Supplementary Material

1 Videos of the performance of the learned neuronal circuit policies

Table S1: Videos

Description	*URL*
TW circuit controls an inverted pendulum at different stages of the training process	`https://youtu.be/cobEtJVw3A4`
TW circuit controls a mountain car at different stages of the training process	`https://youtu.be/J7vXFsZz7EM`
TW circuit performs the parking task	`https://youtu.be/p0GqKf0V0Ew`
Parking task with four degrees-of-freedom	`https://youtu.be/jVQqKoHopTU`

2 Sensory Neuron and Motor neuron equations

A **sensory component** consists of two neurons S_p, S_n and a measurable dynamic system variable, x. S_p gets activated when x has a positive value, whereas S_n fires when x is negative. Mathematically, the potential of the neurons S_p, and S_n, as a function of x, can be expressed as

$$S_p(x) := \begin{cases} -70mV & \text{if } x \leq 0 \\ -70mV + \frac{50mV}{x_{max}}x & \text{if } 0 < x \leq x_{max} \\ -20mV & \text{if } x > x_{max} \end{cases} \tag{1}$$

$$S_n(x) := \begin{cases} -70mV & \text{if } x \geq 0 \\ -70mV + \frac{50mV}{x_{min}}x & \text{if } 0 > x \geq x_{min} \\ -20mV & \text{if } x < x_{min}. \end{cases} \tag{2}$$

This maps the region $[x_{min}, x_{max}]$ of system variable x, to a membrane potential range of $[-70mV, -20mV]$. Note that the potential range is selected to be close to the biophysics of the nerve cells, where the resting potential is usually set around -70 mV and a neuron can be considered to be active when it has a potential around -20 mV [2].

Similar to sensory neurons, a **motor component** is composed of two neurons M_n, M_p and a controllable motor variable y. Values of y is computed by $y := y_p + y_n$ and

$$y_p(M_p) := \begin{cases} y_{max}, & \text{if } M_p > -20mV \\ \frac{y_{max}(M_p + 70mV)}{50mV}, & \text{if } M_p \in [-70, -20]mV \\ 0, & \text{if } M_p < -70mV \end{cases} \tag{3}$$

$$y_n(M_n) := \begin{cases} y_{min}, & \text{if } M_n > -20mV \\ \frac{y_{min}(M_n + 70mV)}{50mV}, & \text{if } M_n \in [-70, -20]mV \\ 0, & \text{if } M_n < -70mV \end{cases} \tag{4}$$

This maps the neuron potentials M_n and M_p, to the range $[y_{min}, y_{max}]$. FWD and REV motor classes of Fig. 1B (main text), are modeled in this fashion.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

3 Neural Circuit Implementation and Setup

In this section we describe how to integrate the neuron and synapse equations into a computational framework to build up the TW circuit. Due to non-linearity of the sigmoid function in Eq. (7)Main-text, the neuron's differential equation, Eq. (1)Main-text, becomes non-linear. Unfortunately, there are no complete theory of solving problems of this class, explicitly [1]. Thus, for simulating neural networks composed of such dynamic neuron models, we adopted a numerical implicit solver.

When a network structure is dense and full of synaptic pathways, the set of ODEs (neuron potentials), defined in Eq. (1)Main-text, becomes *stiff* [3]. Therefore, in order to overcome stability issues we used an *implicit* derivation approximation method as follows [3]:

$$\frac{dv}{dt} \approx \frac{v(t) - v(t - \Delta_t)}{\Delta_t} \qquad \text{for some small } \Delta_t. \tag{5}$$

In this way, we discretize the time variable and transform the set of ODEs into a set of iterative equations.

For each neuron, Eq. (1)Main-text, exposed to chemical synaptic currents in the form of Eq. (7)Main-text, and gap junction currents in the form of Eq. (8)Main-text, if we apply approximation of the Eq. (5)supplementary and assume $v_{pre}(t) \approx v_{pre}(t - \Delta_t)$, we can show that the membrane potential of that neuron at time t, is computable by:

$$\begin{aligned} v(t) = & \\ & [\frac{C_m}{\Delta_t} v(t - \Delta_t) + G_{Leak} V_{Leak} + \\ & \sum_{i \in Ex} \omega_{ex,i} E_{Rev,ex} + \sum_{i \in Inh} \omega_{inh,i} E_{Rev,inh} + \\ & \sum_{i \in GJ} \omega_{gj,i} v_{pre}(t - \Delta_t)] / \\ & [\frac{C_m}{\Delta_t} + G_{Leak} + \sum_{i \in Ex} \omega_{ex,i} + \\ & \sum_{i \in Inh} \omega_{inh,i} + \sum_{i \in GJ} \omega_{gj,i}] \end{aligned} \tag{6}$$

