International Journal of Neural Systems, Vol. 22, No. 4 (2012) 1250013 (20 pages)

DOI: 10.1142/S012906571250013X

World Scientific
www.worldscientific.com

FROM SENSORS TO SPIKES: EVOLVING RECEPTIVE FIELDS TO ENHANCE SENSORIMOTOR INFORMATION IN A ROBOT-ARM

NICETO R. LUQUE*,‖, JESÚS A. GARRIDO†,‖,
JARNO RALLI‡, JUANLU J. LAREDO§ and EDUARDO ROS¶
Department of Computer Architecture and Technology, CITIC
University of Granada, Periodista Daniel Saucedo s/n
Granada, Spain
**nluque@atc.ugr.es*
†*jgarrido@atc.ugr.es*
‡*jarnor@atc.ugr.es*
§*juanlu@geneura.ugr.es*
¶*eros@atc.ugr.es*

In biological systems, instead of actual encoders at different joints, proprioception signals are acquired through distributed receptive fields. In robotics, a single and accurate sensor output per link (encoder) is commonly used to track the position and the velocity. Interfacing bio-inspired control systems with spiking neural networks emulating the cerebellum with conventional robots is not a straight forward task. Therefore, it is necessary to adapt this one-dimensional measure (encoder output) into a multi-dimensional space (inputs for a spiking neural network) to connect, for instance, the spiking cerebellar architecture; i.e. a translation from an analog space into a distributed population coding in terms of spikes. This paper analyzes how evolved receptive fields (optimized towards information transmission) can efficiently generate a sensorimotor representation that facilitates its discrimination from other "sensorimotor states". This can be seen as an abstraction of the Cuneate Nucleus (CN) functionality in a robot-arm scenario. We model the CN as a spiking neuron population coding in time according to the response of mechanoreceptors during a multi-joint movement in a robot joint space. An encoding scheme that takes into account the relative spiking time of the signals propagating from peripheral nerve fibers to second-order somatosensory neurons is proposed. Due to the enormous number of possible encodings, we have applied an evolutionary algorithm to evolve the sensory receptive field representation from random to optimized encoding. Following the nature-inspired analogy, evolved configurations have shown to outperform simple hand-tuned configurations and other homogenized configurations based on the solution provided by the optimization engine (evolutionary algorithm). We have used artificial evolutionary engines as the optimization tool to circumvent nonlinearity responses in receptive fields.

Keywords: Receptive field; evolutionary algorithm; parallelism; population coding; cuneate nucleus; spiking neural network; robot.

1. Introduction

There is an active interdisciplinary field called Neurobotics in which actual robots are controlled by bio-inspired neural processing engines. Besides other potential applications, this kind of set ups are important for understanding neurobiological computational principles (system neuroscience), specifically, some issues under study are how sensorimotor representations are integrated and efficiently used in accurate manipulation tasks,[1–3] how spike timing based on different sensory representations can help to enhance information transmission[4,5] and be efficiently used by biologically plausible neural systems[6–12] (such as cerebellar-like structures).[13]

‖These authors contributed equally to this work.

It is well known that the cerebellum constitutes a fundamental part in motor systems.[13–17] The cerebellum is fed by inputs from the cerebellar cortex, providing a contribution in *fast and precise movements*.[18] This is crucial in the fine control of the temporal evolution of fast ballistic movements,[19] that is, extremely fast movements that are impossible to be modified by feedback circuits because the complete movement muscle sequence control has to be planned in advance.[20,21]

Furthermore, taking a look at the current research in robot labs, a new trend in constructing and controlling light-weight compliant robot arms[22–24] which mimic human arms can be seen. Such new robotic features pursue the search of new ways of control. In fact, controlling the dynamics of any of these kinds of robot arms is an open issue (there is no general established methodology developed yet).[25] Since the cerebellum combines sensory information with the physical current state to generate motor signals, it is a proper candidate for studying how these controlling problems are solved by nature. In that sense, cerebellum architecture, as a control scheme, has received much attention in the literature and different cerebellum computational models have been developed. Among others, the Cerebellar Model Articulation Controller (CMAC),[26] the Adjustable Pattern Generator (APG),[27] the Schweighofer-Arbib Model,[28] or the Multiple Paired Forward-Inverse Model[29,30] represent good state-of-the-art examples. All these models have something in common: they try to mimic the functionality of the cerebellum by making an abstraction of the cerebellum structure while keeping robotic control theory in mind. As a result, the approximations mentioned above configure their sensorimotor inputs to enhance their control aims.

