Designing Worm-inspired Neural Networks for Interpretable Robotic Control

Mathias Lechner[1,3*], Ramin Hasani[1*], Manuel Zimmer[2,4], Thomas Henzinger[3], and Radu Grosu[1]

***Abstract*— In this paper, we design novel liquid time-constant recurrent neural networks for robotic control, inspired by the brain of the nematode, *C. elegans*. In the worm's nervous system, neurons communicate through nonlinear time-varying synaptic links established amongst them by their particular wiring structure. This property enables neurons to express liquid time-constants dynamics and therefore allows the network to originate complex behaviors with a small number of neurons. We identify neuron-pair communication motifs as *design operators* and use them to configure compact neuronal network structures to govern sequential robotic tasks. The networks are systematically designed to map the environmental observations to motor actions, by their hierarchical topology from sensory neurons, through recurrently-wired interneurons, to motor neurons. The networks are then parametrized in a supervised-learning scheme by a search-based algorithm. We demonstrate that obtained networks realize interpretable dynamics. We evaluate their performance in controlling mobile and arm robots, and compare their attributes to other artificial neural network-based control agents. Finally, we experimentally show their superior resilience to environmental noise, compared to the existing machine learning-based methods.**

I. INTRODUCTION

The *C. elegans* nematode, with a rather simple nervous system composed of 302 neurons and 8000 synapses [1], exhibits remarkable controllability in it's surroundings; it expresses behaviors such as processing complex chemical input stimulations [2], sleeping [3], realizing adaptive behavior [4], [5], performing mechano-sensation [6], and controlling 96 muscles [7]. How does *C. elegans* perform so much with so little? What are the underlying computational principles to gain such high degrees of controllability? and how can we design worm-like artificial intelligent (AI) systems based on these principles to obtain better AI controllers? To answer these questions, we take a computational approach.

It has been recently shown that neural networks constructed by a worm-inspired neuronal and synaptic model, realize liquid time-constant (LTC) recurrent neural networks (RNN)s with universal approximation capabilities [8], [9]. LTC-RNNs can capture complex dynamics with a few number of neurons, due to the existence of sigmoidal nonlinearities on the synapses which enable neurons to express varying time-constants. The open question is how to build such network topologies systematically in robotic control domains while enhancing the interpretability of the system?

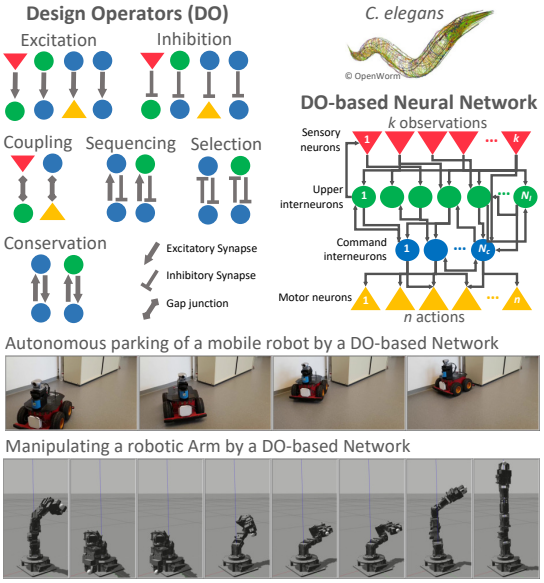

Fig. 1. We design compact, interpretable and noise-robust neural controllers inspired by the relational structures of the worm's brain, to control robots.

In the present study, we propose a worm-inspired network design methodology that utilizes the LTC-RNN model to create interpretable neural controllers in robotic tasks. The method proposes a set of rules to construct hierarchical architectures, from sensory neurons, through an interleaved set of interneurons, to motor neurons, by means of binary relational structures named *design operators* (DO), illustrated in Fig. 1. This design procedure imposes a high degree of sparsity (around 80%), and builds up an attention mechanism through distinct network pathways that enhances interpretability. Along with the recent successes of the search-based learning techniques for neural networks in control environments [10], [11], [12], we adopt a random-search memetic algorithm to parametrize the synaptic weights. The key contributions of this paper are as follows:

1) Introducing novel network-design principles for the liquid time-constant neuronal models, and equipping the designed network with a search-based learning algorithm, to govern robotic tasks.
2) Deploying DO-based networks in experiments with real and simulated robotic environments.
3) Experimental demonstrations of the superiority of the performance of DO-based networks in terms of their compactness, robustness to noise and their interpretable dynamics, compared to common RNNs.

*Equal Contributions
[1]Technische Universität Wien (TU Wien), 1040 Vienna, Austria
[2]Department of Neurobiology, University of Vienna, 1090 Vienna,Austria
[3]Institute of Science and Technology (IST), 3400 Klosterneuburg, Austria
[4]Research Institute of Molecular Pathology (IMP), Vienna Biocenter (VBC), 1030, Vienna, Austria

II. Related Works

Brain-inspired Robotic Control. The way nervous systems of living creatures process information has been extensively used in robotic control as a source of inspiration [13], [14], [15], [16]. In particular, networks of biophysically modelled neurons [17], [18] are deployed in applications such as navigation of mobile robots [15], [19], control of unmanned aerial vehicles (UAV) [20] and legged robots [21], [22], [23]. Obtained networks can be topologically divided into two categories: 1) Networks that are put together by hand in a *piece-by-piece* and *trial-and-error* fashion [21], [23], [15], [20]. These approaches lack fundamental design principles. 2) Networks that deploy fully-connected structures and rely purely on the learning phase to determine functions. Similar to Deep learning models, interpreting dynamics of these networks becomes a challenge [24], [25]. Our networks address both challenges by incorporating a systematic design together with a set of rules that improves interpretability.

