Neural Dynamics of Hierarchically Organized Sequences: a Robotic Implementation

Boris Durán
Informatics Research Center
University of Skövde
Skövde, Sweden
Email: boris.duran@his.se

Yulia Sandamirskaya
Institut für Neuroinformatik,
Ruhr-Univesität Bochum,
Bochum, Germany.
Email: sandayci@rub.de

***Abstract*—Robotic researchers face fundamental challenges when designing autonomous humanoid robots, which are able to interact with real dynamic environments. In such unstructured environments, the robot has to autonomously segment objects, detect and categorize relevant situations, decide when to initiate and terminate actions. As humans are very good in these tasks, inspiration from models of human sensory-motor and cognitive processes may help design more flexible and autonomous robotic control architectures. Recently, we have extended a neurally-inspired model for sequential organization with a representation of hierarchies of behaviors. Here, we implement this model on a robotic platform and demonstrate its functionality under constraints of a real-world implementation. The architecture generates hierarchically organized behavioral sequences on the Aldebaran's humanoid robot NAO. The key dynamic components of serial organization – such as the intention, condition of satisfaction (CoS), and interactions within the hierarchy – are coupled to robotic sensors and motors and bring about flexible and autonomous behavior. We also demonstrate how continuous in time neural-dynamic parts of the controller may be seamlessly integrated with preprogramed algorithmic behaviors, introducing flexibility, autonomy, and ability to learn, while avoiding unnecessary complexity of the architecture.**

I. Introduction

Control of a humanoid robot with many degrees of freedom is a challenging engineering task, which has been advanced recently due to improvement in hardware and software (e.g, [1], [2]). However, robotic systems often fail in unconstrained, real-world environments shared with a naive human user. Such environments have rich perceptual structure and dynamics, which means that the robotic controller has to be flexible and adaptive. On the perceptual side, the robot must be able to autonomously detect relevant states of the environment and transitions between them. On the motor side, the system must decide autonomously when a particular action is appropriate and coordinate activation and deactivation of the respective internally represented behaviors. The engineering solutions to the robotic control today do not show the desired flexibility, which leads to a demand on new approaches to robot control [3].

As humans are very good in dealing with complex dynamic environments, one obvious strategy is to gain inspiration from how human central nervous system solves the control problems. While the artificial perceptual and motor system are fairly advanced nowadays and in some domains outperform humans, the immense flexibility, with which humans make use of their abilities is still not matched by robotic systems.

In this paper, taking inspiration from a neural-dynamic framework that is used to understand human cognitive functions in a close link to perception and action, we apply a neurally inspired model for generating serially ordered behaviors [4] to control the behavior of a robot. We increase the complexity of the tasks to be executed by introducing a hierarchical structure in the serial order architecture.

The architecture is based on dynamic neural fields (DNFs) – a mathematical framework that bridges sensory-motor processes, continuous in time and space, and discrete cognitive entities, such as representations of objects and representations of the states in motor systems [5]. This framework is a variant of the dynamical systems approach to cognition and robotics [6]. In this approach, attractors of a continuous time dynamical system correspond to behaviorally relevant states. The dynamic neural fields defined over continuous dimensions such as color, location, or motor command extend the traditional dynamical systems framework and enable link to unprocessed sensory information. Instabilities of the dynamics of neural fields account for segregation of visual features into objects and stabilizing spatial representations of objects and events [7].

Within the DNF framework, behavior of a human, or a robot, may be understood in terms of elementary behaviors (EBs), which may be activated and deactivated in a sequence. This principle is similar to the classical behaviorist modular architecture, proposed by Brooks and followed up in the field of behavioral robotics [8], [9], [10]. However, EBs in our framework are inspired from modelling human cognition and are featured by two elements – a representation of the intention of an action and the action's condition of satisfaction as stable states of a neural dynamics. Because of their attractor properties, both of these structures may be linked to the sensory-motor system of the robot and control its action even in noisy environments [11]. Each activated intention shapes attractors of sensory-motor dynamics and eventually results in generation of an overt action. When the behavioral goal of an action is completed, the condition of satisfaction system is activated and inhibits the active EB. Here, we demonstrate how this model of sequence generation in humans may be extended to sequences with hierarchical structure and implement it on the humanoid robot NAO. Hierarchies are fundamental mechanisms created to embrace the complexity of real-world tasks [12] and to generalize new task domains. The importance of hierarchical representations has also been emphasized in the robotic context [13], [14], [15].

