

# Actions & Imagined Actions in Cognitive robots

Vishwanathan Mohan<sup>1</sup>, Pietro Morasso<sup>1</sup>, Giorgio Metta<sup>1</sup>, Stathis Kasderidis<sup>2</sup>

<sup>1</sup>Robotics Brain and Cognitive Sciences department, Italian Institute of Technology, Genoa, Italy

<sup>2</sup>Computational Vision and Robotics Lab, FORTH, Heraklion, Greece

**Abstract-** Natural/Artificial systems that are capable of utilizing thoughts at the service of their actions are gifted with the profound opportunity to mentally manipulate the causal structure of their physical interactions with the environment. A cognitive robot can in this way virtually reason about how an unstructured world should ‘change’, such that it becomes a little bit more conducive towards realization of its internal goals. In this article, we describe the various internal models for real/mental action generation developed in the GNOSYS Cognitive architecture and demonstrate how their coupled interactions can endow the GNOSYS robot with a preliminary ability to virtually manipulate neural activity in its mental space in order to initiate flexible goal directed behavior in its physical space. Making things more interesting (and computationally challenging) is the fact that the environment in which the robot seeks to achieve its goals consists of specially crafted ‘stick and ball’ versions of real experimental scenarios from animal reasoning (like tool use in chimps, novel tool construction in Caledonian crows, the classic trap tube paradigm and their possible combinations). We specifically focus on the progressive creation of the following internal models in the behavioral repertoire of the robot a) a passive motion paradigm based forward inverse model for mental simulation / real execution of goal directed arm (and arm+tool) movements; b) a spatial mental map of the playground; c) an internal model representing the causality of pushing objects and further learning to push intelligently in order to avoid randomly placed traps in the trapping groove. After presenting the computational architecture for the internal models, we demonstrate how the robot can use them to mentally compose a sequence of ‘Push-Move-Reach’ in order to Grasp (an otherwise unreachable) ball in its playground.

## 1 Introduction

The world we inhabit is an amalgamation of structure and chaos. There are regularities that could be exploited. Species biological or artificial, which do this best have the greatest chances of survival. We may not have the power of an ox or the mobility of an antelope but still our species surpasses all the rest in our flair by inventing new ways to think, new ways to functionally couple our bodies with the

structure afforded by our worlds. Simply stated, it is this ability to ‘explore, identify, internalize and exploit’ the possibilities afforded by the structure in one’s immediate environment to counteract limitations ‘of perceptions, actions and movements’ imposed by one’s embodied physical structure, and to do this in accordance with one’s ‘internal goals’, that forms the hallmark of any kind of cognitive behavior. In addition, natural/artificial systems that are capable of utilizing ‘thoughts’ at the service of their ‘actions’ are gifted with the profound opportunity to mentally manipulate the causal structure of their physical interactions with the environment. Complex bodies can in this way decouple behavior from direct control of the environment and react to situations that ‘do not really exist’ but ‘could exist’ as a result of their actions on the world. However, the computational basis of such cognitive processes have still remained elusive.

This is a difficult problem, but there are many pressures to provide a solution – from the intrinsic viewpoint of better understanding ourselves to creating artificial agents, robots, smart devices and machines that can reason and deal autonomously with our needs and with the peculiarities of the environments we inhabit and construct. This has led researchers towards several important questions regarding the nature of the computational substrate that could drive an artificial agent to exhibit flexible, purposeful and adaptive behavior in complex, novel and sometimes hostile environments. How do goals, constraints and choices ‘at multiple scales’ meet dynamically to give rise to the seemingly infinite fabric of reason and action? Is there an internal world model (of situations, actions, forces, causality, abstract concepts)? If yes, ‘How’ and ‘What’ is modeled, represented and connected? How are they invoked? What are the planning mechanisms? How are multiple internal models coordinated to generate (real/mental) sequences of behaviors ‘at appropriate times’ so as to realize valued goals? How should a robot respond to novelty and how can a robot exhibit novelty? What kind of search spaces (physical and mental) are involved and how are they constrained? This chapter is in many ways an exploration into some of these questions expressed through the life of a moderately complex robot ‘GNOSYS’ playing around in a moderately complex playground (which implicitly hosts artificially reconstructed scenarios inspired from animal cognition), trying to use its perceptions, actions and imaginations ‘flexibly and resourcefully’ so as to cater ‘rewarding’ user goals.

In spite of extensive research in multiple fields, scattered across multiple scientific disciplines, it is fair to say that the present day artificial agents still lack much of the resourcefulness, purposefulness, flexibility and adaptability that humans so effortlessly exhibit. Cognitive agent architectures are found in the current literature, ranging from purely reactive ones implementing the cycle of perception and action in a simplistic hardwired way to more advanced models of perception, state estimation and action generation (Brooks, 1986; Georgeff, 1999, 1987; Toussaint, 2006; Shanahan, 2005, 2006; Grossberg, 2008; Sun 2007, CLARION architecture), architectures for analogy making (Hopfstader, 1984; French, 1995; Kokinov and Petrov 2001), causal learning (Pearl, 1998; Geffner, 1992), probabilistic/ statistical inference (Yuille et al, 2006, Pearl 1988) and brain based devices (Edelman et al 2007, DARWIN Series). Even though symbols and

symbol manipulation have been the main stay of cognitive sciences (Newell and Simon, 1976) ever since the days of its early incarnations as AI, the disembodied nature of traditional symbolic systems, the need to presuppose explicit representations, symbol grounding and all other associated problems discussed in Sun (2000) have been troubling many cognitive scientists (Varela and Maturana, 1974; Churchland, 1986).

This led to the realization of the need for *experience* to precede *representation*, in other words the emergence of representational content as a consequence of sensory-motor interactions of the agent with its environment, a view that can be traced back to many different contributions spanning the previous decades, e.g. Wiener's Cybernetics (1948), Gibson's ecological psychology (1966), Maturana and Varela's autopoiesis (1974), Beer's neuroethology (1990), Clark's situatedness (1997). In this view, adaptive behaviour can best be understood within the context of the (biomechanics of the) body, the (structure of the organism's) environment, and the continuous exchange of signals/energy between the nervous system, the body and the environment. Hence the appropriate question to ask is not what the neural basis of adaptive behaviour is, but what the contributions of all components of the coupled system to adaptive behaviour and their mutual interactions are (Morasso 2006c).

In other words, the ability to autonomously explore, identify, internalize and exploit possibilities afforded by the structure in one's immediate environment is critical for an artificial agent to exercise intelligent behaviour in a messy world of objects, choices, relationships. Intelligent agents during the course of their lifetimes gradually master this ability of coherently integrating the information from the bottom (sensory, perceptual, conceptual) with the drives from the top (user goals, self goals, reward expectancy), thereby initiating actions that are maximally rewarding. A major part of this process of transformation takes place in the mental space (Holland and Goodman, 2003) where in the agent, with the help of an acquired internal model, executes virtual actions and simulates the usefulness of their consequences towards achieving the active goal. Hence, unlike a purely reactive system where the motor output is exclusively controlled by the actual sensory input, the idea that a cognitive system must be capable of mentally simulating action sequences aimed at achieving a goal has been gaining prominence in literature. This also resonates very well with emerging biological evidence in support of the simulation hypothesis towards generation of cognitive behaviour, mainly *simulation of action*: we are able to activate motor structures of the brain in a way that resembles activity during a normal action but does not cause any overt movement (Metzinger and Gallese, 2003; Grush 2004); *simulation of perception*: imagining perceiving something is actually similar to the perceiving it in reality, only difference being that, the perceptual activity is generated by the brain itself rather than by external stimuli (Grush, 1995); *anticipation*: there exist associative mechanisms that enable both behavioural and perceptual activity to elicit other perceptual activity in the sensory areas of the brain. Most important, a simulated action can elicit perceptual activity that resembles the activity that would have occurred if the action had actually been performed (Hesslow, 2002).

Computationally this implies the need to have two different kinds of loops in the agent architecture, firstly a situation-action-consequence loop or forward model that allows contemplated decision making (without actual execution of action) and secondly a Situation-Goal-Action loop to solve the inverse problem of finding action sets which map the transformation from initial condition to active goal. That such forward models of the motor system occur in the brain has been demonstrated by numerous authors. For example Shadmehr (1999) has shown how adaptation to novel force fields by humans is only explicable in terms of both an inverse controller and a learnable forward model. More recent work has proposed methods by which such forward models can be used in planning (where actual motor action is inhibited during the running of the forward model) or in developing a model of the actions of another person (Oztop, Wolpert and Kawato, 2004). Engineering Control frameworks of attention, using modules of control theory (Taylor, 2000) extended so as to be implemented using neural networks, have been extensively applied to modeling motor control in the brain (Morasso, 1981; Wolpert, Ghahramani & Jordan 1994, Morasso & Sanguineti 1997; Gribble et al 1998; Desmurget & Grafton, 2000; Miall & Wolpert, 1996; Kawato, 1999; Wolpert & Kawato, 1998; Imamizu, 2000), with considerable explanatory success. Such planning has been analysed in these and numerous other publications for motor control and actions but not for more general thinking, especially including reasoning. Nor has the increasingly extensive literature on imagining motor actions been appealed to: it is important to incorporate how motor actions are imagined as taking place on imagined objects, so as to 'reason' what objects and actions are optimally rewarding. Others have also emphasized the need to combine working memory modules for imagining future events with forward models, for example the process termed 'prospection' in (Emery & Clayton, 2004). Guided by the experimental results from functional imaging and neuropsychology, computational architectures have recently begun to emerge in the literature for open-ended, goal-directed reasoning in artificial agents, most importantly incorporating the creation and use of internal models and motor imagery. A variety of computational architectures incorporating these ideas have been proposed recently, for example an architecture that combines internal simulation with a global workspace (Shanahan 2005), IAM (Internal Agent Model) theory of consciousness Holland (2003), learning a world model using interacting self-organizing maps (Toussiant, 2006, 2004), learning motor sequences using recurrent neural networks with parametric bias (Tani et al, 2007).