Motion Planning. to (optimally) solve the motion-planning problem, various model driven techniques have been proposed, such as rapidly-exploring random trees [26], [27], [28], cell decomposition [27], [29], [30], potential fields [27], [31], [29], [30], satisfiability modulo theories (SMT) [32], [33], [34] and model predictive control (MPC) [35], [36]. These approaches are often human expert labor intensive, to distill task-specific solutions. We aim to ease the effort by introducing a combination of systematic design equipped with machine learning techniques. In Parallel to model-driven control systems, deep reinforcement learning (RL) has achieved significant successes in agent control [37], [38], [39], [10]. In a deep RL setting, parameters of the neural network are tuned by a learning algorithm for the control agent to take actions that maximize the total reward. While they perform as good or surpass the performance of the manually designed agents, their explainability becomes a challenge, which is not desirable in safety-critical applications. Our methodology overcomes this challenge by imposing sparse network connectivity and by using an interpretable neuronal model [9].

III. Preliminaries

Here, we revisit the liquid time-constant (LTC) RNN model, used to design DO-based networks and recap its properties.

A. Neuron and Synapse Model

Neuronal communication is modelled by ordinary differential equations (ODE) describing the biophysical dynamics of a passive cell [9]:

$$\begin{aligned}
\dot{V}_i(t) &= \left[I_{i,L} + \sum_{j=1}^{n} \hat{I}_{i,j}(t) + \sum_{j=1}^{n} I_{i,j}(t)\right] / C_{i,m} \\
I_{i,L}(t) &= \omega_{i,L}\left[E_{i,L} - V_i(t)\right] \\
\hat{I}_{i,j}(t) &= \hat{\omega}_{i,j}\left[V_j(t) - V_i(t)\right] \\
I_{i,j}(t) &= \omega_{i,j}\left[E_{i,j,R} - V_i(t)\right] g_{i,j}(t) \\
g_{i,j}(t) &= 1 / \left[1 + \exp(-\sigma_{i,j}\left(V_j(t) - \mu_{i,j}\right))\right]
\end{aligned} \tag{1}$$

Algorithm 1: Network simulator

Input: Network V, Sensory neurons S, Motor neurons M

$v[0 \dots n] \leftarrow V_{leak}$;
while *True* **do**
  $v[s \in S] \leftarrow$ read_sensor_values();
  $I[0 \dots n] \leftarrow 0$;
  for $e = (pre,post) \in E$ **do**
    $I[\text{post}] \leftarrow I[\text{post}] + (E_{rev}(e) - v[\text{post}])w(e) \cdot \sigma(v[\text{pre}])$;
  end
  for $i \in 0, \dots n$ **do**
    $v[i] \leftarrow$ ODE_update(v, I);
  end
  set_output$(v[m \in M])$;
end

where $V_i(t)$ and $V_j(t)$ are the membrane potential of the post and pre-synaptic neurons, $E_{i,L}$ and $E_{i,j}$ are the reversal potentials of the leakage-channel and chemical-synapse, $I_{i,L}$, $\hat{I}_{i,j}$, and $I_{i,j}$ are the currents flowing through the leakage-channel, electric-synapse, and chemical-synapse, with conductances $\omega_{i,L}$, $\hat{\omega}_{i,j}$, and $\omega_{i,j}$, respectively, $g_{i,j}(t)$ is the dynamic conductance of the chemical-synapse, and $C_{i,m}$ is the neuron's capacitance. $E_{i,j}$ determines the polarity of a synapse either being inhibitory or excitatory. We also utilize the artificial model for sensory and motor neurons introduced in [9], in order to exchange information between networks and their environment. In order to solve the ODE in real-time efficiently, we used a fixed step solver [40]. Our simulator (Algorithm 1) runs with $\mathcal{O}(|\#\text{ neurons}| + |\#\text{ synapses}|)$ time complexity for each time step Δ_t.

B. Universal Approximation capability of the model

It has been recently shown that internal dynamics of a neural network built by the neuronal and synaptic model introduced by Eq. 1, realizes an LTC-RNN [9]. In addition to varying their state-dynamics, LTC-RNNs adapt their time-constant of activity particularly corresponding to their input currents through a sigmoidal nonlinearity. It has been proved that any finite-time horizon of n-dimensional continuous dynamical systems, can get approximated by the internal state and the n-output states of an RNN with units of the form Eq. 1 (see Theorem 1 of [9]). Here, we take advantage of the universal approximation capability of LTC-RNNs, to introduce a systematic way to design interpretable and noise-resilient controllers for robots.

IV. Design Operators

In this section, we characterize a set of relational components with which we construct DO-based networks. The wiring patterns among two neurons, discovered in *C. elegans*, are called binary motifs [41], [42], [43]. Equipping these motifs with the neuronal model of Eq. 1, results in six design operators (DO)s: Excitation, inhibition, coupling, sequencing, conservation, and selection, presented in Fig. 2.

Definition 1: Given the neurons in circuit $G = \{N_1, N_2, \dots N_n\}$, a Design Operator is a relation formed among the activation state of neurons, based on their distinct connectivity structure and their synaptic parametrization.