In this paper, we demonstrate the functioning of a neural-dynamic architecture that implements hierarchically organized sequences. This is a first, proof-of-concept robotic implementation of our model and will be extended through learning in subsequent implementations. The paper is structured as follows. First, we present the robotic scenario and implementation details of our neurally-inspired architecture. Then we proceed with the description of the architecture, present the results of robotic experiments, and finish with a short conclusion.

II. Methods

A. The scenario and resources

Aldebaran's humanoid platform NAO was used for the implementation of hierarchical sequences. NAO may be controlled using Aldebaran's Naoqui framework, which enables access both to low-level sensors and motors of the robot as well as predefined behavioral modules. One of the goals we pursue in this implementation is to demonstrate that in the DNF framework, the neural-dynamic parts of the architecture may be seamlessly integrated with algorithmically implemented behaviors, which do not rely on the low-level sensory input.

In the presented architecture, we used the visual input from the top camera, the control of the yaw motion of the NAO's head, and several built-in behavioral modules, such as walk-to, stand-up, approach, and point. The complete set of elementary behaviors (EBs) used in the implementation for the assembly of higher order sequences are listed in Table I. The built-in behavioral modules, such as *Walk*, *Stand-up*, *Approach*, and *Point*, were activate and deactivate by the dynamical nodes of the architecture.

The higher-order task, which the robot had to accomplish, was to retrieve a sequence of colors, whereas each of these color-seek behaviors entailed a sequence of EBs from the lower level of the hierarchy, e.g. stand-up, approach, point, return. Dynamical nodes of the architecture activated and deactivated these behaviors in appropriate moments in time. We describe the architecture in detail next.

B. Sequence generation with neural dynamics

In our architecture, a sequence is comprised of elementary behaviors. Each elementary behavior features three main components – an intention node, a condition of satisfaction (CoS) node, and a memory node. Each node is characterized by its activation, which follows a continuous-time equation with bistable dynamics (with an "on" and an "off" state). If a node is active ("on" state), it sends activation through weighted connections to those nodes it is coupled to. Within an EB, the intention node provides enough input to activate the memory node, which in turn sends an input to the CoS node, but not enough as to activate it. The intention node also impacts on the intention field, described below and/or on the sensory-motor systems, shaping their dynamics to produce an action, which the EB encodes. When the action is completed, the CoS node detects the respective perceptual state and inhibits the intention of the EB. When the intention is deactivated, the memory node sustains its activation and provides input to the intention node of the next EB. A global inhibition between all intention nodes ensures that only one of them is active at a time. Please, refer to our previous publication [16] and [11] for mathematical details of the model.

TABLE I
ELEMENTARY BEHAVIORS USED TO CONSTRUCT HIERARCHICAL SEQUENCES.

Behaviors	Dynamics	Callback
Stand up	Discrete	Blocking
Search	Continuous	Non-blocking
Approach/walk toward	Discrete	Blocking
Point/reach	Discrete	Non-blocking
Return/walk backward	Discrete	Blocking
Sit down	Discrete	Blocking

Some EBs also have an intention dynamic neural field (DNF) and a CoS DNF. DFNs are variants of the dynamical systems that are defined over continuous dimensions, such as color, orientation, or location in space. The intention field encodes dynamically a perceptual or motor parameter of the action, which then may shape the sensor-motor dynamics in a graded fashion. The CoS field detects a match between the preactivating input from the intention field and the input from perception and thus signals the successful accomplishment of an action [4].