The idea of using internal models to aid generation of intelligent behavior also resonates very well with compelling evidence from several neuropsychological, electrophysiological and functional imaging studies, which suggest that much of the same neural substrates underlying modality perception are also used in imagery; and imagery, in many ways, can 'stand in' for (represent, if you will) a perceptual stimulus or situation (Zattore et al, 2007; Berhmann, 2000; Fuster 2003). Studies show that imagining a visual stimulus or performing a task that requires visualization is accompanied by increased activity in the primary visual cortex (Kosslyn et al, 2006; Klein and Le Bihan, 2000). The

same seems to be true for specialized secondary visual areas like fusiform gyrus, an area in the occipito-temporal cortex which is activated both when we see faces (Op de Beeck and Kanwisher 2008) and also when we imagine them (O'Craven et al, 2000). Lesions that include this area impair both face recognition (Damasio et al, 1990) and the ability to imagine faces. Brain imaging studies also illustrate heavy engagement of the motor system in mental imagery i.e. we are able to activate motor structures of the brain in a ways that resembles activity during a normal action but does not cause any overt movement (Parsons et al, 2005; Rizzolati et al 2001, Grush, 2004). EEG recordings on subjects performing mental rotation tasks have revealed activation of premotor and parietal cortical areas, indicating that they may be performing covert mental simulation of actions by engaging the same motor cortical areas that are used for real action execution (Williams et al, 1995). FMRI studies have similarly found activation of the supplementary motor area as well as of the parietal cortex during mental rotation (Cohen et al, 1996). Similar results have also been obtained from experiments that involve auditory imagery of melodies that activates both the superior temporal gyrus (an area crucial for auditory perception) and the supplementary motor areas. Further, metallization also affects the autonomic nervous system, the emotional centers and the body in same ways as actual perceptual experiences (Damasio, 2000).

To summarize, the increasing complexity of our society and economy places great emphasis on developing artificial agents, robots, smart devices and machines that can reason and deal autonomously with our needs and with the peculiarities of the environments we inhabit and construct. On the other hand, considerable progress in brain science, emergence of internal model based theories of cognition and experimental results from animal reasoning has resulted in tremendous interest of the scientific community towards investigation of higher level cognitive functions using autonomous robots as tools. Rapid increase in robots' computing capabilities, quality of their mechanical components and subsequent development of several interesting (and complicated) robotic platforms, for example, Cog (Brooks, 1997) with 21 DOFs (Degrees of Freedom), DB (Atkeson et al, 2000) with 30 DoFs, Asimo (Hirose and Ogawa, 2007) with 34 DoFs, H7 (Nishiwaki et al, 2007) with 35 DoFs, iCub (Metta, Natale et al 2007) with 53 DoFs raise the challenge to propose concrete computational models for reasoning and action generation capable of driving these systems to exhibit purposeful, intelligent response and develop new skills for structural coupling with their environments.

The computational machinery driving the action generation system of the GNOSYS robot presented in this chapter contributes solutions to a number of issues that need to be solved to realize these competences:

- a) account for forward / inverse functions of sensorimotor dependencies for a range of motor actions / action sequences
- b) provide a proper neural representation to realize goal directed planning, virtual experiments, and reward related computations

- c) capable of learning the state representations (sensory/motor) by exploration (and importantly without hand coded states, unrealistic assumptions in data acquisition)
- d) models that are scalable (wrt dimensionality) and have an organized way to deal with novelty in state space
- e) plastic and capable of representing/dealing with dynamic changes in the environment
- f) capable of accommodating heterogeneous optimality criteria in a goal dependent fashion (and not being governed by a single predefined minimization principle to constrain solution space / resolve redundancy)
- g) built in mechanisms for temporal synchrony and maintenance of continuity in perception, action and time
- h) a clear framework for the integration of three important streams of information in any cognitive system: the top down (simulated sensorimotor information), the bottom up (real sensorimotor information) and the active goal.
- i) Using the measure of coherence between these informational streams to alter behavior from normal dynamics to explorative dynamics, with a goal to maintain psycho-logical consistency in the sensorimotor world
- j) demonstrate the effectiveness of the architecture in a physical instantiation that allows active sensing/autonomous movement in ecologically realistic environments that permits comparisons to be made with experimental data acquired from animal nervous systems, animal reasoning tasks

The rest of the chapter is organized as follows: section 2 presents a general overview of the environmental set up we constructed for training/ validating the reasoning-action generation system of the GNOSYS robot, experiments from animal reasoning that inspired the design of the playground and the intricacies involved in different scenarios that the environment implicitly affords to the robot during phases of user goal/curiosity driven explorative play. Section 3 presents an concise overview of the forward/inverse model for simulating/executing a range of goal directed arm (and arm+tool) movements. Section 4 describes how a spatial map of the playground and an internal model for pushing objects is learnt by the GNOSYS robot with specific focus on acquisition, dynamics, generation of goal directed motor behavior and dealing with dynamic changes in the world. How these internal models can operate unitedly in the context of an active goal is the major focus of section 5. A discussion concludes.

## 2 The GNOSYS playground

Emerging experimental studies from animal cognition reveal many interesting behaviours demonstrated by animals that have shades of manipulative tactics, mental swiftness and social sophistication commonly attributed to humans. Such experiments generally focus on many open problems that are of great interest to the cognitive robotics community, mainly attention, categorization, memory, spatial cognition, tool use, problem solving, reasoning, language and consciousness. Seeing a tool using chimp or a tool making corvid [1,2] often falls short of astonishing us unless we question their computational basis or try to make robots do similar tasks that we often take as granted for humans. The advantages of creating a rich sensorimotor world for a cognitive robot are several a) Facilitate exploration-driven development of different sensorimotor contingencies of the robot; b) development of goal-dependent value systems; c) Allow realistic and experience driven internal representation of different cause effect relations, outcomes of interventions;

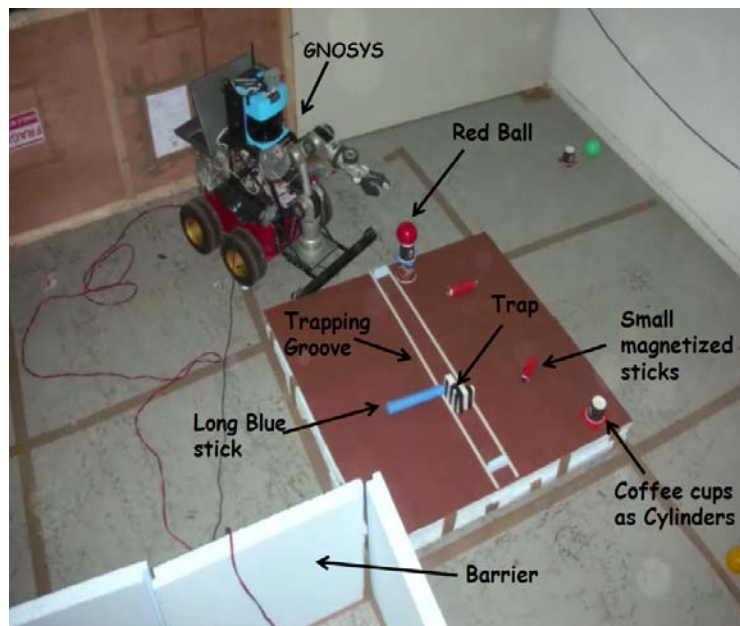


Figure 1. 3X3m GNOSYS playground with different randomly placed goal objects and tools.

d) Aid the designer to understand various computational mechanisms that may be in play (and should be incorporated in the cognitive architecture) based on the amazingly infinite ways by which goals may be realized in different scenarios; e) serve as a test bed to evaluate the performance of the system as a whole and make comparisons of the robot's behaviour with that of real organisms, other cognitive

architectures. Guided by experiments from animal reasoning, we constructed a playground for GNOSYS robot that 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. As seen in figure 1, The GNOSYS playground is a 3×3m enclosure (every square approx. 1m<sup>2</sup>) with goal objects placed at arbitrary locations on the floor and on the centrally placed table. Objects like cylinders of various sizes, sticks of different lengths (possible tools to reach/push otherwise unreachable goal objects), and balls are generally placed randomly in the environment. Among the available sticks, the small red sticks are magnetized. Hence the robot can discover (through intervention) an additional affordance of making even longer sticks using them. Further, as seen in the figure 1, a horizontal groove is cut and run across the table from one side to another, which enables the robot to slide sticks (a grasped tool) along the groove to push out a rewarding object to the edge of the table (this could eventually result in spatial activations that drive the robot to move to the edge of the table closest to the object). Moreover, traps could be placed all along the groove so as to prevent the reward from moving to the edges of the table when pushed by the robot (similar to the trap tube paradigm) hence blocking the action initiated by the robot and forcing it to change its strategy intelligently (and internalize the causal effect of traps);

The environment was designed to implicitly host three specific experiments from animal cognition studies (and their combinations):

a) *the n-stick paradigm* : It is a slightly more complicated version of the task in which the animal reasons about using a nearby stick as a tool to reach a food reward that was not directly reachable with its end-effector [3]. The two-sticks paradigm for example, involves two sorts of sticks: Stk1 (short), and Stk2 (long), one of each being present on a given trial, only the small one being immediately available, and the food reward only being reachable by means of the larger stick. We can easily see that a moderately complex sequence of actions involving tool use, pushing, reaching, grasping is required to grasp a goal object under the two stick paradigm scenario. Both sticks and long cylinders could be opportunistically exploited by the robot as tools in different environmental scenarios.

b) *Betty's Hook shaping task*: If the previous task was about exploiting tools, this experiment relates to a primitive case of making a simple tool (based on past experience) to realize an otherwise unrealisable goal. This scenario is a 'stick and ball' adaptation of an interesting case of novelty in behaviour demonstrated by Betty, the Caledonian crow who lived and "performed" in Oxford under the discrete scrutiny of animal psychologists[1,2]. She exploited her past experience of playing with flexible pipe cleaners to make a hook shaped wire tool out of a straight wire in order to pull her food basket from a transparent vertical tube. The magnetized small sticks were introduced in the playground so that the robot could learn (accidentally) their special utility and use them creatively when nothing else works. Computationally it implies making a cognitive architecture that enables a robotic artefact to reason about things that do not exist, but could exist as a result of its actions on the world.

c) *Trap Tube Paradigm*: The trap tube task is an extremely interesting experimental paradigm that has been conducted on several species of monkeys and children (between 24-65 months), with an aim to investigate the level of understanding they have about the solution they employ to succeed in the task [4,5]. Of course a robot that is capable of realizing goals under the previous two scenarios is going to fail in the when traps are introduced in the groove (at least during the initial trials). This failure contradicts robot's earlier experiences of carrying out the same actions, for which it was actively rewarded. Can this contradiction at the level of reward values be used to trigger higher levels of reasoning and/or exploration activities in order to seek the cause of failure? To achieve this computationally, the robot must have at least the following three capabilities:

- (a) achieving awareness that, for some reason, the physical world works differently from the mental (simulated) world;
- (b) identifying the new variables in the environment that determine this inconsistency (in the trap tube case the robot should discover that the essential novelty are the holes/traps introduced by the experimenter);
- (c) initiating new actions that can block the effect of this new environmental variable (change the direction of pushing the ball i.e. away from the hole/trap in the simplest case).