In Eq. (6)Supplementary, $\omega_{ex,i}$, $\omega_{inh,i}$, $\omega_{gj,i}$, respectively stand for the overall conductance of the excitatory synapse, inhibitory synapse and the gap junction, where $\omega_{ex,i} = g_{ex,i}(v_{pre})$, $\omega_{inh,i} = g_{inh,i}(v_{pre})$, and $\omega_{gj,i} = \hat{\omega}$. Variables together with their boundaries, and constants in Eq. (6)Supplementary, are summarized in the Table S2.

Table S2: Parameters and their bounds of a neural circuit

Parameter	*Value*	*Lower bound*	*Upper bound*
C_m	Variable	1mF	1F
G_{Leak}	Variable	50mS	5S
E_{rev} excitatory	0mV		
E_{rev} inhibitory	-90mV		
V_{Leak}	Variable	-90mV	0mV
μ	-40mV		
σ	Variable	0.05	0.5
ω	Variable	0S	3S
$\hat{\omega}$	Variable	0S	3S

Formally, Eq. (6)Supplementary, was realized in a hybrid fashion which combine both implicit and explicit Euler's method; The overall neuron equation, Eq. (1)Main-text is approximated by the implicit Euler's method while the parts substituted from Eq. (7)Main-text and Eq. (8)Main-text, were estimated by an explicit Euler's method.

The motivation for implementing such hybrid solver, was to make the resulting algorithm of simulating the neuronal network be separable into the following steps:

1. Compute all the incoming currents, $I_{in}^{(i)}$, form all synapses to the cell using the most recent values of $v(t)$
2. Update all $v(t)$ by Eq. (6)Supplementary

This is significantly effective when implementing a neural network on a real-time controller.

4 Neuron traces for simulations included in Figure 5

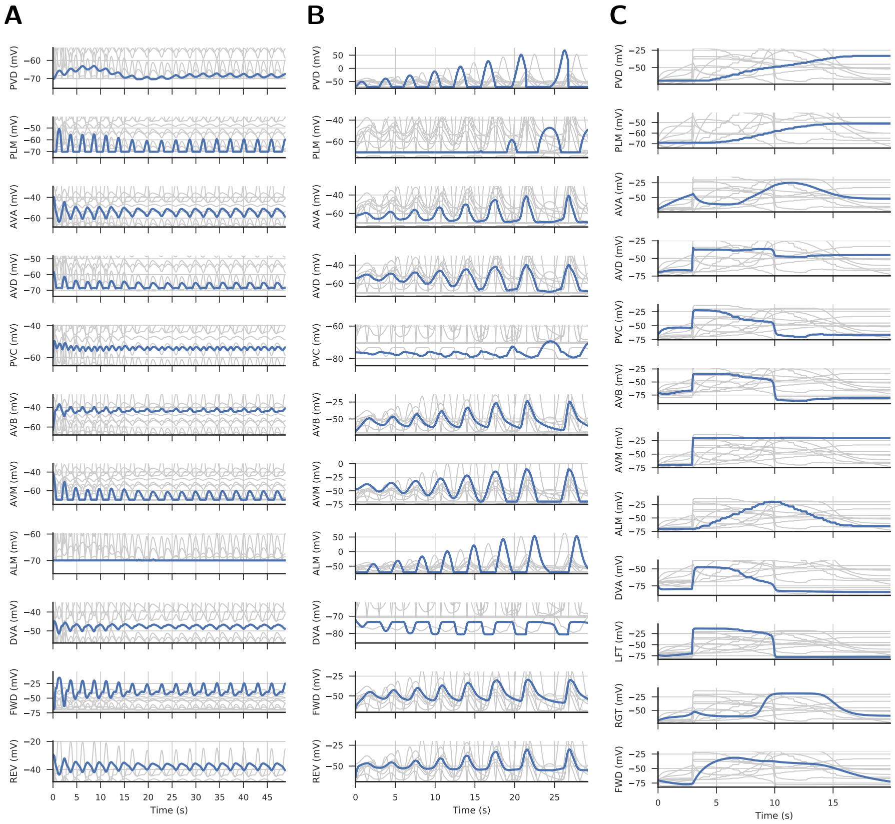

Figure S1: Neuron activity represented in figure 6. A) Inverted pendulum, B) Mountain car C) Parking

