A Dynamic Field architecture for the generation of hierarchically organized sequences

Boris Durán[1], Yulia Sandamirskaya[2], and Gregor Schöner[2]

[1] University of Skövde, Interaction Lab, P.O.Box 408, 54128, Skövde, Sweden. boris.duran@his.se,
[2] Ruhr-Univesität Bochum, Institut für Neuroinformatik, Universitätstr. 150, 44780 Bochum, Germany.

Abstract. A dilemma arises when sequence generation is implemented on embodied autonomous agents. While achieving an individual action goal, the agent must be in a stable state to link to fluctuating and time-varying sensory inputs. To transition to the next goal, the previous state must be released from stability. A previous proposal of a neural dynamics solved this dilemma by inducing an instability when a "condition of satisfaction" signals that an action goal has been reached. The required structure of dynamic coupling limited the complexity and flexibility of sequence generation, however. We address this limitation by showing how the neural dynamics can be generalized to generate hierarchically structured behaviors. Directed couplings downward in the hierarchy initiate chunks of actions, directed couplings upward in the hierarchy signal their termination. We analyze the mathematical mechanisms and demonstrate the flexibility of the scheme in simulation.

Keywords: Sequences, Dynamic Field Theory, Hierarchies, Intentionality

1 Introduction

Perhaps you have once thought about a new paper while cooking your dinner and keeping an eye on the toddler next door, being interrupted by a phone call and then returning to your last thought. If so, then you were demonstrating a hallmark of human embodied cognition, our capacity to generate well-organized sequences of mental and physical acts while faced with a complex, time-varying environment. Enabling artificial cognitive systems to generate sequences requires theoretical understanding of sequence generation shaped into process models that can be linked to real sensory information, that control real effectors, and that operate in the real world.

Although considerable theoretical work has focussed on sequence generation, often with methods from the theory of neural networks [1–3], the demands of embodiment have received little attention. A critical issue in acting out sequences is the conflict between a need for stability to resist change while performing a step in a sequence, and a need for release from stability to transition to the next step. Previously, we have proposed a neural dynamics that addresses this issue [4]. Each step in a sequence is an attractor state that steers in closed loop the action of an agent toward the intended action goal. A specialized neural structure represents the “condition of satisfaction”, that continuously compares the current intention to perceptual information. When perception matches intention, a dynamical instability releases the current attractor from stability and switches to the next attractor. This proposal was elaborated in simple models [4, 5], in which a set of ordinal units represents the serial order of a sequence. Each ordinal unit projects onto the sensory and motor systems required to achieve each action goal. Although we have recently generalized this idea to enable flexible reorganizations of sequences in a robotic setting [6], this proposal faces a fundamental limitation. Every time a new sequence is learned, new units must be linked to the relevant sensory and motor systems, increasing combinatorially the number of these units and of the associated connections.

Cognitive scientists and roboticists have recognized that scaling the complexity of real-world sequence generation requires hierarchical structures, in which chunks of subsequences and individual action units can be reused flexibly [7–10]. Such hierarchies are conceptually founded on an information processing view of neural computation in which each stage of a computation has a well defined outcome that then triggers the next computational step. This conception makes strong demands on each element of the hierarchy so that it may perform its computation independently of the state of the agent and of its environment. This leaves the stability/instability dilemma unaddressed. Our previous effort was to reduce such demands by making each stage of a sequence a stable state that can operate in closed loop with online sensory information.

In this paper we ask how this principle may be generalized to emulate hierarchies. We propose a neural dynamics in which ordinal neurons may call entire chunks or subsequences at a lower level of a hierarchy. When they do so, they not only trigger the transition to the first element of the chunk, but also set the “condition of satisfaction” that signals the termination of the chunk and organizes the return to the calling point in the hierarchy. This leads to a dynamical hierarchy, in which the progression downward the hierarchy is governed by a dynamics of intention, and the return to higher levels of the hierarchy is governed by a dynamics of conditions of satisfaction. The present paper elaborates the critical mathematical mechanisms and demonstrates these in simulation while a demonstration on a robotic hardware is not yet part of the present contribution. The approach is similar in spirit to earlier work by Stringer and colleagues [11], but enables generalization beyond the motor sequences treated there by making the analysis and synthesis of the dynamical instabilities more explicit.

