Grounded representations through deep variational inference and dynamic programming

Juan Sebastian Olier[§][†][*], Emilia Barakova[§], Matthias Rauterberg[§],
Lucio Marcenaro[†] and Carlo Regazzoni[†][‡]
[§] Department of Industrial Design, Eindhoven University of Technology, The Netherlands
[†]Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture
University of Genoa, Italy.
[‡]Intelligent Systems Lab, University Carlos III de Madrid, Spain
[*] J.S.Olier.Jauregui@tue.nl, sebastian.olier@ginevra.dibe.unige.it

Abstract—**In this work, we present a method for building grounded representations by structuring the sensorimotor data of an agent. The aim is to encode sensory inputs into internal states that describe action-environment couplings, or relations that connect elements in a scene to action concepts. Thus, the environment is represented regarding the sensorimotor integrations required to interact with elements in it. Such representations are acquired for a particular task in which they serve to infer important states and generate control commands accordingly. Representations are learned in an unsupervised process and are assembled into probability distributions to capture uncertainty. That is achieved through variational methods using deep learning and dynamic programming, and an architecture partially inspired by active inference. During an interaction, future representational states and sensorimotor data are actively predicted, while the incoming sensory information is incorporated through prediction error. Results in a navigation task show that situated representations emerge as sensorimotor relations that are interpretable as action concepts, and which allow interpreting important elements in the environment to generate satisfactory actions. We argue that the acquisition of such capabilities relates to the prediction processes enabled by two mechanisms; first, state-transition models explicitly dependent on the generation of control commands, and secondly, a system trained through dynamic programming, which generates further predictions that relate sensory data to expected state changes.**

I. Introduction

Building adequate representations is an essential skill for intelligent agents to interact with the world properly. In particular, sufficient representational capabilities can allow to predict future states and define the actions to be taken in a specific situation. Those features imply representations that dynamically relate sensory inputs to possible future outcomes, or which allow embedding intuition or common sense. Nonetheless, it is not clear how such representational mechanisms can be acquired and grounded in the sensorimotor systems.

Furthermore, given that skills such as predicting future states and producing adequate actions are task-dependent, no general characteristics of elements in the environment concerning actions can be assumed. Thus, no semantics or externally given symbol-like structures about data can be regarded as part of the grounding process. That is, the acquisition of grounded representations should work in an unsupervised manner, and the process should be the same independently of the task. Similarly, since at a given time the agent only has access to its sensory data and its current internal state, no explicit information about future states can be considered for acquiring prediction capabilities.

In this work, we address that problem by linking sensorimotor activities to abstraction and representational mechanisms from which motion and sensory inputs are predicted. To that aim, we follow the theoretical development we introduced in [1], where we support a view on conceptual representations as dynamic structures that evolve during interactions, and not as concrete and stable arrangements based on categories.

Such views are linked to the ideas of embodied cognition (EC), where a clear relation between action, perception, and representations is highlighted. Particularly, in [2] it is proposed that in EC the need for concrete or symbolic representations are excluded by the idea of bodies perceptually coupled to the environments. In consequence, symbolic representations of objects are not seen as required since they can be replaced by the dynamics of actions related to them in a given situation.

Similarly, Engel et al. [3] argue that cognition should be studied with respect to action generation as a capability for creating structure. In that way, internal states acquire meaning by their role in behavior production. Thus, one should not aim at symbolic or static representations, but at dynamic representational mechanisms that evolve with interaction and make sense for a particular task and context.

To define a concrete mechanism for connecting representations to the sensorimotor system, we consider some ideas from Grounded cognition (GC) [4]. In GC simulation capabilities are central and are understood as the reenactment of sensorimotor modalities. Moreover, concepts are seen as situated, meaning that a situation has to be simulated during processing. Thus, a conceptual representation cannot be context-agnostic by default and should be dynamically connected to the sensorimotor system through simulations. Accordingly, and in line with [1], conceptual representations are here understood as dynamic processes that are flexible, context-dependent and adapt to specific situations. Then, the meaning of elements in an environment is to be described regarding behaviors

978-1-5386-3715-9/17/$31.00 ©2017 IEEE

associated to, or caused by them. In other words, meaning lays on the sensorimotor integration required to interact with such perceived elements.