Next we describe how hierarchies of sequences of EBs are built in this framework and provide details of our robotic implementation.

C. The hierarchy

The hierarchy in our implementation is composed of two layers: the top layer consists of EBs representing a sequence of colors to search, and the bottom layer consists of a group of EBs representing motor actions. Each EB in the top layer activates a sequence in the bottom layer, Fig. 1.

The intention nodes of the EBs in the top layer activate a single intention field, defined over color dimension. The matching *CoS* field detects when the color is "found". This state corresponds to two conditions to be fulfilled: first, the sought color is centered in the camera image and, second, the lower-level sequence of EBs is completed. The second condition is detected by the CoS nodes of the EBs at the lower level, and summed-up by the dynamics of the CoS node in the upper level.

At the lower level, only the *Search* EB includes a dynamic field to represent the intended orientation of the head and the respective CoS dynamic field. Other EBs, e.g. *Approach*, *Walk*

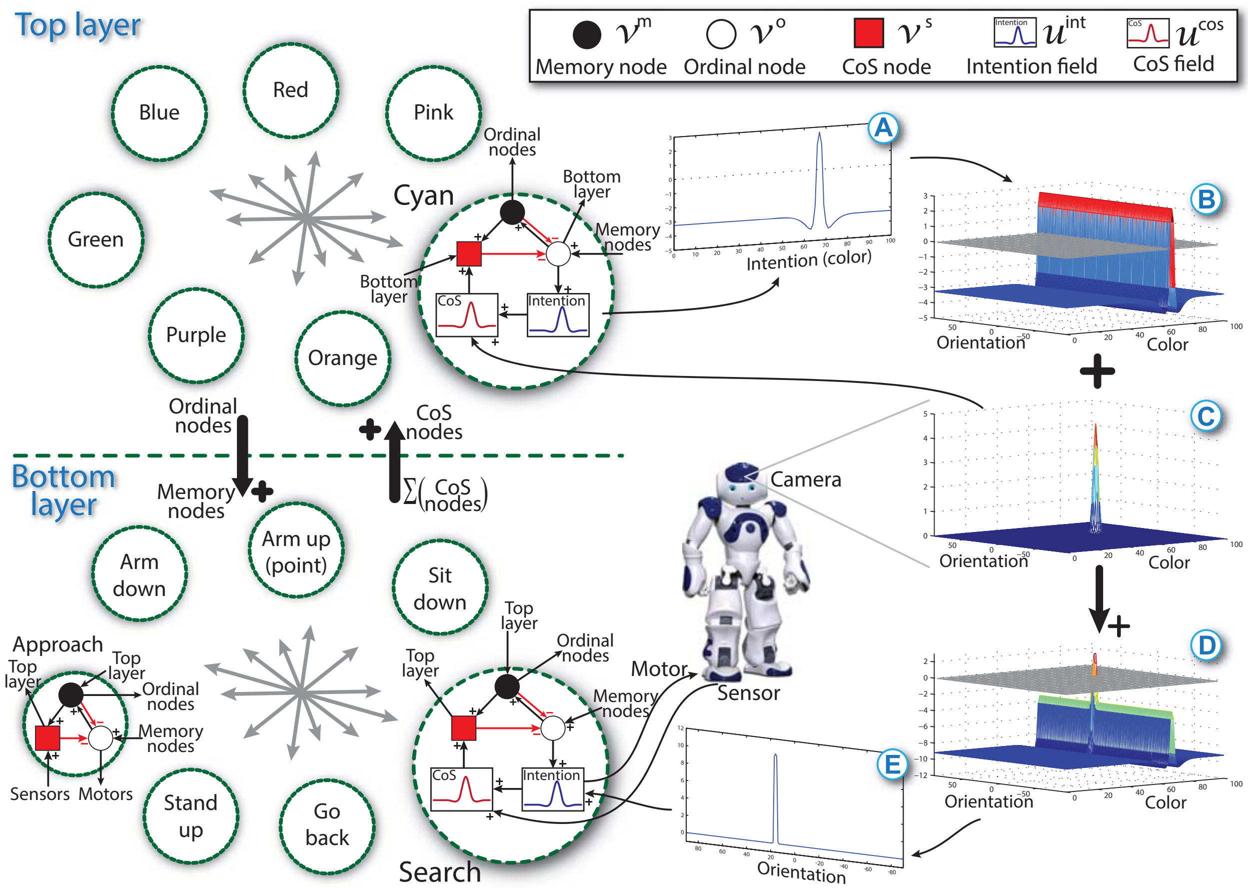

Fig. 1. Schematics for the implementation of the hierarchical serial order model.

backward, or *Point*, do not make use of the graded dynamic neural fields and are simply activated and deactivated by the intention nodes. In a more general architecture, where these EBs would have to be more flexible and open to learning, the implementation of motor parameters and perceptual pre-conditions could be done using DNFs, which are subject to neurally-inspired learning algorithms. This will require a more efficient implementation of the model in C/C++ which falls out of the scope of the present work, which is based on a Matlab prototype of the architecture.

D. The sensory-motor system