5 Experimental Setup Parameters

5.1 Mapping of the Environment to the TW Circuit (Table S3)

Table S3: Mapping the environmental variables to the sensory and motor neurons of the TW circuit, in different experiments

Experiment	*Environment variable*	*Type*	*Positive neuron*	*Negative neuron*
Inverted Pendulum	φ	Sensor (pendulum angle)	PLM	AVM
	x	Sensor (cart position)	ALM	PVD
	a (Control)	Motor (move right/left)	FWD	REV
Mountain Car (OpenAI Gym)	x	Sensor (car position)	PLM	AVM
	$\dot{x}$	Sensor (car's linear velocity)	ALM	PVD
	a (Control)	Motor (move right/left)	FWD	REV
Mountain Car (rllab)	x	Sensor (car position)	PLM	AVM
	$\dot{x}$	Sensor (car's linear velocity)	ALM	PVD
	a (Control)	Motor (move right/left)	FWD	REV
Cart-Pole	φ	Sensor (pole angle)	PLM	AVM
	$\dot{\varphi}$	Sensor (pole angular velocity)	ALM	PVD
	a (Control)	Motor (move right/left)	FWD	REV
Parking of a Rover	x	Sensor (estimated x position)	PVD	
	y	Sensor (estimated y position)	PLM	
	s	Sensor(start signal)	AVM	
	θ	Sensor (estimated angular pose)	ALM	
	a_1 (Control)	Motor (angular velocity)	FWD	REV
	a_2 (Control)	Motor (linear velocity)	FWD/REV	

5.2 Experimental setup (Table S4)

Table S4: Experiment Parameters

	Inverted Pendulum	Mountaincar (OpenAI Gym)	Mountaincar (rllab)	Cart-pole	Parking
Iterations	25,000	50,000	50,000	50,000	20,000
Horizon	1000	1000	500	500	320
Sample size	20	20	20	20	1
Filter size	10	20	20	10	1

With the aim of gaining a better performance, and utilizing parallel hardware resources and to increase the statistical confidence, all experiments are performed by an ensemble of 12 agents. Due to the stochasticity of the training algorithm, not all agents were able to solve the tasks in the given time frame. The observed success-rates are: Mountaincar (OpenAI gym) 25%, Mountaincar (rllab) 42%, Cart-pole 42%, Inverted Pendulum 100% and Parking 100%.

5.3 Neuron's and synapse's parameter-boundaries in the optimization setting (Table S5 and S6)

Table S5: Types of parameters that are optimized and range of valid values

Type	*Lower bound*	*Upper bound*
ω	0	3
σ	0.05	0.5
C_m	0.001	1
G_{Leak}	0.05	5
V_{Leak}	-90	0

5.4 Sensory neuron mappings

As introduced in the main text, input and output values are mapped to the potential of sensory respectively motor neurons by an affine mapping. This affine mapping is defined by the minimum and maximum value of the particular input or output value. For each of the five RL environments we set these boundary values separately, according to the table S6:

Table S6: Input and output boundary values used to define the affine sensory and motor mappings

Environment	*Variable*	*Minimum*	*Maximum*
	x	-1	+1
Inverted pendulum	φ	-0.12	+0.12
	a	-0.3	+0.3
	x	-0.02	+0.02
Mountaincar (OpenAI Gym)	$\dot{x}$	-0.12	+0.12
	a	-1	+1
	x	-0.8	+0.8
Mountaincar (rllab)	$\dot{x}$	-1.5	+1.5
	a	-1	+1
	φ	-0.15	+0.15
Cart-pole	$\dot{\varphi}$	-1	+1
	a	-1	+1
	x		+1
	y		+1
Parking	θ		+1
	Start signal		+1
	Linear velocity		+1
	Angular velocity	-1	+1