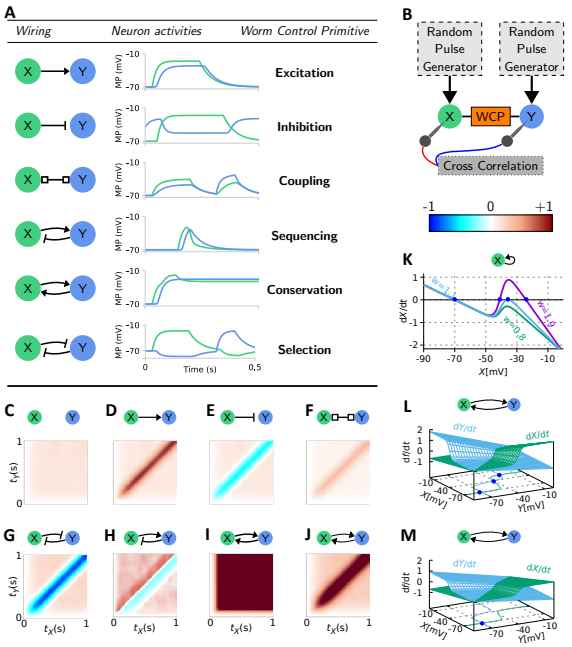

Fig. 2. Design Operators. (A) Time series of the activity of DOs. (B) Sampling process used in the correlation analysis. (C-J) Cross-correlation. (I-J) Cross-correlation of the conservation DO with strong and weak synaptic weights, respectively. (K) Bifurcation analysis of self-conservation (unary) DO (at most three stationary points). (L) Bifurcation analysis for conservation DO with strong synaptic weights. The isolines for $dX(t)/dt = 0$ and $dY(t)/dt = 0$, of the membrane potential of neurons X and Y, plotted on the base plane have three intersections (stationary points). (M) Bifurcation analysis for conservation DO with weak synaptic weights. The isolines plotted on the base plane have only one intersection (stationary point).

DOs are fundamentally different compared to network motifs. Motifs [44], are frequently repeated structural patterns in biological networks, whereas DOs are both structural and relational dependencies of neurons. Motifs' significance is mainly limited due to the lack of information about the synaptic polarities in biological networks [41]. However, We adopted the concept of DOs from the functional dynamics of neurons, captured by investigating calcium imaging of the neuronal activity of the *C. elegans*' brain [45]. More importantly, the general goal of network motifs is to explain the mechanisms underlying behavior of a biological network through the interaction of basic building blocks [44]. A design operator, in contrast, may result in the emergence of many behaviors, for a given structure due to its output dependencies on the alternation of the synaptic parameters.

We quantify this dependence by performing cross correlation and bifurcation analyses. A short simulation for each DO is depicted in Fig. 2A. An excitation/inhibition DO occurs through one excitatory/inhibitory chemical synapse and leads to the stimulation/inhibition of the post-synaptic neuron. A coupling DO occurs through one electrical synapse and establishes the coupling of the activity of the two neurons. A sequencing DO imposes a sequential activation of the neurons. A conservation/selection realizes a synchronizing/antagonizing activity.

The correlation analysis of the DOs given in Fig. 2C-J, was obtained by subjecting the neurons to independent random-pulse generators, as in Fig. 2B, and collecting their outputs. In an excitation/inhibition DO, the neurons are phase-aligned with a positive/negative correlation at the main diagonal, (Fig. 2D, 2E). This means that excitation/inhibition does not introduce delay or memory. In a Coupling DO, the neurons are also phase-aligned Fig. 2F. In a selection DO, the dynamics are antagonistic, and result in a competition for being active, (Fig. 2G). In a sequencing DO, a positive/negative correlation appears above/below the main diagonal, (Fig. 2H). Finally, in a conservation DO, the activity is correlated, independently of phase differences. It thus introduces a memory element (Fig. 2I), which vanishes at low synaptic weights, (Fig. 2J).

To understand the dependencies of a DO to its parameters, let us take the memory effect realized by the conservation DO and perform a bifurcation analysis, to explore the number of fixed points, for different synaptic weights [46]. In Fig. 2K, The bifurcation plot represents how a self-excited neuron, as a special case of conservation, determines the dynamics. For large weights (purple line), the neuron has three fixed-points as follows: Stable (left), meta-stable (middle), and stable (right). The stable fixed-points are able to robustly preserve the membrane potential values, as being intuitively inactive (resting) at around $-70mV$, and active at around $-20mV$. This ability vanishes for a low synaptic weight (green line), as the only fixed-point corresponds to the resting potential. In Fig. 2L-M, we show the same analyses for two neurons. For large synaptic weights, the leftmost and rightmost fixed-points of the isolines plotted on the base plane, robustly preserve the potential. For low synaptic weights, this ability of the DO demolishes. In the next section, we describe how to design neural controllers based on DOs.

V. Design a DO-based Neural Network

In this section, we introduce our methodology for designing DO-based neural networks. For a sequential robot control task with identifiable finite action primitives, a network can be designed to fit the given behavior. Given a robotic environment with p sequential action primitives, we design a DO-based network by the principles described below:

Rule 1: Add p command neurons for p motion primitive.
- If two primitives have direct temporal dependencies, add a Sequencing DO between their underlying command neurons.
- If two primitives function in parallel, add Conversation DO.
- If two action primitives are mutual exclusive, add a Selection DO between their corresponding command neurons.
- If a single action primitive should persist over a certain time, add a Self-Conservation DO.

Rule 2: Identify trigger conditions of primitives; add an Upper interneuron for each of the conditions and connect the interneuron to the command neurons by Excitation DOs.