The cerebellum *supervises and supplies corrective adjustments in motor commands*[31,32] which are generated in other encephalon zones. It receives continuous information from peripheral body parts (position, movement rhythm, interacting external forces, etc.) and, according to *sensorial information*, compares the physical state of each body part against the desired state which the motor system is trying to achieve.[33–35] In the framework of errors (detected through this continuous comparison), proper corrective signals are transmitted to the motor system increasing or decreasing specific muscle activity.[36] The cerebellar *sensorial input* is carried by mossy fibers[37] (MFs), which constitute one of the major cerebellar afferent systems.[38] MFs carry information from different sources; MFs from the pontine nuclei report on motor and sensory areas of the cerebellar cortex,[39–43] MFs from cells in the CN handle information from forelimb muscle spindles[44] related to position and movement,[45,46] MFs from collaterals of cortical fibers carry a copy of descending motor commands to the cerebellum, and finally, MFs from the visual cortex supply information about movements in the visual space.[47] Therefore, different kinds of MFs drive detailed information related to the external world and the desired/actual body movements/positions. As a result of that, it can be postulated that sensorial *cerebellum inputs play a critical role in cerebellum functionality.*

Our system uses neural population coding[48] for sensorimotor representation. Each neuron presents a distribution of responses over some set of inputs, and the responses of many neurons are combined to determine some information about the input state.[48,49] Using this kind of coding, each input stimulus is represented by a set of spikes. However, the occurrence of these spikes strongly depends on the current generated by sensors and on which spikes subsequently reach the first-layer cells.

In a reaching movement, the arm direction is encoded by means of neurons whose input current changes with the cosine of the difference between the stimulus angle and the cell's preferred direction[50] (Cosine tuning). Each population vector cell contributes a vector in the direction of its preferred direction in relation to its current. Nevertheless, a simple reaching movement involves extracting spatial information including visual acquisition of the target, coordination of multi-modal proprioceptive signals, and a proper motor command generation to drive proper motor response towards the target.[51] Usual reaching movements towards a target that we have already seen involve an internal representation of the target and limb positions, and also a coordinate transformation between different internal reference frames. A spiking population coding seems to be the best way to encode sensorial information to be consistent with biological control requirements.[52,53] This is also important to allow system level studies for the evaluation of the cerebellum

functionality in the framework of accurate movement experiments.[1,2]

However, the integration of computational models with neurophysiologic observations in order to understand the main problems in motor control requires not only the cerebellum functionality to be considered (as is done in the CMAC, APG, and other approaches) but also, its biological architecture (cell-network) has to be taken into account. A necessary translation from analog domain sensor signals into spike-based patterns compatible with a spiking cerebellar network needs to be developed.

This paper tries to reveal the best way in which sensorimotor information in a common robot scenario can be handled to investigate an optimal encoding in terms of somatosensory information.

To that aim, the followed methodology can be briefly described at the following points:

(i) First, we consider the execution of a biologically relevant reaching movement in a robot arm scenario. With that purpose, different trajectories are defined over a joint space. A biologically plausible translation from joint position/velocity measures to their corresponding spike train representation has been defined. Population coding and tuning curves are used to be consistent with reaching arm movements.
(ii) In order to make previous codifications more accurate, a bio-inspired Evolutionary Algorithm (EA) which optimizes the receptive fields towards maximizing sensorimotor information and state discrimination has been used. The combinatorial space to be explored towards an optimization is enormous; the EA seems to be the proper tool that unifies biology evolution and an optimization procedure.[54]
(iii) Finally, the performance of the sensorial representations using different measures that take into account metrical properties of the spike train space has been evaluated.

Therefore, the paper uses a new methodology in which an EA is used as an optimization engine towards reaching an efficient sensory representation to be further processed at the spiking-based cerebellum. As a result, in this case, an EA is used for reverse engineering an abstraction of a biologically plausible model. Rather than finding an optimal fitness value of the cost function, the goal is to arrive at an efficient solution in terms of receptive fields in the sensory space. The cost function includes actual trajectories, which makes the experimental set up heavier but also more informative, enhancing the usefulness of the searching methodology carried on by the EA.

2. Materials and Methods

This section describes the principles of the proposed methodology. An answer to different issues explicitly indicating what/why/how and the basis of the proposed approach is given.

2.1. *Target reaching trajectories*

Anthropomorphic robotic arms, mimicking human arms, usually consist of three links (arm, forearm, and hand) which are connected with each other using motorized joints (shoulder, elbow, and wrist). Reaching involves bringing the endpoint of the robot arm to a desired target position. Therefore, the aim is to "connect" two points, the initial point, defined by the actual endpoint robot arm position and the final point, defined by the endpoint robot arm target position. The control system leads the sequence of motor actions to achieve the target. In a robotic arm (due to the redundancy in the degrees of freedom), there is an infinite number of possible trajectories that allow the arm to reach any given target point. A specific approach will be provided by a planner module. But even focusing on a single joint workspace (see Fig. 1) (shoulder, elbow, wrist), the movement can be performed in different ways, in smoother or abrupt movements (as indicated in Fig. 1).

Taking into account both the ability of humans to generalize motor learning skills with a changeable duration/amplitude in a common workspace and the possibility of reaching a target point in infinite different ways, we designed a set of different trajectories (Fig. 1) that allows us to properly explore the workspace in a very simplified scenario. An optimal evolution of receptive fields in this workspace may provide us a generalized solution, i.e. a solution for this kind of movements.

