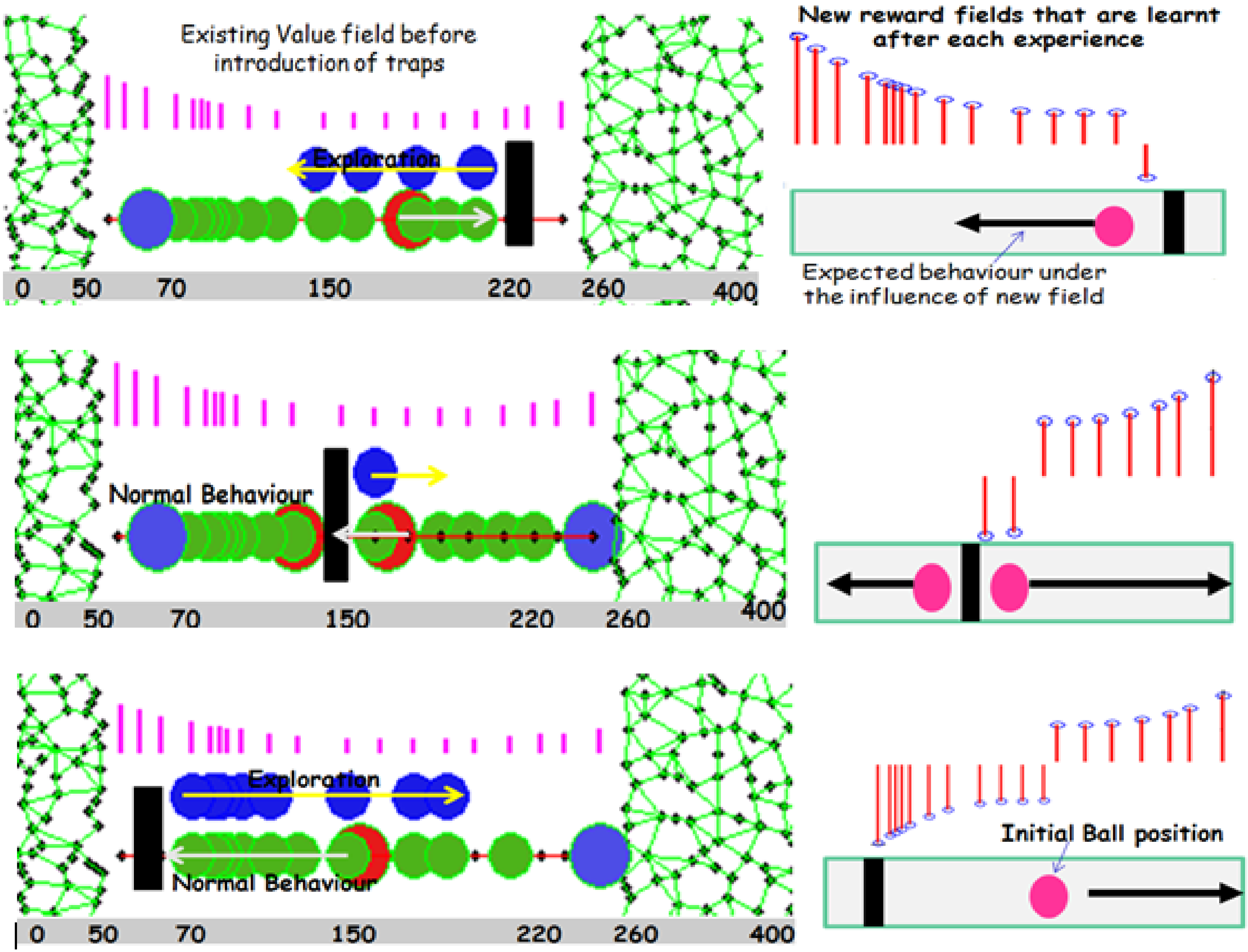
an experience of exploration which characterizes its attempt to nullify the effect of the trap so as to realize the goal

an experience of being rewarded by the user in case of success

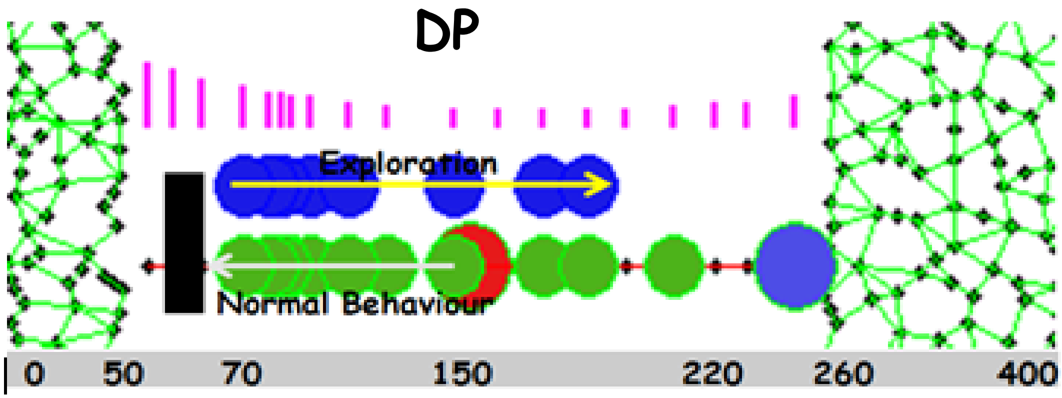
Rewards are distributed using the simple 3 point heuristics

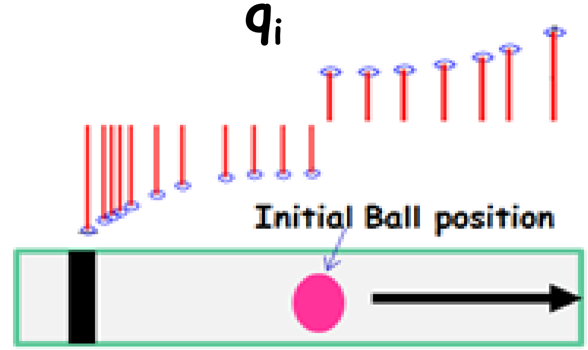
Each experience is represented in the form of a reward field in the pushing sensorimotor space.


Existing Value field before introduction of traps
Exploration
0
50
70
150
220
260
400
New reward fields that are learnt after each experience
Expected behaviour under the influence of new field
Normal Behaviour
0
50
70
150
220
260
400

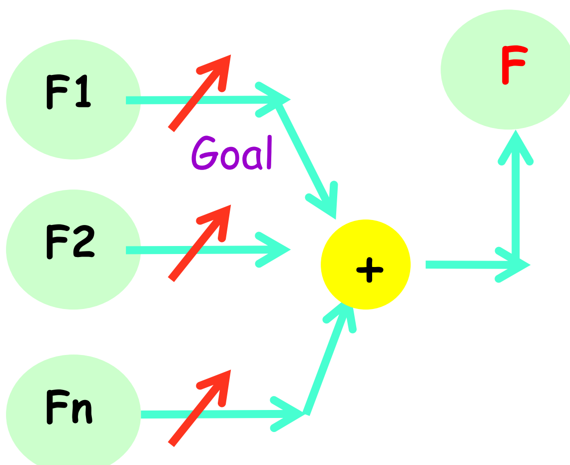


Existing Value field before introduction of traps
Exploration
0 50 70 150 220 260 400
New reward fields that are learnt after each experience
Expected behaviour under the influence of new field
Normal Behaviour
0 50 70 150 220 260 400
Exploration
Normal Behaviour
0 50 70 150 220 260 400
Initial Ball position





Generalizing the learnt new fields



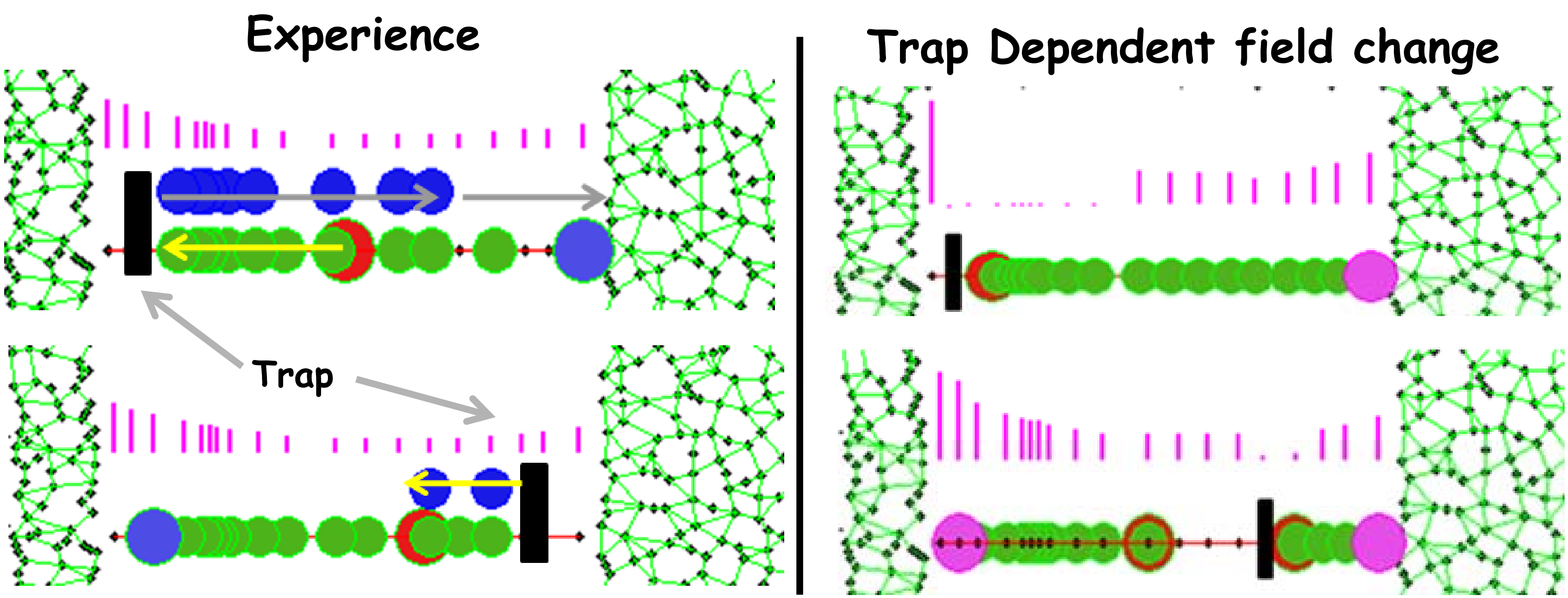
$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max}$$