In this environmental layout, the robot is asked to pursue relatively simple high level user goals like reaching, grasping, stacking, pushing and fetching different objects. The interesting fact is that even though the high level goals are simple, the complexity of the reasoning process (and subsequent action generation) needed to successfully realize these goals increases more than proportionately with the complexity of the environment in which the goal is attempted. Further, using a set of few sticks, balls, traps and cylinders and combining/placing them in different ways, an enormous amount of complex environmental situations can be created, the only limitation being the imagination of the experimenter itself.

### **3 Forward/Inverse model for reaching: The Passive Motion Paradigm**

The action of 'reaching' is fundamental for any kind of goal directed interaction between the body and the world. Tasks and goals are specified at a rather high, often symbolic level ("Stack 2 cylinders", "Grasp the red ball" etc.) but the motor system faces the daunting and under-specified task of eventually working out the problem at a much more detailed level in order to specify the activations which lead to joint rotations, movement trajectory in space, and interaction forces. In addition to dealing with kinematic redundancies, the generated action must be compatible with a multitude of constraints: internal, external, task specific and their possible combinations. In this section, we describe

the forward/inverse model for reaching that coordinates arm/tool movements in the GNOSYS robot during any kind of manual interaction with the environment.

The central theme behind the formulation of the forward inverse models is the observation that motor commands for any kind of motor action, for any configuration of limbs and for any degree of redundancy can be obtained by an “internal simulation” of a “passive motion” induced by a “virtual force field” (Mussa Ivaldi et al, 1988) applied to a small number of task-relevant parts of the body. Here “internal simulation” identifies the relaxation to equilibrium of an internal model of limb (arm, leg etc, according to the specific task); “passive motion” means that the joint rotation patterns are not specifically computed in order to accomplish a goal but are the indirect consequence of the interaction between the internal model of the limb and the force field generated by the target, i.e. the intended/attended goal. The model is based on non-linear attractor dynamics where the attractor landscape is obtained by combining multiple force fields in different reference systems. The process of relaxation in the attractor landscape is similar to coordinating the movements of a puppet by means of attached strings, the strings in our case being the virtual force fields generated by the intended/attended goal and the other task dependent combinations of constraints involved in the execution of the task.

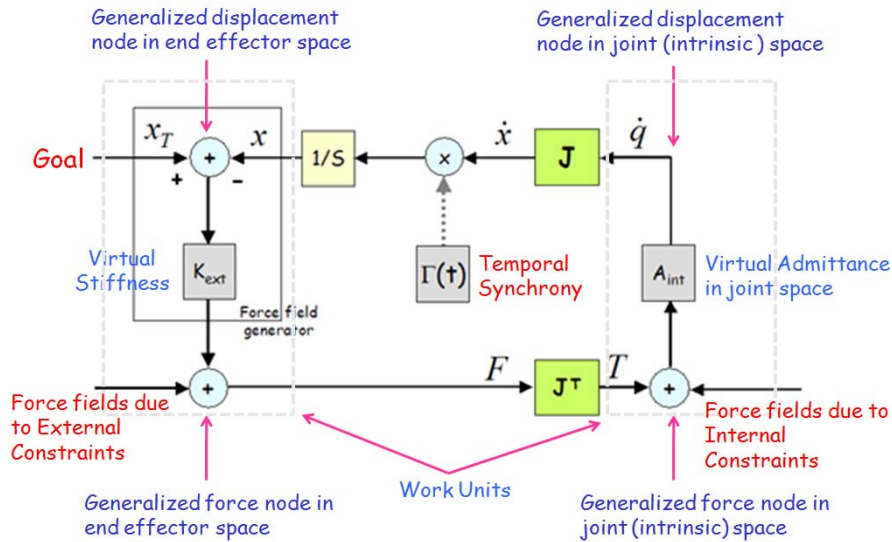


Figure 2. Basic computational scheme of the PMP for a simple kinematic chain.  $x$  is the position/orientation of the end-effector, expressed in the extrinsic space;  $x_T$  is the corresponding target;  $q$  is the vector of joint angles in the intrinsic space;  $J$  is the Jacobian matrix of the kinematic transformation  $x=f(q)$ ;  $K_{ext}$  is a virtual stiffness that determines the shape of the attractive force field to the target; “external constraints” are expressed as force fields in the extrinsic space; “internal constraints” are expressed as force fields in the intrinsic space;  $A_{int}$  is a virtual admittance that distributes the relaxation motion to equilibrium to the different joints;  $\Gamma(t)$  is the time-varying gain that implements the terminal attractor dynamics.

As shown in figure 2, the basic structure of the forward inverse models is composed of a fully connected network of nodes either representing forces or representing flows (displacements) in different motor spaces (end-effector space, joint space, muscle space, tool space etc). We also observe that a displacement and force node belonging each motor space is grouped as a work (force. displacement) unit (WU). There are only two kinds of connections 1) between a force and displacement node belonging to WU that describes the elastic causality of the coordinated system (determined by the stiffness and admittance matrices) and 2) between two different motor spaces that describes the geometric causality of the coordinated system (Jacobian matrix).

Let  $x$  be the vector that identifies the pose of the end-effector of a robot in the extrinsic workspace and  $q$  the vector that identifies the configuration of the robot in the intrinsic joint space:  $x = k(q)$  is the kinematic transformation that can be expressed, for each time instant, as follows:  $\dot{x} = J(q) \cdot \dot{q}$  where  $J(q)$  is the Jacobian matrix of the transformation. The motor planner/controller, which expresses in computational terms the PMP, is defined by the following steps that are also represented graphically by the PMP network of fig 1:

- 1) Associate to the designated target  $x_T$  an attractive force field in the extrinsic space

$$F = K_{ext} (x_T - x) \quad (1)$$

where  $K_{ext}$  is the virtual impedance matrix in the extrinsic space. The intensity of this force decreases monotonically as the end-effector approaches the target.

- 2) Map the force field into an equivalent torque field in the intrinsic space, according to the principle of virtual works:

$$T = J^T F \quad (2)$$

Also the intensity of this torque vector decreases as the end-effector approaches the target.

- 3) Relax the arm configuration in the applied field:

$$\dot{q} = A_{int} \cdot T \quad (3)$$

where  $A_{int}$  is the virtual admittance matrix in the intrinsic space: the implicit or explicit modulation of this matrix affects the relative contributions of the different joints to the reaching movement.

- 4) Map the arm movement into the extrinsic workspace:

$$\dot{x} = J \cdot \dot{q} \quad (4)$$

- 5) Integrate over time until equilibrium

$$x(t) = \int_{t_0}^t J \dot{q} d\tau \quad (5)$$

Kinematic inversion is achieved through well posed direct computations and no

predefined cost functions are necessary to account for motor redundancy. While the forward model maps tentative trajectories in the joint space into the corresponding trajectories of the end-effector variables in the workspace), the inverse model maps desired trajectories of the end-effector into feasible trajectories in the joint space. The timing of the relaxation process can be controlled by using a TBG (Time Base Generator [8] and the concept of terminal attractor dynamics [9]: this can be simply implemented by substituting the relaxation equation (4) with the following one:

$$\dot{q} = \Gamma(t) \cdot B \cdot T \quad (6)$$

where a possible form of the TBG or time-varying gain that implements the terminal attractor dynamics is the following one (it uses a minimum-jerk generator with duration  $\tau$ ):

$$\Gamma(t) = \frac{\dot{\xi}}{1 - \xi} \quad (7)$$

where

$$\xi(t) = 6(t/\tau)^5 - 15(t/\tau)^4 + 10(t/\tau)^3 \quad (8)$$

In general, a TBG can also be used as a computational tool for synchronizing multiple relaxations in composite PMP networks, coordinating relaxation of movements of two arms or even the movements of two robots. The algorithm always converges to an equilibrium state, in finite time (that is set using the TBG) under the following conditions:

A. When the end-effector reaches the target, thus reducing to 0 the force field in the extrinsic space (1);

B. When the force field in the intrinsic space becomes zero (2), although the force field in the extrinsic space is not null and this can happen in the neighbourhood of kinematic singularities;

Case (A) is the condition of success termination. But also in case (B), in which the target cannot be reached for example because it is outside the workspace, the final configuration has a functional meaning for the motion planner because it encodes geometric information valuable for re-planning (breaking an action into a sequence of sub-actions like using a tool of appropriate length [6]).

Multiple constraints can be concurrently imposed in a task-dependent fashion by building composite F/I models ( in other words simply switching on/off different task relevant force field generators). In the composite F/I model of figure 3, there are three weighted, superimposed force fields that shape the spatio-temporal behavior of the system.

1. To the end-effector (to reach the target);
2. To the wrist (for proper orientation);
3. A force field in joint space as internal constraints of Joint limits.