The described characteristics of grounded representations are considered as particularly relevant since they are necessary for developing more complex phenomena such as the emergence of language, dynamic reasoning, or planning skills. Thus, given such inherent potentials, the primary focus of this paper is on how to ground the kind of representations described. Specifically, we present a method to structure sensorimotor data of an agent, by constructing states that represent the environment regarding behavioral concepts, or as the agent-environment coupling in a situated task. All of that is achieved in an unsupervised form, meaning that the agent only has access to its sensory data and its internal states at a given time; no additional information about the environment, or about future states is given.

A. Active inference and representations

To achieve the described features in section I, we follow the arguments in [1] supporting the proposals by Friston et al. [5], [6] as the framework for a computational solution. In [5], [6] Friston and colleges argued that the primary function of the brain is to suppress prediction error. In particular, they suggest that motor control has a role in reducing uncertainty when understood as a mechanism to fulfill proprioceptive expectations. Those ideas, known as active inference (ActInf), relate to predictive coding [7], where upper elements in a hierarchical structure send down predictions about lower levels, and feed-forward connections carry residual prediction errors. Moreover, given variabilities in the world, the noise of sensors and actuators, and the uncertainty about temporal relations of elements in the environment, ActInf assumes that internal states and predictions are to be encoded in probabilistic terms.

ActInf proposes a hierarchical structure, where states of higher levels infer causes of the dynamics of lower ones. Thus, more abstract representations, i.e. higher levels, are to be more stable over time than the lower ones, which are in turn associated with more immediate behavioral reactions. How to ground such representational states is one of the central parts of this work. In particular, we address the challenge is the acquisition of the generative and recognition models for encoding and predicting sensory data. We use the principles of ActInf to propose a method that constructs internal representational states concerning causes of observations and describing action-environment couplings in a particular situation and task.

II. Approach

We propose a model, to which we refer as predictive grounding architecture (PGA), and aims to encode causes of observed behaviors of an agent performing a particular task in states of two different representational levels. Such states are denominated Z_t and S_t respectively, as depicted in figure 1.

States in the lower level (Z_t) represent causes of observed input data at a given time by taking into account the immediate dynamics of the interaction. The states of the second representational level (S_t) infer causes of the dynamics of Z_t over time. In a particular situation, Z_t is to represent the state of the interaction at time t. In turn, S_t would encode the dynamics of Z_t in more steady states, for example in action concepts such as, avoid an obstacle or go straight, which would be stable for longer periods. The information processing in PGA can be thought as a three steps process.

1) Update: Incoming sensory data and previous states are used to estimate the current state of the world (Z_t).
2) Prediction, divided in two parts.
   i) Action generation: Actions U_{t+1} are predicted from Z_t since control signals are seen as a way of fulfilling expectations in the following time step.
   ii) Prior estimation: Z_t and U_{t+1} are used to predict the next state in the form of a prior (Z_{t+1}).
3) Input estimation: given the prior, expected sensory inputs are estimated as the distribution $P(X_{t+1}|Z_{t+1})$.

Such process is affected by S_t through the inference of actions ($U_{t+1}|Z_t, S_t$)), which could be interpreted as a switching variable in a probabilistic graphical model (see figure 1). In other words, the estimated causes of the dynamics at a given time step, i.e. S_t, are used for defining the action to take, and through that, estimate a prior and predict future sensory inputs.

Moreover, to address the relation of state updates to prediction error, we assume the input to the first representational level to be proportional to such error as elaborated in II-B.

As a result of this process, it is expected that important elements in the environment are actively interpreted, and adequate behaviors dynamically generated for the interaction on which the system is trained. As partially inspired by ActInf, it is assumed that if a PGA achieves low prediction error for motor and sensory data, then the architecture is building sufficient internal states to describe the interaction.