2 The model

In Dynamic Field Theory (DFT), the states of a cognitive system are represented as attractor states of Dynamic Fields (DFs) [12], defined over behaviorally relevant perceptual, motor, or abstract spaces [13]. Localized peaks are induced in

DFs by external inputs and are stabilized by lateral interactions. Discrete dynamic nodes may be introduced in this framework with an analogous bi-stable dynamics. Bistability of dynamic fields and dynamic nodes enables not only their coupling to real sensory-motor systems, but also coupling to each other through weighted connections, while preserving stability and robustness of the overall dynamics.

Here, we describe two subsequent layers of the hierarchical DF architecture for sequence generation (Fig. 1). Each layer contains a number of elementary behaviors (EBs). The EBs at the lower layer of the hierarchy directly drive overt actions and the EBs at the upper layer may activate subsequences, or chunks, at the lower layer. An EB consists of five elements – three discrete neural activation variables (the memory, ordinal, and condition-of-satisfaction nodes) and two dynamic fields (the intention field and the condition-of-satisfaction field). Their dynamics and couplings are described next.

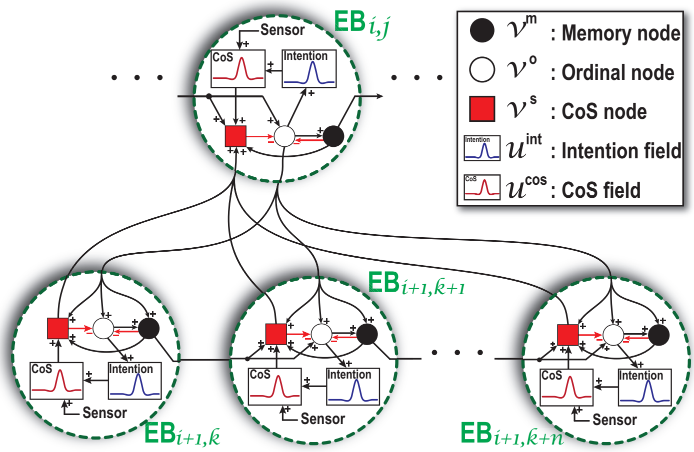

Fig. 1: Illustration of the structure of a single elementary behavior (EB) and the inter- and intra-layer coupling for the hierarchical organization of sequences.

Ordinal nodes. An EB (index j within layer i) is activated by its ordinal node, $v^o_{i,j}$, that generates output through a sigmoidal non-linearity, $\sigma(\cdot)$. In the dynamics

$$\tau^o \dot{v}^o_{i,j} = -\, v^o_{i,j} + h^o + c^{o,o}\sigma(v^o_{i,j}) - c^- \sum_{j'(\neq j)} \sigma(v^o_{i,j'}) - c^{o,s}\sigma(v^s_{i,j})$$
$$+ \sum_k \sum_{j'} W^{o,m}_{i,j,j',k}\sigma(v^m_{i,j'})\sigma(v^o_{i-1,k}) + \sum_k W^{o,o}_{i,j,k}\sigma(v^o_{i-1,k}) \qquad (1)$$

τ^o is the time constant, h^o is a negative resting level, $c^{o,o}$ is the strength of self-excitatory input, c^- is the strength of the mutual inhibition between ordinal nodes in a layer. $c^{o,s}$ is an inhibitory input from the condition-of-satisfaction node of the EB, which deactivates the ordinal node when the EB is finished. $W^{o,m}_{i,j,j',k}$ is a connection weight from the memory node j' to the ordinal node j, both in layer, i that controls the sequential activation of ordinal nodes within layer. There is one weight matrix per possible chunk (index k), each being activated by an ordinal node at the upper layer, $v^o_{i-1,k}$. Finally, $W^{o,o}_{i,j,k}$ are connection weights from ordinal nodes in the upper layer to ordinal nodes of the current layer, which bring ordinal nodes belonging to a given chunk, k, closer to the activation threshold.

Memory nodes. The memory nodes, $v^m_{i,j}$, become activated when the associated ordinal node has been activated. They remain active through strong self-excitatory coupling until the entire chunk, of which they are part, has been executed. An active memory node provides excitatory input to the next ordinal node in a sequence, so that the memory nodes through their connections to the ordinal nodes encode the serial organization of EBs within a layer. In the dynamics

$$\tau^m \dot{v}^m_{i,j} = -\, v^m_{i,j} + h^m + c^{m,m}\sigma(v^m_{i,j}) + c^{m,o}\sigma(v^o_{i,j}) + \sum_k W^{m,o}_{i,j,k}\sigma(v^o_{i-1,k}) \quad (2)$$

notations is analogous to that of Eq.(1). Note that there's no mutual inhibition between memory nodes.