The same TBG coordinates all the three relaxation processes. This composite PMP network is effective in tasks like grasping a stick placed in the table, with a



## 4 Spatial Map and Pushing Sensorimotor space

A large body of neuroanatomical and behavioral data acquired from experiments conducted on mammals (primarily rodents), suggest involvement of a range of neural systems being involved in spatial memory and planning, like the head direction cells (Blair et al, 1998), spatial view cells (Georges-Francois et al, 1999), hippocampal place cells (O'Keefe and Dostrovsky, 1971) that exhibit a high rate of firing whenever an animal is in a specific location in the environment corresponding to the cell's "place field" and the recently found grid cells located in the entorhinal cortex in rats, known to constitute a mental map of the spatial environment (Hafting et al, 2005). Like animals, the GNOSYS Robot also faces problems related to learning a mental map of the spatial topology of its environment and use it in coordination with the forward/inverse models for arm to realize goals in more complex scenarios. In addition, it also needs to learn the causality of pushing objects in the trapping groove using sticks. The spatial map and the pushing internal model essentially share the same computational substrate with the only difference being the sensorimotor variables that are at play in the two internal models. Hence we describe the two internal models jointly in this section. The computational architecture for the development of these internal models and the associated dynamics (that organize goal oriented behavior) is novel and brings together several interesting ideas from the theory of self organizing systems [15], their extensions to growing maps [12], neural field dynamics [13], sensorimotor maps [10,11], reinforcement learning [16] and temporal hebbian learning [14]. For reasons of space, we restrict ourselves to the following issues in this chapter:

- a) Learning the Sensorimotor space (through self organization of sequences of randomly generated sensory motor data)
- b) Dynamics of the Sensorimotor space (SMS): How activity moves bidirectionally between sensory and motor units
- c) Value Field Dynamics: How activity moves bidirectionally between sensory and motor units in a 'goal directed fashion'
- d) Dealing with dynamic changes in the world, cognitive dissonance (for example, learning to nullify the effect of traps in the trapping groove)

### 4.1. Acquisition of the Sensorimotor space

The sensorimotor variables for the spatial map are relatively straight forward, the sensory space composed of the global location of the robot in the playground (x–y coordinates and orientation) coming from the localization system [18], the motor space is two dimensional, composed of translation commands appropriately converted into speed set commands communicated to low level hardware. For the pushing internal model, sensory information coded is the location of the object (being pushed). This information is derived after a visual scene analysis using the GNOSYS visual modules and reconstructed into 3D space coordinates using a

motor babbling based algorithm [6]. The function of the visual modules is out of scope for discussion in this article and interested reader may refer to [18] for further information on this issue. The motor space consists of the following variables (shown in figure 5) : a) Location of the tool with respect to Goal and b) The amount of force applied to the object. This is approximately proportional to the change in the DOF  $\theta_1$  and  $\theta_5$  of the KATANA arm of the robot.

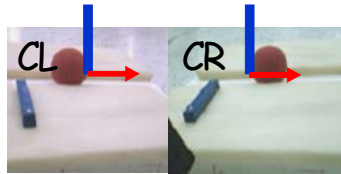


Figure 5. Pushing to the right in the case of CL will not induce any motion on the ball. Pushing to the right in the case CR will displace the ball based on the amount of force applied (i.e. approximately equal the displacement of the stick in contact with the ball along the trapping groove).

Figure 6 shows the general computational structure for the pushing and moving related internal models. The central element of the architecture is a growing intermediate neural layer common to both perception and action, called the sensorimotor space, henceforth SMS (Toussaint et al 2006).

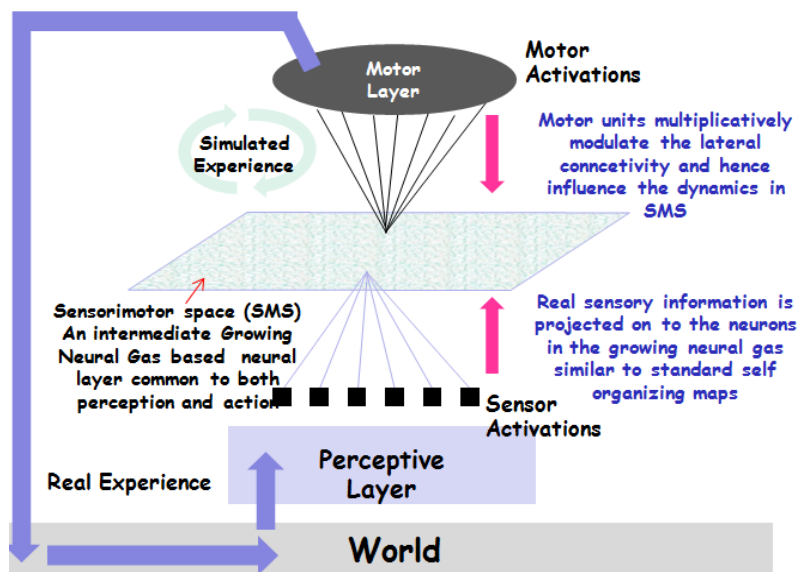


Figure 6. General computational structure for the spatial map and pushing internal model

This neural layer not only self organizes sequences of sensorimotor data generated by the robot through random motor explorations (through the loop of real experience) but also sub symbolically represents the forward inverse functions of various sensorimotor dependencies (that is encoded in the connectivity structure). Further, it also serves as a proper computational substrate to realize goal directed

planning (using quasistationary value fields) and perform ‘what if’ experiments in the mental space (through the loop of simulated experience shown in figure 6). During the process of learning the SMS, the simulated experience loop is turned off. In other words, the only loop active in the system is the loop of real experience. To learn the spatial mental map, the agent is allowed to move randomly in the play ground with a maximum translation of 14 cms and maximum rotation of 20 degrees in one time step ( in order to achieve the representational density necessary to perform motor tasks in future that require high precision). These movements generate the data i.e. sequences of sensory and motor signals  $S^{(t)}$  and  $M^{(t)}$  using which both the sensory weights, lateral connections between neurons and the motor weights of the motor modulated lateral connections are learnt. Both the SMS and the complete lateral connectivity structure is learnt from zero using sequences of sensor and motor data generated by the robot through a standard growing neural gas algorithm [12] and extended to encode motor information into the connectivity structure like the sensorimotor maps of Toussaint et al [10,11]. Hence, in addition to incrementally self organizing the state space based on incoming sensorial information ( like a standard GNG ), the motor information is also fully integrated with the SMS at all times of operation. As seen in figure 5, motor units project to lateral connections in between the neurons in the SMS and influence their dynamics. This allows motor activity to multiplicatively modulate these lateral connections hence cause anticipatory shifts in neural activity in the SMS similar to that which would have occurred if the action was actually performed. Moreover, provided that the world is consistent, both mental simulation (top down through motor modulated lateral connections) and real performance (bottom up through self organizing competition) should activate the same neural population in the SMS, the coherence between them forming the basis for the stability of the sensorimotor world of GNOSYS robot. Figure 7 shows the lateral topology of the SMS of the spatial map learnt by the robot after this initial phase of self organization on sequences of sensory motor data.

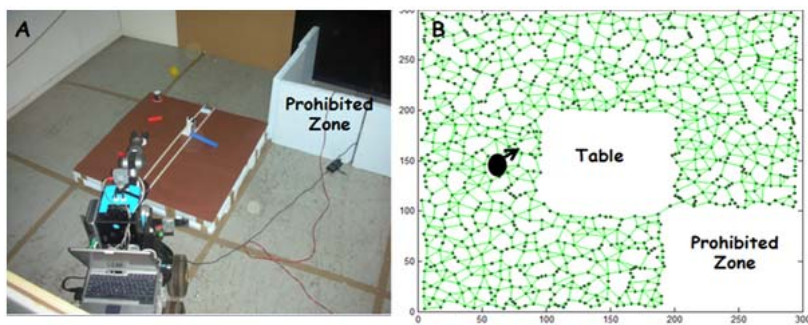


Figure 7. Lateral topology of the spatial map after 23350 iterations of self organization after which the map becomes almost stationary. Number of neurons=933;

Similar to the development of the SMS for spatial map, a growing SMS for pushing in the trapping groove was built using the data generated by repeated sequences of reaching a goal object with a stick (using the F/I Model pair for reaching), pushing in different directions (with different amount of force) and then tracking the new location of the ball. We simplify this scenario by considering pushing to be only functional along the horizontal axis. Figure 8 shows the internal spatial map of the Gnosys playground along with the SMS for pushing in the trapping groove. The other panels show a typical pushing sequence for data generation.

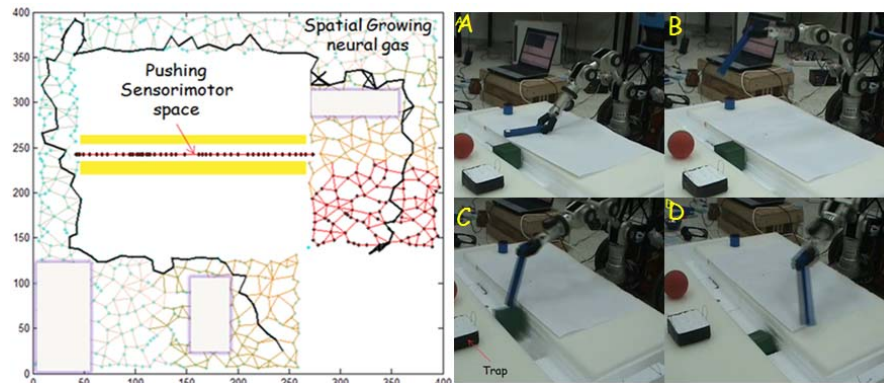


Figure 8. Lateral topology of the spatial map and the pushing SMS in the trapping groove. Panels A-D: A typical push sequence.

#### 4.2 Dynamics of the sensorimotor space

After learning the SMS through self organization of sequences of sensory and motor data generated by the robot, we now focus on the dynamics of SMS that determines how activations move back and forth between the sensorimotor-action spaces, and realize goal directed behavior. A zoomed view of the interactions between two neurons in the scheme of figure 6 is shown in figure 9. The dynamical behavior of each neuron in the sensorimotor space is as follows: To every neuron 'i' in the SMS we associate an activation  $x_i$  governed by the following dynamics:

$$\tau_x \dot{x}_i = -x_i + S_i + \beta_{ij} \sum_{i,j} (M_{ij} W_{ij}) x_j \quad (9)$$

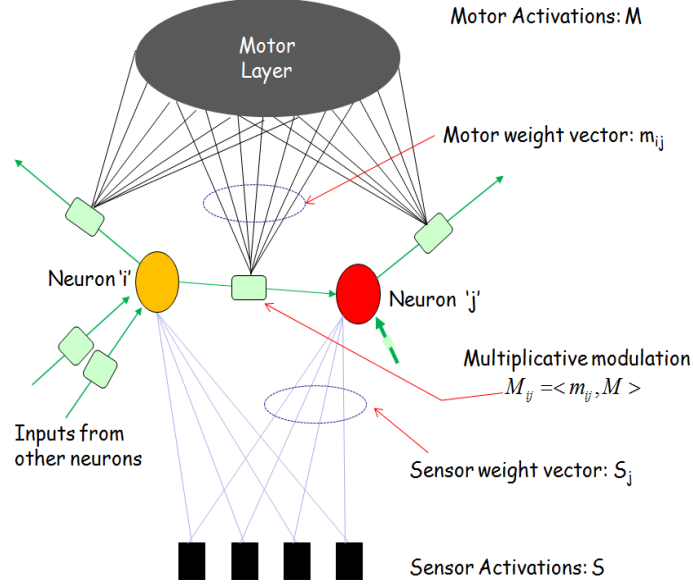


Figure 9. Zoomed view of interactions between two neurons in the SMS, interactions between perceptive layer, motor layer and the SMS.