6 TW circuit can realize more degrees of freedom

In our first parking experiment, the TW circuit is able to make a turn left and move the rover forward with only one command neuron being active. This means that the circuit is able to solve the task with having binary activation states (active, not active) of the command neuron. To test the flexibility of the TW circuit and underlying neuron model, we set up a second parking experiment. In this experimental setup, we connected the command neuron AVA to two motor neurons responsible for turning right and moving backwards, and AVB to two motor neurons responsible for turning left and moving forward. In this setup, the controller is principally able to move the robot to 4 different directions: Forward, Backward, turn left and turn right. Furthermore, the TW circuit is not able to move the rover forward and turn right with only command neuron being active. If the TW circuit tends to make a right turn and move the rover forward at the same time, (which is necessary to solve this task), the circuit must be able to do this via a synchronized cooperation of the two command neurons. With this configuration, our goal was to test whether the TW circuit can express multiple output primitives with only two command neurons, by operating them in more than two potential states. We conclude that the training algorithm was able to parametrize the TW circuit, such that the agent can keep the trajectory checkpoints, correctly. A video on this scenario can be viewed at `https://youtu.be/p0GqKf0V0Ew`.

7 Flattened time-series plots and correlation detection histograms for all the experiments

Based on Definition 1, in the cross correlation domain, neurons are positively depend on each other if their manifold realizes a positive slope. Similarly, a negative slop in a flattened plot, represent a negative correlation respectively. if there are vertical and horizontal lines in the plot, then the two neuronal dynamics are independent from each other.

In Figures S2, S4 and S6, we plotted the time series of individual neuronal dynamics in respect to each other, for the inverted pendulum, the mountain car and the parking task, respectively. Accordingly, interpretable manifolds of activity are realized, such that we can reason about the dynamics of the neuronal circuit policies learned on each particular task. The colorbar, blue to yellow, represents time in each subplot. It is interesting to note how the dynamics of the TW circuit with the exact same architecture can get adapted to various tasks, by adapting its interneuron dynamics to that exposed from the sensory neurons.

To quantitatively reason about neuronal dynamics, we computed the histogram of all the slops in the flattened data plots, between all points. Based on the Definition 1, we can observe how neuronal dynamics are dependent on each other for a given task. The histograms are shown in Figures S3, S5 and S7.

It is worth noting that there are upper-interneurons (particularly PVC and AVD) which switch their dynamical state for the realization of different sensory to motor neuron pathways.

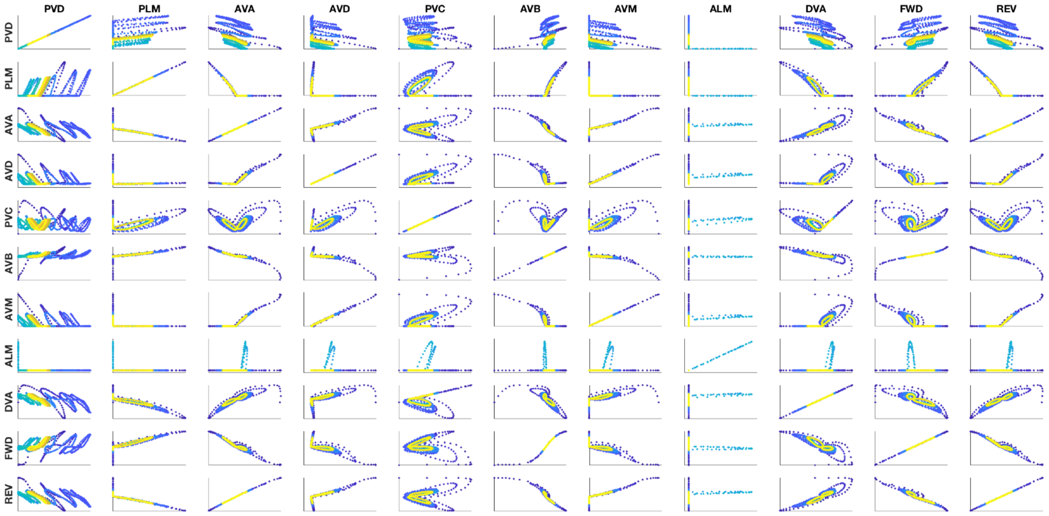

Figure S2: Flattened time-series data of neuron pairs in Inverted pendulum. The colors from dark blue to yellow, represents the evolution of simulation time in each subplot. Neuronal dynamical state is declared by the membrane potential of a neuron throughout the simulation time. Positive slops represent positive correlation, negative slops shows negatively correlated dynamics and a circle would realizes no correlation.