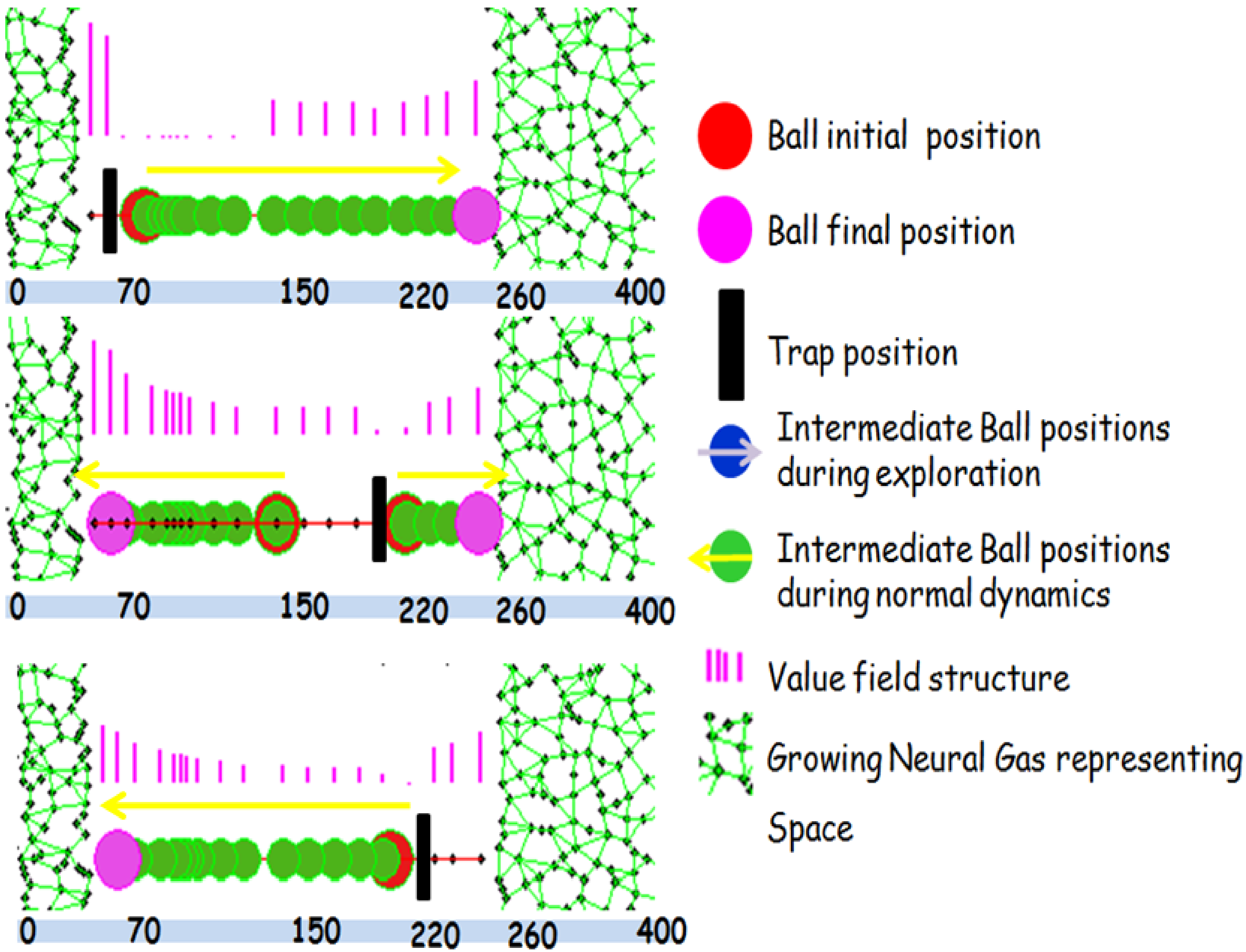
$$R_i = DP + \boxed{Q}$$

$$Q = q_1 + q_2 + ... + q_n$$

$$\boxed{q_i = \varphi_n . U_i . \frac{1}{\sqrt{2\pi}\,\sigma_G} e^{\frac{-(G-Gi)^2}{2\sigma_G{}^2}}}$$

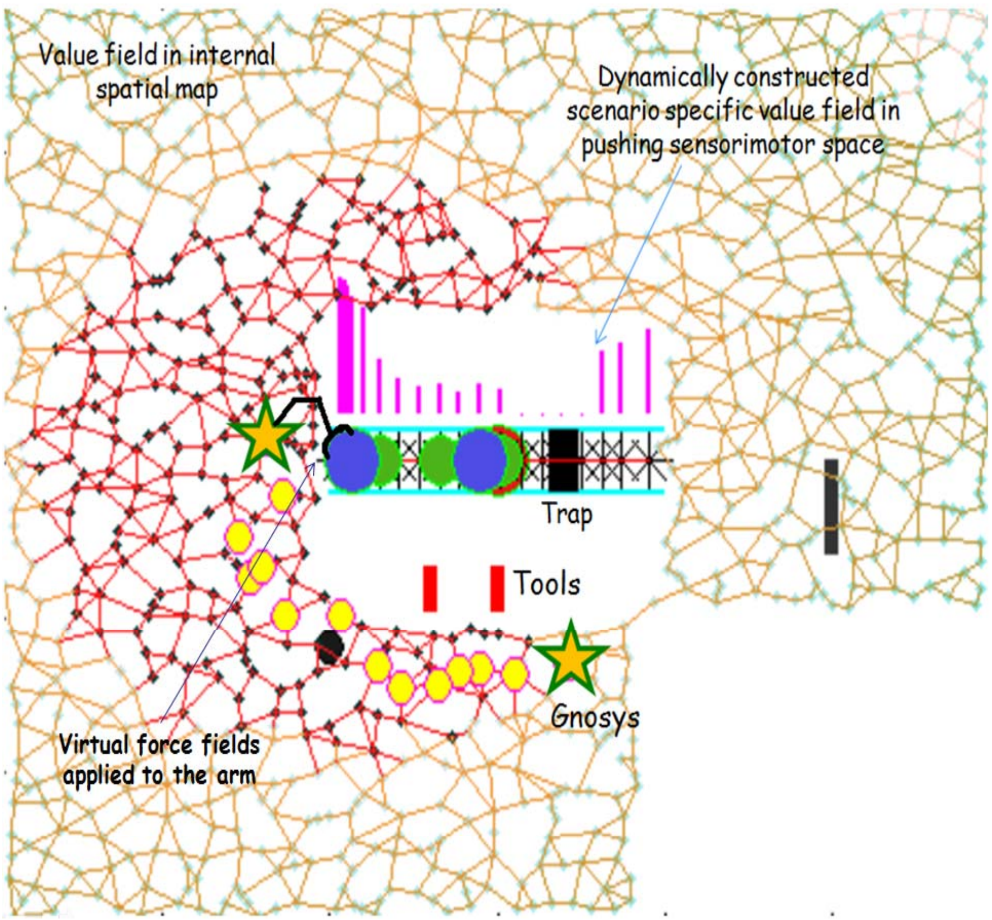
$$R = R_{default} + \sum_{T=1}^{N} \sum_{E=1}^{m} R_E . \frac{1}{\sqrt{2\pi}\,\sigma_T} e^{\frac{-(Trap_T - Trap_E)^2}{2\sigma_T{}^2}}$$


Experience
Trap
Trap Dependent field change



0
70
150
220
260
400
Ball initial position
Ball final position
Trap position
Intermediate Ball positions during exploration
Intermediate Ball positions during normal dynamics
Value field structure
Growing Neural Gas representing Space

A virtual sequence of 'Push-Move-Reach'



> Pushing Internal Model

A virtual trajectory of the goal object

> Internal Spatial Map

A virtual trajectory of the body

> Arm F/I model (PMP)

(Which now recieves two crucial pieces of information to trigger PMP X_T adnd X_{ini})

A virtual trajectory of the end effector

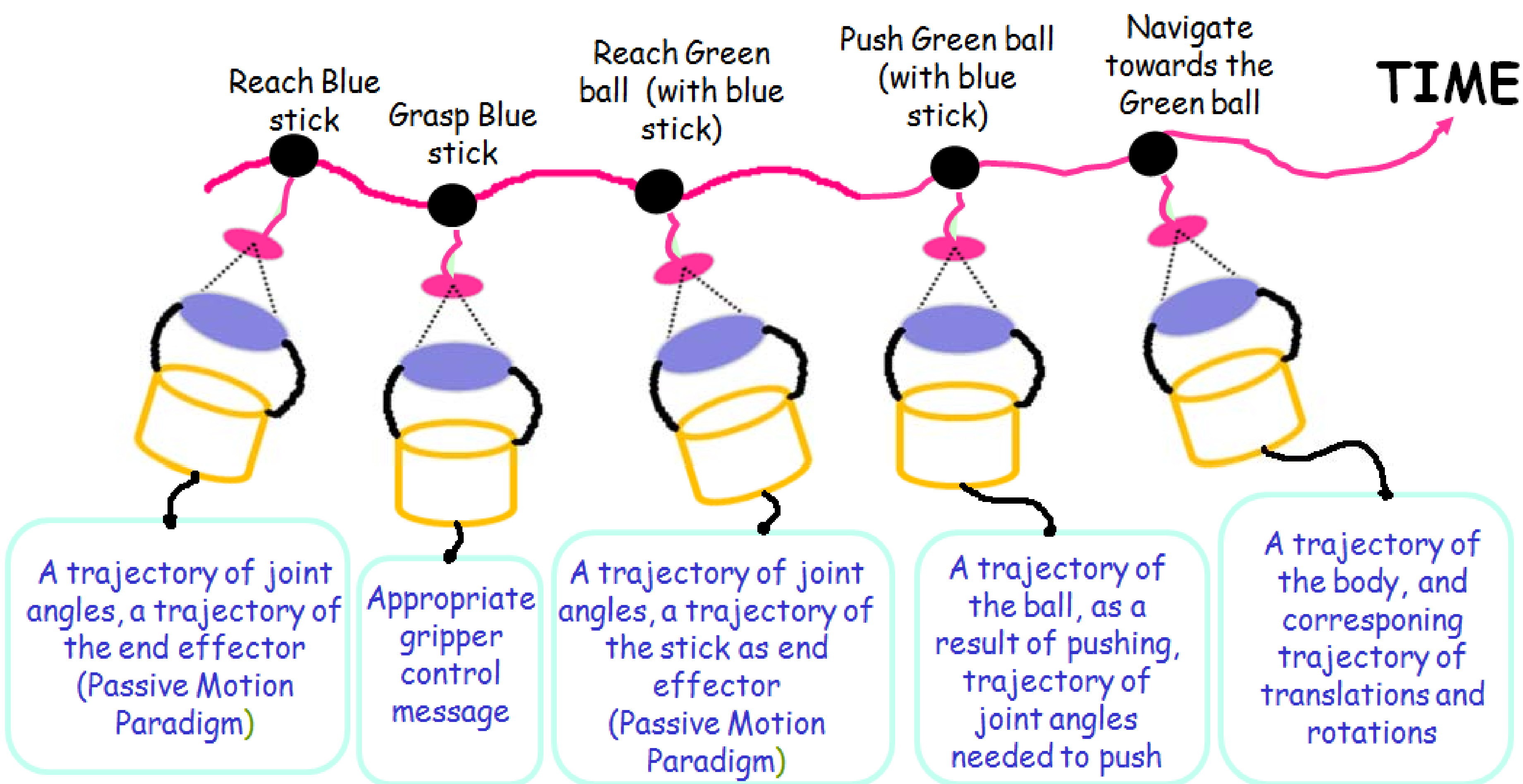
"... since there is a trap there, it is advantageous to push in that direction; if I push in that direction, the ball may eventually go to that side of the table; in case I move my body closer to that edge, I may be in a position to grasp the ball and get some rewards ...".

Abstract Reasoning: From 'Force-Flows' to 'Situation-Action'