Rule 3: For each of the Upper interneurons, identify the signals on which its condition triggers. This signals can come from other neural circuit modules or be defined based on the

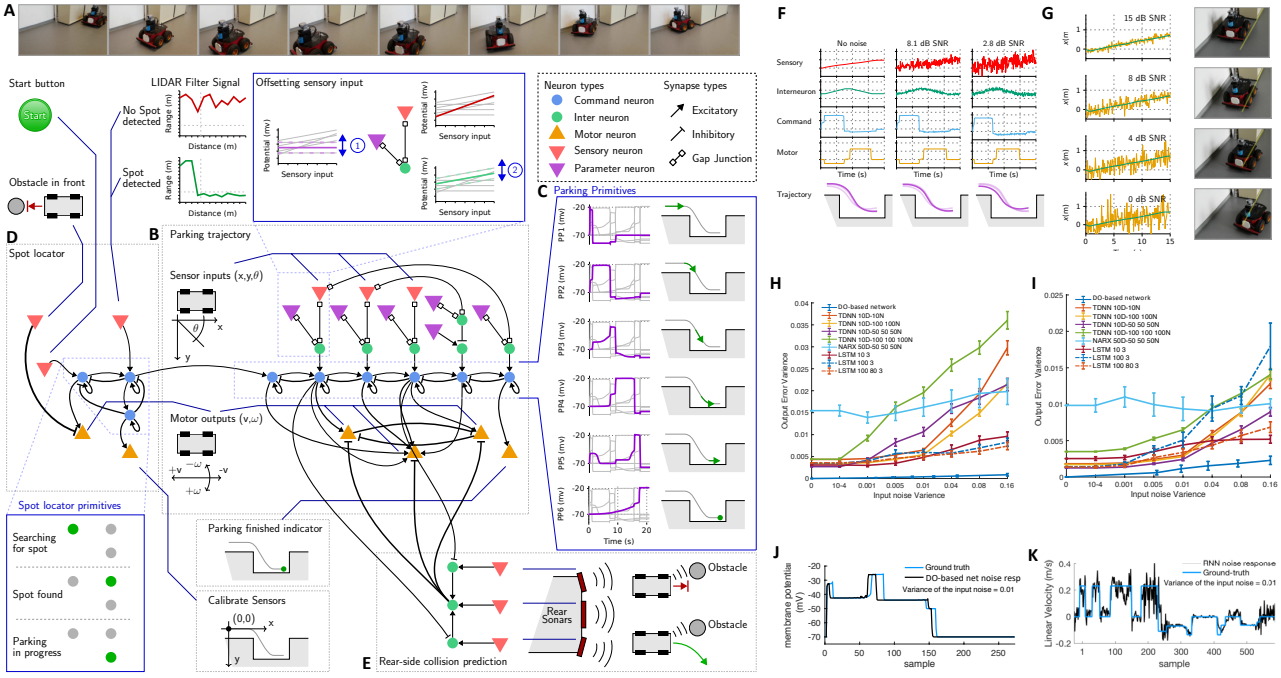

Fig. 3. (B-E) The architecture of the DO-based network designed parking. (A) Rover searches for a spot and performs the parking. (B) Parking-trajectory circuit. (C) Six motion primitives for parking controlled by six command neurons. (D) The spot-locator circuit (E) The rear-side collision-avoidance circuit. (F-K) Input noise resilience. (F) Noise injection analysis. The increasing noise (from left to right) is directly applied to the sensory neurons. (G) The effect of noisy input data on the function of the network. (H) The variance of the linear-velocity output error, while increasing the variance of the input noise. (I) The same analysis for the angular-velocity output. (J) The response of the network in the presence of noisy input. (K) The response of the TDNN 10D−100 100N RNN (10D = 10 delay elements and 100 100N = 2 layer each with 100 neurons), to the same noisy input. TDNN = time-delayed neural network [47]. NARX = nonlinear auto-regressive network with exogenous input [48]. LSTM = long short term memory [49].

characteristics of the environment. Signals link their activity to their downstream interneurons, by Coupling DOs.

Rule 4: k Sensory neurons are deployed for k observation variables. Sensory neurons are coupled with their downstream upper interneurons, by coupling DOs. Their connections are established such that particular pathways from the input to output, direct the attention [50] of the network towards specific actions.

Rule 5: Devote n motor neurons corresponding to n control action. Command neurons positively correlating with the control action, synapse into the motor neurons by Excitation DO and inhibit their negatively correlating downstream motor neurons, by inhibitory DOs.

Described rules enable us to design highly sparse neural networks with auditable internal dynamics. By learning the parameters of these networks, we can guide the architecture to perform and generalize well in the control of robots.

VI. Synaptic Parametrization

To optimize the parameters of a DO-based network, we adopted a Random-search memetic learning algorithm. Recently it has been shown that random search optimization strategies [51], can perform as good as gradient-based approaches, with additional advantages such as skipping the gradient issues [10], [11], [12].

Our learning algorithm uses a population of (synaptic) parameter particles and repeats the following two steps until convergence: Generate a new population by randomly perturbing the current one. Resample the obtained population according to the cost of the particles (Network behavior deviation for this particle, from a desired behavior). Algorithm 2 outlines the working principles of the learning system.

VII. Autonomous Parking of a Mobile Robot

In this section, we design DO-based neural networks to perform an autonomous parking procedure with a Pioneer P3-AT robot [52], by the design rules introduce in Sec. V.