These trajectories are realistic both in terms of robotics (cubic polynomials, linear segments with parabolic blends[55]) and biological plausibility (smooth trajectories with a bell-shaped velocity

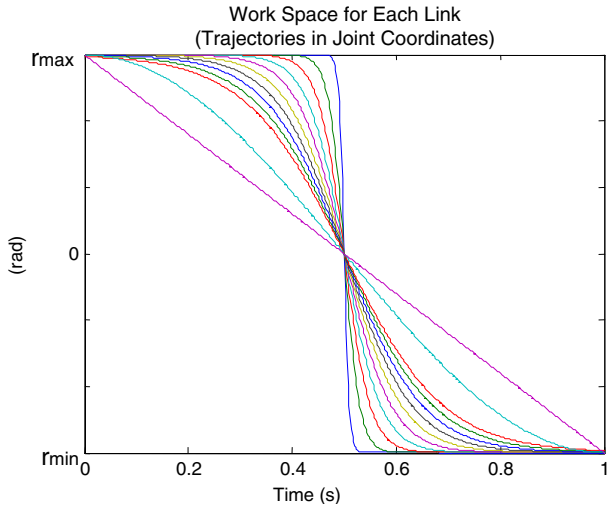


Fig. 1. Trajectory benchmark. Within a single joint workspace (as shown in this plot) the actual movement can be done also in different ways, through different position/velocities profiles. $r_{\max}$ and $r_{\min}$ represent the maximum and the minimum values of the joint angle.

profile (with different smooth profiles and different trajectory ranges)).[56,57]

2.2. *From analog signals to spike patterns: Receptive fields*

When interacting with the real world, a representation of the external environment and the internal state of our body is supplied by the somatosensory system to the central nervous system. The afferent (sensory) information signals are propagated from peripheral nerve fibers to the central nervous system (spinal cord and brain).[58] Each region of the skin is related to an individual cutaneous sensory nerve fiber (or a population of them). These skin regions are called receptive fields. Therefore, each nerve fiber has its associated receptive field, which overlaps with other receptive fields from other fibers. This overlapping is not fixed; the average overlapping degree between receptive fields is related to its body-location.[58]

When a target reaching movement is executed, different body-parts, as muscles, tendons, and joints are articulated depending on their body-location along the followed trajectory. Sensory receptors (proprioceptors) are activated according to movement; thus, a time-varying set of stimuli is produced, and its corresponding neural population varying activity is generated. In contrast, in a robot scenario, the only available sensory information is the one supplied by a single encoder for each link. That involves a translation between the joint position/velocity measures to a time-varying set of stimuli. This is illustrated in Fig. 2. At this point, to find out an optimal biologically plausible encoding scheme that allows "biological decoders" to take advantage of the codification is a nontrivial point. It is assumed that the firing rate of an individual sensory receptor follows a neural response which is characterized by Eq. (1) (also equivalent to a cosine tuning curve, that is, neurons' firing rate varies as the angle between a sensory receptors' preferred direction or angle varies).[59] Therefore, a reaching movement execution will be represented with a sparse population of active cells which are changing with time. This coding mechanism facilitates the representation of the current sensorial state during the trajectory execution in an unambiguous way.

The output of each receptive field (RF) in Fig. 2 is given by Eq. (1):

$$I_{Ri}(t) = r_{\min} + r_{\max} \sum_{n} \mathrm{e}^{-(\theta - \theta_{\mathrm{pref}_i} - 2\pi n)^2 / 2\sigma_i^2}, \tag{1}$$

where $[r_{\min}, r_{\max}]$ is the joint range in radians, θ is the actual position, θ_{pref} is the RF preferred direction which in this work is simplified by the RF centroid (RF responses maximally during a trajectory execution near this centroid, and its response decreases when the trajectory execution increasingly differs from the preferred direction or RF centroid in this case), σ is the width of the RF, i is the identifier of each RF (each one is linked to its corresponding mossy fiber), $2\pi n$ is a subtractive term used to refer the actual position to the first-360° (the maximal range of any revolute joint is ideally 360°), and finally, I_{Ri} is the input current from the corresponding RF_i.

RFs are distributed along the range of each joint (Fig. 2) and they have certain overlap (as in the case of peripheral nerve receptive fields).

Each value of a proprioceptor output signal is integrated using a leaky integrate-and-fire neuron model shown in Eq. (2), which determines the output activity that drives the cuneate nucleus activity in the same way as the mossy fiber activity from cells in the CN handles information from forelimb muscle