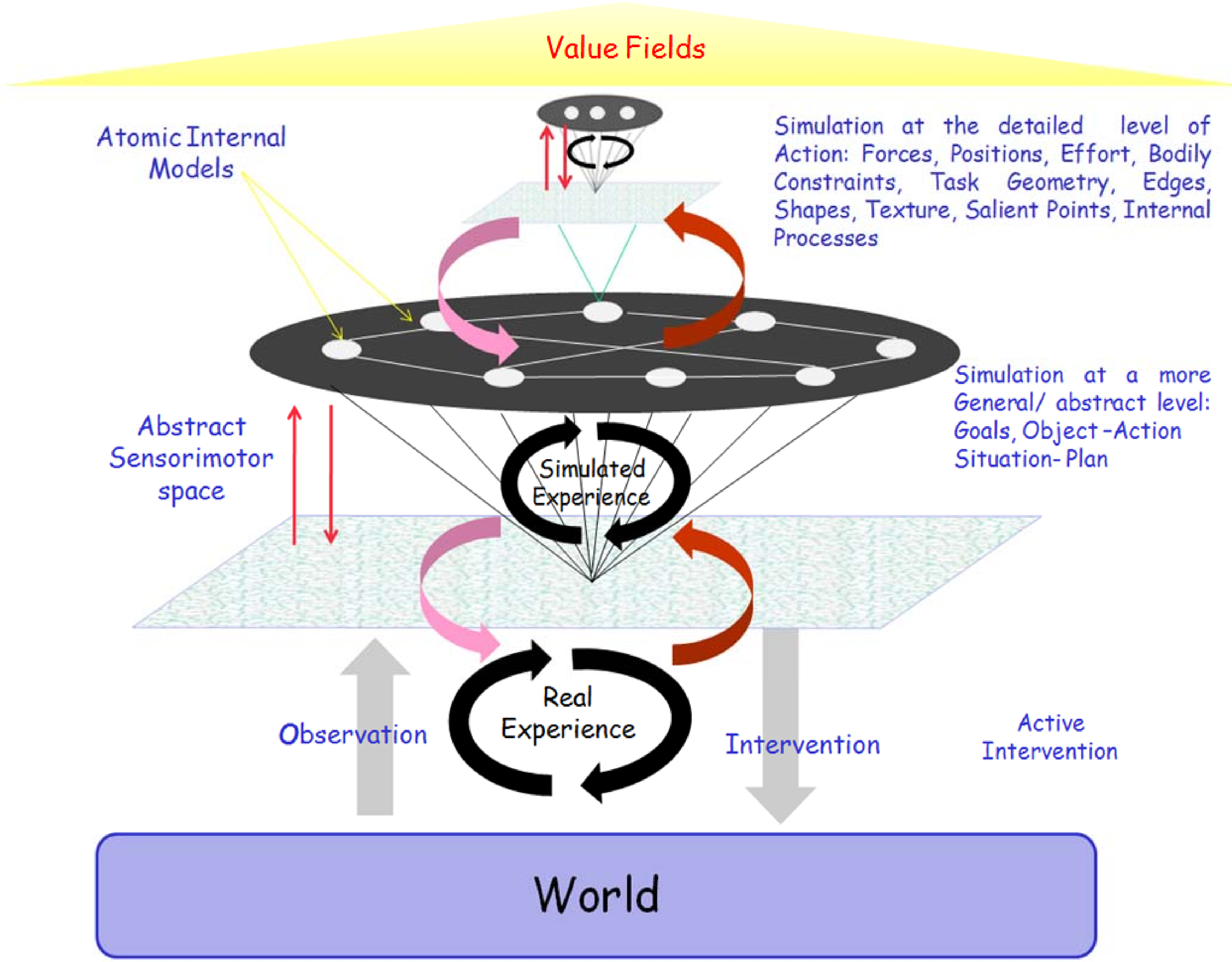
Transform

High level 'user goal'
+ World/Body state
+ Existing Knowledge

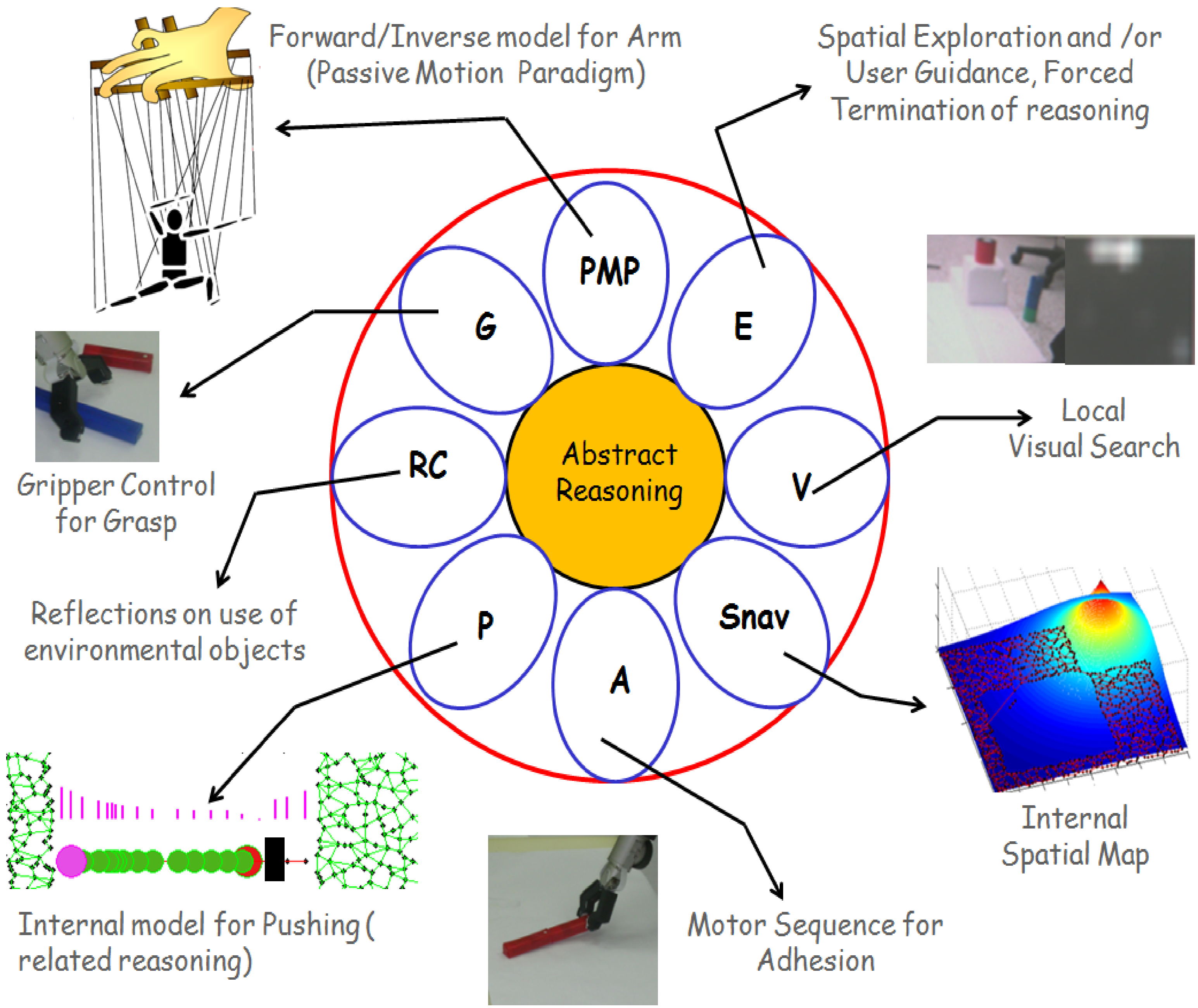
→ Concrete goals for the various internal action models and support processes in the system



From 'Force-Flows' to 'Situation-Action'



Action Primitives (functional and relatively well developed)



Dynamic memory structures for efficient information organization to manage a continuously changing world

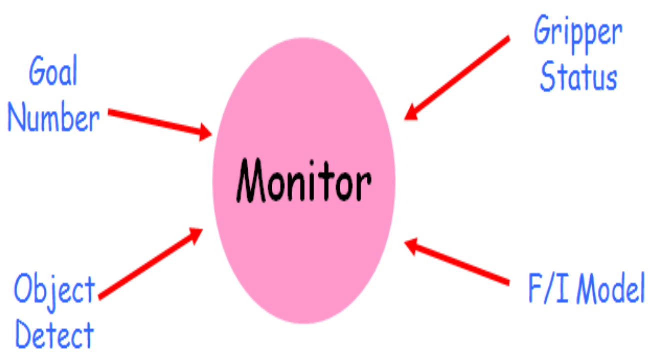
1) Action Primitives (functional and relatively well developed)

2) Sensory Datagram

E1G	E2SG	E3	E4	E5	E6	E7	E8	F/I	GO	OD	M	A	D	-	-

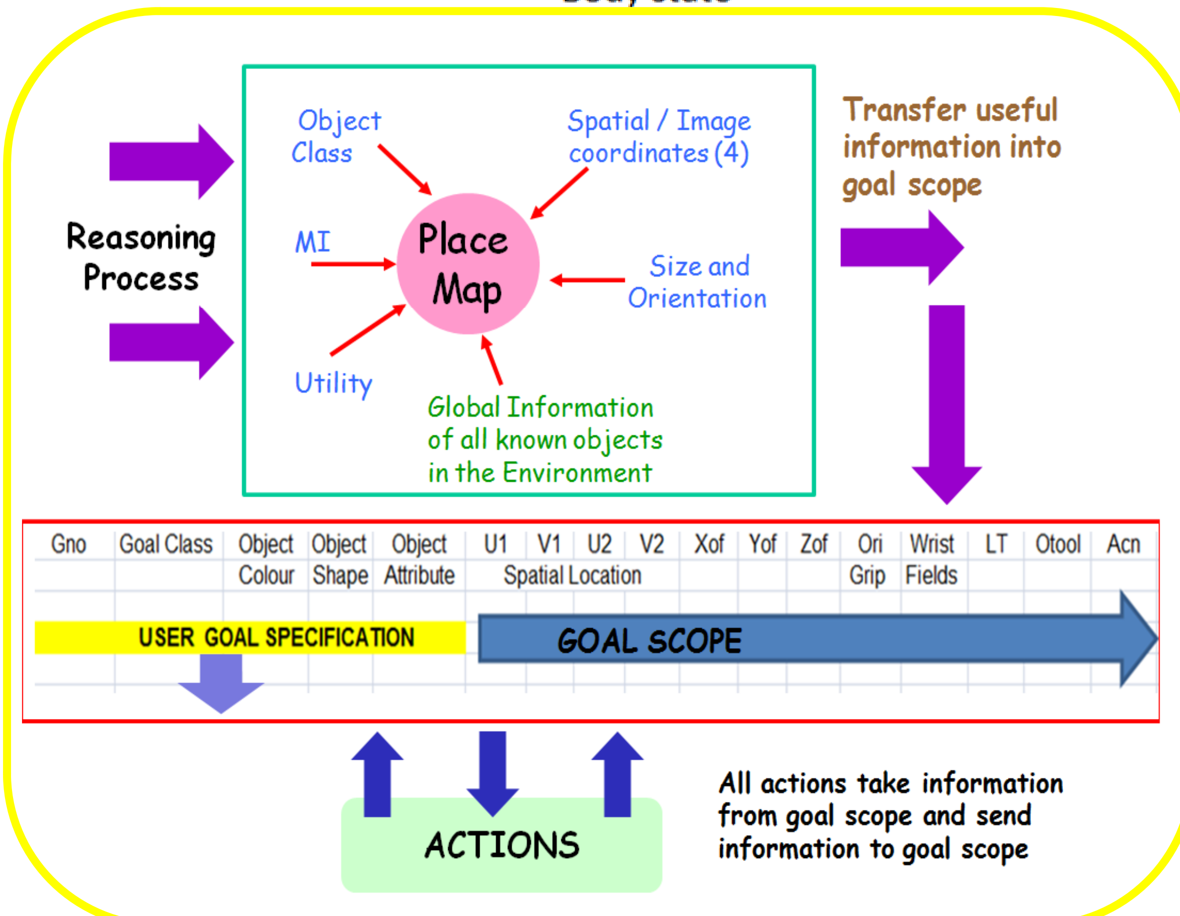
Environmental state (E1G–E8) · Body state (F/I–last)