Condition-of-satisfaction nodes. An active CoS node, $v^s_{i,j} : \sigma(v^s_{i,j}) > 0$, signals that the behavioral goals of the EB have been realized. In the activation dynamics

$$\begin{aligned}\tau^s \dot{v}^s_{i,j} = &-\, v^s_{i,j} + h^s + c^{s,s}\sigma(v^s_{i,j}) - c^- \max_j \left(\sigma(v^s_{i,j})\right) + c^{s,m}\sigma(v^m_{i,j}) \\ &+ \sum_k W^{s,o}_{i,j,k}\sigma(v^o_{i-1,k}) + c^{s,cos}\int \sigma(u^s_{i,j}(x'))dx' + \frac{c^+}{l_c}\sum_l \sigma(v^s_{i+1,l}) \quad (3)\end{aligned}$$

the first three terms of Eq.(3) form the generic dynamics of an Amari node, the fourth term is a global inhibition within a layer. Input from the memory node of the EB is scaled by the constant, $c^{s,m}$, and input from the active ordinal node at an upper layer is scaled by the chunk-dependent synaptic weights $W^{s,o}_{i,j,k}$. These two inputs enable activation of the CoS node by contributions that signal completion of the EB encoded in the last two terms of Eq.(3). A positive input from the CoS field, $u^s_{i,j}(x)$, Eq.(5), signals accomplishment of a motor action at the lowest layer. In the upper layer, the CoS node also detects the end of a chunk by collecting input from the CoS nodes within the chunk at the lower layer. This contribution has a constant factor c^+ and is scaled inversely to the length of the chunk l_c. The chunk's length is estimated by summing the non-zero synaptic connections between the memory and the ordinal nodes for the lower layer, $W^{o,m}$, Eq.(1).

When a CoS node becomes active, it inhibits the respective ordinal node and triggers an instability in the dynamics of the EB, which leads to the sequential transition at the current layer of the hierarchy and to a reset of the memory, the CoS, and the ordinal nodes at the lower layer in a reverse detection instability. The CoS nodes thus organize the generation of sequences in time.

Intention field. An intention field, $u^{\text{int}}_{i,j}(x)$, is associated with every EB and receives localized input from the ordinal nodes. An active ordinal node thus specifies parameter values of the intended action. Typical parameters, x, are feature values that specify a perceptual goal (e.g., the color of an object that must be visually searched) or movement parameters (e.g., the direction in which an effector is moved). The dynamics of the intention fields reads:

$$\tau^{\text{int}}\dot{u}^{\text{int}}_{i,j}(x) = -u^{\text{int}}_{i,j}(x) + h^{\text{int}} + \int \sigma(u^{\text{int}}_{i,j}(x'))w^{\text{int}}(x-x')dx' + \sigma(v^{o}_{i,j})W^{\text{int}}_{i,j}(x). \quad (4)$$

The intention field receives input from the associated ordinal node, $v^{o}_{i,j}$, through a synaptic weights function, $W^{\text{int}}_{i,j}(x)$, that would typically be learned [4], but will be set to particular localized gaussian inputs in the simulations below. This input pushes the intention field through the detection instability and induces a localized activation peak. The peak is not sustained, so that it decays when localized input from the ordinal node is removed. As long as the intention field is activated, it impacts on the down-stream sensory-motor structures (dynamic fields and motor dynamics) and sets attractors in these structures that result in the overt behavior of the agent.

Condition-of-satisfaction field. The CoS field, $u^{\text{cos}}_{i,j}(x)$, detects if the intended action or perceptual state of the calling EB, expressed by a peak in the intention field, has been realized. It does so by matching the pattern induced by input from the intention field to sensory information. When a match is detected, a peak forms in the CoS field and activates the CoS node, ultimately triggering a cascade of instabilities that leads to the deactivation of the current EB and the activation of the next EB within a layer. The dynamics of the CoS field reads:

$$\begin{aligned}\tau^{\text{cos}}\dot{u}^{\text{cos}}_{i,j}(x) = -u^{\text{cos}}_{i,j}(x) + h^{\text{cos}} + c^{\text{cos}} \int \sigma(u^{\text{cos}}_{i,j}(x'))w^{\text{cos}}(x-x')dx' \\ + c^{\text{cos,int}} \int \sigma(u^{\text{int}}_{i,j}(x'))w^{\text{cos,int}}(x,x')dx' + I^{\text{cos}}(x). \quad (5)\end{aligned}$$

The CoS field receives input from two sources. The intention field provides localized input at the locations along the dimension of the CoS field specified by the coupling kernel, $w^{\text{cos,int}}(x,x')$. If this input overlaps with the perceptual input, $I^{\text{cos}}(x)$, the CoS field goes through a detection instability and an activation peak builds-up in this field. The peak is not self-sustained and decays when either of the two inputs ceases after the behavior has been realized.