One of the main components of the sensory-motor dynamics of the architecture is a two-dimensional color-orientation field, plot D in Fig. 1. This field receives input from the intention of the higher level in the hierarchy and from the robot's camera. The active *intention field* of the higher level in the hierarchy, plot A in Fig. 1, provides the first stimulus as a sub-threshold ridge along the color dimension, plot B in Fig. 1. The information from NAO's top camera is used to create a sub-threshold 2D gaussian stimulus located at the hue site specified by the intention node of the current EB and the horizontal location of the detected object, plot C in Fig. 1. Only when those two sub-threshold inputs overlap, the 2D field becomes active creating a peak at the desired location.

This peak is then projected into the orientation dimension by integrating the 2D field along the color dimension, plot E in Fig. 1. The intention field of the *Search* EB in the lower level of the hierarchy receives input from the previous projection. When activated, this intention field induces a motor action in the pan/yaw motor of NAO's head by setting an attractor for the dynamics that controls pan/yaw of the robot. This dynamics makes the head to center the object in middle of NAO's field of view (FOV). The *CoS* field for the *Search* EB receives input from the current state of NAO's pan motor and is tuned to create a peak when the object is in the center of the FOV. At this point the *Search* behavior is completed and the system transitions to the next EB, as defined by the connection weights from its memory node. When the lower-level sequence is accomplished, the CoS node of the higher level is activated and the robot proceeds with the motor sequence associated with the next color.

III. Results

In the robotic experiments, NAO is positioned in the *Sit down* position in front of a wall with three colored pieces of paper sticked to the wall. NAO's task was to approach the wall and to point to the pieces of paper following the sequence encoded in the top layer of the hierarchy, and using