3) Monitor



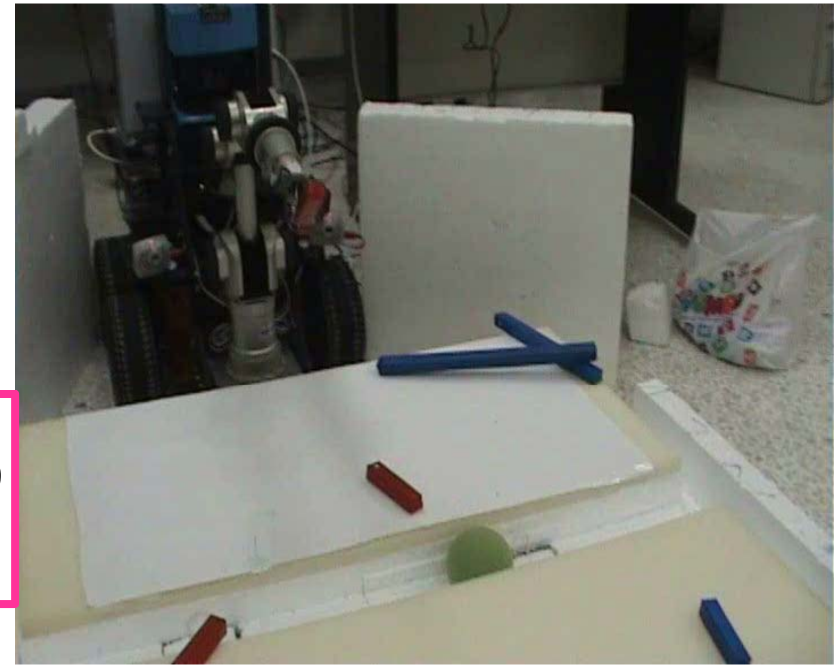
4) Place Map

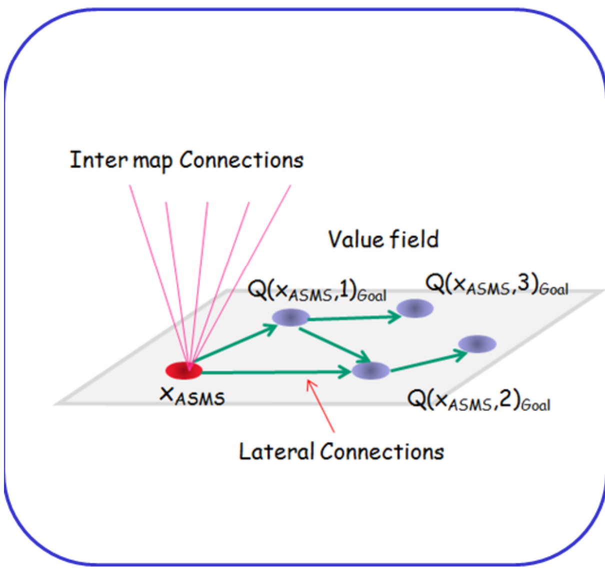
5) Goal Space



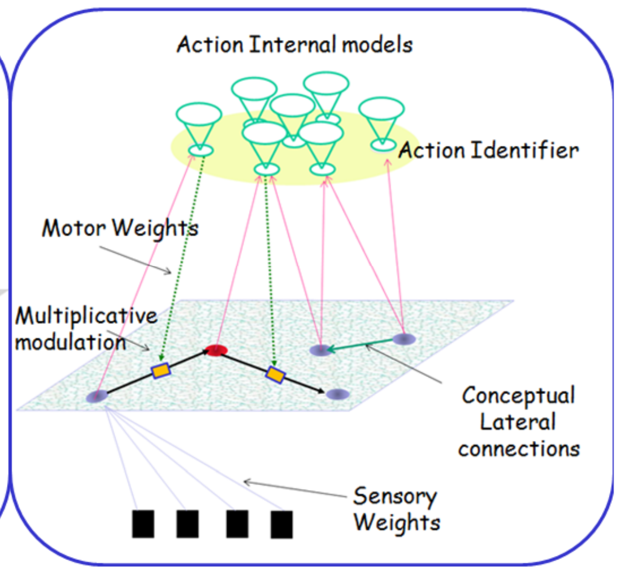
Connectivity Structures

1) Sensory weights
   (Perceptive layer to Sensorimotor map)
2) Lateral weights
   (Activty spread, Value computation)
3) Motor Modulated lateral conections
   (Action -perception)
4) Intermap Connections
   (Perception- action, Generally w/o linked value)
5) Conceptual Lateral connections
   (Peception-Perception, w/o motor influence)
6) Growing reward matrices (Whats useful)

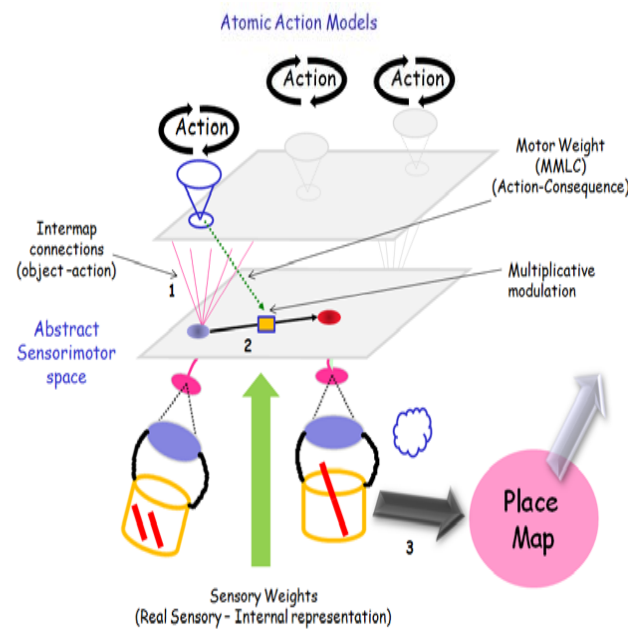




Situation-Goal-Value-Action loop in ASMS



Situation-Action-Consequence loop in ASMS



Continuosuly Learning System (Reasoning and Exploration are always loosely coupled, no seperate phases of learning etc)

Accidental Trial and Error

Reaching other objects using sticks, cylinders

Pushing, Pushing+Traps

Coupling two sticks, observing consequence

Intermap Conceptual Lateral Conectivity

What's Possible

User Goal Driven

Gradual Increments in Environmental complexity

Bidirectional Activity Transfer

MMLC, LatCon Reward Matrix

What's Useful

Interactive Knowledge Transfer

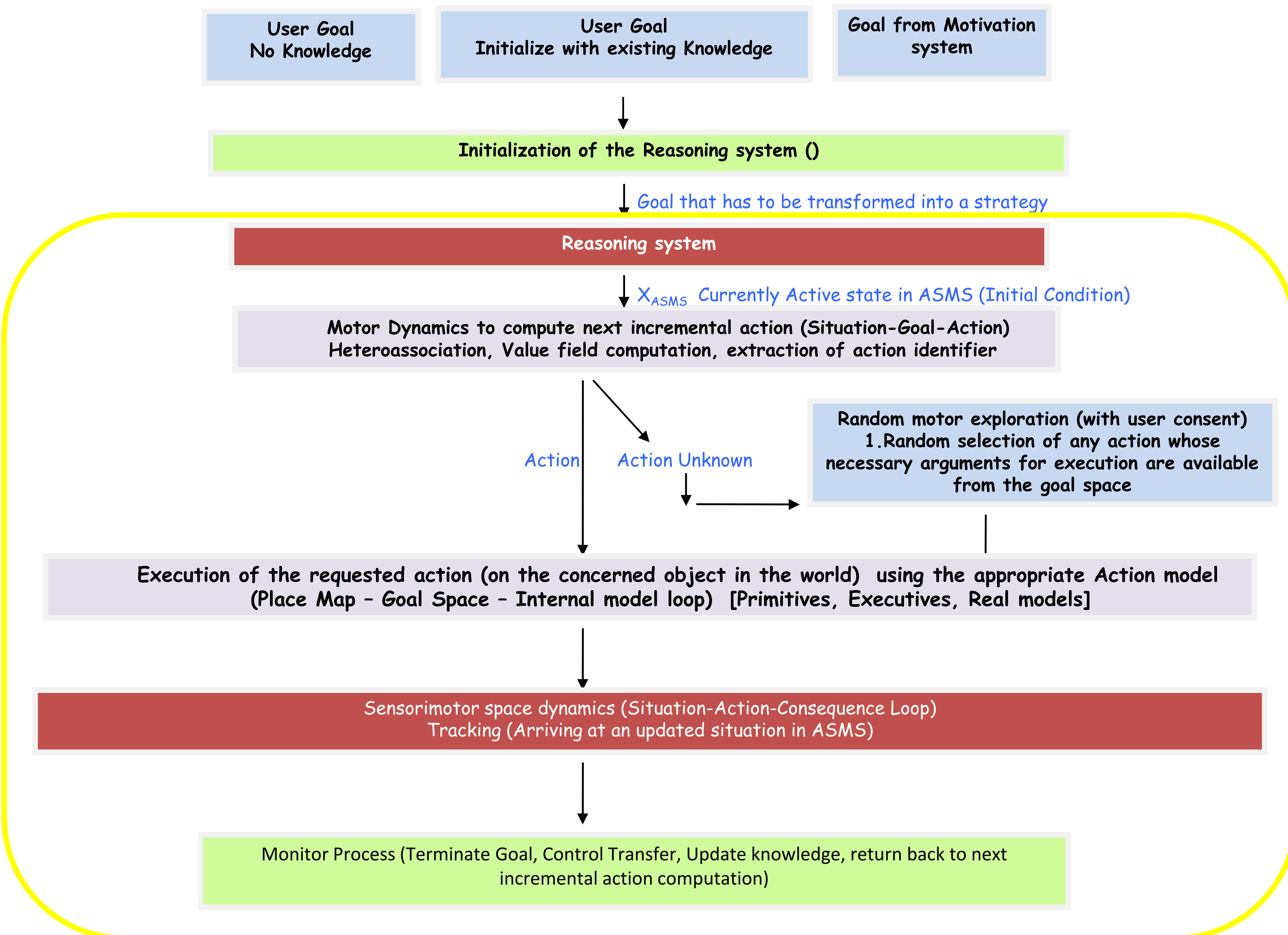
L1 Sensory
(Self organization, GNG)

L2 Sensory-Motor
(Heteroassociative Memory)

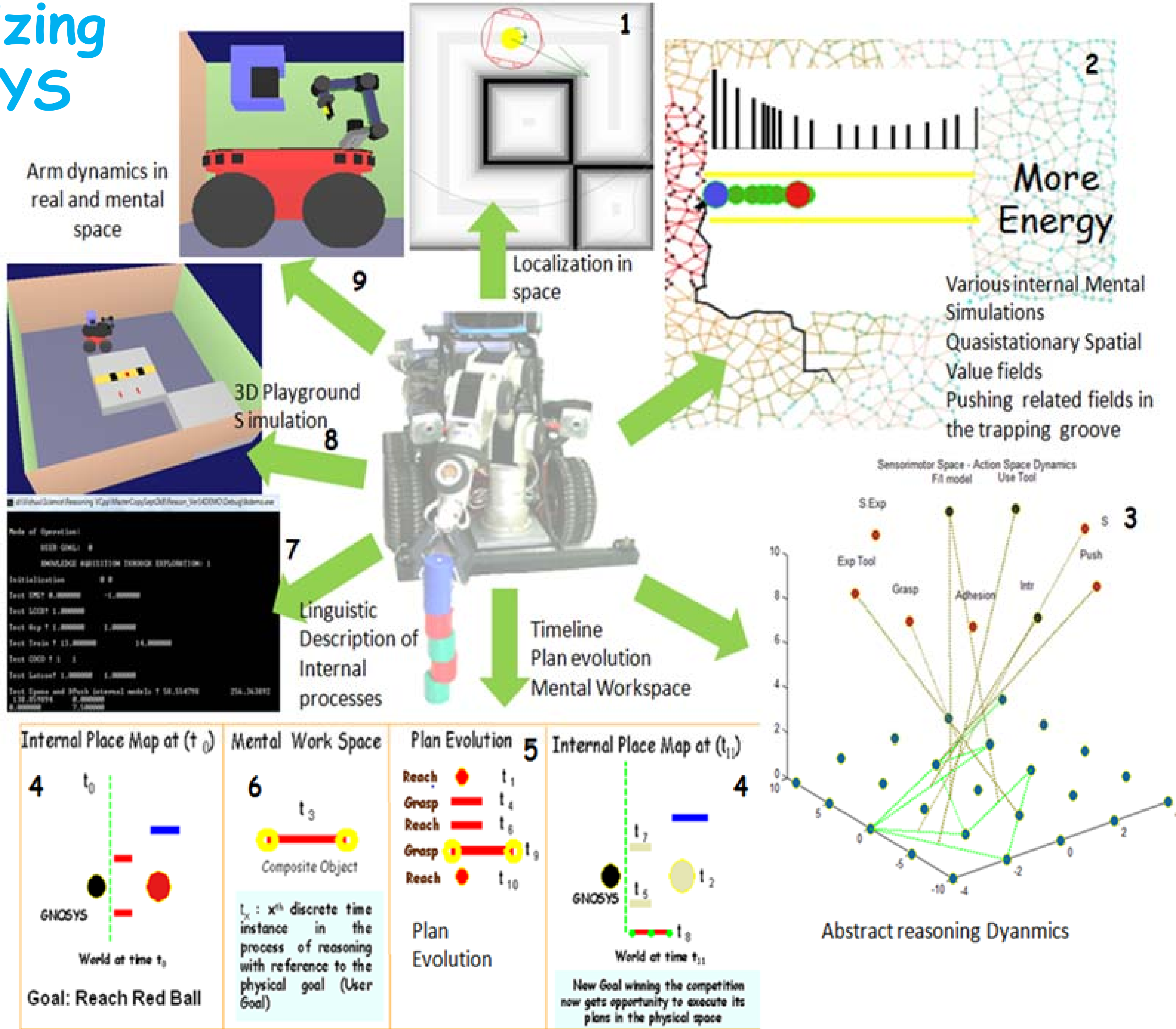
L3 Value Dependent
(Influence of Goals)

L4 Accidental/ TE
(Curiosity+ Motivation)