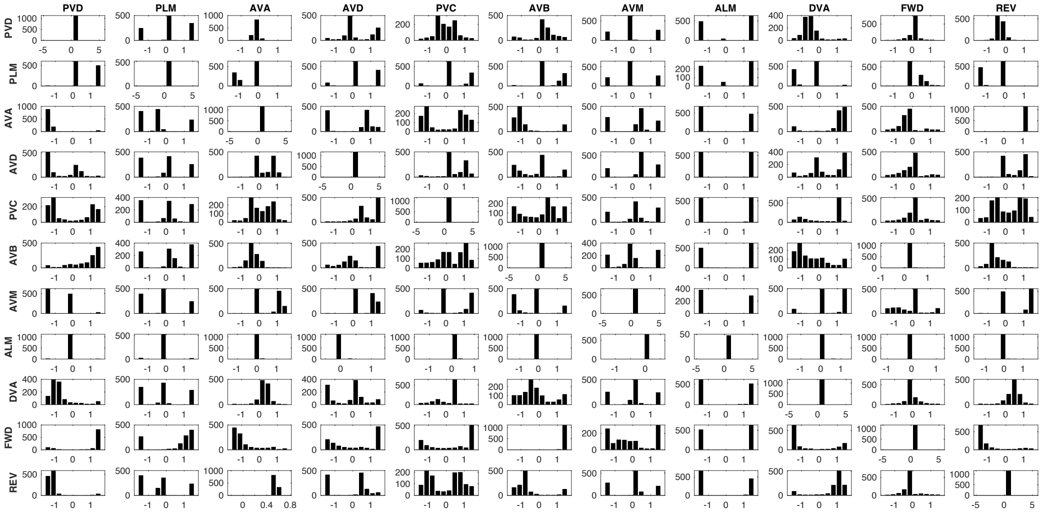

Figure S3: Neural correlation histograms in inverted pendulum. We computed the slops between all pair data-points in the corss-correlation plots and computed their distribution for each neuron pair in order to reason about the correlation of the activity of neurons with each other. Histograms are computed with bin size of 10. Y axis stands for the slops' counts in each bin. X axis shows the $arctan$ of the slop values in radian in a range $[-\pi/2, \pi/2]$. To count neurons as positive contributors to a motor neuron decision, the sum of the counts of the positive radian bins must be greater than half of the counts on the negative side.

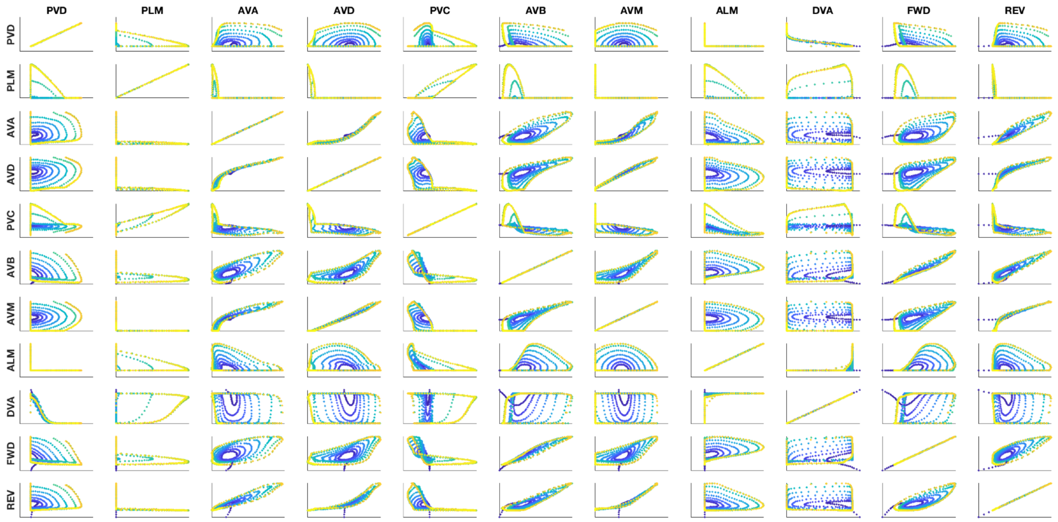

Figure S4: Flattened time-series data of neuron pairs in Mountain car. The colors from dark blue to yellow, represents the evolution of simulation time in each subplot. Neuronal dynamical state is declared by the membrane potential of a neuron throughout the simulation time. Positive slops represent positive correlation, negative slops shows negatively correlated dynamics and a circle would realizes no correlation.