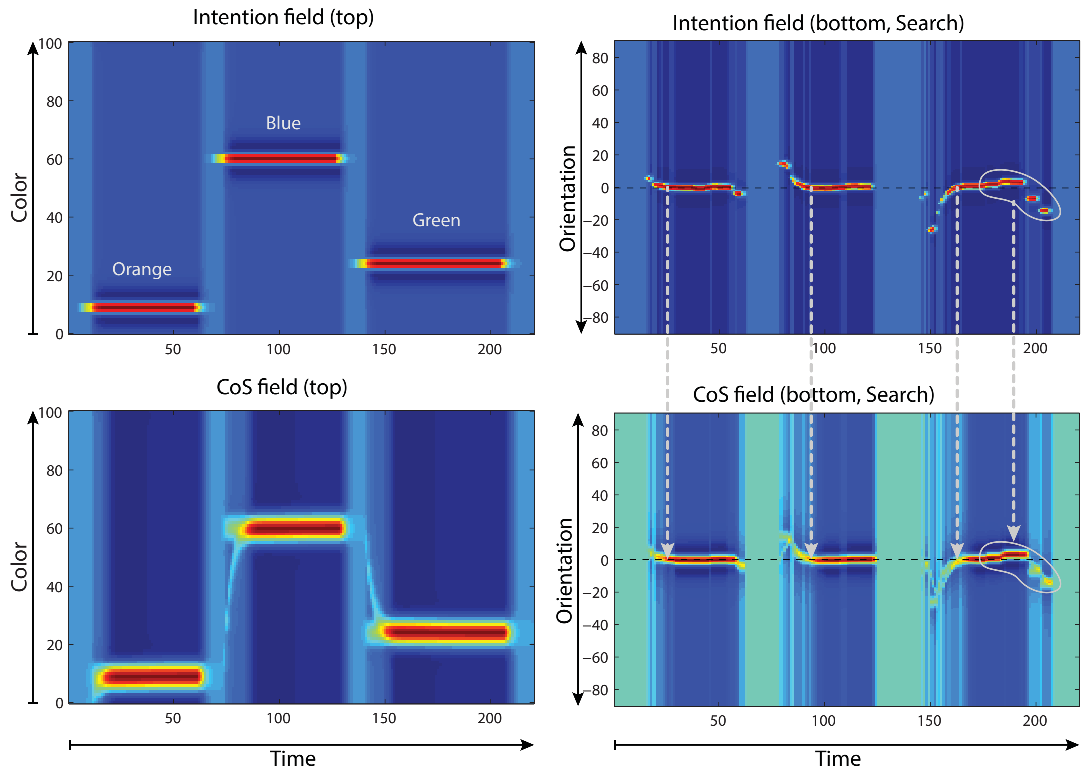

Fig. 2. Left, activity of a common Intention (top) and CoS (bottom) fields for all EBs in the top layer of the serial order hierarchy. Right, activity of the Intention (top) and CoS (bottom) fields for the *Search* EB in the bottom layer of the serial order hierarchy. See text for annotations.

the respective sequence of motor actions from the bottom layer of the hierarchy.

The order of the colors in the sequence can be observed in the traces of positive activation in the intention field of the higher-level of the hierarchy, Fig. 2 top-left. The activation traces correspond to three different EBs, which induce localized activation in the intention field of the higher-level of the hierarchy, defined over color dimension. The bottom-left plot shows the time-course of activation of the correspondent CoS field for colors. The similarity of traces comes from the fact that color is provided as information to the color-segmentation dynamics of the perceptual dynamic field.

A. Integration of sensory-motor behaviors

The right column of Fig. 2 shows the activation of the intention field for the *Search* EB at the bottom level of the hierarchy. The intention field of this EB is defined over the dimension of orientation of the robot's head. On the plots, one may observe movement of the activation in the intention field, produced by movement of the NAO's camera head, tracked by activation in perceptual color-orientation field, Fig. 1.

On the bottom-right plot, offset of red regions mark moments in time, when the CoS field for orientation of NAO's head is activated after receiving enough input from: 1) the orientation sensor in the robot, and 2) the intention field of the *Search* EB (top-right plot). Activation in the CoS field represents the completion of the *Search* behavior, i.e. when the object with the current color is located in the center of NAO's field of view. The combination of these dynamical elements exemplifies the use of DNFs in a complete sensori-motor loop implementation on a robotic platform.

B. Temporal stabilization and robustness of the EBs

The encircled activations in the right-hand side plots of Fig. 2 show the persistence of activity in the *Search* EB through the execution of other motor actions. This can take different amounts of time depending on the geometry of the visual scene.

A zoom over the activation during the execution of the sequence correspondent to the last EB in the top layer is shown in the bottom plot of Fig. 5. Due to the nature of this robotic platform some behaviors such as *Stand up* or *Sit down* can generate jerky motions from the sudden contact with other surfaces; and *Approach* or *Walk back* does not always occur in straight lines due to the smooth contact between NAO's feet and the floor. Moreover, among NAO's repertoire of motor primitives (bottom layer's EBs) there are several discrete actions that block the flow of information from sensors and into motors. The main drawback of this constraint being the loss of continuity in the computation of dynamic fields. Such jumps can be observed in the last group of activations of the right-hand side plots of Fig. 2 and its detailed analysis presented in Fig. 5.