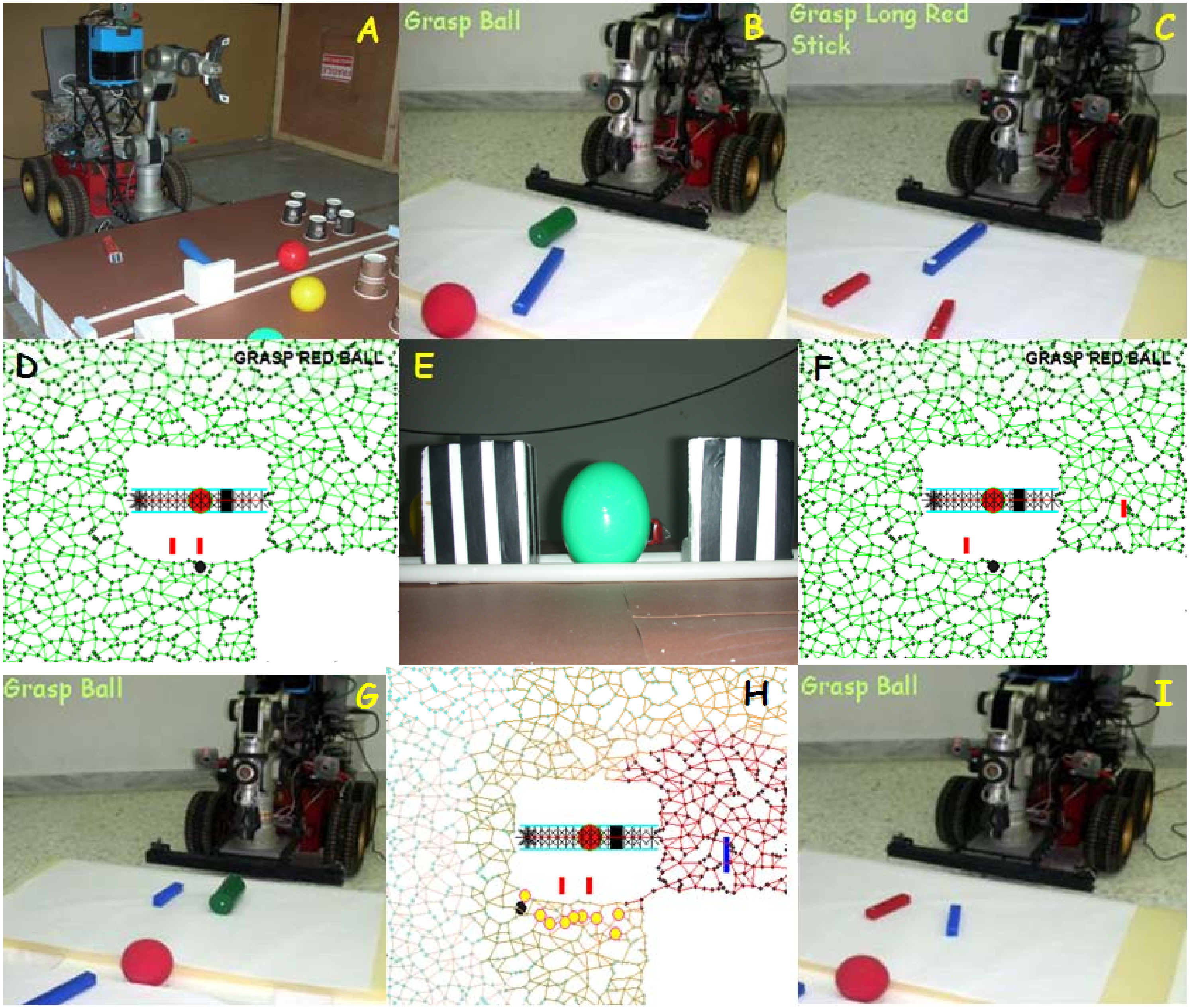
L5 Interactive
(Help seeking/Giving)


User Goal
No Knowledge
User Goal
Initialize with existing Knowledge
Goal from Motivation system
Initialization of the Reasoning system ()
Goal that has to be transformed into a strategy
Reasoning system
X_{ASMS} Currently Active state in ASMS (Initial Condition)
Motor Dynamics to compute next incremental action (Situation-Goal-Action)
Heteroassociation, Value field computation, extraction of action identifier
Action
Action Unknown
Random motor exploration (with user consent)
1.Random selection of any action whose necessary arguments for execution are available from the goal space
Execution of the requested action (on the concerned object in the world) using the appropriate Action model
(Place Map – Goal Space – Internal model loop) [Primitives, Executives, Real models]
Sensorimotor space dynamics (Situation-Action-Consequence Loop)
Tracking (Arriving at an updated situation in ASMS)
Monitor Process (Terminate Goal, Control Transfer, Update knowledge, return back to next incremental action computation)