Some ideas of PGA relate to the work of Tani and colleges [8], [9], but PGA focuses on internal probabilistic states, and the higher representational level is learned through dynamic programming and not through stated time constants for leaky recurrent states. Moreover, in PGA actions are fundamental for state transitions, thus not just a different signal to be predicted. Another approach based on predictive coding has been presented in [10], where a hierarchical system based on convolutional and recurrent neural networks is trained to minimize prediction error. That approach shows good predictive capabilities, but all representations are deterministic, thus do not explicitly encode uncertainty. Additionally, in the model in [10] upper levels represent error information in lower ones, which does not necessarily lead to extract more abstract or stable states. To address those issues, PGA is based on variational deep generative models. Particularly, we focus on [11], [12] where systems are optimized end-to-end to build continuous variational representations, and on [13], [14] where Bayesian filters are learned through back propagation. Moreover, PGA is partially based on the method in [1]. The main differences are that in [1] only one representational level has been explored, and time dependencies are addressed

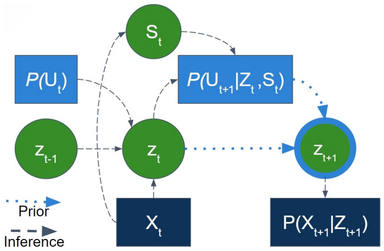

Fig. 1. Schema of PGA. Z_t is the first level of representation, and S_t the second one. X_t is the input data and U_t refers to the control.

by encoding sequences with LSTMs. Here a markovianity is assumed; thus the recursion takes into account only the last step as in a Bayesian filter.

For PGA we assume an agent with sensory inputs, including proprioceptive information, and defined motor capabilities.

A. Predictive grounding architecture

For building grounded representations we focus on the principles of variational auto-encoders (VAE)[11]. In a VAE latent variables Z capture variations in observations, denoted by X. An inference model $P(Z|X)$ given by $Z \sim \mathcal{N}\left(\mu_z, diag(\sigma_z^2)\right)$ is assumed, and is parametrized by a neural network (NN). The prior $P(Z)$ is assumed as a normal distribution. The conditional generative model $P(X|Z)$ is also approximated via a NN, and assumed to be Gaussian. To train a VAE the Kullback-Leibler (KL) divergence between $P(Z|X)$ and $P(Z)$ is minimized, and $log(P(X|Z))$ maximized.

B. First representational level

For a diagram of the first level see figure 2 (left).

As mentioned before, the input to the first representational level is given in terms of prediction error. In this case, as elaborated in [6], the error is related to the precision or inverse covariance. The covariance matrices of predicted data are assumed to be diagonal. Thus, the error for a sensory input X_t given a prediction parametrized by μ_{x_t}, and $\sigma^2_{x_t}$, is given by: $\epsilon_{X_t} = \left(\mu_{x_t} - X_t\right)/\sigma_{x_t}$.

The three steps conforming the process, update, prediction and input estimation, are described as follows:

1) Update: In PGA, a central mechanism for encoding the dynamics of a task is the modeling of time dependencies in the state transitions. To that aim, the calculation of Z_t depends on Z_{t-1}, along with the input data and the the last action taken. Thus, the update is of the form: $Z_t|X_t, U_t, Z_{t-1}$. The sub index for the control is t and not $t-1$ since U_t is interpreted as the action taken to get from Z_{t-1} to Z_t. The parameters of $P(Z_t|X_t, U_t, Z_{t-1})$ are given by $[\mu_{z_t}, \sigma_{z,t}] = \varphi^Z(\epsilon_{X_t}, U_t, Z_{t-1})$.

2) Prediction: The parameters μ_{u_t} and $\sigma_{u,t}$ of the distribution $P(U_{t+1}|Z_t, S_t)$ are approximated by $\varphi^U(Z_t, S_t)$. That is a distribution over actions to be taken to go from Z_t to Z_{t+1}.