We observe that the instantaneous activation of a neuron in SMS is a function of three different components. The first term induces an exponential relaxation to the dynamics (and is analogous to spatially homogenous neural fields of Amari et al, 1977). The second term is the net feed forward (or alternatively bottom up) input coming from the sensors at any time instant. The Gaussian kernel compares the sensory weight  $s_i$  of neuron  $i$  with current sensor activations  $S^t$ .

$$S_i = \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(s_i - S)^2}{2\sigma_s^2}} \quad (10)$$

Finally, the third term represents the lateral interactions between different neurons in the SMS, selectively modulated by the ongoing activations in the motor space. Hence, through this input the motor signals can couple with the dynamics of the SMS. If  $M$  is the current motor activity, and  $m_{ij}$  the motor weight encoded in the lateral connection between neuron  $i$  and  $j$ , the instantaneous motor modulated lateral connection  $M_{ij}$  between neurons  $i$  and  $j$  is defined as (and shown in fig. 8)

$$M_{ij} = \langle m_{ij}, M \rangle \quad (11)$$

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 and hence influences the dynamics of SMS. 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. Inversely, by multiplicatively modulating lateral

interactions between neurons in the SMS as a function of the motor activity in the action space, it is possible to predict the sensorial consequences of executing a motor action. Interaction between action space and sensorimotor space by virtue of motor modulated lateral connectivity thus embeds ‘Situation-Action-Consequence’ loops or Forward Models into the architecture and offers a way of eliciting perceptual activity in the sensorimotor space, similar to that which would have occurred if the action was performed in reality.

The element  $\beta_{if}$  in equation 9 is called the bifurcation parameter and is defined as follows

$$\beta_{if} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(S_{Anticip}-S)^2}{2\sigma^2}} \quad (12)$$

This parameter basically estimates how closely the top down (predicted) sensory consequence  $S_{Anticip}$  of the virtual execution of any incremental motor action  $M$  correlates with the bottom up (real sensory information)  $S$ .  $S_{Anticip}$  can be easily computed by only considering the effect of top down modulation in equation 4 and finding the neuron ‘ $k$ ’ in the SMS that shows maximum activation  $x_k$  among all neurons.

$$x_k = \sum_{k,j} (M_{kj} W_{kj}) x_j, \text{ for all } k, j \in (1, N)$$

Since sensory weights of every neuron is approximately tuned to the average sensory stimulus for which it was the best match, the anticipated sensory consequence  $S_{Anticip}$  is nothing but the sensory weights of the neuron  $k$  that shows maximum activation the under effect of top down modulation. The bifurcation parameter hence is a measure of the accuracy of the internal model at that point of time.  $\beta_{if} \rightarrow 0$  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’.

What should the robot do when it detects the fact that the world is functioning in ways that are contrary to its anticipations? The best possible solution is to work on real sensory information and engage in a incremental cycle of exploration to adapt the sensorimotor space, learn some new lateral connections, grow new neurons and eliminate few neurons (like the initial phase of acquiring the SMS ). This flexibility is incorporated in the dynamics in the following fashion: As we can observe from equation 9 that as  $\beta_{if} \rightarrow 0$ , the top down contribution to the dynamics also gradually decreases, in other words the system responds real sensory information only. Hence in this case only the real experience loop (of figure 5) is functional in the system. Now comes the next problem of how to trigger motor exploration dynamically, and this is the third important function of the bifurcation parameter. The bifurcation parameter controls the gradual switch between random exploration and planned behavior, by controlling the amount of randomness ( $r$ ) in motor signals in the dynamics of the action space as evident in equation 14.

$$\bar{a} = \beta_{if} \left( \sum_{i=1}^N x_i \overline{m_{k,i}} \right) + \zeta(\bar{r}) \quad (14)$$

The second term in equation 14 triggers random explorative motor actions, where  $r$  is a vector of small random motor commands (in the respective motor DoFs) and  $\zeta = 1 - \beta_{if}$ . So under normal operations (when  $\beta_{if}$  is close to 1) the amount of randomness is very less and the motor signal are incrementally planned to achieve the goal at hand using the first term of equation 14. We will enter into details of this component after formulating the value field dynamics in the next subsection. We also note that  $x_i$  in equation 9 are time dependent activations and the dot notation  $\tau_x \dot{x}_i = F(x)$  is algorithmically implemented using an Euler integration step:

$$x(t) = x(t-1) + \frac{1}{\tau_x} (F(x(t-1))) \quad (15)$$

In sum, a consequence of the dynamics presented in this section is that at all times, information flows circularly between the sensorimotor space and the action space. While the current goal, connectivity structure and the activity in the sensorimotor space project upwards to the action space and determine incremental motor excitations that are needed to realize the goal, motor signals from the action space influence top down multiplicative modulations in the lateral connections of the sensorimotor space hence causing incremental shifts in the perceptual activity. In the next subsection, we will describe the how representational scheme described in the previous section and the dynamics described in this section serve as a general substrate to realize goal directed planning (in simple terms, the problem of how goal couples with the internal model and influences the dynamics of the SMS).

### 4.3 Value Field Dynamics: How goal influences activity in SMS

In addition to the activation dynamics presented in the previous section, there exists a second dynamic process that can be thought as an attractor in the SMS that performs the function of organizing goal oriented behavior. The quasi-stationary value field  $V$  generated by the active goal together with the current (nonstationary) activations  $x_i$  (equation 9) allow the system to incrementally generate motor excitations that lead toward the goal.

Value field dynamics acting on the sensorimotor space is defined as follows:

$$\tau_v \dot{v}_i = -v_i + R_i + \gamma (W_{ij} v_j)_{\max} \quad (16)$$

$$R_i = DP + Q \quad (17)$$

Let us assume that the dynamical system is given a goal  $G$  that corresponds to reaching a state  $s_G$  in the sensorimotor space. *Just like the sensory signals couple with the neurons in the SMS through feed forward connections, the motor signals couple with the neurons in the SMS through motor modulated lateral connections, the goal  $G$  couples with the SMS by inducing reward/value excitations in all the neurons in the SMS.* As seen in equation 16, the instantaneous value  $v_i$  of the  $i^{\text{th}}$  neuron in the SMS at any time instance, is a function of three factors 1) the instantaneous reward  $R_i$  2) the contribution of the expected future reward, where

$\gamma$  (approx 0.9 ) is the discount factor and 3) the lateral connectivity structure of the SMS. Equation 17 shows the general structure of the instantaneous reward function we used in our computational model. The first term in the reward equation DP expresses the default plan if available ( for example, take the shortest or least energy path in the case of the spatial map). We will see in the later sections that it is in fact not really necessary to have a default plan in the reward structure and further there can be situations where new reward functions must be learnt by the system in order to initiate flexible behavior in the world. The second element in the reward function models these additional Goal dependent qualitative measures in the reward structure that are learnt through user/self penalization/rewards.

$$Q=Q_1 + Q_1 + \dots + Q_n \quad (18)$$

Every component Q can be thought as a learnt additional value field (having a scalar value at each neuron of the SMS) and the net value field is a superposition of the Q component and the DP component. In this sense the net attractor landscape is shaped by a task specific superposition of value fields (similar to combinations of different force fields I the reaching F/I model), and behavior is nothing but an evolution of the system in this dynamically composed attractor landscapes. The Q components of the reward structure further play an important role in dealing with heterogeneous optimality, dealing with dynamic changes in the world, taking account of traps during pushing etc. We will now present two examples to explain how different components in the model described by equations 9-18 interact under the presence of a goal.

#### 4.4 Reaching spatial goals using the spatial sensorimotor space

Coming to the problem of reaching spatial goals using the spatial SMS, let us consider that the spatial goal induces a reward excitation to every neuron in the SMS (similar to Toussaint et al 2006) as given by equation 19, where  $s_i$  is the sensory codebook weight of the  $i^{\text{th}}$  neuron, and G is the spatial goal in the playground that has to be reached by the robot and Z is chosen such that  $\sum_i R_i = 1$

$$R_i = \frac{1}{Z} e^{\frac{-(s_i - G)^2}{2\sigma_R^2}} \quad (19)$$

Under the influence of this reward excitation, the value field on the spatial SMS with relax quickly to its fixed point

$$v_i^* = R_i + \gamma (W_{ij} v_j)_{\max} \quad (20)$$

The coupling between the value field and the dynamics of the SMS can now be understood by revisiting the expression for action selection (equation 14). The element  $m_{k,i}$  represents the motor weights of a lateral connection between neuron  $i$  and its immediate neighbor  $k_i$  such that  $k_i = \text{argmax}_j (w_{ij} V_j)$ . In simple terms, the value field influences the motor activity by determining the neighboring neuron (to the currently active neuron) that holds maximum value in the context of the

currently active goal. In other words, it determines how valuable any motor excitation  $m_{hi}$  is with respect to the goal currently being realized. The motor action that is generated is hence the activation average of all the motor reference vectors  $m_{k,i}$  coded in the motor weights, for all  $N$  neurons and at that time instance. In sum, the goal induces a value field, that influences the computation of the incremental motor action to move towards the goal for the next time step, this motor activation in turn influences the dynamics of the SMS and causes a shift in activity, now the next step of valuable motor activation is computed and this process progresses till the time the system achieves equilibrium. Hence, the information flows between the SMS and the motor system is in both ways: In the “tracking” process as given by equation 9 the information flows from the motor layer to the SMS: Motor signals activate the corresponding connections and cause lateral, predictive excitations. In the action selection process as given by equation 14 information moves from the SMS back to the motor layer to induce the motor activations that will enable the system to move closer to the goal. In sum, the output of this circular dynamics involving SMS, action space and the goal induced value field is a trajectory: a trajectory of perceptions in the sensorimotor space and a trajectory of motor activations in the action space. Figure 10 shows the trajectories generated by the robot while moving to different spatial goals in the playground.