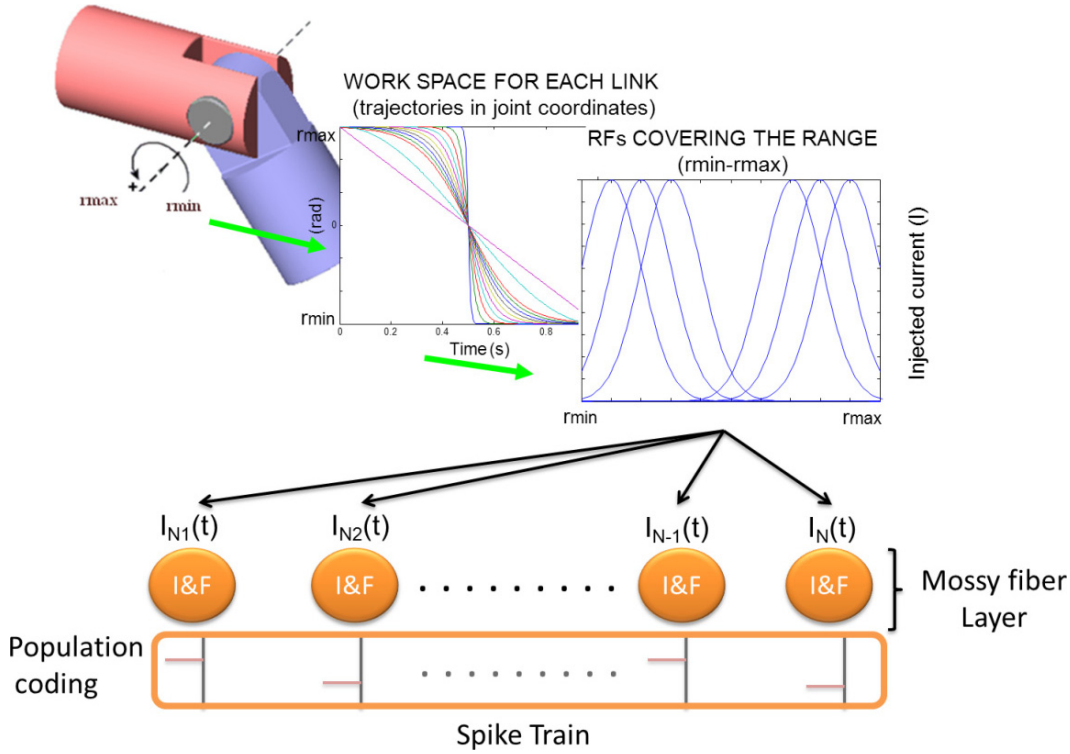


Fig. 2. Population coding of receptor (proprioceptors) signals. The position of a revolute joint given by an encoder along a trajectory is translated into a population coding by means of a set of tuning curves which represent the current injected to Integrate & Fire (I&F) neurons by different sensory receptors (propioceptors). Tuning proprioceptor curves are overlapped mimicking peripheral nerve receptive fields in the human arm. Each value of a proprioceptor output signal (I current) is integrated using an I&F neuron whose output spikes represent the activity provided by the mossy fibers. At the end, in each step time, a spike train is obtained from the mossy fibers that represent the sensory inputs to the cerebellum module.

spindles.

$$\tau_{mi}\frac{dv_i}{dt} = -v_i(t) + R_i I_{Ri}. \quad (2)$$

Related to the integrated and fire cell dynamics,[60] τ_{mi} is the resting time constant, v_i, the membrane potential, I_{Ri}, the input current from the corresponding receptive field, and R_i is related to the resting conductance of the membrane. Finally, the i sub-index term defines the identifier of the related mossy fiber. Therefore, the mossy fiber layer will consist of a group of leaky I&F neurons connected to their corresponding target granule cells.

At this point, the problem in this population coding scheme is not only how to distribute propioceptors (centroid and width) along the workspace, but also how many RFs should be used in order to enhance the information transfer between sensor signals and their spike representation.

2.3. *The evolutionary algorithm as optimization engine*

The distribution of peripheral nerve receptive fields in a human arm is the result of a continuous test and trial process of biological evolution through millions of years. Taking a look around our surrounding environment, there are many examples of well-adapted organisms (in fact, as many as living forms), pointing out that evolution is a universal solver which overcomes difficulties presented by nature. Hence, evolutionary algorithms[61] seem to be a proper tool to optimize the receptive fields of our cerebellum architecture according to artificial evolution and keeping the analogy (though at a very high abstraction level) with the way in which nature solved the biological problem.

Strictly speaking, evolutionary algorithms are a set of bio-inspired techniques for optimization based on the Darwinian process of natural selection. As

in the evolution of species, those individuals (solutions) showing to be the fittest ones are preferentially selected for mating, so that their offspring will inherit their genes through the course of generations. Iteratively, selection acts as a filter for genes and just those belonging to the best solutions are able to overcome the selection pressure and recombine forming higher order solutions. It is within that process where the stochastic-based search of an evolutionary algorithm has been shown to succeed in many optimization problems.[62,63] A genetic algorithm (i.e. a sub-class of an evolutionary algorithm) is used to obtain, through evolution, a near-optimal peripheral nerve receptive field distribution.