When Z_t and S_t are the input to φ^U, and to other networks, the distributions are sampled. During training, as in [11], Z_t is sampled as $z_t = \mu_{z_t} + \sigma_{z_t} * \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, I)$. For testing, the maximum likelihood is used, i.e. μ_{z_t}. S_t always corresponds to the maximum likelihood.

Once Z_t and U_{t+1} have been estimated, the prior about the next state is given by: $Z_{t+1} \sim \mathcal{N}\left(\mu_{o_t}, diag(\sigma^2_{o_t})\right)$ where $[\mu_{o_t}, \sigma_{o,t}] = \varphi^{prior}(Z_t, U_{t+1})$.

3) Input estimation: The generative model does not predict sensory inputs from Z_t directly but from the prior $Z_{t+1}|Z_t, U_{t+1}$. Thus, sensory inputs at $t+1$ are estimated as $P(X_{t+1}|Z_{t+1})$, which, as in a VAE, is assumed to be Gaussian: $X_{t+1} \sim \mathcal{N}\left(\mu_{x_{t+1}}, diag(\sigma^2_{x_{t+1}})\right)$ where $\left[\mu_{x_{t+1}}, \sigma_{x_{t+1}}\right] = \varphi^X(Z_{t+1})$. φ^Z, φ^U, φ^{prior} and φ^X are all approximated by NNs.

C. Second representational level

A simple diagram of the second representational level can be seen in figure 2 (right). The first level encodes the dynamics of the agent-environment coupling between two time steps. It is necessary to predict further in time to build more general descriptions of data, and to infer environment-behaviour relations. Nonetheless, at time t, no information about future states can be assumed since the agent only has access to its sensory inputs and internal states. Hence, such knowledge about the future must be incrementally acquired with experience.

To address that and build such representations we develop an algorithm based on dynamic programming and inspired by deep reinforcement learning[15], and based on the method in [16]. The differences with [16] reside in the interpretation of S_t, its connection to the first level, and the training mechanism, since in [16] S_t is does not predict changes in Z_t.

The method assumes a linear process in which given U_{t+1} the difference between two states is expressed as $Z_{t+1} - Z_t = \Delta Z_t$. If such process is repeated for τ time steps, the final state would be $Z_{t+\tau} = Z_t + \Delta Z_\tau$. In general:

$$\Delta Z_\tau = \Delta Z_t + \sum_{i=1}^{\tau}(\Delta Z_{t+i}) \qquad (1)$$

If τ is assumed to be big in relation to the average actions in the task at hand, the attempt to predict ΔZ_τ from the information at time t would have high levels of uncertainty. Thus, instead of predicting the difference to a final state, one would like to estimate the expected change over time from the knowledge at time t. As in [15], that can be achieved by introducing an exponential discount factor $\gamma < 1$ in the sum of equation 1. In that way, a lower weight or relevance is given to predictions further in time. That yields:

$$\Delta Z_\tau = \Delta Z_t + \sum_{i=1}^{\tau}(\gamma^i * \Delta Z_{t+i}) \qquad (2)$$

In equation 2 ΔZ_τ can be interpreted as the expected change in state given the interaction and the information at time t. As in [15], the higher γ is, the further in time the states are being predicted, and thus more uncertainty included.

Letting $\phi^s(X_t) = \Delta Z_\tau$, it can be shown easily that:

$$\gamma * \phi^s(X_{t+1}) = \sum_{i=1}^{\tau+1}(\gamma^i * \Delta Z_{t+i}) \qquad (3)$$

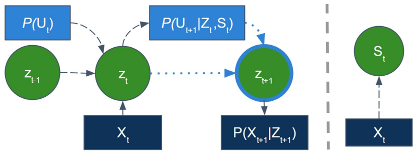

Fig. 2. Separated architectures of the two representational levels (Left: first level, right: second level). This figure illustrates the separated descriptions.