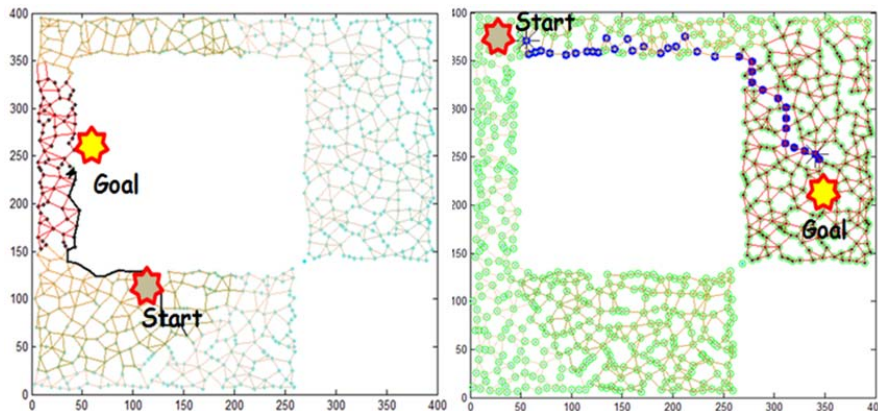


Figure 10. Movements to different spatial goals in the GNOSYS play ground. The goal dependent value field (quasistationary) is shown superimposed on the spatial map. As seen in the figure, using the simple reward structure of equation 19 (i.e only the DP component and no learnt value fields), neurons closer to the goal induce greater rewards.

#### 4.5 Learning the reward structure in ‘Pushing’ sensorimotor space

In order to realize any high level goal that requires a pushing action to be initiated, it is not just important to be able to simulate the consequences of pushing, but also to be able to ‘push in ways that are rewarding’. In other words, after learning the pushing SMS as described in 4.1, we now have the task of

making the robot learn the reward structure involved in a pushing action, so that it can coordinate the pushing in a goal directed fashion. In the set up of pushing in the trapping groove, we can estimate that pushing the goal to the either edges of the table should be maximally rewarding, since it ensures that the robot can move around the table and grasp the goal. We note here that no default plan (DP component) needs to be defined. Rather, the reward structure can be learnt directly 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) after which the robot is presented by a reward by the user. These trials can also be done in the mental space by initiating virtual pushing commands, simulating the consequence, virtually evaluating the possibility of reaching the now displaced goal (using the GNG for spatial navigation, and the forward/inverse model for reach/grasp) and finally self evaluating its success. Full reward is given to the neuron that fired last (that represents the location from where chances of reaching the goal is maximum) and gradually scaled versions the total reward is distributed to all the other neurons in the pushing SMS that were sequentially active during the trial. Energetic issues can also have their effects in the learnt reward structure, since there are multiple solutions to get the reward successfully by pushing in different directions. Influence of energetic issues in the reward field can be introduced by adding a decaying element in the promised net reward for achieving a goal successfully (equation 21), which is a function of the amount of energy spent in the process of getting the goal (For example, if the ball is pushed towards the right, more energy will be spent in navigation to achieve the goal of grasping the ball).

$$\begin{aligned}
 R_T &= R_{net} \text{ if } Dist_{iter} < \delta \\
 R_T &= R_{nst} e^{-(Dist_{iter}/125)} \text{ if } Dist_{iter} < \delta \\
 \delta &= \frac{Goal - Initpos}{1.5}.
 \end{aligned} \tag{21}$$

Where  $R_T$  is the actual reward received in the end of the  $T^{\text{th}}$  trial in case of success,  $R_{net}$  is the net reward promised in each trial (we kept all promised rewards for success as 50),  $Dist_{iter}$  is an approximate calculation of the distance navigated by the robot to get the goal, that is estimated based on the number of neurons in the spatial SMS that were active in the trajectory from initial position of the robot to the goal. We must note that this distance travelled  $Dist_{iter}$  is a consequence of the pushing action that preceded navigation and not a result of the constraints in spatial navigation in the playground). In other words, if the robot pushed the goal to the right, it needs to navigate a much greater distance that it would have had to in case it had pushed the goal to the left. This is reflected in the number of neurons that are sequentially activated during the path from source to goal i.e.  $Dist_{iter}$ . Since navigation has a high cost in terms of battery power consumed and since navigating greater distances than that was necessary directly implies spending more energy than that was necessary, the term  $Dist_{iter}$  is one of the parameter that helps in distributing rewards based on the energetic efficiency of the solution. The other term  $\delta$  is the ratio of the shortest distance between ‘the initial position

(*Initpos*) of the robot from the final location of goal after pushing (Goal)' and 'representational density of neurons covering the spatial SMS ' that we conservatively approximated as 1.5. After every trial of pushing, the reward received by each neuron in the pushing SMS is added to its previous accumulated reward value. After about 50 trials, we averaged the rewards received by each neuron in each trial, in order to generate the final reward structure for pushing. This reward structure can now be used to compute the value field which then drives the pushing SMS dynamics. This works exactly the same way as the spatial map dynamics i.e based on the value field, the next incremental motor action for pushing the goal object (a ball) is computed. This then modulates the lateral connections to cause a shift in activity that corresponds to the anticipated movement of the ball in the trapping groove. Now based on this new predicted location of the ball in the pushing SMS, and the value field, the next incremental pushing action is computed and so on till the time the system attains equilibrium. The final anticipated spatial position of the ball after the pushing SMS dynamics is complete, in turn induces a quasistationary value field in the spatial map that triggers the spatial SMS dynamics so as to eventually pull the body towards it. Figure 11 shows a combined sequence of pushing and moving in the respective sensorimotor spaces.

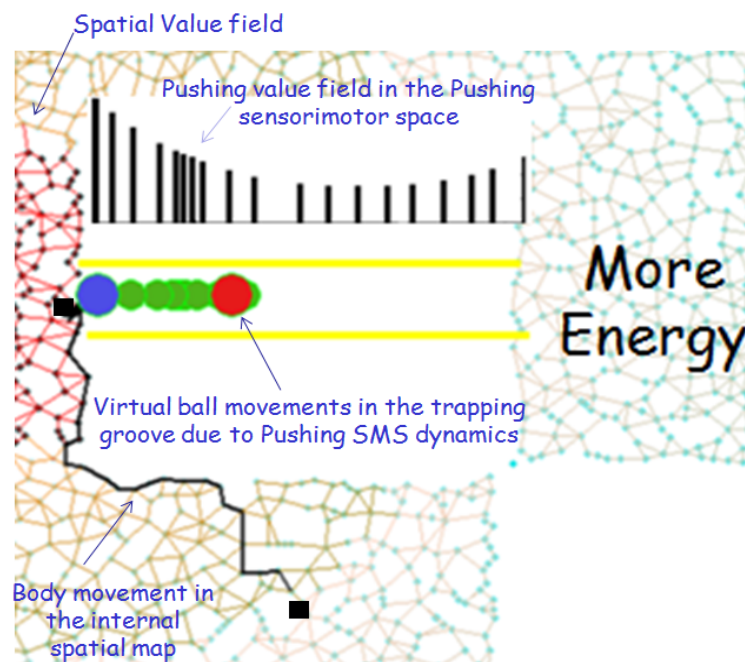


Figure 11. Combined sequence of pushing and moving in the mental space. Note that the pushing reward structure encourages pushing to the left (that is more energy efficient). The final anticipated position of the ball once pushing SMS reaches its equilibrium is a spatial goal for the spatial SMS. This spatial goal induces a quasistationary value field in the spatial SMS there by triggering the dynamics in the spatial map, hence pulling the body closer to the goal.

We can observe from figure 11 that the pushing value field encourages the robot to push towards the left since it is an energy efficient strategy and hence more rewarding. However, this may not always be true if there are dynamic changes in the world (like introduction of traps) during which always pushing the goal to the left may result in a failure to get the reward. Under such cases new ‘experiences based value fields need to be learnt (Q components in eqn. 17) that dynamically shape the field structure appropriately taking into account these issues.

We now introduce these additional constraints on the pushing scenario by placing traps randomly at different locations along the trapping groove. Traps were indicated to the robot through visual markers so that their location in the groove can be estimated by reconstructing the information coming from the visual recognition system. When traps are introduced initially in the trapping groove, the behavior of the system is only governed by the previously learnt reward structure. Hence the robot follows the normal strategy as in the previous section. As seen in the three trials after introduction of traps shown in figure 19, the ball is pushed as a function of the value field learnt in the previous section ( shown in pink on top of the trapping groove ) and is always constant. This normal behavior continues till the time a contradiction is encountered between the anticipated position of the ball as a result of an incremental pushing action and the real location of the ball coming from the 3d reconstruction system. In other words, the ball is not really in the place where the robot thinks it should be as a result of the pushing action initiated by it. A contradiction automatically implies that there are new changes taking place in the world whose effects are not represented internally by the system. Such contradictions result in a phase of active exploration (since  $\beta_{if} \rightarrow 0$ , eqn.9 and 14) at least till the time the system is pulled back to the normal behavior by the already existing value field. Now the robot initiates incremental random pushing in different directions till the time the ball begins to move as anticipated, in which case pushing is once again governed by the preexisting plan. The path of the ball during random pushing and normal behavior is shown in figure 19 for four different cases . In the first case, since the initial location of the ball is close to the right end, following the normal behavior, the ball was pushed rightwards where it collides with the trap placed at around 220, this motion of the ball is shown in green with the white arrow. Now there is an active phase of random pushing for a while with the ball moving forwards and backwards, till the time it reaches a position from where the preexisting value field takes over. The motion of the ball due to explorative pushing is shown in blue with the direction indicated by the yellow arrow. Once it is on the other end of the table, it can be easily reached. Case 3 and case 4 are also similar to the first case, however with a different environmental configurations. In the case 2 the trap was placed around 150 and the initial location of the ball shown is approximately 135. In this case there was no exploration at all because the previously existing value field automatically causes the ball to be pushed to the left and the goal was achieved. In fact, the robot was blind about the existence of the trap in the sense that it was not the trap that caused the direction of pushing but the preexisting reward field it had developed earlier. This may be also be a limitation of the approach because the knowledge is

represented more in the form of associations of experiences (like the capuchins) rather than a still higher level of understanding of the real physical causality. Is there such a still higher level of understanding or is it just associative rules learnt by experience that are exploited intelligently is still an issue of debate, which we will not enter in this section.