However, the proposed DFT-based methodology cope with these otherwise problematic interactions with the environment

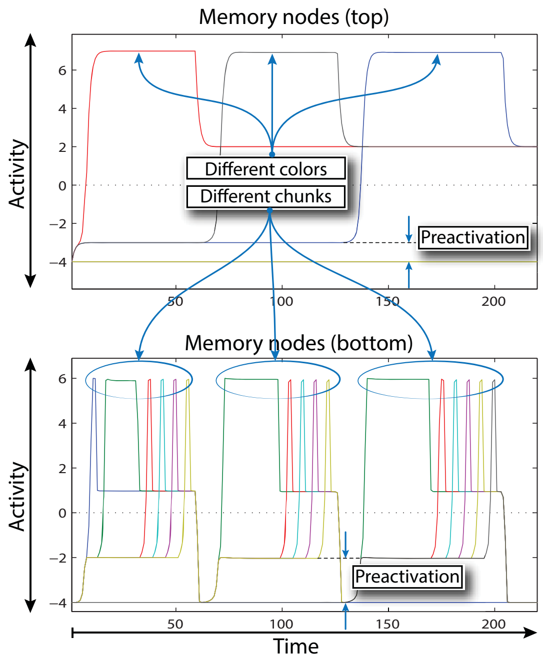

Fig. 3. Activity of memory nodes through time. Each active color (top) starts a chunk of sequential actions (bottom).

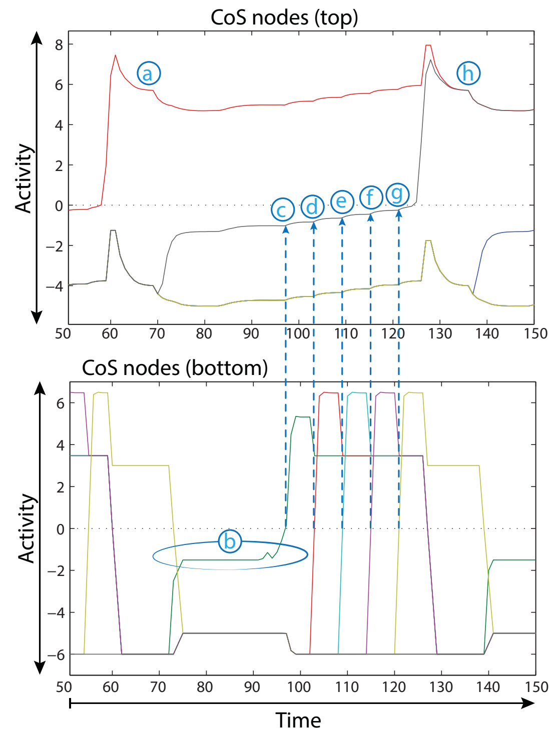

Fig. 4. Time window of CoS nodes in both layers for a single color's actions. Each activation of a CoS node in the bottom layer adds a small input to the CoS node in the top layer until it reaches its own instability and becomes active. See text for a more detailed explanation.

by tracking continuously and dynamically the different inputs and generating continuous and dynamic outputs.

C. Reusing sequential chunks and processing along the hierarchy

Figure 3 provides an overview of the interaction between layers. In this figure, *Memory* nodes are used to show the activation of the color sequence (top plot) in the top layer and the individual sequences of motor behaviors (bottom plot) in the bottom layer. For example the first color makes use of: *Stand up, Search, Approach, Point, Arm down, and Walk back*; whereas the second color re-uses all but the *Stand up* behavior since NAO is already in a standing position. This shows one of the attractive characteristics of the proposed hierarchical serial model, i.e. reusing chunks of sequences, and its successful implementation.