Algorithm 2: Random-Search Memetic Algorithm

Input: A cost function f to minimize, Population size n
Output: Parameter θ such that θ is a minimum of f

$P \leftarrow$ random Population of size n;
$\theta_{best} \leftarrow rand()$;
while *New* θ_{best} *found recently* **do**
  for $i \leftarrow 1$ **to** n **do**
    Local-random-search(f,$P[i]$);
    if $f(P[i]) < f(\theta_{best})$ **then**
      $\theta_{best} \leftarrow P[i]$;
  for $i \leftarrow 1$ **to** n **do**
    $q \leftarrow$ select best parameters from P;
    $Q[i] \leftarrow q + rand()$;
  $P \leftarrow Q$;
return θ_{best};

The task includes 3 control procedures as finding a parking spot, performing a parking trajectory and simultaneously avoiding possible obstacles. For each task, a DO-based network is designed and presented in Fig. 3D, 3B, and 3E, respectively. The core circuit (the parking trajectory), in Fig. 3B, follows Rule 1 to include 6 command neurons for six motion primitives shown in Fig. 3C second column, and configures sequencing DO amongst them. Upper interneurons establish Coupling DO with the Parameter neurons which condition their activation, based on Rule 2 and 3. (See Fig. 3B top side box). Based on Rule 4, sensory neurons only synapse into their downstream pathways on which they impose a high impact. For instance, the angular position (θ) sensor, connects to interneuron pathways that are involved in the control of the robot's turns. 3 motor neurons are set to control right turning, left turning and moving backward, then Rule 5 is applied for their connectivity.

The spot locator neural circuit, shown in Fig. 3D, designed to move the robot forward until a filtered Light Detection and Ranging (LIDAR) signal flags a proper parking location. We pre-processed the LIDAR signal with a non-linear Finite-Impulse-Response (FIR) filter before feeding it into the network [53]. Once the spot is found, the circuit activates the Parking trajectory agent to initiate the parking process. While the parking is in action, a rear-side collision avoiding circuit (Fig. 3E), translates the Sonar sensory inputs to preventive signals to the motor neurons by the inhibition DO. A parking reference trajectory is provided as a set of points $T = \{(x_t, y_t, \theta_t) \mid t \in \{0, \ldots, M\}\}$. Correspondingly, we set a cost (objective) function as:

$$R = \sum_{i \in T} (i(t).x - s(t).x)^2 + (i(t).y - s(t).y)^2 + (i(t).\theta - s(t).\theta)^2, \quad (2)$$

where $s(t)$ is the state (x_t, y_t, θ_t) of the rover at time t. We then call the learning algorithm, to minimize this function with respect to the synaptic weights. A video of the resulting neural network's performance can be viewed at `https://youtu.be/zOnqVaSl9nM`.

VIII. Manipulating a Robotic Arm

In this section, we extend our experimentation to the design of a DO-based network to move a Cyton-Epsilon-300 robotic arm with eight degrees of freedom (seven joints and a gripper) [54] in Fig. 4C, to a certain location, grab an object, move the object to a second location and then release the object (Fig. 4A). Action primitives are divided into 5 tasks schematically explained in Fig. 4A right box, and based on Rule 1, 5 command neurons are wired together to control each action. Upper Interneurons in this setting, adopt a fully connected topology to translate 14 sensory observations to trigger commands for the command interneurons while respecting Rules 2 and 3. Each sensory neuron that corresponds to the angle of individual joints (see the setting in Fig. 4D), forms coupling DO with two downstream interneurons randomly. This sparsity imposes an attention mechanism on the network and makes the interpretation of the decision pathways easier. Command neurons controlling

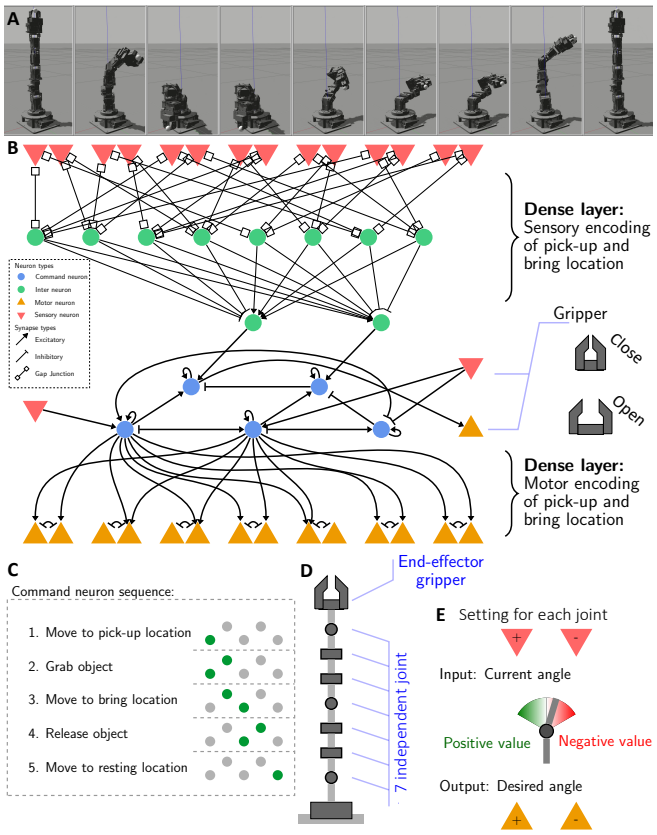

Fig. 4. Manipulating a robotic arm. (A) Grasping and releasing an object at distinct positions. (B) Arm controller neural network (C) Command neurons' sequential activation (D) Illustration of the controlled end-effector composted of 7 joints and one gripper. (E) Indented cooperation of the command neurons.