What should the robot do with these sequences of new experiences, the experience of a new environment, a contradiction which it did not encounter before while solving similar goal, a phase of exploration to try to find an alternative solution that eventually results in success and rewards? We suggest that it should represent them as a memory and in the form of the  $q_i$  components in the reward structure given by equation 19. Further, the reward that was received on success needs to be distributed to the contributing neurons in the pushing SMS. This distribution is done as follows: in case of rewards, the most distal element receives the maximum reward and all contributing elements receive gradually scaled versions, circular solutions being actively penalized. The panels on the right show the new reward fields ( $q_i$ ) learnt after each trial. In case1 for example,

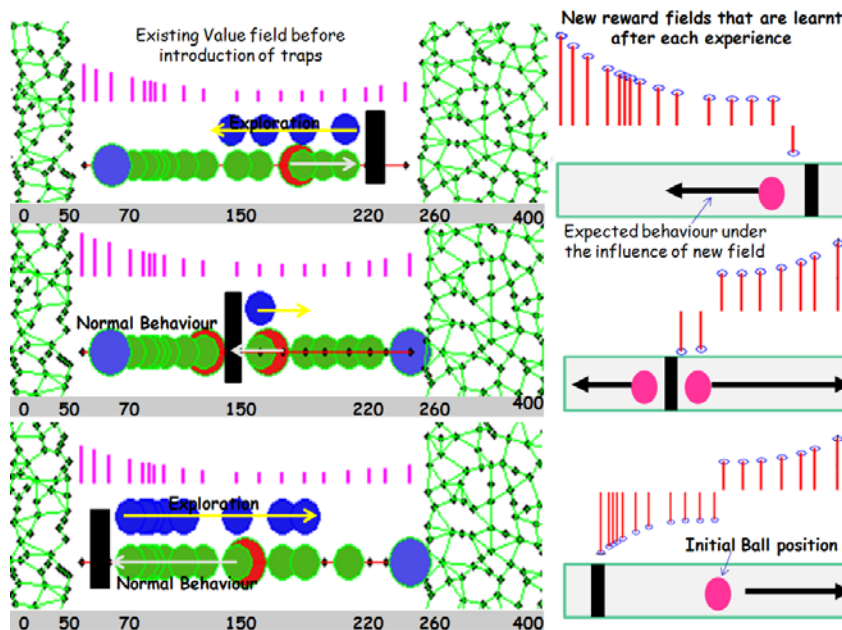


Figure 12. Three trials of pushing under the influence of traps placed at different locations along the groove are shown in the figure. The panels on the right show the new reward components  $q_i$  learnt after being rewarded due to successful realization of the goal partly because of random explorative pushing. In every trial the robot has an experience, 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 and an experience of being rewarded by the user/self in case of success. This experience is represented in the form of a reward field in the pushing sensorimotor space. For example, in trial 3, what is represented is the simple fact that if the initial position of the ball is around 150, and the position of the trap is around 65, it is more rewarding to push towards the right and navigate all around the table to reach closer to the ball. These experiences, based on their relevance to the goal being attempted will influence the behavior of the robot in the future.

the reward structure representing this experience reflects the fact that if the initial position of the ball is around 180 and the location of the trap is somewhere around 220, it is rewarding to push leftwards. For case 4, it reflects the fact that if the trap is somewhere around 60, and the initial position of the ball is around 150, it is more rewarding to push to the right. We also note that there is no need to predecide how many trials of such learning has to take place. Learning in the system takes place when it is needed, i.e. when there is a contradiction and things are not working as expected. After eight different single trap configurations the behavior produced was intelligent enough that no further training was required. The additionally learnt  $q_i$  components of reward field also begin to influence the value field dynamics now and hence the value field structure is no longer constant like it was in figure 19. It changes based on the configuration of the problem. The net reward structure is a superposition of the default plan which was learnt previously in the absence of traps and the new experience related fields that were learnt after introduction of traps, scaled appropriately based on their relevance to the currently active goal.

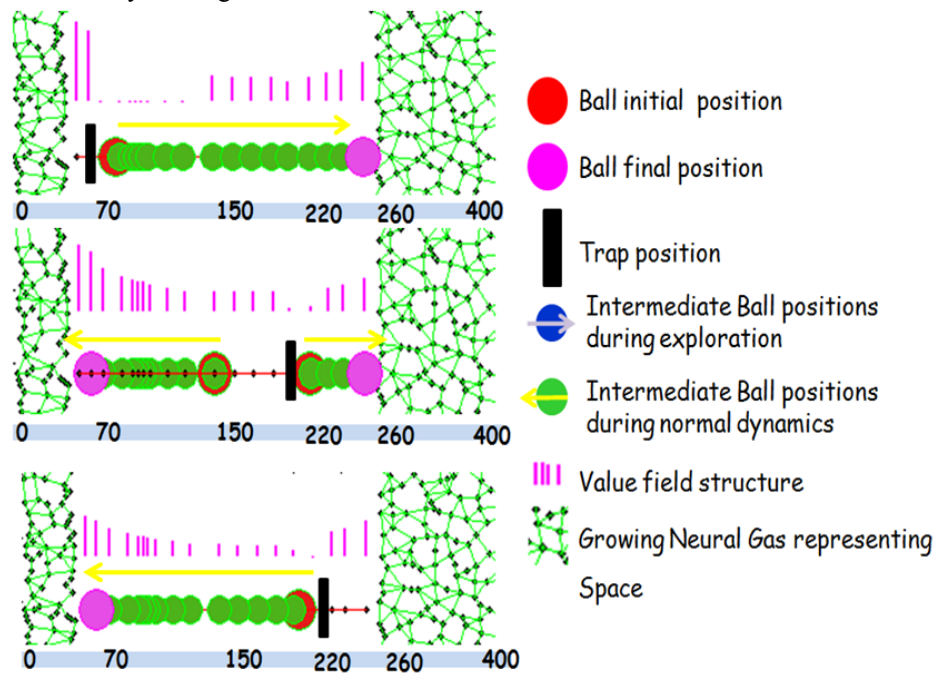


Figure 13. Pushing in the presence of traps in the trapping groove. In the previous cases of pushing shown in figure 12, the value field superimposed on the pushing sensorimotor space was constant. In this figure we can observe goal/trap specific changes in the value field. Experiences encountered in the past and represented in terms of fields are superimposed in a task relevant fashion, to give rise to a net resultant field that drives the dynamics of the system. Also we see that in this case pushing direction is a function of both the relative position of the hole, and the starting position of the reward/ball.

$$R = R_{default} + \sum_{T=1}^N \sum_{E=1}^m R_E \cdot \frac{1}{\sqrt{2\pi}\sigma_T} e^{-\frac{(Trap_T - Trap_E)^2}{2\sigma_T^2}} \quad (22)$$

Here  $R_{default}$  is the pushing reward structure learnt in the previous section. T stands for number of traps. E stands for the number of experiences during which new reward field were learnt (eight in our case).  $R_E$  is the  $E^{th}$  reward field. And the final term computes how relevant an experience E is with respect to the situation considering trap T present alone in the environment. Figure 20 shows examples of the pushing in the trapping groove for single trap configurations, after the learnt reward fields began contributing to the value field structure and hence actively influencing the behavior.

## 5 A Goal directed, mental sequence of ‘Push-Move-Reach’

How can the internal models for reaching, spatial navigation and pushing cooperate in simulating a sequence of actions leading towards the solution of a high level goal? Let us consider a scenario where the robot is issued a user goal to grasp a Green ball as shown in panel 1 of figure 14. In the initial environment the ball is placed in the center of the trapping groove, unreachable from any direction. In addition, one trap is placed in the trapping groove as an additional constraint. It is quite a trivial task for even children to mentally figure out how to grasp the ball through a sequence of ‘push-move reach’, using the available blue stick as a tool and avoiding the trap. However the amazing complexity of such seemingly easy tasks is only realized when we question the computational basis of these acts or make robots act in similar environmental scenarios. How can the robot use the internal action models presented in sections 3 and 4 to mentally figure out a plan to achieve its goal? Of course it can employ the F/I model for reaching to virtually evaluate the fact that the ball is not directly reachable with the end effector, but reachable using the long blue stick (which is directly reachable to its end effector). Using the pushing internal model, the robot can now perform a virtual experiment to evaluate the consequence of pushing the ball using the stick. The value field in the Pushing SMS (Panel B) incrementally generates actions that are needed to push the ball in the most rewarding way. We note that the pushing value field shown in panel B also includes trap specific adaptations, though a simple learnt pushing value field like the one shown in figure 8 is equally applicable when traps are not present. On the other hand these motor activations modulate the lateral connectivity in the pushing SMS and anticipate the position of the ball as the result of the virtual pushing. On reaching equilibrium, the output of the pushing internal model is a set of trajectories: the trajectory of the ball in the SMS and the trajectory of motor actions that is needed to push the ball, in the action space. The anticipated final position of the ball in the trapping groove induces reward excitations on the neurons in the spatial sensorimotor space and triggers the spatial

dynamics. The spatial dynamics functions exactly the same way moving in a dynamically generated value field in the internal spatial map, taking into account the set of constraints that are relevant to the task. The output of the spatial dynamics is once again a set of trajectories: the trajectory of the body in the spatial sensorimotor space and the trajectory of motor commands that needs to be executed in order to move the body closer to the spatial goal (i.e. the anticipated