Given the assumption of a large τ, the last term of the sum in equation 3 can be neglected, and thus the sum in 2 be replaced by $\gamma * \phi^s(X_{t+1})$ yielding:

$$\phi^s(X_t) = \Delta Z_t + \gamma * \phi^s(X_{t+1}) \tag{4}$$

The recursion in equation 4 could be directly used to train a NN with dynamic programming as in [15]. Nonetheless, as S_t is to encode variability in the relations between input data and predicted states in terms of probability; we consider $S_t \sim \mathcal{N}\left(\mu_{S_t}, diag(\sigma^2_{S_t})\right)$, where $[\mu_{S_t}, \sigma_{S_t}] = \varphi^s(X_t)$. Then μ_{S_t} corresponds to $\phi^s(X_t)$ in equation 3, and σ_{S_t} encodes the mentioned variability. φ^s is approximated by a NN.

D. Training

For the first level three objectives are optimized through stochastic gradient decent (SGD), the KL divergence between the prior and Z_t, and the likelihoods of control prediction, and sensory inputs. This is done by maximizing the following equation for every sample training sequence of T time steps:

$$\mathcal{L}_\mathcal{T} =$$
$$\tfrac{1}{T}\sum_{t=1}^{T}[-KL\left(P(Z_t|\epsilon_{X_t}, U_t, Z_{t-1})||P(Z_t|U_t, Z_{t-1})\right) +$$
$$log(P\left(X_{t+1}|Z_{t+1}\right)) + log(P\left(U_{t+1}|Z_t, S_t\right))$$

A second part for φ^s is performed simultaneously by means of two different networks, namely φ^s and $\varphi^{s'}$, used to optimize

$$\varphi^s(X_t) = \Delta Z_t + \gamma\varphi^{s'}(X_{t+1})$$

The weights of $\varphi^{s'}$ are kept constant for k time steps (in this work $k = 10$), during which the weights of φ^s are updated to maximize $log(P(S_t|X_t)$; that is, the likelihood of the distribution $P(S_t)$ generating $\Delta Z_t + \gamma * \varphi^{s'}(X_{t+1})$ given X_t. After k steps, the weights from φ^s are copied to $\varphi^{s'}$. Thereby, the training maximizes $log(P(S_t|X_t)$ using as data points (X_t, X_{t+1}) pairs.

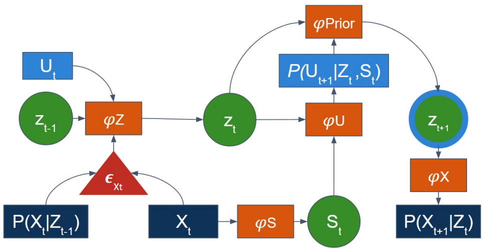

Fig. 3. Implementation schematic. Orange boxes represent NNs, green circles are variational representations, light blue boxes are control signals, and dark blue boxes correspond to inputs, either measured or estimated.

E. Implementation details

The NNs described are implemented as follows (figure 3):

$\varphi^Z$is divided in three parts; φ^{Z_p}, φ^{Z_v} to encode the video and the proprioceptive inputs respectively. A third part fuses the information from φ^{Z_p} and φ^{Z_v}.

φ^{Z_v} is implemented via a convolutional neural network (CNN) with 4 convolutional layers, and filters of size 3 by 3 in banks of sizes [16, 32, 64, 128]; the first three convolutions are followed by max-pool operators of size 2 by 2. The output of the last convolution is followed by a fully connected layer (FC) of size n. φ^{Z_p} is implemented with 3 FC of size n. The output of φ^Z is constructed by fusing the outputs of φ^{Z_v} and φ^{Z_p}, which are stacked and forwarded through what we call a Gaussian coder (GaussCod). A GaussCod is formed by two FC of size n, followed by two separated FC of size z_t, one for the means and other for the variance.

φ^U and φ^{prior} are GaussCods, with input size given by the sum of sizes of Z_t and S_t, and of z_t and U_t respectively.

φ^X is divided in two parts, φ^{X_v} for visual data, and φ^{X_p} for proprioception. For φ^{X_v} a deconvolutional neural network [17] is used. The input to φ^{X_v} is constructed by a FC generating an output from Z_{t+1} with the same size of the output of φ^{Z_v}, which is then reshaped to a tensor form. The sizes of the filters are the same as in φ^{Z_v} though for up-sampling the stride of the convolution are set to 2 where needed. The last deconvolution is performed by two filter banks, one for the means and one for the variances. For φ^{X_p} a GaussCod is used, with output of the size of proprioception signals, and input with size of Z_t.