the joints densely synapse into their 14 downstream motor neurons by Excitation DOs. The designed neural circuit is presented in Fig. 4B. The learning process of the network has been performed hierarchically. First, the sub-circuit including sensory neurons to upper interneurons was trained in a separate environment with the objective to activate when the arm reached a desired position. Then the motor neuron network is optimized in isolation with a supervised setting that with an objective of moving the arm to a desired position if the corresponding command neuron is activated. After stacking up the entire network, we fine-tuned all synaptic weights with Algorithm 2. A video demonstrating the performance of the neural controller on the arm in Gazebo can be viewed at `https://youtu.be/p8D3JTb8qLM`.

IX. Experimental Evaluation

In this section, we point out the distinctions of DO-based networks compared to that of artificial recurrent neural networks and assess their performance. The parking network shown in Fig. 3B comprises 39 neurons together with 49 trainable parameters. Compared to standard artificial neural networks generating similar dynamics, they are significantly smaller. We conducted an experiment to compare the parking performance of standard RNNs in generating the outputs

TABLE I

COMPARING PARKING PERFORMANCE OF DO-BASED NETWORKS IN TERMS OF THEIR NETWORK SIZE WITH STANDARD RNNS. RMSD = ROOT MEAN SQUARED DEVIATION, $\mathbf{v}$ = LINEAR VELOCITY OUTPUT, $\boldsymbol{\omega}$ = ANGULAR VELOCITY OUTPUT AND $\mathbf{E}$ = TERMINATION OUTPUT

RNN Type	Neurons Per-Layer	Params Total	v RMSD	ω RMSD	E RMSD
TDNN-10D	10-3	943	$8.9{\cdot}10^{-3}$	$4.1{\cdot}10^{-3}$	$7.7{\cdot}10^{-3}$
TDNN-10D	100-100-3	19503	$7.0{\cdot}10^{-3}$	$4.2{\cdot}10^{-3}$	$9.4{\cdot}10^{-3}$
TDNN-10D	50-50-50-3	9803	$6.2{\cdot}10^{-3}$	$3.3{\cdot}10^{-3}$	$8.4{\cdot}10^{-3}$
NARX-50D	50-50-50-3	13153	$3.5{\cdot}10^{-3}$	$2.6{\cdot}10^{-3}$	$4.8{\cdot}10^{-3}$
LSTM	10-3	968	$3.5{\cdot}10^{-3}$	$3.5{\cdot}10^{-3}$	$3.5{\cdot}10^{-3}$
LSTM	100-3	45248	$2.0{\cdot}10^{-3}$	$2.0{\cdot}10^{-3}$	$2.0{\cdot}10^{-3}$
LSTM	100-80-3	102928	$1.6{\cdot}10^{-3}$	$1.6{\cdot}10^{-3}$	$1.6{\cdot}10^{-3}$
DO-based	**39**	**49**	**Ground-truth**		

of a DO-based network, given the sensory inputs. Table I, summaries the performance of various RNN topologies.

A. DO-based networks realize complex dynamics with highly sparse and compact network architectures

DO-based nets are 19 times smaller in terms of their trainable parameters than the smallest RNN (time-delayed neural network (TDNN) with 10 delay elements). The reason for the capability of realizing complex dynamics with a fewer number of elements lays in their neuronal and synaptic model (Eq. 1). The model realizes liquid time constant dynamics, meaning that each neuron varies its time constant based on its presynaptic inputs. This is due to the synaptic model's nonlinearity which becomes a rich resource for fitting complex dynamics with a fewer number of neurons.

B. DO-based networks are interpretable

Designing neural networks based on DOs, allows us to establish neuronal pathways with certain functionality, inside the network. For instance, in the parking network, the function of every node is known given their underlying design rules. In a more general case, such as the Arm neural controller, the layer-wise design principles (Rules 1 to 5) increases the level of transparency of the network. In fact, the design principles realize an empirical attention mechanism, to govern interpretable dynamics, where every input to the output pathway, contains interneurons with dedicated actions.

C. DO-based networks are highly resilient to noise

For the parking network, we performed two white Gaussian noise-injection experiments. The first exposes all sensory neurons to increasing internal noise and observes how the robot parks and how the noise propagates through the network to the output (Fig. 3F). The second watches how environmental input noise effects the performance of the network compared to other types of RNNs (Fig. 3G-I),

Fig. 3F-I, show the gradual worsening of the parking behavior. Fig. 3F and 3J, expose a remarkable property of the DO-based networks: The noise is filtered out as it propagates from the sensory-neurons layer to the motor-neurons layer. The performance of network gets defected by a phase shift however it still can perform a decent parking trajectory even at a noise-level as large as the output signal itself. The capacitive nature of neurons in the neuronal model acting as a filter, presumably explains this robustness. A video of the parking performance in the presence of input noise can be viewed at `https://youtu.be/tM9xBQFzBks`.

As illustrated in Fig. 3H and 3I, DO-based Networks considerably outperformed other RNN structures in terms of expressing noise-resilient output dynamics. The figures further demonstrate the sensitivity of the RNNs to noise attacks. While the input noise in DO-based networks causes a slight phase-shift in the output, as shown in Fig. 3J, the noise passes unhindered through all the layers of an RNN, and resulted in distortion of the outputs, as shown in Fig. 3K. Hence, DO-based neural networks distinct their performance in terms of robustness to input noise, compared to other recurrent neural network topologies.