3 Results

To demonstrate how the dynamic field architecture may generate sequences with a hierarchical structure, we describe here an exemplar simulation. We show two layers of the hierarchy: at the top layer, a sequence of three EBs is generated, each driving a subsequence (chunk) of EBs at the lower layer. In particular, we want to make three points here: (1) show how individual EBs and whole chunks may be reused at different positions in the sequences; (2) show how simulated sensory input to the CoS DF may be integrated with the CoS that signals the completion of a subsequence; (3) demonstrate context-dependent, variable timing of the subsequences. In this simulation, the ordinal nodes within a layer project onto a single intention field for simplicity of presentation, but separate intention fields for every EB may also be used if the application requires it [6, 5].

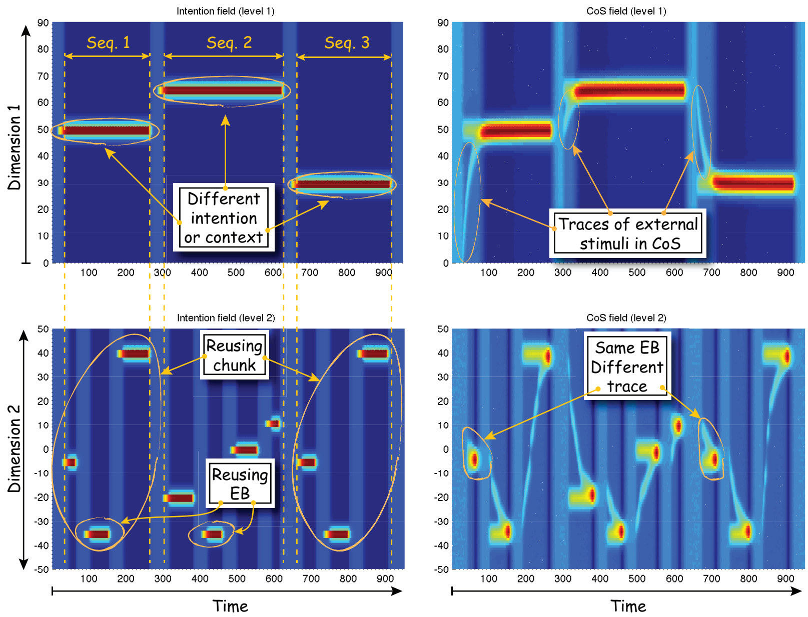

Fig. 2: Activity of fields through time.

Fig. 2 shows on the top-left and activation time-course of the intention field at the upper layer of the hierarchy. The three traces of positive activation correspond to three different action goals for the three EBs at the upper layer. The bottom-left plot shows activation of the intention field at the lower layer. The dashed vertical lines mark the chunks, activated by the upper layer. Note that an EB may be reused in different chunks. Also, entire chunks may be reused at different points in the upper-layer sequence (chunks 1 and 3).

The right column of Fig. 2 shows the time-courses of activation of the CoS fields of the upper (top) and lower (bottom) layer of the hierarchy. Traces of the sensory input can be seen in the CoS fields, which have simple time courses in this simulation, but could form more complex trajectories in a dynamic environment. We modulated the dynamics of the simulated sensory input to emulate the variability in timing of the real-world sensory input. This illustrates that sequence generation is robust against differences in the duration of each EB (see, e.g., first EB of the reused junk across the two repetitions of the chunk). The CoS field of the upper layer is activated as soon as the sensory input overlaps with input from the intention field. The CoS node (not shown), however, is only activated when the chunk at the lower level is finished. Only then is the intention field of the upper layer deactivated (top-left plot) and, consequently, the CoS field of the upper layer deactivated as well (top-right plot).