One of the key components of the hierarchical serial order model is the graded inclusion of a lower layer's CoS node activations into the dynamics of an upper layer CoS node. For a more detailed and complete mathematical description of this component and the overall approach the reader is referred to [16]. However, Fig. 4 can be used to graphically explain the process behind these dynamics in both layers.

The curve close to the "a" tag in Fig. 4 shows the activity of an already fulfilled step in the sequence. The tag "b" shows the pre-activation of the CoS node within the *Search* EB, preparing it to receive input from its CoS field, which in this case is given by the orientation of NAO's head. Around time step 95 the CoS node starts receiving a non-zero stimulus from its CoS field, i.e. NAO is close to placing the object in the center of the field-of-view (FOV). Tag "c" marks the moment when the *Search* behavior has been completed adding a small input to the activity of the upper layer's CoS node. Tags "d", "e", "f", and "g" mark the completion of the other behaviors for this sequence, each one adding a small input to the top layer's CoS node. However it is the last input "g" the one that drives the dynamics of the system into a new instability finally activating the CoS node of the current EB in the top layer, "h".

Figure 5 zooms over the sequence of actions for the final color EB. In this specific sequence, tag "a" encircles a continuous *Search* behavior until "b" when the action is considered to be completed. The object remains in the center of NAO's FOV until "c" when NAO has performed an *Approach* behavior and realizes that the object is a bit off-center. At the end of other EBs ("d", "e", "f") is also possible to see such displacements but it is much more obvious after completing the *Sit down* behavior, "g". This process shows the fast adaptation of dynamic fields to unexpected changes in the environment.

IV. Conclusion

The current work presents an implementation of a previously reported model for the generation of hierarchically structured sequences of elementary behaviors, based on dynamic field theory (DFT), [16]. The robotic platform used for this implementation is Aldebaran's NAO, a small humanoid robot that provides enough basic sensori-motor functionality for testing the proposed model. A two-layer hierarchical model was created to represent a group of colors (top layer) and a group of

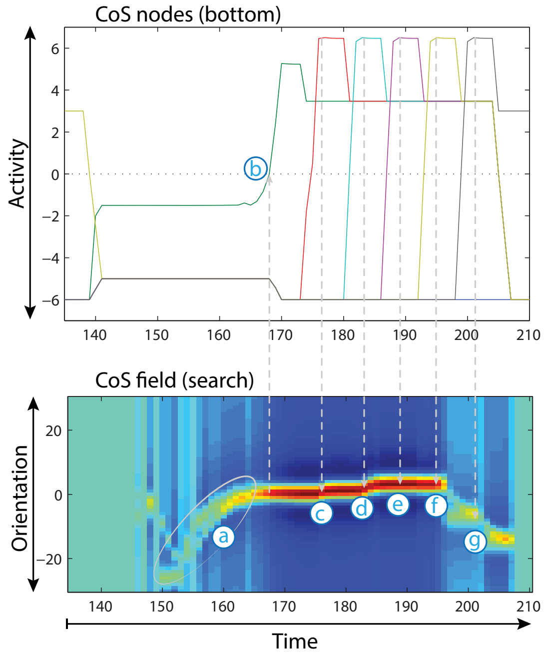

Fig. 5. Time window of CoS nodes in the bottom layer (top) and the CoS field of the *Search* EB (bottom) for the last color's actions. Identifying jumps in the dynamics of a field. See text for a more detailed explanation.

motor primitives (bottom layer). Experiments were performed on a task consisting of a sequence of 3 colors, each activating a subsequence of actions from the bottom layer. The elementary behaviors of the architecture were linked both to low-level sensory-motor dynamics and to algorithmic preprogrammed behaviors, integrating continuous-time dynamics with discrete software modules.

The current implementation in NAO shows that not only environmental inputs, but also the interaction of hardware and environment generates unexpected changes in the sensorimotor loop. Our DFT-based model demonstrates that it is possible to keep track of those changes in a dynamic and effortless way. The current implementation validates the theory behind the design of hierarchical serial order, which emphasizes the importance of stability of the representations of intentions and conditions of satisfaction of elementary behaviors, as well as the robustness of transitions between EBs within and between levels of hierarchy.