$\phi^s(X_t)$ is built with a CNN as the one for φ^{Z_v}, followed by a GaussCod with input of size n coming form a FC after the last convolution, and output the size of S_t.

For the evaluated task $n = 64$, while Z_t and S_t are bivariate Gaussian distributions.

III. Evaluation

PGA is trained on data from a simulated environment. The data comes from a two-wheels robot controlled through differential speeds. The sensory data is acquired from a mounted camera, and a two axes accelerometer (proprioception). The control corresponds to the speed applied to the wheels. The environment is a closed squared space surrounded by walls, where objects of different sizes and visual characteristics can be found in no particular order.

The training data correspond to the sensory-motor information from a robot while navigating. The robot moves based on a specified controller that drives it by avoiding the obstacles and walls when approaching them and going towards the most open space visible otherwise. Some noise is added to the speed and acceleration signals to simulate randomness in the actuators. For simulating we use Webots [18].

Two data sets are created for training and testing. The difference lays in the position and visual characteristics of obstacles. The positions are changed randomly to generate different trajectories. Both training and testing sets include 5000 samples of each signal. For training sequences of 16 time steps are fed in mini batches of the same size. The control

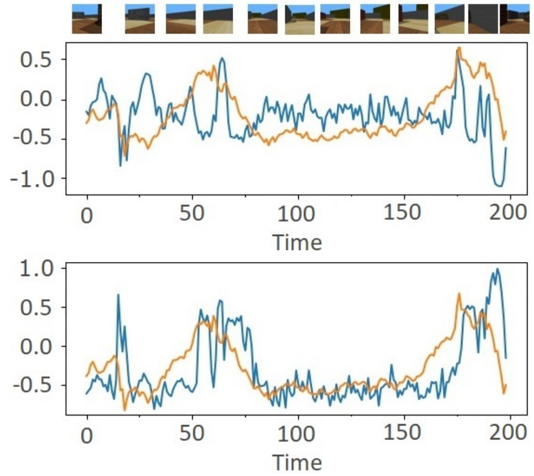

Fig. 4. Upper part, the normalized motion, as the differential speed of the wheels (blue), and the normalized predicted state form $S_t + Z_t$ (orange). In the bottom part, Z_t (blue) and $S_t + Z_t$ (orange), both normalized. The upper pictures relate the graphs to the visual input over time.

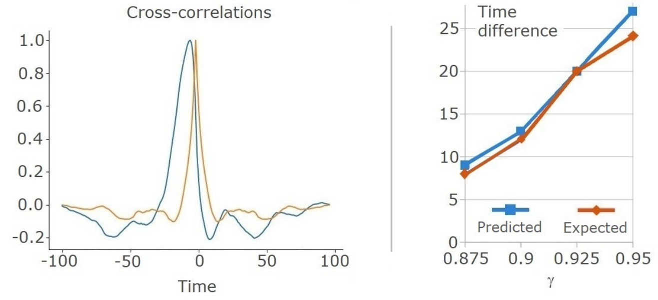

Fig. 5. Left: normalized auto-correlation of Z_t(orange), and normalized cross-correlation between Z_t and $Z_t + S_t$ (blue). Right: maximum distance between auto-correlation and cross-correlation peaks for different values of gamma. Original control signals are compared to predicted ones generated from $Z_t + S_t$ and the expected ones at time τ generated from Z_τ.

signal has size 2, corresponding to the wheels; the size of proprioception is 2 (X and Y axes of the accelerometer). The visual input is an RGB image of size 32 by 32.

On that data, PGA is expected to construct useful representations to enact internalized action-environment couplings with no assumption about specific semantics in the data. The primary goal is to show how grounded representations can be built from the abstraction of such couplings.