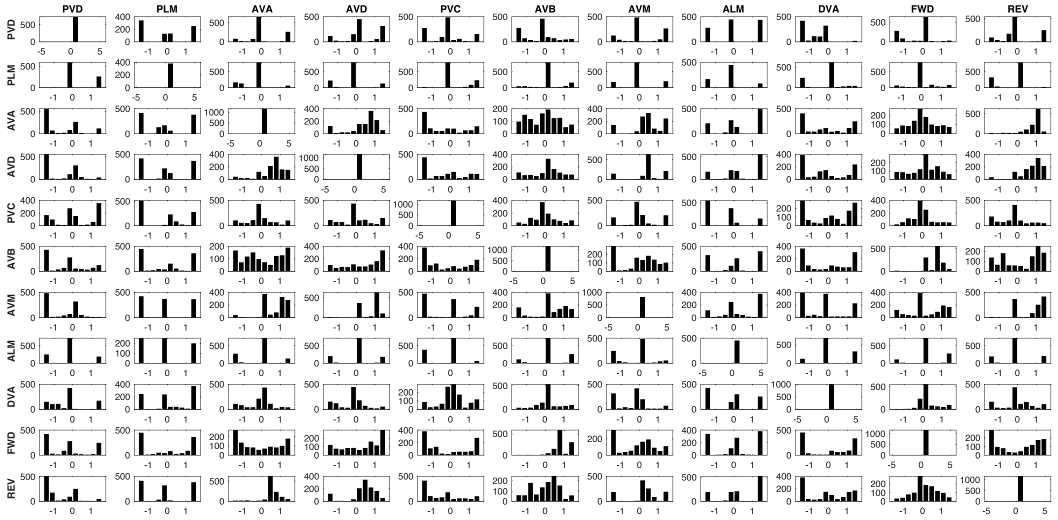

Figure S5: Neural correlation histograms in Mountain car. We computed the slops between all pair data-points in the corss-correlation plots and computed their distribution for each neuron pair in order to reason about the correlation of the activity of neurons with each other. Histograms are computed with bin size of 10. Y axis stands for the slops' counts in each bin. X axis shows the $arctan$ of the slop values in radian in a range $[-\pi/2, \pi/2]$. To count neurons as positive contributors to a motor neuron decision, the sum of the counts of the positive radian bins must be greater than half of the counts on the negative side.

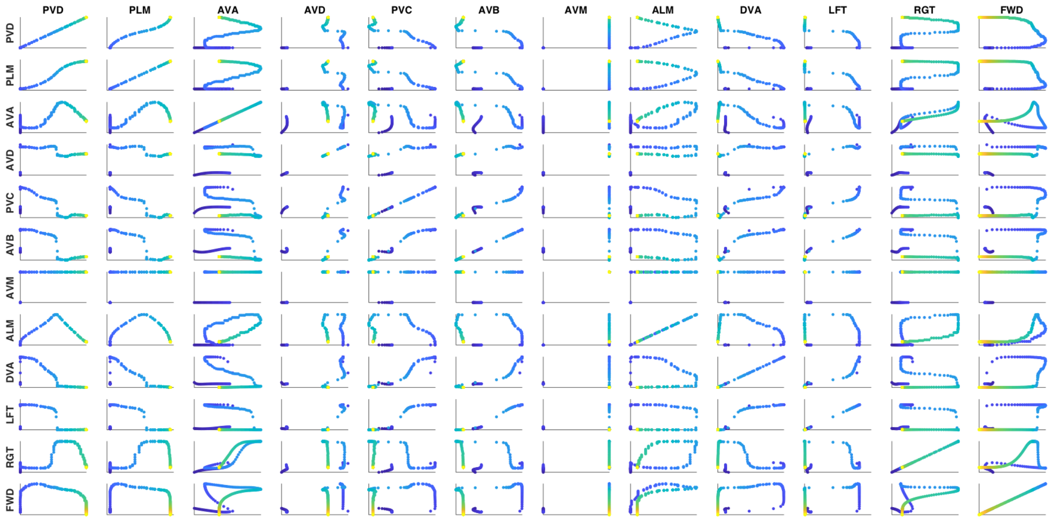

Figure S6: Flattened time-series data of neuron pairs in the parking task. The colors from dark blue to yellow, represents the evolution of simulation time in each subplot. Neuronal dynamical state is declared by the membrane potential of a neuron throughout the simulation time. Positive slops represent positive correlation, negative slops shows negatively correlated dynamics and a circle would realizes no correlation.