In Table 1, the pseudo-code of an evolutionary algorithm where a population of plausible solutions (P) is iteratively improved from random is shown. This evolvable population (P) consists of Individuals (Ind) as indicated in Eq. (3).

$$P = \{Ind_1, Ind_2, \ldots, Ind_j\}, \quad \text{where} \quad j = 1, \ldots, Ind_{\max}, \tag{3}$$

where the candidate solutions (Individuals) are encoded by Eq. (4).

$$Ind = \{RF_{\max}, \{e_1, e_2, \ldots, e_i\}\}, \quad \text{where} \quad i = 1, \ldots, RF_{\max}. \tag{4}$$

Finally, e is a receptive field defined in Eq. (5).

$$e = (\theta, \sigma), \tag{5}$$

where θ represents the centroid of the receptive field along the sensory space (preferred coordinate of receptive field e) and σ, the width of the receptive field (Fig. 3). Therefore, according to Fig. 2, each candidate solution presents a spike train response in time when a trajectory is executed. A set of executed trajectories produce a set of spike train responses. R denotes the set of spiking responses of a possible candidate solution along the followed trajectories as expressed in Eq. (6).

The heuristic-based search consists of projecting each individual encoding (Ind) into the problem

Table 1. Pseudo code of a Generational EA.

```
/* The initial population is a random sampling
of the search landscape*/
P <= Randomly generated initial population
Fitness(P)
    /*For a number of predefined generations*/
Repeat until termination
    /*Every generation, we create a new population
(P_aux) of evolved individuals*/
  Repeat P times
    Ind1 Ind2 <= Select 2 of the fittest individuals
      in P
    New_Ind1 New_Ind2 <= Crossover(Ind1,Ind2)
    New_Ind1 <= Mutate(New_Ind1)
    New_Ind2 <= Mutate(New_Ind2)
    P_aux.add(New_Ind1,New_Ind2)
  End Repeat
    /* Evaluate individuals in population P_aux */
  Fitness(P_aux)
  /* To keep elitism, we replace the worst individual
in P_aux with the best individual in P */
  P_aux(individual_worst) <= P(individual_best)
  P <= P_aux
End Repeat
```

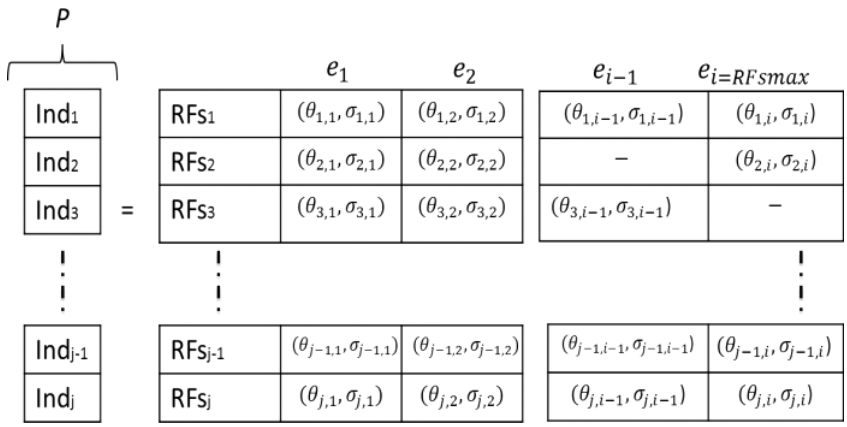


Fig. 3. Visual interpretation of the Population (P) to be evolved. Each individual consists of a vector containing a variable number of RFs defined by their own preferred coordinate θ_{pref} and the width of the receptive field-associated.

space (i.e. using the *Fitness* function). Then, the fittest individuals are selected for recombination. As in the case of natural reproduction, *crossover* applies to a couple of individuals (Ind_1 and Ind_2), merging their encodings to produce descendants (New_{Ind_1} and New_{Ind_2}). Furthermore, there are also small *mutations* in descendants in order to escape from local-optima attracting regions. Within this process, the best solution in every iteration (known as *epoch*) is preserved for the next one.

2.4. *Van Rossum distance-based fitness function. Metrics for evaluating information transfer*

At this point, it is necessary to define a metric (goal function) to measure whether a solution A is better than a solution B. This is intimately related to the functionality assumed by the system. To that end, a metrical information transfer measure which is employed in order to assess the fitting of an evolved peripheral nerve receptive field distribution has been chosen. We have found no previous approach related to information transfer (Shannon's mutual information, for instance) that takes into account the whole metrics of the spike response space.[64] In order to determine the quantity of information transmission carried on by a large population of spikes (taking into consideration their metrical properties), a new entropy definition based on Ref. 65 is used. That facilitates the comparison between different spike populations generated by different receptive field distributions, as shown by Eq. (6).

$$H^*(R) = -\sum_{r\in R}\frac{1}{card(R)}\log\left(\sum_{r'\in R}\frac{1}{card(R)}\alpha(r,r')\right), \tag{6}$$

where R is the set of spiking responses of a possible configuration of receptive fields along the followed trajectories, $card(R)$ is the cardinal number of R, and finally, α is a similarity real function between the responses (r, r'). We use as a similarity function between two spike trains the van Rossum distance[66] D_{vr} defined in Eq. (8). The van Rossum-based Real function $\alpha(r, r')$ will take values in the interval $[0, 1]$ as a response to r, r' stimuli, as shown in Eq. (7).