Overall, our results demonstrate how a neurally-inspired DNF architecture for production of hierarchically organized action sequences may be implemented in a robotic setting. The robustness of the DNF approach, when integrating algorithmic and dynamical components of the architecture on a rather simple and noisy robotic platform, is verified in the experiments. In our future work we will extend this framework, implementing a neurally-inspired learning mechanism that may autonomously shape the hierarchy of actions based on observation.

The use of Matlab allowed us to implement the different algorithms in a fast and compact coding style. The advantage of using this tool was even greater at the moment of visualizing the results in real time. However this also meant a great decrease in the performance of the whole task. Future work will focus on transporting the proposed model into a more powerful software implementation. Specifically, a multi-thread environment is needed in order to let field and neuron dynamics continuously update their values while sensor inputs and motor outputs are handled in separated processes. A more efficient C/C++ implementation of the framework is envisioned in order to implement all action components with dynamic neural fields.

REFERENCES

[1] T. Asfour, K. Regenstein, P. Azad, J. Schr"oder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Aramr-iii: An integrated humanoid platform for sensory-motor control," *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 169–175, 2006.

[2] R. Bischoff and V. Graefe, "Hermes-a versatile personal robotic assistant," *Proceedings of the IEEE*, vol. 92, no. 11, pp. 1759–1779, 2004.

[3] M. Asada, K. F. MAcDorman, H. Ishigo, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, pp. 185–193, 2001.

[4] Y. Sandamirskaya and G. Schöner, "An embodied account of serial order: How instabilities drive sequence generation," *Neural Networks*, vol. 23, no. 10, pp. 1164–1179, December 2010.

[5] G. Schöner, *Toward a New Grand Theory of Development? Connectionism and Dynamic Systems Theory Re-Considered.* New York: Oxford University Press, 2009, ch. Development as Change of System Dynamics: Stability, Instability, and Emergence., pp. 25–49.

[6] W. Erlhagen and E. Bicho, "The dynamic neural field approach to cognitive robotics." *Journal of Neural Engineering*, vol. 3, no. 3, pp. R36–R54, Sep. 2006. [Online]. Available: http://dx.doi.org/10.1088/1741-2560/3/3/R02

[7] S. K. U. Zibner, C. Faubel, I. Iossifidis, and G. Schöner, "Dynamic neural fields as building blocks for a cortex-inspired architecture of robotic scene representation," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 1, pp. 74–91, 2011.

[8] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14–23, 1986.

[9] R. C. Arkin, *Behavior-Based Robotics.* MIT Press, Cambridge, M, 1998.

[10] M. Mataric, "Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics," *Imitation in animals and artifacts*, pp. 391–422, 2002.

[11] Y. Sandamirskaya, M. Richter, and G. Schöner, "A neural-dynamic architecture for behavioral organization of an embodied agent," in *IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL EPIROB 2011)*, 2011.

[12] D. Wang and M. Arbib, "Timing and chunking in processing temporal order," *Transactions on Systemas, Man, and Cybernetics*, vol. 23, pp. 993–1009, 1993.

[13] M. N. Nicolescu and Mataric, "A hierarchical architecture for behavior-based robots," *Proceefins of the First INternational Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 227–233, 2002.

[14] J. Bryson, "Hierarchy and sequence vs. full parallelism in action selection," *From animals to animats*, vol. 6, pp. 147–156, 2000.

[15] M. Atkin, G. King, D. Westbrook, B. Heeringa, and P. Cohen, "Hierarchical agent control: A framework for defining agent behavior," in *Proceedings of the fifth international conference on Autonomous agents*. ACM, 2001, pp. 425–432.

[16] B. Duran, Y. Sandamirskaya, and G. Schöner, "A dynamic field architecture for the generation of hierarchically organized sequences," in *International Conference on Aritificial Neural Networks*, Submitted.