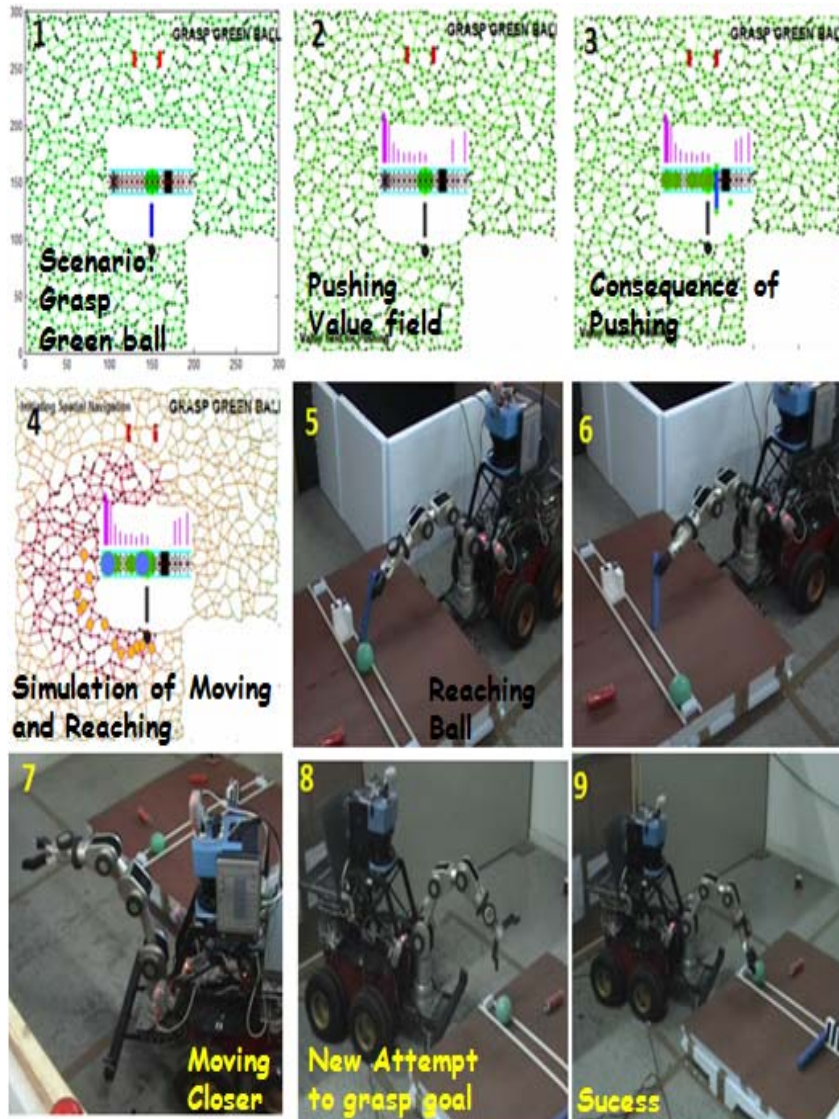


Figure 14. Panel 1-4: Simulation of Virtual Push-Move-Reach actions to realize an otherwise impossible goal; Panel 5-9 initiation of real motor actions

final position of the ball which was the output of the pushing internal model). Once the dynamics of spatial growing neural gas becomes stationary, Gnosys has the two crucial pieces of information needed to trigger passive motion paradigm (forward/inverse model for the arm): the location of the target (predicted by Pushing model), and the initial conditions (location of the body/end-effector predicted by the equilibrium configuration of the dynamics in the internal spatial map). As we saw in section 2, the output of the forward inverse model is also a set of trajectories: the trajectory of the end-effector in the distal space and the trajectory of the joint angles in the proximal space. Starting from a mentally simulated initial body/end-effector position (coming from spatial sensorimotor map), the robot can now mentally simulate a reaching action directed towards a mentally simulated position of the goal target (coming from the pushing sensorimotor space), using the forward inverse model for reaching (passive motion paradigm). In sum, using the three internal models presented in this article, GNOSYS now has the seamless capability to mentally simulate sequences of actions (in different sensorimotor spaces) and evaluate their resulting perceptual consequences: *“... since there is a trap there, it is advantageous to push in this direction; if I push in this 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 ...”*.

## 6 Discussion

The functional role played by explorative sensorimotor experience acquired during play towards the overall cognitive development of an agent (natural/artificial) is now well appreciated by experts from diverse disciplines like child psychology, neuroscience, motor control, machine learning, linguistics, cognitive robotics among others. No wonder, playing is the most natural thing we do and there is much more to it than just having fun. In this article, we initially introduced the playground we designed for the GNOSYS robot and described the scenarios from animal reasoning that inspired its creation. Three internal models for action generation (reaching, spatial Map and pushing), all critical for initiating intelligent motor behavior in the playground were presented. We further showed how using the acquired internal models, Gnosys can virtually simulate sequences of ‘actions and perceptions’ in multiple sensory motor state spaces in order to realize a high level goal in a complicated environmental set up. The core action models like Pushing, Moving, Reaching, Grasping form a closely connected network, predictions of one slowly driving the other (or providing enough information to make the other mental simulation possible). One key feature regarding various internal models (arm, spatial map, pushing, abstract reasoning system) created in the GNOSYS architecture is the fact that all of them are structurally and functionally identical, use the same protocols for acquisition of

information, same computational mechanisms for planning, adaptation and prediction. The only difference is that they operate on different sensorimotor variables, move in the presence of different value fields, towards different goals (local to their computational scope), using different resources of the body/environment. The output of the system ultimately is a set of temporally chunked trajectories (of end effector, body, external object etc) all shaped due to combinations of superimposed fields applied to respective sensorimotor spaces.

While extending the architecture beyond the internal action models presented in this paper, we note that the computational complexity in the problem of realizing an user goal like ‘Reaching a Red Ball’ in a complex environment results from the fact that before reaching the red ball itself with end effector, there may be several intermediate sequences of real/virtual ‘Reaching’, ‘Grasping’, ‘Pushing’, and ‘Moving’ etc directed at ‘potentially useful’ environmental objects, information regarding which is not specified by the root goal itself (which was just ‘reach the red ball’). So before realizing the root goal, the robot has to ‘track down’ and ‘realize’ a set of useful subgoals that ‘transform’ the world in ways that would then make the successful execution of the root goal possible. Hence, even though the high level goals are simple, the complexity of the reasoning process and actions needed to achieve them can increase more than proportionately with the complexity of the environment in which they need to be accomplished. So how can the robot reduce/distribute a high level goal into temporally chunked atomic goals for the different internal models? How can the robot do this flexibly for a large set of environmental configurations each having its own affordances and constraints? What happens if the constraints in some environments do not allow the goal to be realised (For example there are two traps in the trapping groove and the goal is placed in between them) ? Can the robot mentally evaluate the fact that it is in fact impossible to realize the goal in that scenario ? Will it Quit without executing any physical action at all? If yes, does it have a reason to Quit ? and Can we see the reasons that caused the Quitting by analysing the field structure? We are currently developing and evaluating the extended GNOSYS reasoning-action generation architecture to possibly attack some of these questions.

#### ACKNOWLEDGMENT

This research was partly supported by the EU FP6 project GNOSYS and EU FP7 projects iTalk (Grant No:214668) and HUMOR (Grant No: 231724).

#### REFERENCES

- [1] Emery, N.J. & Clayton, N.S. (2004). The Mentality of Crows: Convergent Evolution of Intelligence in Corvids and Apes. *Science* 306: 1903-1907.
- [2] Weir, A.A.S., Chappell, J., Kacelnik, A. (2002). Shaping of Hooks in New Caledonian Crows. *Science* 297:981-3.
- [3] Visalberghi, E., Limongelli, L. (1996). Action and understanding: tool use revisited through the mind of capuchin monkeys. In A. Russon, K. Bard, & Parker, S. (Eds.), *Reaching into thought. The minds of the great apes* (pp. 57-79). Cambridge: Cambridge University Press.

- [4] Visalberghi, E. (2000). Tool use behaviour and the understanding of causality in primates. In E. Thommen & H. Kilcher (Eds.), *Comparer ou prédire: exemples de recherches en psychologie comparative aujourd'hui* (pp. 17-35). Fribourg (Switzerland): Les Editions Universitaires.
- [5] Visalberghi, E., Tomasello, M. (1997). Primate causal understanding in the physical and in the social domains. *Behavioral Processes*, 42, 189-203.
- [6] Mohan, V., Morasso, P. (2007b). Towards reasoning and coordinating action in the mental space. *International Journal of Neural Systems*, 17, 4, 1-13.
- [7] Mussa Ivaldi, F.A, Morasso, P., Zaccaria, R. (1988). Kinematic Networks. A Distributed Model for Representing and Regularizing Motor Redundancy. *Biological Cybernetics*, 60, 1-16.
- [8] Tsuji, T., Morasso, P., Shigehashi, K., Kaneko, M. (1995). Motion Planning for Manipulators using Artificial Potential Field Approach that can Adjust Convergence Time of Generated Arm Trajectory. *Journal Robotics Society of Japan*, 13, 285-290.
- [9] Zak, M. (1988). Terminal attractors for addressable memory in neural networks. *Phys. Lett. A*, 133, 218-222.
- [10] Toussaint, M. (2004). Learning a world model and planning with a self-organizing dynamic neural system. *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, 929-936, MIT Press, Cambridge.
- [11] Toussaint, M. (2006). A sensorimotor map: Modulating Lateral connections for Anticipation and Planning. *Neural Computation* 18, 1132-1155.
- [12] Fritzke, B. (1995). A growing neural gas network learns topologies. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems*, 7(pp. 625-632). Cambridge, MA: MIT Press.
- [13] Amari, S. (1977). Dynamics of patterns formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27, 77-87.
- [14] Abbot, L., Sejnowski, T.J. (1999). *Neural Codes and Distributed representations*. MIT Press.
- [15] Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer.
- [16] Sutton, R., Barto, A. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.
- [17] Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, 27, 377-396.
- [18] GNOSYS project documentation: [www.ics.forth.gr/gnosys](http://www.ics.forth.gr/gnosys)
- [19] Georgeff, M. P. (1999). The belief-desire-intention model of agency. In *Intelligent Agents*, V (eds J. P. Müller, M. P. Smith and A. S. Rao) LNAI Volume 1555, pp.1-10. Springer, Berlin.
- [20] Hesslow, G. (2002). Conscious thought as a simulation of Behavior and Perception, *Trends in Cognitive Sciences*, Vol 6.
- [21] Hesslow, G., Jirenhed, D.A. (2007). The Inner World of a Simple Robot. *Journal of Consciousness Studies*, 14:85-96
- [22] Holland, O., Goodman, R. (2003). Robots with internal models: a route to machine consciousness? *Journal of Consciousness Studies*, Special Issue on Machine Consciousness, vol 10, No 4.
- [23] Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current Opinion in Biology* 9:718-727.