IV. Results

A. Prediction and causes representation

Prediction achieved by $\phi^s(X_t)$ directly form the input image captures the situations relevant to the task with a degree of anticipation. To show that, in figure 4 the value of the principal component of the expected values of $S_t + Z_t$ over time is shown (orange). Latent variables are bivariate, but, results show that the main part of the information about motion is encoded in a single component, which is used here for ease of illustration.

In the upper graph of figure 4, the values of a component of $S_t + Z_t$ (orange) over a sequence of 200 time steps is compared to the differential speed of the wheels in the same period. In the lower graph of the figure, the same component of $\phi^s(X_t) + Z_t$ is compared the equivalent component in Z_t. In the most upper part of figure 4, snapshots from the visual inputs are arranged in time to visualize the internal states interpreted as causes inferred in relation to the actual elements in the environment.

The comparisons in the graphs show how the predictions produced by $S_t + Z_t$ stably separate different action concepts such as turning or going straight. In particular, the stability of predictions ($S_t + Z_t$) is clear in comparison to the different representational states in Z_t produced to generate the needed motion for turning (periods roughly from 25 to 75, and 150 to 200 in figure 4). Those states of Z_t are grouped together in more stable representations by $S_t + Z_t$ which predicts and generalizes movement sequences over time.

Comparing also to the snapshots in the upper part of figure 4, it can be seen how higher values can be semantically related to obstacles or walls in the example. The corresponding motion reactions are predicted several time steps in advance by $S_t + Z_t$. In other words, the network is learning to represent elements and anticipate their consequences from different distances, and not only in their immediacy. This could be relevant also for more complex agents that may need to prepare actuators when anticipating a motion in advance; or can be interpreted as a way of planning. But more importantly, this shows the emergence of a semantic-like representation directly from streams of sensorimotor data. Those results can also be interpreted as a filtering capability over the state space of Z_t, which relates to a representational space of causes for the observed behaviors.

To further evaluate the prediction capability of ϕ^s, a cross-correlation between $Z_t + S_t$ and Z_t is compared to the auto-correlation of the last. In figure 5 (left), it can be appreciated that the center of the cross-correlation is shifted to the left, meaning an anticipation. Moreover, the wider peak shows the encoded uncertainty. All graphs in figure 5 are the result of averaging a total of 100 sequences of size 100 in the test data.

A similar analysis is performed for the control signals; however, it is not direct in that case as the control prediction depends on a $\tau > t$, which is not specified, but depends on γ. Thus, in order to determine such τ in relation to control predictions we compare the auto-correlation of the original control signals to cross-correlations between predicted and original signals. The cross-correlation of two kinds of produced controls are compared, the expected one generated from Z_τ, for an estimated $\tau > t$, and the predicted one generated from $Z_\tau^{'} = Z_t + S_t$. For all possible $\tau > t$ in a test sequence the argmax of the cross-correlations are subtracted form the argmax of the auto-correlation of the original signals. Then, the maximum distance between them determines the value of τ. That process is repeated for PGAs with different values of γ. The results appear in figure 5 (right). For the range of γ evaluated an almost linear relation is found.

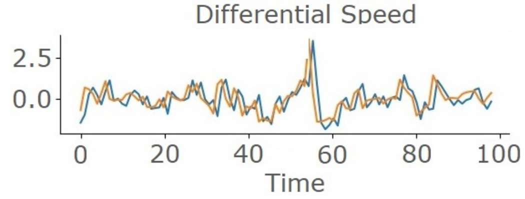

Fig. 6. Original differential speed (blue), and the predicted by PGA (Orange).

B. Control

The capabilities of the system to reproduce the expected behaviors are tested by comparing the generated and expected control signals. In figure 6 an example is depicted. The time step shift is due to the fact that the generation is predicting U_{t+1} at time t. The result of averaging over the absolute difference between predicted and actual control over the whole testing set gives an error of 3.9%.