X. CONCLUSIONS AND DISCUSSIONS

We introduced a novel methodology for constructing compact, interpretable and noise-resilient neural networks for controlling robots, inspired by the relational structures (Design Operators) of the nervous system of *C. elegans*. We experimentally illustrated the superiority of the performance of our compact DO-based neural networks in terms of robustness to noise, compared to standard RNNs.

DO-based networks construct a hierarchical network structure from sensory neurons, through an interleaved set of interneurons, to motor neurons. Their wiring structure is realized by a systematic set of rules, at multi-scale network resolutions. Furthermore, the synaptic and neuronal model constructs a sigmoidal nonlinearity on each synaptic link, resulting in the creation of varying time-constants of the network's nodes. This property enables DO-based networks to construct complex dynamics with a few number of elements.

Synaptic parameters of DO-based networks are then learned in a supervised fashion, utilizing a search-based optimization algorithm. The learning process enhances the scalability of the DO-based networks. Application of this type of circuits is broad in fitting finite-time horizon of n-dimensional continuous dynamical systems since their neuronal semantics realize universal approximation capabilities.

Many alternative approaches to the construction of DO-based networks can be taken. Model reduction methods on a densely connected network can be applied to obtain automatically generated neural networks while respecting Rules 1 to 5. DO-based neural networks are bio-physically realistic artificial RNNs. We believe that their working principles presumably results in the development of better and safer AI systems, specifically in safety-critical robotic applications.

ACKNOWLEDGMENT

R.H., M.L. and R.G. are partially supported by the Horizon-2020 ECSEL Project grant No. 783163 (iDev40), and the Austrian Research Promotion Agency (FFG), Project No. 860424. M.Z. was supported by the Wellcome Trust / HHMI, International Research Scholar Award

(#UNS47805) and the Simons Collaboration on the Global Brain (#543069). T.H. was supported in part by the Austrian Science Fund (FWF) under grants S11402-N23 (RiSE/SHiNE) and Z211-N23 (Wittgenstein Award).

REFERENCES

[1] B. L. Chen, D. H. Hall, and D. B. Chklovskii, "Wiring optimization can relate neuronal structure and function," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 12, pp. 4723–4728, 2006.

[2] C. I. Bargmann, "Chemosensation in c. elegans," *WormBook*, pp. 1–29, 2006.

[3] A. L. Nichols, T. Eichler, R. Latham, and M. Zimmer, "A global brain state underlies c. elegans sleep behavior," *Science*, vol. 356, no. 6344, p. eaam6851, 2017.

[4] E. L. Ardiel and C. H. Rankin, "An elegant mind: learning and memory in *Caenorhabditis elegans*," *Learning & memory*, vol. 17, no. 4, pp. 191–201, 2010.

[5] R. Hasani, M. Fuchs, V. Beneder, and R. Grosu, "Non-associative learning representation in the nervous system of the nematode caenorhabditis elegans," *arXiv preprint arXiv:1703.06264*, 2017.

[6] M. Chalfie, J. E. Sulston, J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The neural circuit for touch sensitivity in caenorhabditis elegans," *Journal of Neuroscience*, vol. 5, no. 4, pp. 956–964, 1985.

[7] Q. Wen, M. D. Po, E. Hulme, S. Chen, X. Liu, S. W. Kwok, M. Gershow, A. M. Leifer, V. Butler, C. Fang-Yen, *et al.*, "Proprioceptive coupling within motor neurons drives c. elegans forward locomotion," *Neuron*, vol. 76, no. 4, pp. 750–761, 2012.

[8] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant recurrent neural networks as universal approximators," *arXiv preprint arXiv:1811.00321*, 2018.

[9] R. M. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Re-purposing compact neuronal circuit policies to govern reinforcement learning tasks," *arXiv preprint arXiv:1809.04423*, 2018.

[10] T. Salimans, J. Ho, X. Chen, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017.

[11] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[12] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," *arXiv preprint arXiv:1803.07055*, 2018.

[13] A. Brabazon, M. O'Neill, and S. McGarraghy, *Natural Computing Algorithms*, 1st ed. Springer Publishing Company, Incorporated, 2015.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, pp. 436–444, May 2015.

[15] M. Folgheraiter, G. Gini, A. Nava, and N. Mottola, "A bioinspired neural controller for a mobile robot," in *2006 IEEE International Conference on Robotics and Biomimetics*, Dec 2006, pp. 1646–1651.

[16] M. D. Capuozzo and D. L. Livingston, "A compact evolutionary algorithm for integer spiking neural network robot controllers," in *2011 Proceedings of IEEE Southeastcon*, March 2011, pp. 237–242.

[17] R. Hasani, V. Beneder, M. Fuchs, D. Lung, and R. Grosu, "Sim-ce: An advanced simulink platform for studying the brain of caenorhabditis elegans," *arXiv preprint arXiv:1703.06270*, 2017.

[18] P. Gleeson, D. Lung, R. Grosu, R. Hasani, and S. D. Larson, "c302: a multiscale framework for modelling the nervous system of caenorhabditis elegans," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 373, no. 1758, p. 20170379, 2018.

[19] H. Hagras, A. Pounds-Cornish, M. Colley, V. Callaghan, and G. Clarke, "Evolving spiking neural network controllers for autonomous robots," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 5, April 2004, pp. 4620–4626 Vol.5.