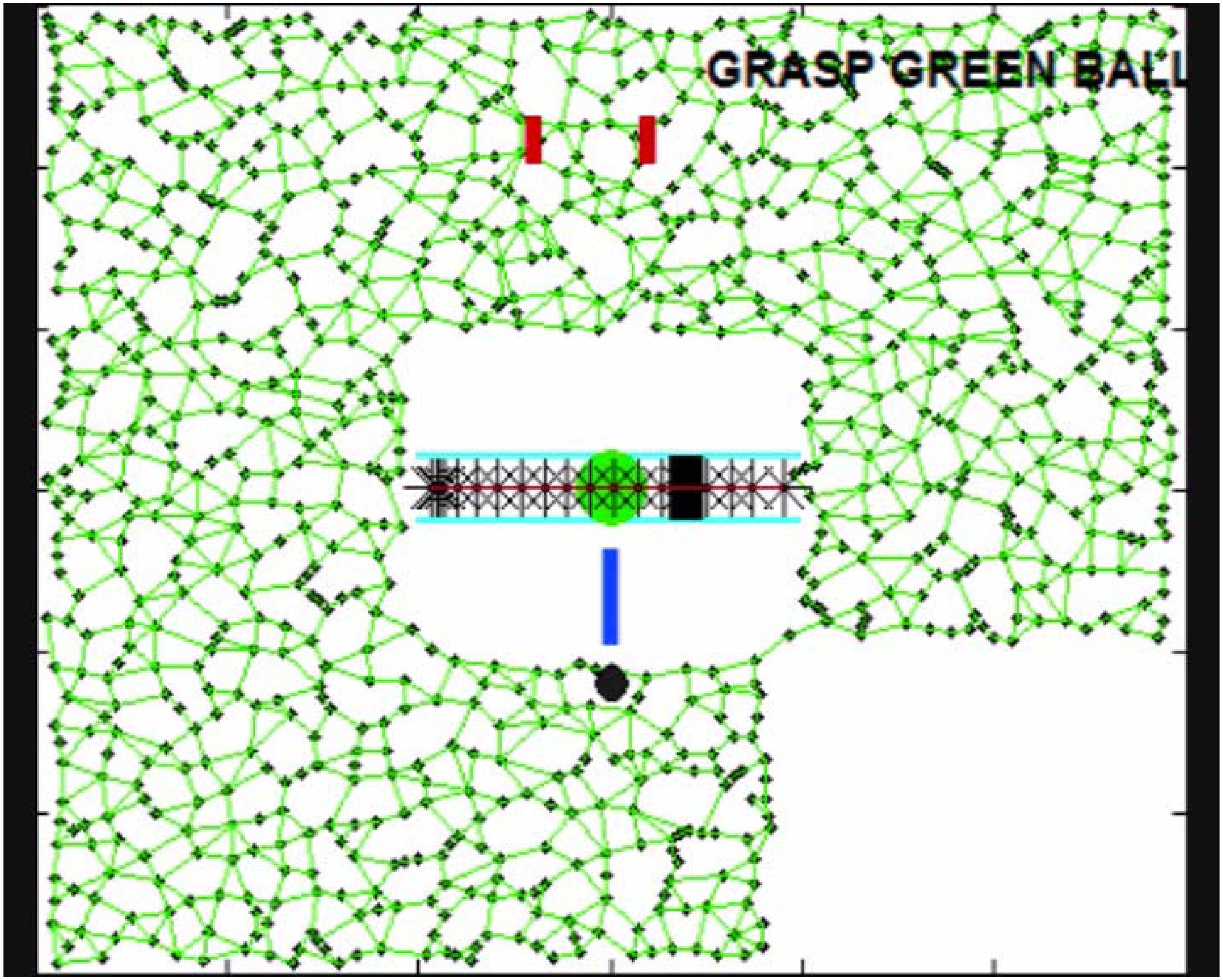
Visualizing GNOSYS



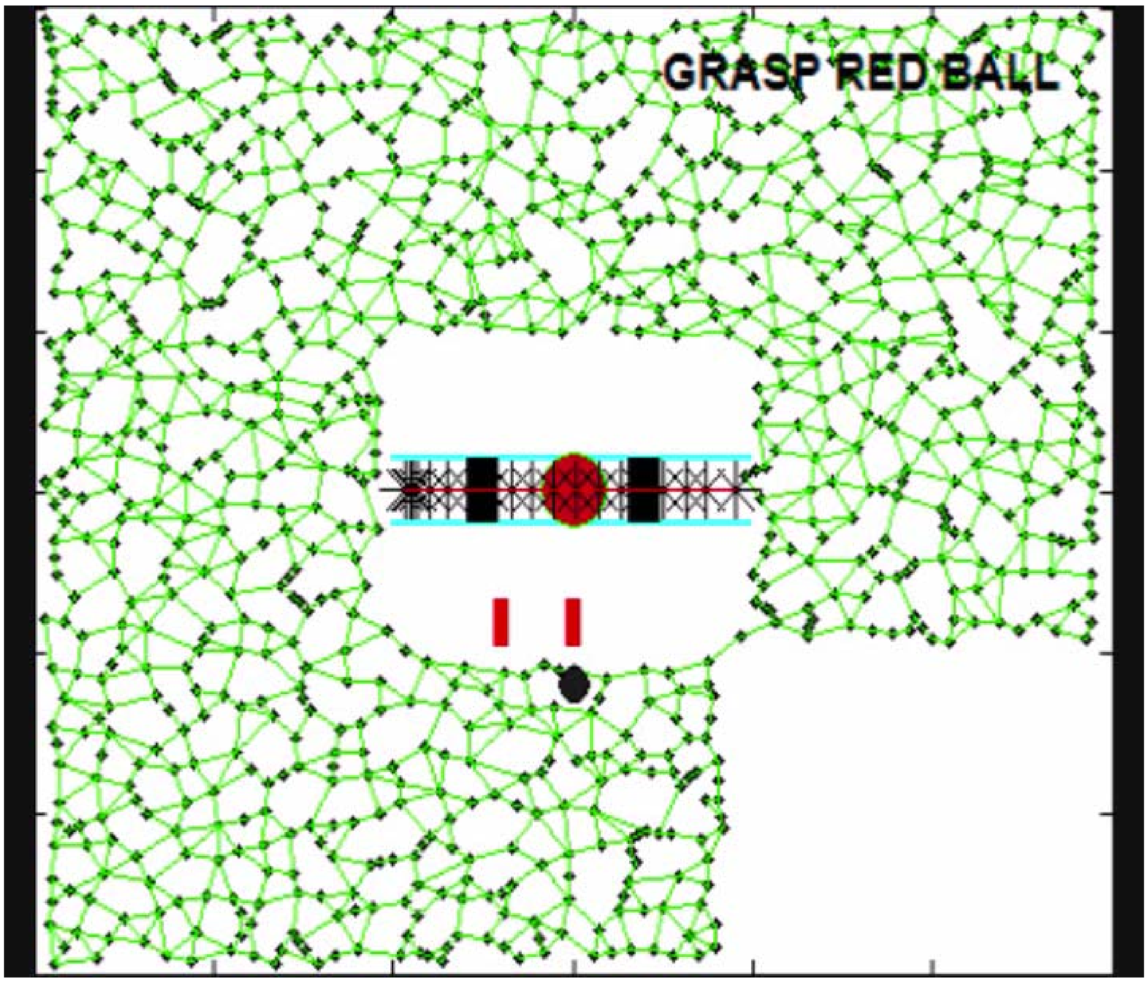
Some Test Scenarios

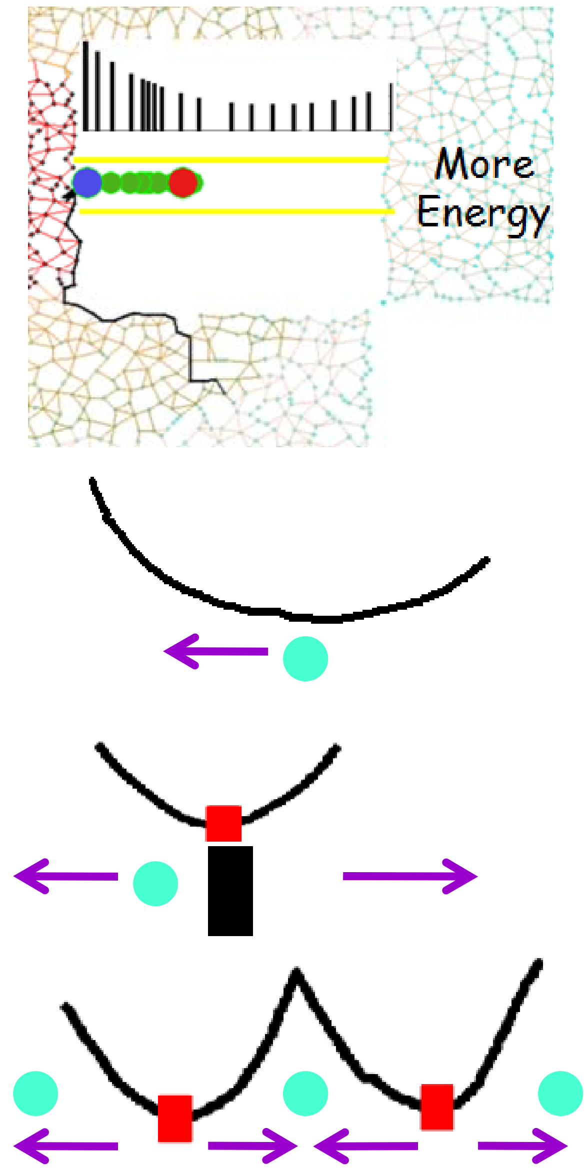


‘Grasp Green Ball’--- GNOSYS Demo

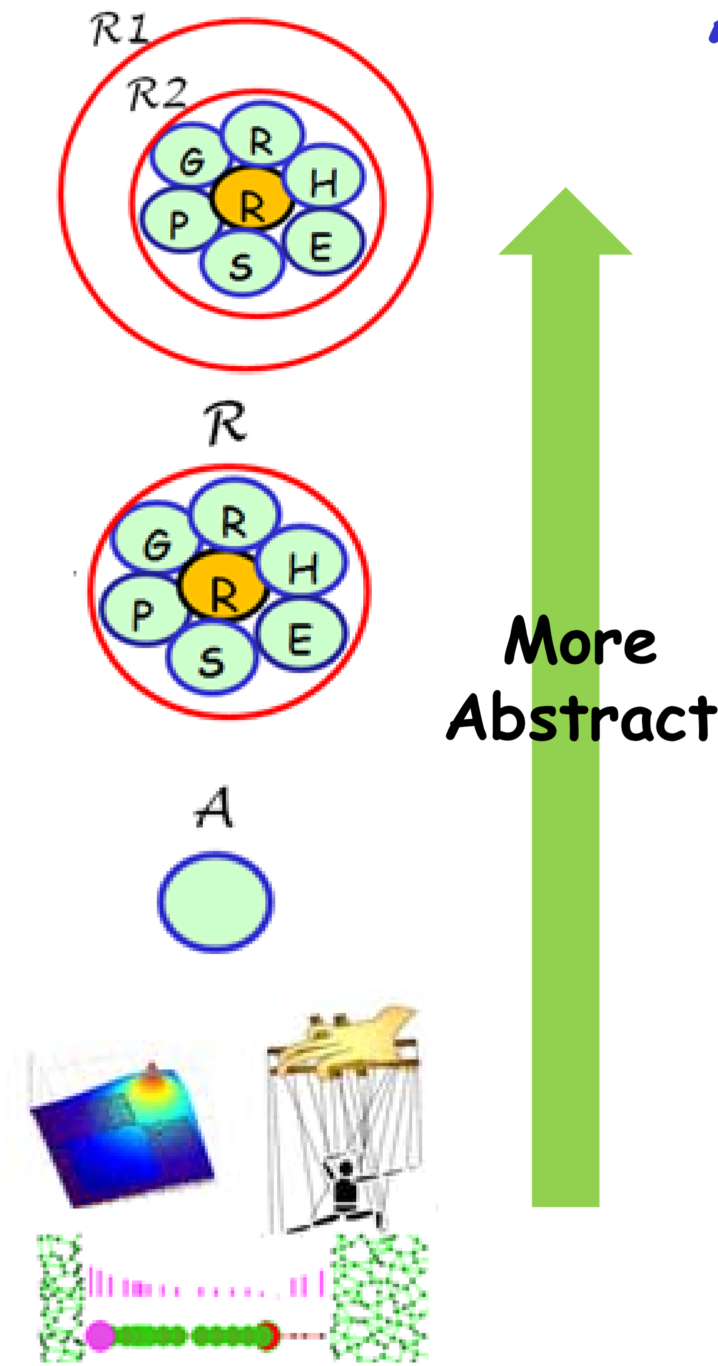


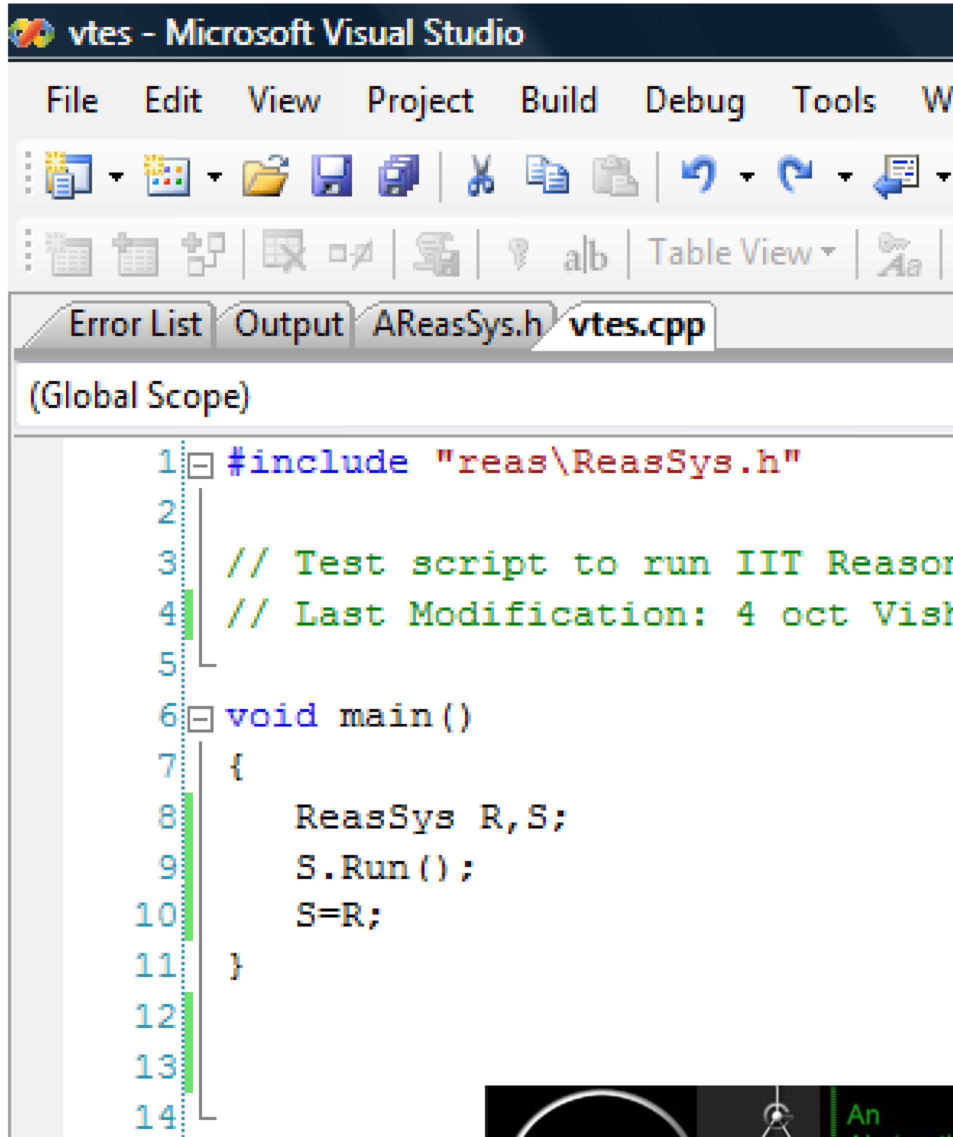
'Quitting' and having a REASON to QUIT





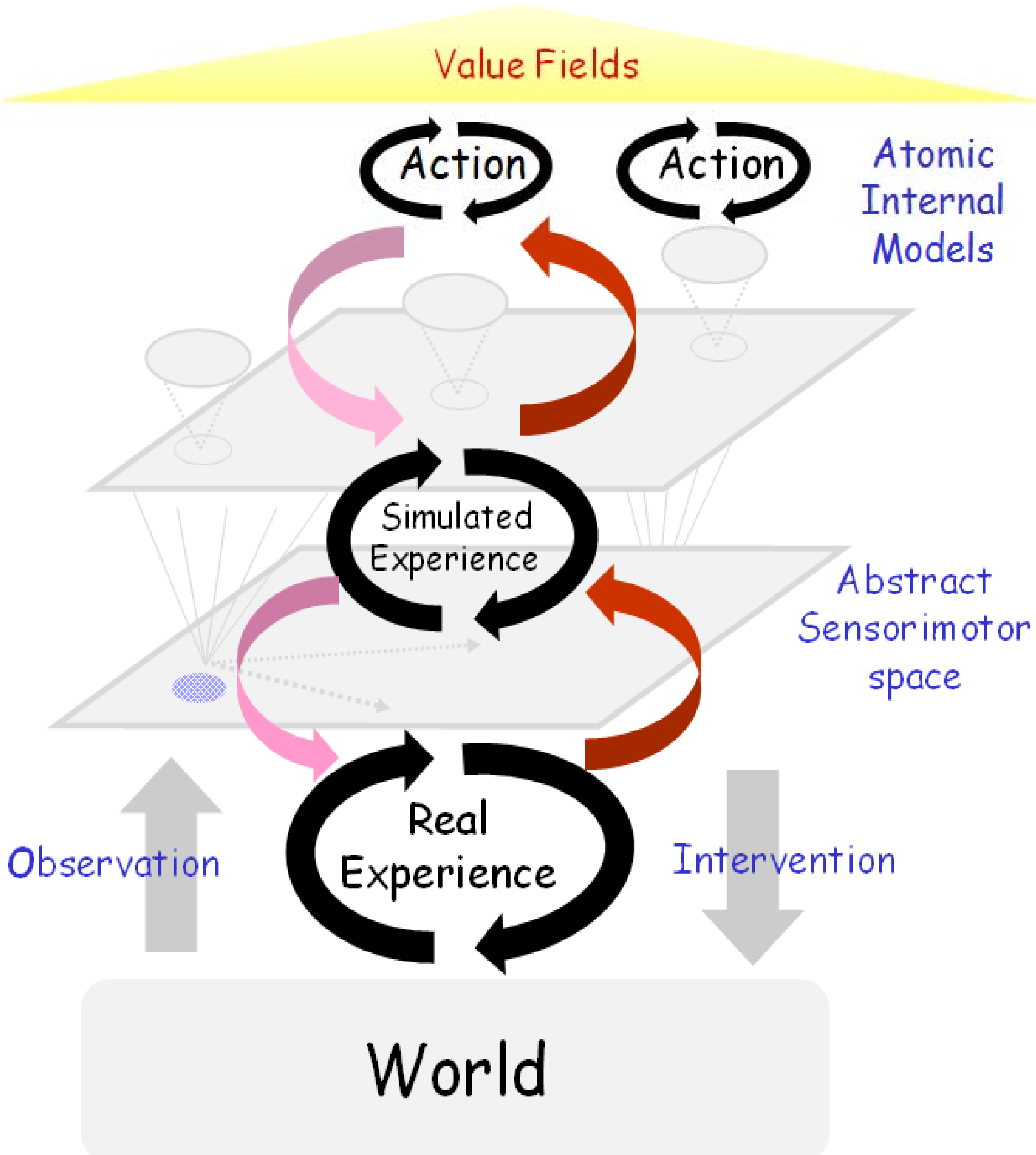
Abstraction and Modularity





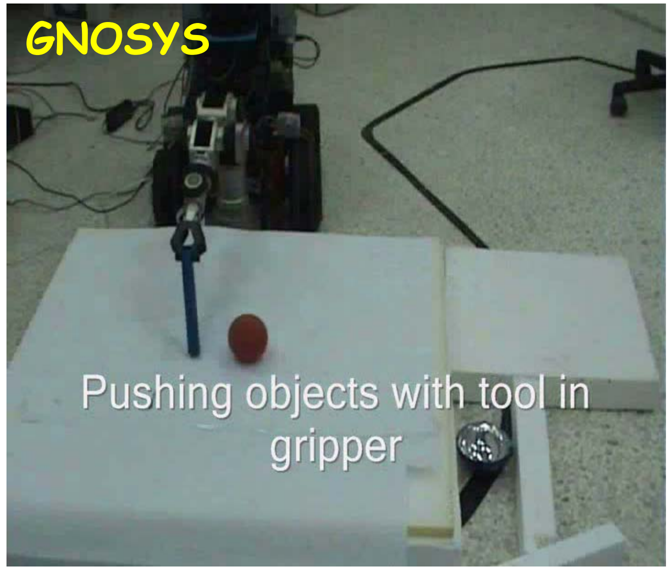
CNOSYS An Abstraction Architecture for Cognitive Agents