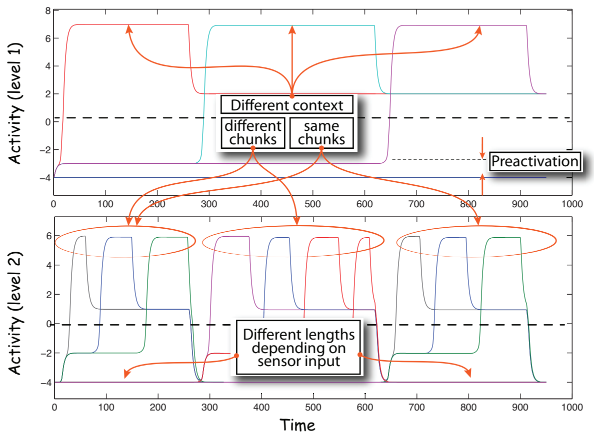

Fig. 3: Activity of memory nodes through time.

Fig. 3 shows the activity of the memory nodes at the upper (top plot) and the lower (bottom plot) layers. Each memory node is activated by the ordinal node of the associated EB. The memory node is brought to the level of self-sustained activation when the EB ceases and stays at this level until the end of the chunk. At the end of the chunk, the ordinal node of the upper layer is inhibited and the memory nodes of the lower layer decay to the negative resting level. The memory node of the upper layer stays at a self-sustained level of activation until the end of the sequence. Note the variable duration of EBs, even for the same chunk at positions 1 and 3 of the upper-layer sequence.

4 Discussion

The neural dynamics proposed in this paper provides a general and robust mechanism for transitioning from the stable state that each stage of a behavioral sequence represents to its successor state. The mechanism enables inserting a chunk or subsequence at any point and then returning to the ongoing sequence at the higher level of an implied hierarchy. Transitions to a lower level of the hierarchy are based on the same principle as transitions within each level and enable sustaining the stable state at each step for variable amounts of time until perceptual information matches the expected outcome of the stage, inducing a detection instability in the associated “condition of satisfaction” (CoS) system. This concept of a CoS is also used to organize the transition from the end of a chunk to the next stage of a sequence at the higher level, from which the chunk was initiated. The hierarchical organization enables reuse of the same chunk at different points in the sequence. Moreover, different chunks may overlap, reusing elementary behaviors or subsequences. This core mechanism opens the neural dynamics of sequence generation to the much richer class of hierarchically organized sequences.

References

1. Grossberg, S.: 6. The Adaptive Self-organization of Serial Order in Behavior: Speech, Language, and Motor Control. In: Pattern Recognition by Humans and Machines: Speech Perception. (1986) 187–396
2. Houghton, G.: The problem of serial order: a neural network model of sequence learning and recall. In: Current research in natural language generation. Academic Press Professional, Inc. San Diego, CA, USA (1990) 287–319
3. Cooper, R., Shallice, T.: Contention scheduling and the control of routine activities. Cognitive Neuropsychology **17**(4) (2000) 297–338
4. Sandamirskaya, Y., Schöner, G.: An embodied account of serial order: How instabilities drive sequence generation. Neural Networks **23**(10) (December 2010) 1164–1179
5. Sandamirskaya, Y., Schöner, G.: Serial order in an acting system: a multidimensional dynamic neural fields implementation. In: Development and Learning, 2010. ICDL 2010. 9th IEEE International Conference on. (2010)
6. Sandamirskaya, Y., Richter, M., Schöner, G.: 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)
7. Dehaene, S., Changeux, J.P.: A hierarchical neuronal network for planning behavior. Proceedings of the National Academy of Sciences of the United States of America **94**(24) (November 1997) 13293–8
8. M, N.N., J, M.M.: A hierarchical architecture for behavior-based robots. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems. (2002)
9. Bryson, J.J.: The study of sequential and hierarchical organisation of behaviour via artificial mechanisms of action selection. Citeseer (January 2001)
10. Cooper, R., Shallice, T.: Hierarchical schemas and goals in the control of sequential behavior. Psychological Review **113**(4) (2006) 887–916
11. Stringer, S.M., Rolls, E.T., Taylor, P.: Learning movement sequences with a delayed reward signal in a hierarchical model of motor function. Neural Networks **20**(2) (2007) 172–181
12. Amari, S.: Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics **27** (1977) 77–87
13. Schöner, G.: Dynamical systems approaches to cognition. In: Cambridge Handbook of Computational Cognitive Modeling. R. Sun, UK: Cambridge University Press (2008) 101–126