[20] A. Westphal, D. Blustein, and J. Ayers, "A biomimetic neuronal network-based controller for guided helicopter flight," in *Biomimetic and Biohybrid Systems: Second International Conference, Living Machines 2013, London, UK, July 29 – August 2, 2013. Proceedings*, N. F. Lepora, A. Mura, H. G. Krapp, P. F. M. J. Verschure, and T. J. Prescott, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 299–310.

[21] R. D. Beer, H. J. Chiel, R. D. Quinn, K. S. Espenschied, and P. Larsson, "A distributed neural network architecture for hexapod robot locomotion," *Neural Computation*, vol. 4, no. 3, pp. 356–365, 1992. [Online]. Available: http://dx.doi.org/10.1162/neco.1992.4.3.356

[22] N. S. Szczecinski, D. M. Chrzanowski, D. W. Cofer, A. S. Terrasi, D. R. Moore, J. P. Martin, R. E. Ritzmann, and R. D. Quinn, "Introducing mantisbot: Hexapod robot controlled by a high-fidelity, real-time neural simulation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 3875–3881.

[23] N. S. Szczecinski, A. J. Hunt, and R. D. Quinn, "Design process and tools for dynamic neuromechanical models and robot controllers," *Biological Cybernetics*, vol. 111, no. 1, pp. 105–127, Feb 2017.

[24] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, vol. 3, no. 3, p. e10, 2018.

[25] R. Hasani, A. Amini, M. Lechner, F. Naser, D. Rus, and R. Grosu, "Response characterization for auditing cell dynamics in long short-term memory networks," *arXiv preprint arXiv:1809.03864*, 2018.

[26] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[27] L. E. Kavraki and S. M. LaValle, "Motion planning," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 109–131.

[28] F. T. Pokorny, D. Kragic, L. E. Kavraki, and K. Goldberg, "High-dimensional winding-augmented motion planning with 2d topological task projections and persistent homology," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 24–31.

[29] J.-C. Latombe, *Robot Motion Planning*, ser. The Springer International Series in Engineering and Computer Science, 1991.

[30] E. Masehian and D. Sedighizadeh, "Classic and heuristic approaches in robot motion planning-a chronological review," *World Academy of Science, Engineering and Technology*, vol. 23, pp. 101–106, 2007.

[31] S. Stavridis, D. Papageorgiou, and Z. Doulgeri, "Dynamical system based robotic motion generation with obstacle avoidance," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 712–718, 2017.

[32] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach," 2016.

[33] S. Nedunuri, S. Prabhu, M. Moll, S. Chaudhuri, and L. E. Kavraki, "Smt-based synthesis of integrated task and motion plans from plan outlines," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 655–662.

[34] W. N. N. Hung, X. Song, J. Tan, X. Li, J. Zhang, R. Wang, and P. Gao, "Motion planning with satisfiability modulo theories," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 113–118.

[35] V. Cardoso, J. Oliveira, T. Teixeira, C. Badue, F. Mutz, T. Oliveira-Santos, L. Veronese, and A. F. De Souza, "A model-predictive motion planner for the iara autonomous car," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 225–230.

[36] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[37] M. Zhang, X. Geng, J. Bruce, K. Caluwaerts, M. Vespignani, V. SunSpiral, P. Abbeel, and S. Levine, "Deep reinforcement learning for tensegrity robot locomotion," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 634–641.

[38] S. Gu, E. Holly, T. P. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation," *CoRR*, vol. abs/1610.00633, 2016. [Online]. Available: http://arxiv.org/abs/1610.00633

[39] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," *arXiv preprint arXiv:1710.06537*, 2017.

[40] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge University Press, 2007.

[41] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, ser. Chapman & Hall/CRC Mathematical and Computational Biology. Taylor & Francis, 2006. [Online]. Available: https://books.google.at/books?id=pAUdPQlCZ54C

[42] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002. [Online]. Available: http://science.sciencemag.org/content/298/5594/824

[43] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of evolved and designed networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, 2004. [Online]. Available: http://science.sciencemag.org/content/303/5663/1538

[44] U. Alon, "Network motifs: theory and experimental approaches," *Nature Reviews Genetics*, vol. 8, no. 6, pp. 450–461, 2007. [Online]. Available: http://dx.doi.org/10.1038/nrg2102

[45] S. Kato, H. S. Kaplan, T. Schrdel, S. Skora, T. H. Lindsay, E. Yemini, S. Lockery, and M. Zimmer, "Global brain dynamics embed the motor command sequence of *Caenorhabditis elegans*," *Cell*, vol. 163, pp. 656–669, October 2015.

[46] H. Poincar, "L'quilibre d'une masse fluide anime d'un mouvement de rotation," *Acta Mathematica*, vol. 7, pp. 259–380, Sept 1885.

[47] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[48] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[51] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005, vol. 65.

[52] *Datasheet: Pioneer 3-AT robot*, OMRON ADEPT MOBILEROBOTS, LLC., 2011. [Online]. Available: http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx

[53] N. Moshchuk and S.-K. Chen, "Spot locator for autonomous parking," in *ASME 2009 International Mechanical Engineering Congress & Exposition*, 2009.

[54] *Cyton Epsilon 300 Arm Specifications*, Robai Corporation, 2015. [Online]. Available: http://www.robai.com/assets/=Cyton-Epsilon-300-Arm-Specifications_apr2015.pdf