$$\begin{aligned}&\alpha(r,r') = 1 \leftrightarrow r = r' \quad \text{otherwise}\\ &\alpha(r,r') = D_{\text{vr}}^{-1}.\end{aligned} \tag{7}$$

Equation (6) means that the quantity of entropy in a system is proportional to the logarithm of possible different microstates presented by this system. Maximizing the entropy involves maximizing the quantity of possible microstates in the system. Keeping in mind this concept and looking backwards to our previously defined receptive field system, some relevant points can be clarified:

(a) Each set of evolvable receptive fields produces a set of spiking stimuli for the previously described trajectory benchmark.
(b) A population coding that represents the different sensorial states in an unambiguous way when each of the trajectories belonging to the benchmark is executed is desirable (Fig. 4).
(c) Maximizing the number of possible population coding microstates improves the representation of different sensorial states. Each microstate might be unambiguously represented in just one single way.
(d) In order to differentiate a couple of spike train sets, van Rossum distance is used. If two sets of spike trains are equal, the entropy is zero, the more the difference between both sets, the higher the entropy will be, i.e. the number of microstates representing different sensorial states increases in proportion with the entropy. According to Eq. (6), entropy depends on α, and α depends on van Rossum distance as well. Therefore, an optimal representation will be ensured only if we evolve receptive fields to maximize the minimal distance between any of two single spike-based states of the whole generated set of spike trains for any benchmark trajectory.

2.4.1. *Similarity function*

As it was previously indicated, we have chosen a similarity function based on the van Rossum distance.[66] This function is related to the distance introduced by Victor and Purpura,[67,68] but is computationally more efficient, Eq. (8), and has a more natural physiological interpretation.

$$D^2_{\text{vr}}(r,r')_{tc} = \frac{1}{t_c}\int_0^{\infty}[r(t) - r'(t)]^2 dt, \tag{8}$$

where our spike train (r) is defined by a set of first spikes generated along a certain time window by the implemented spiking neural network. It is