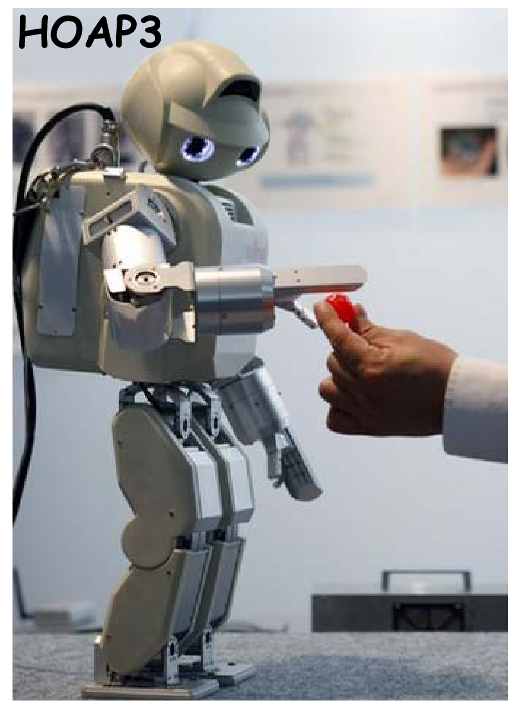
Circularity and Recursivity



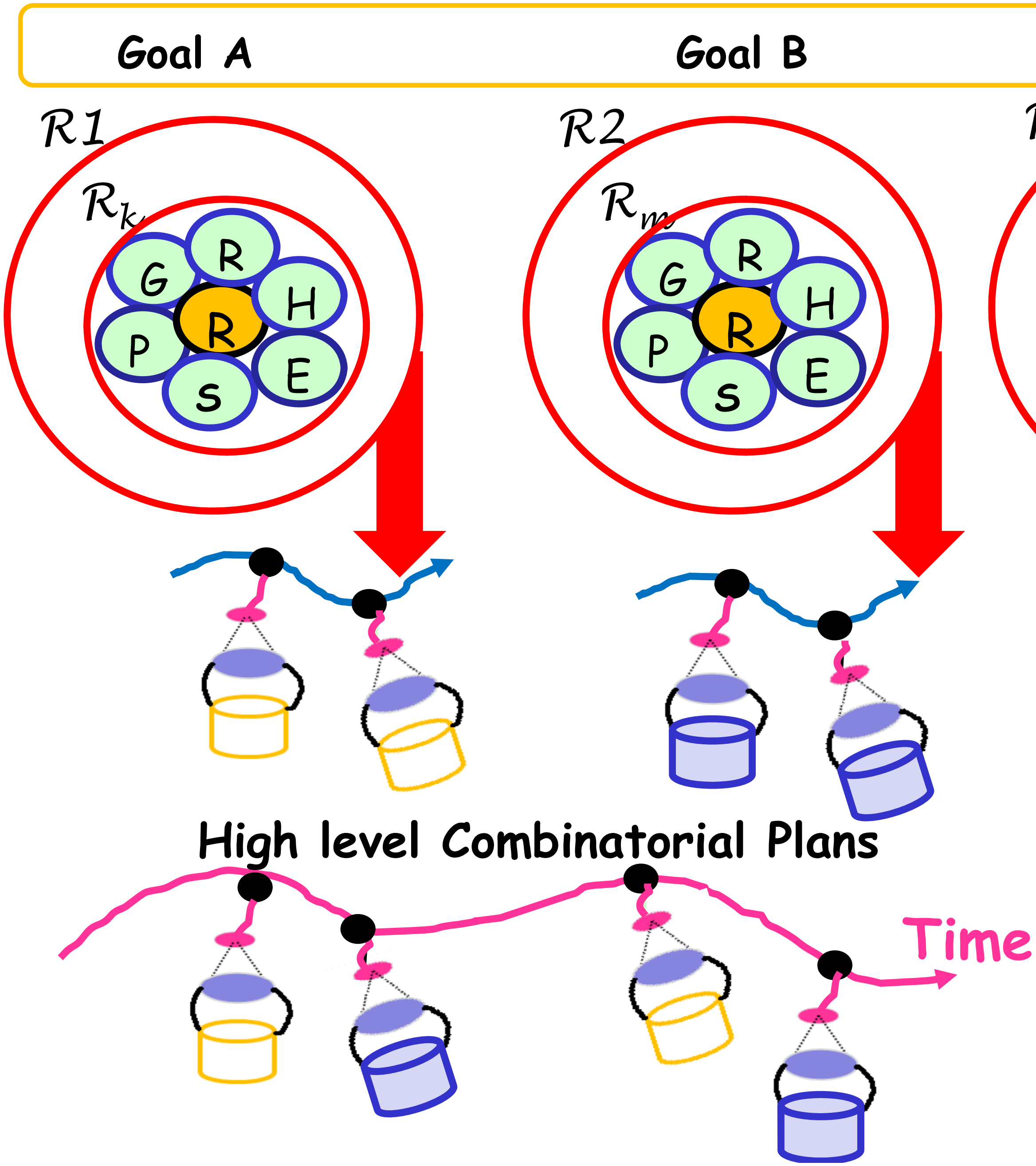

```
Root Goal
   ↓
Reason
  {
    Reason
      {
        Reason
          {
          }
        }
      }
```

Portablility and Scalability





When Goals Compete for the Body…..



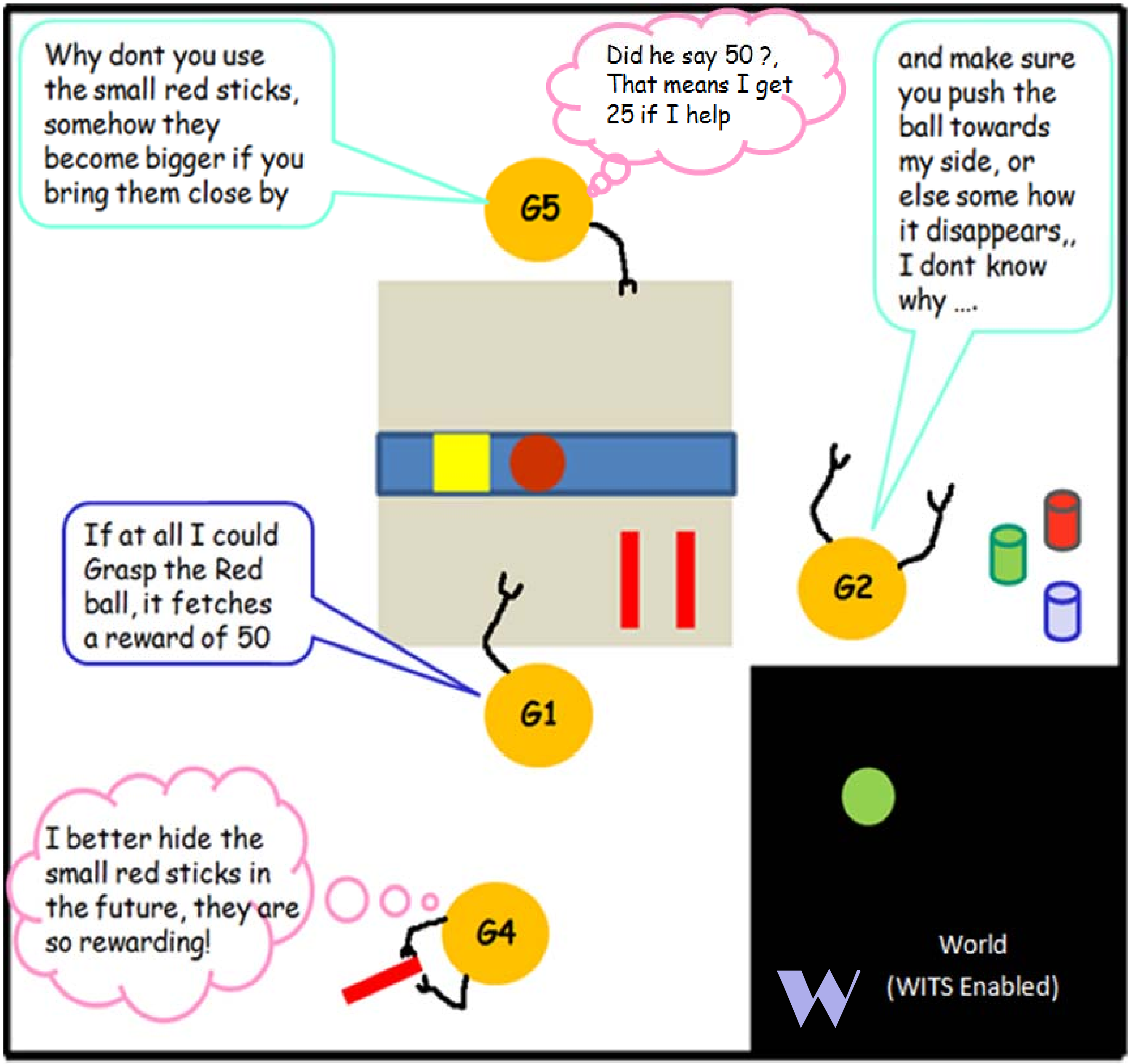
Force - Flows

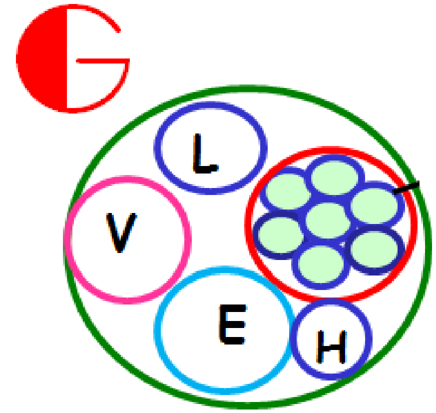
Situations - Actions

Goals - Plans

Motivation- Reasoning-
Executive loop

When bodies Compete for the World…..





W → G

Agents Compete

G → R

Goals Compete

R → A

Actions Compete

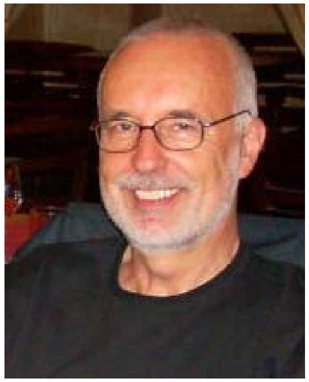
A → F

Fields Compete

GNOSYS : An Abstraction Architecture for Cognitive agents

www.ics.forth.gr/gnosys or neurolab webpage (DIST, Univ.Genoa) for Publications,Deliverables, Movies