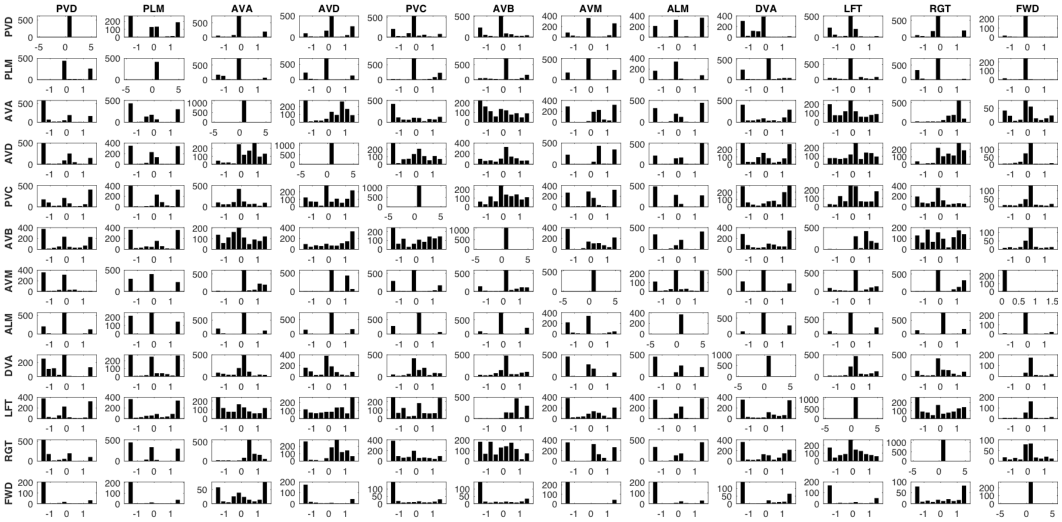

Figure S7: Neural correlation histograms in the parking task. We computed the slops between all pair data-points in the corss-correlation plots and computed their distribution for each neuron pair in order to reason about the correlation of the activity of neurons with each other. Histograms are computed with bin size of 10. Y axis stands for the slops' counts in each bin. X axis shows the $arctan$ of the slop values in radian in a range $[-\pi/2, \pi/2]$. To count neurons as positive contributors to a motor neuron decision, the sum of the counts of the positive radian bins must be greater than half of the counts on the negative side.

Table S7: τ_{min} and τ_{max} for the experiments based on findings of Lemma 1.

	τ_{min} (s)	τ_{max} (s)
Inverted Pendulum		
DVA	9.7e-4	2.5e-3
PVC	0.043	0.71
AVD	0.037	0.11
AVB	4.6e-3	4.6e-3
AVA	2.4e-4	3.9e-4
FWD	3.7e-4	2.4e-3
REV	8.7e-3	8.7e-3
Mountain Car		
DVA	0.5	6.49
PVC	0.11	0.3
AVD	1.32e-4	4.75e-4
AVB	0.05	4.62
AVA	0.014	0.087
FWD	0.069	2.95
REV	0.47	0.47
Parking test		
DVA	0.066	0.17
PVC	0.013	0.37
AVD	1.98e-4	9.65e-4
AVB	2.95	2.95
AVA	0.12	0.56
LFT	2.9e-4	2e-3
RGT	0.61	0.72
FWD	4.045	4.05

8 Time-constant values for the Figure 4 of the main text

References

[1] Teschl Gerald. *Ordinary Differential Equations and Dynamical Systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[2] Ramin M Hasani, Victoria Beneder, Magdalena Fuchs, David Lung, and Radu Grosu. Sim-ce: An advanced simulink platform for studying the brain of caenorhabditis elegans. *arXiv preprint arXiv:1703.06270*, 2017.

[3] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.