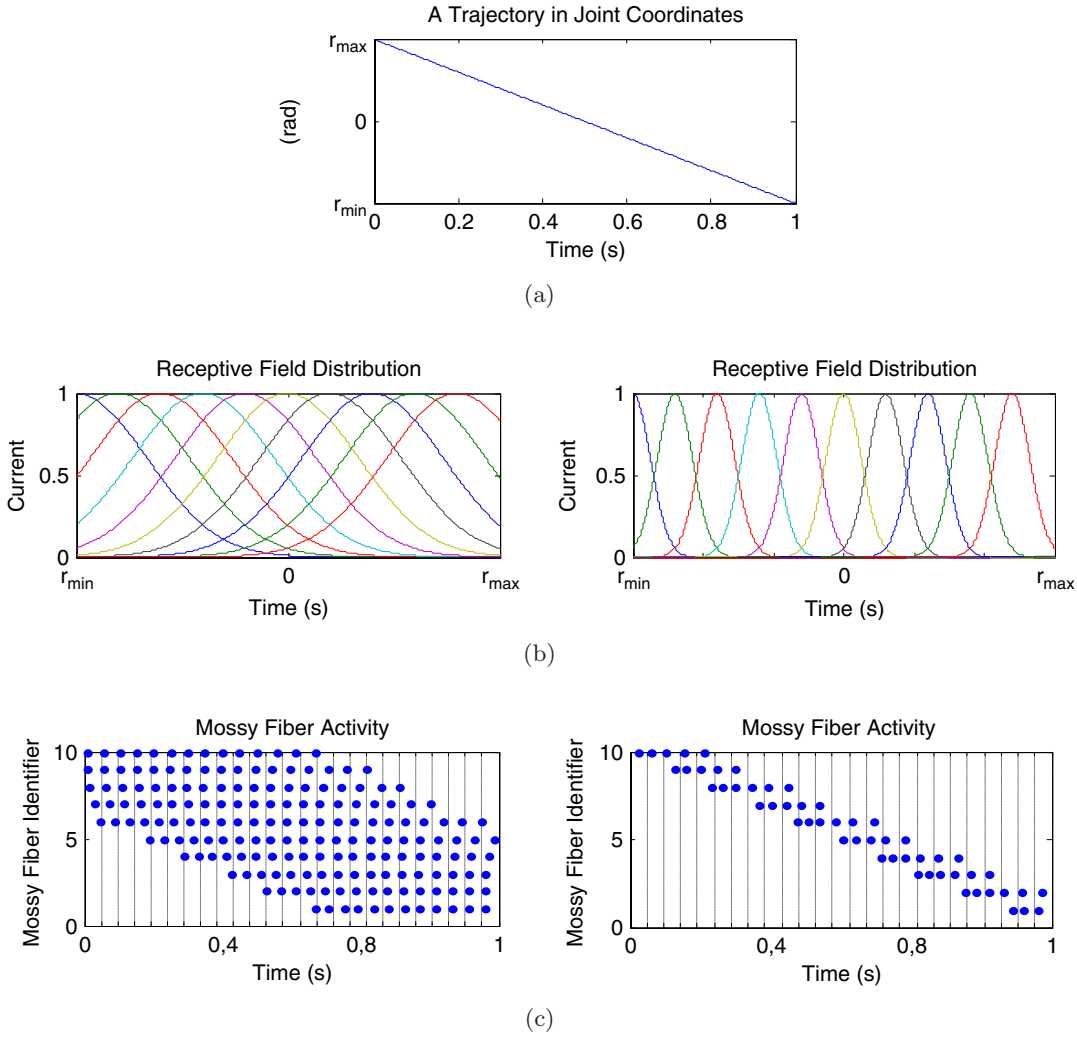


Fig. 4. Different mossy activities corresponding to two different receptive fields when a rectilinear trajectory is followed. (a) Trajectory which is followed by a link of a robot-arm. (b) Two configurations of receptive fields mapping the analog joint coordinate. (c) Two spike populations (population coding) representing each sensorial state (vertical columns) along the executed trajectory. Each sensorial state highly depends on the input receptive field distribution.

assumed that all spikes generated by the spiking neural network are identical; being the timing of its spikes the key information in a spike train. Therefore, it is reasonable to model a spike train as a sequence of identical, instantaneous Dirac delta functions ($\delta(t)$), representing individual spikes as expressed in Eq. (9a).

$$r(t) = \sum_{i}^{M} \delta(t - t_i), \tag{9a}$$

$$r(t) = \sum_{i}^{M} H(t - t_i) \cdot \mathrm{e}^{-(t-t_i)/t_c}. \tag{9b}$$

In Eq. (9b), each Dirac delta function equation (9a) is substituted by an exponential function $\mathrm{e}^{-(t-t_i)/t_c}$. H is the Heaviside step function ($H(x) = 0$ if $x < 0$ and $H(x) = 1$ if $x \geq 0$) and M is the number of events in the spike train. In Eq. (8), distance D_{vr} is calculated as the integration of the difference between r and r', which are spike-driven functions with exponential terms, as indicated in Eq. (9b). Note that the resulting distance and, indeed, its interpretation depends upon the exponential decay constant, t_c in Eq. (9b). The distance also depends upon the number of spikes in the trains; it can be normalized dividing the number of spikes by M.

2.4.2. *Similarity measure*

The previously introduced similarity function D_{vr} depends on the t_c parameter (van Rossum cost parameter).[66] This parameter determines the penalization cost of two spikes when calculating the distance between them; if the distance is higher than t_c, the penalty will be one, the lower the distance is, the lower the penalty will be. According to Refs. 64 and 65, the CN population code is able to discriminate different stimuli around 35 ms after the first afferent spike; therefore, a t_c value of 40 ms is assumed. The codification has to respond in less than 40 ms to be consistent with biology, larger values shall be punished in the spike metric measure using this decay constant. Human micro-neurography recordings[52,53,64] (for distribution latencies of the first afferent spike, see Figs. 3 and 4 in Ref. 52) show that generated spike trains from different continuous stimuli have time lengths around 35 ms on average. Hence, in order to be biologically coherent, a spike train (microstate) is generated for each 40 ms time window providing sensor estimates through a spike-based pattern (Fig. 4(c)). The goal function to be calculated per executed trajectory is given by Eq. (10).

$$\Phi_{\min} = \min_{\substack{i \neq j \\ i,j}} (D_{\text{vr}}(r_i, r_j)), \quad r \in R \quad \text{and} \quad i, j \in \mathrm{N}^+$$
$$R = \{r_n\}, \quad \text{where } n = \text{number of trajectories}, \tag{10}$$

where r_i and r_j represent a pair of spike trains as a response to two different stimuli. A 40 ms time window activity after the stimulus presentation is taken into account to determine the stimulus response (i.e. in a 1 s trajectory, 25 time windows of 40 ms are obtained; therefore, consequently 25 spike trains corresponding to 25 microstates are obtained too). R is the whole set of spike patterns $\{r_n\}$ generated when following n trajectories. D_{vr} (r_i, r_j) represents the inter-stimulus distances between responses of two different stimuli. We try to find out the minimal distance between any pair of spikes in the whole set of time windows $\Phi_{\min}$. This process is implemented one by one in each benchmark trajectory $\Phi^n_{\min}$ obtaining Eq. (11):

$$\frac{1}{n}\sum_{1}^{n}(\Phi^n_{\min}), \quad \text{where } n = \text{number of trajectories}. \tag{11}$$

On the other hand, to be consistent with biology, intra-stimulus distance has been implemented by means of using a slightly stochastic threshold voltage in integrated and fire neurons (Eq. (2)). This means that the same stimulus may lead to a slightly different response (Fig. 5). The same input (trajectory) is presented three times to our receptive field; the whole obtained spike set is used in Eq. (11). Therefore, the effect of firing probability in stochastic leaky I&F neurons is compensated in the cost function.