V. Conclusions and future work

In this work, we have introduced a method for constructing grounded representations, through which action-environment couplings inherent to a given task are internalized. Representations are used to generate predictions and control signals. To that aim, the ideas of active inference and predictive coding have been used as inspiration. As mentioned, in [9] and [10] these ideas have also been employed. Our approach differentiates from [9], [10] by focusing on creating representations in terms of probability distributions, thus explicitly capturing temporal relations and uncertainty in the data. Moreover, in PGA the different representational levels have a direct relation to the sensory data, allowing for processing different abstraction levels in parallel while linking them through the generation of action estimates. The dependency of the state transition on the control is central to the idea of concepts as situated and dynamic phenomena, since building such link implies capturing an action-environment coupling that holds meaning in the context of the task at hand.

Possible future work includes the evaluation of the advantages that more levels could have in building more abstract representations, particularly if the agent is trained with data from more than one task. In that case, one could interpret a third level regarding motivations for one task or the other. Equally, extending the time dependencies to the second representational levels could also bring more stability.

Moreover, the usefulness of built representations could be demonstrated by using them for a task different to the one on which the system has been initially trained. Similarly, PGA could also be modified to be trained with, for example, reinforcement learning. In such scenario, a learned action policy would be completely linked to the grounded representations of the environment, and dependencies on previous states could be beneficial in comparison to actions produced only from observations. Finally, the results achieved with the NN φ^s show the possibility of learning to classify visual inputs in terms of action concepts from unlabelled data.

Acknowledgment

-This work was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012.

-Carlo Regazzoni has contributed to produce this work partially under the program UC3M-Santander Chairs of Excellence.

References

[1] Juan Sebastian Olier, Emilia Barakova, Carlo Regazzoni, and Matthias Rauterberg, "Re-framing the characteristics of concepts and their relation to learning and cognition in artificial agents," *Cognitive Systems Research*, vol. 44, pp. 50 – 68, 2017.

[2] Andrew Wilson and Sabrina Golonka, "Embodied cognition is not what you think it is," *Frontiers in Psychology*, vol. 4, pp. 58, 2013.

[3] Andreas K Engel, Alexander Maye, Martin Kurthen, and Peter König, "Where's the action? the pragmatic turn in cognitive science," *Trends in cognitive sciences*, vol. 17, no. 5, pp. 202–209, 2013.

[4] Lawrence W Barsalou, "Grounded cognition," *Annual Review of Psycholgy.*, vol. 59, pp. 617–645, 2008.

[5] Karl John Friston, Thomas H. B. FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo, "Active inference: A process theory," *Neural Computation*, vol. 29, no. 1, pp. 1–49, 2017.

[6] Karl Friston, Rick Adams, Laurent Perrinet, and Michael Breakspear, "Perceptions as hypotheses: Saccades as experiments," *Frontiers in Psychology*, vol. 3, pp. 151, 2012.

[7] Rajesh PN Rao and Dana H Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[8] Jun Tani, Karl Friston, and Simon Haykin, "Self-organization and compositionality in cognitive brains [further thoughts]," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 606–607, 2014.

[9] Jungsik Hwang, Minju Jung, Jinhyung Kim, and Jun Tani, "A deep learning approach for seamless integration of cognitive skills for humanoid robots," in *The Sixth Joint IEEE International Conference Developmental Learning and Epigenetic Robotics (ICDL-EPIROB)*. IEEE, 2016.

[10] William Lotter, Gabriel Kreiman, and David Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[11] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[12] Danilo J Rezende, Shakir Mohamed, and Daan Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1278–1286.

[13] Rahul G Krishnan, Uri Shalit, and David Sontag, "Deep kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.

[14] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt, "Deep variational bayes filters: Unsupervised learning of state space models from raw data," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[16] Juan Sebastian Olier, Damian Andres Campo, Lucio Marcenaro, Emilia Barakova, Carlo Regazzoni, and Matthias Rauterberg, "Active estimation of motivational spots for modeling dynamic interactions," in *The 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2017)*. IEEE, 2017.

[17] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[18] Webots, "http://www.cyberbotics.com," Commercial Mobile Robot Simulation Software.