John Taylor

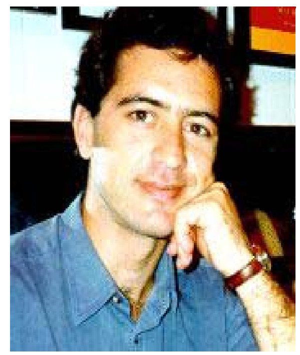

Pietro Morasso


Giulio Sandini


Giorgio Metta


Stathis Kasderidis


Hans Peter Mallot


Panos Trahanias


Harris Baltzakis


Christo Panchev


Wolfgang Heubner

KING'S College LONDON

FOUNDATION FOR RESEARCH AND TECHNOLOGY HELLAS

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

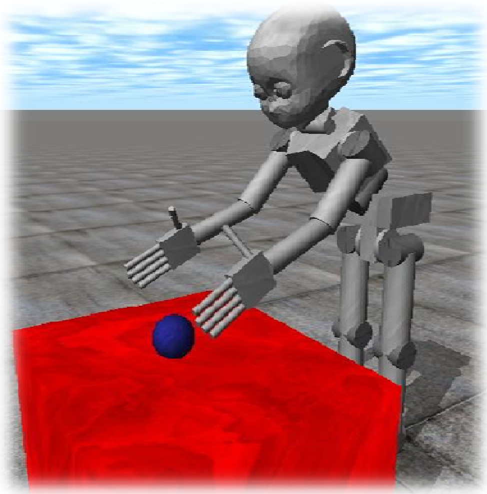
ZENON AUTOMATION TECHNOLOGIES

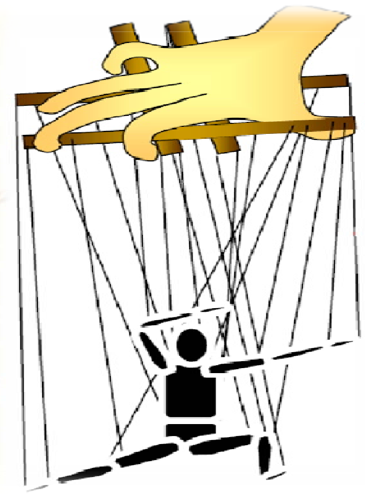
dist

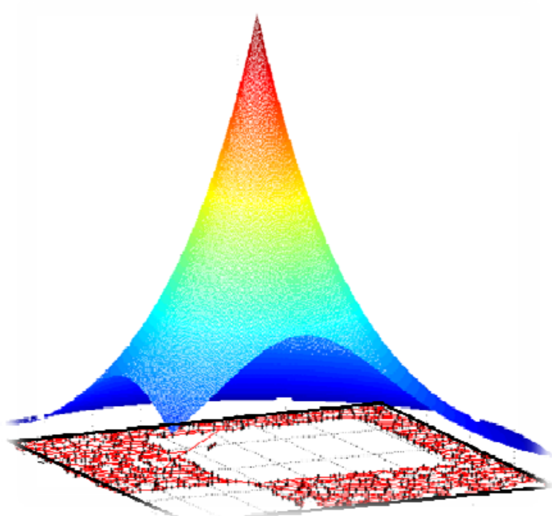
iit ISTITUTO ITALIANO DI TECNOLOGIA

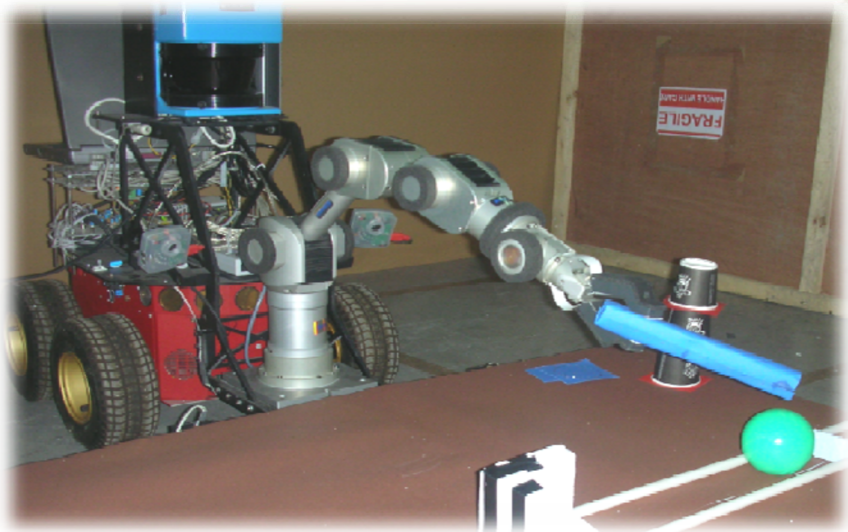
CogSys Cognitive Systems

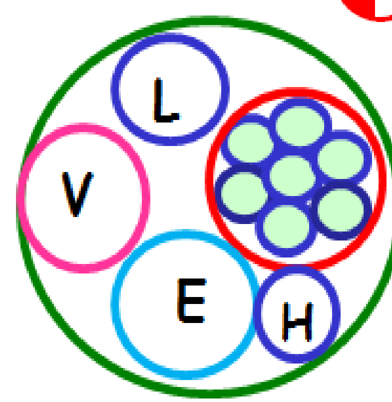
Thank You + Questions ?



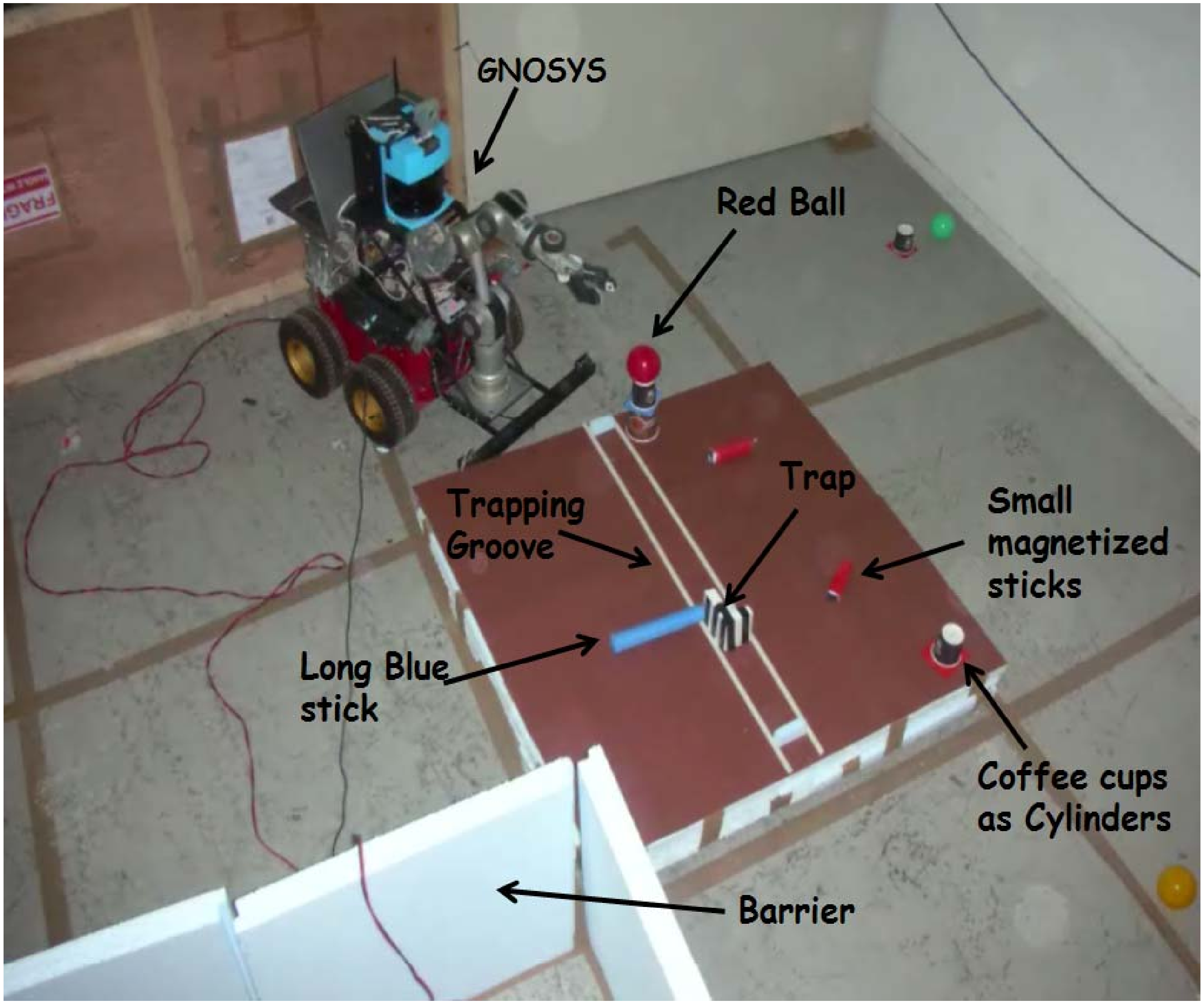








In the real world



Bimanual Reaching

DLT Based Calibertion

Stacking Coffee Cups

Bimanual stacking

Using Sticks, Pushing Internal Model

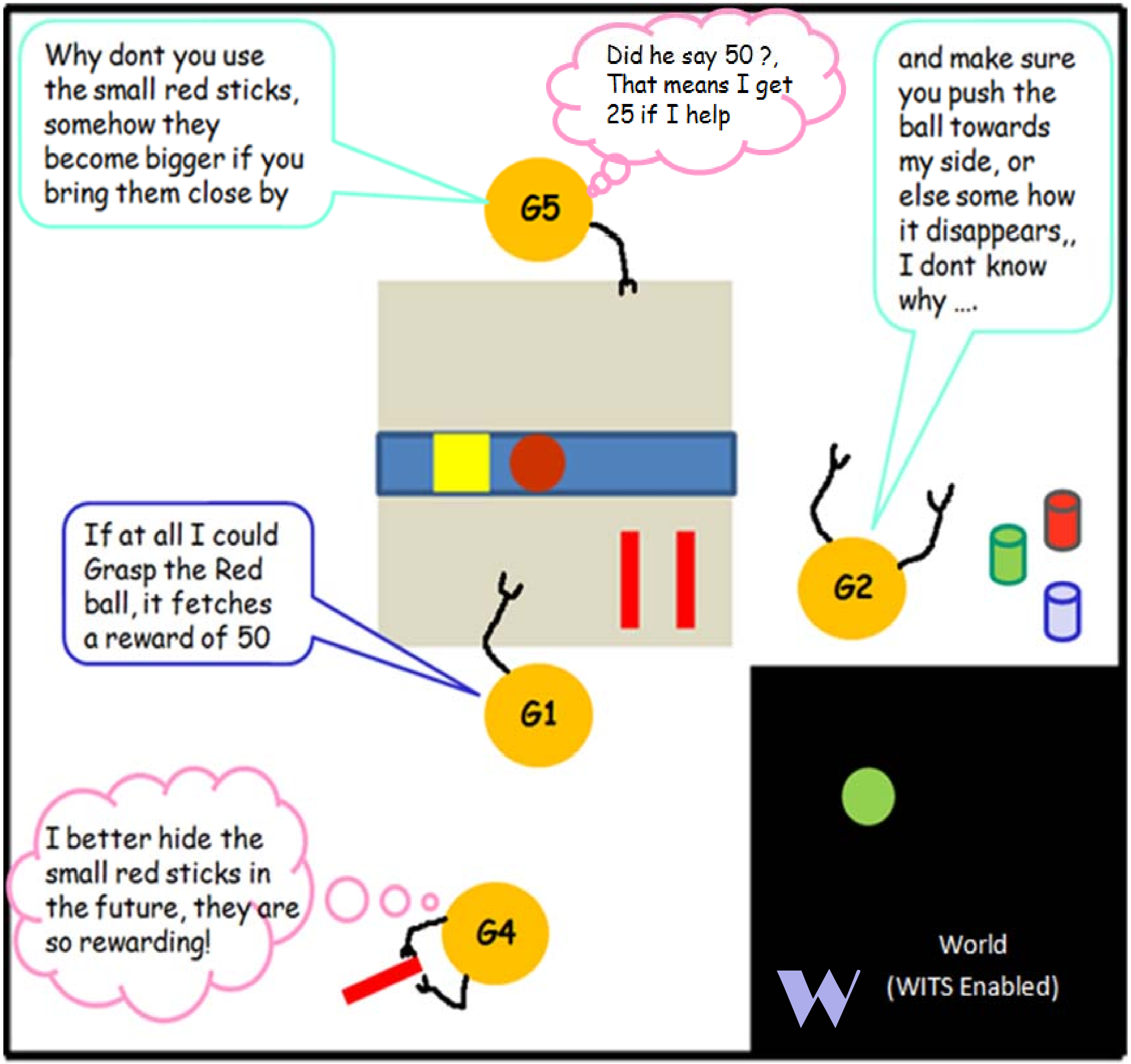
Addition of Internal Spatial Map (Mobile iCub ?)

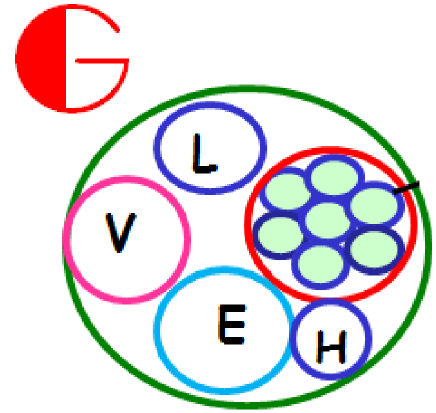
Reasoning Tasks

Parametrized PMP for Verb Grounding

Lingusitic goals

When bodies Compete for the World…..





W → G

Agents Compete

G → R

Goals Compete

R → A

Actions Compete

A → F

Fields Compete