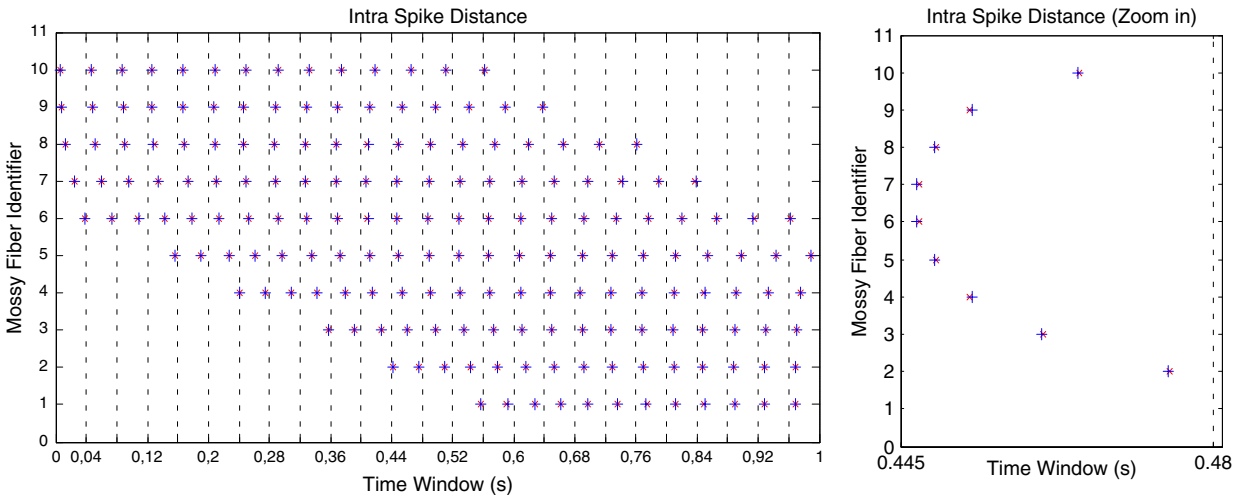


Fig. 5. Intra-Stimulus Distance. A 40 ms length time window showing two slightly different responses (cross and star markers) of the receptive field configurations shown in Fig. 4(b) (left side) for the same input due to stochasticity in the neural model.

2.4.3. *Fitness function*

The target problem, the spike representation to be used, and how to measure the fitness of a particular solution (receptive field distribution) have been defined. Hence, the final global fitness function to be optimized is given by Eq. (12).

$$\text{fitness} = \max\left(\frac{1}{n}\sum_{1}^{n}(\Phi_{\min}^{n})\right),$$
$$n = \text{number of trajectories.} \tag{12}$$

2.5. *Experimental setup*

As it was explained in the previous section, the optimization criterion is to find a combination of receptive fields capable of maximizing Eq. (12). That translates into a hard combinatorial and deceptive problem in which a good solution can be found near a poor region of the searching landscape. Specifically, up to 30 receptive fields are considered, each having 6000 positions (i.e. each vector element that goes from $r_{\min} = -6$ rad to $r_{\max} = 6$ rad in steps of 0.002 rad is considered as a possible centroid's position) and different coverage width (i.e. each value from $\sigma = 0.02$ to $\sigma = 10$ in steps of 0.001 is considered as a possible width of the associated receptive field). That is, the combinatorial space can be roughly estimated to be around 10^{169}.

Furthermore, the time to simulate the fitness of a single solution is computationally expensive and can last several seconds even in current processors. Therefore, to alleviate the burden of a deterministic exploration of the combinatorial space, we have used an evolutionary algorithm in which receptive fields are represented using three vectors; the first one encodes the number of receptive fields to be used, the second one contains the position of the receptive fields covering the range that can be achieved by the joint, and the last one contains the width of the respective receptive fields. An initial random solution (in terms of the number of receptive fields, their position over the defined range of possible values per joint, and their width) is supplied to the evolutionary algorithm. Through evolution, the evolutionary algorithm drives the population to promising regions of the searching space towards near-optimal solutions.

It is remarkable that an evaluation of a single trajectory takes 0.645 s in an Intel Core Quad Q6600 2.4 GHz 4 GB RAM (the evaluation has been performed using MATLAB). An evaluation of the previously described benchmark takes 7.75 s; the evaluation of the whole population of 100 individuals takes almost 13 min. An evaluation of such a population of over 300 or more epochs/generations will take days. That means that finding an optimal solution in a reasonable time becomes a problem. Fortunately, the nature of the evolutionary algorithm is inherently suited to be parallelized, offering a straightforward way to be scaled up improving performance in terms of convergence time.[69,70] The main idea is to speed-up the execution times by sharing the workload of the individuals among a pool of processors. To overcome the issue of computational time, a global parallel evolutionary algorithm has been implemented. This approach takes advantage of the parallelism at an evaluation level in the case of a very demanding fitness evaluation function (as is the case in this work). Global parallelization consists in the parallel evaluation of the individuals (i.e. candidate solutions),[71] usually following a master–slave model. The algorithm runs on the master node and the individuals are sent for evaluation to the slaves. Additionally, the master is responsible for collecting the results and applying the genetic operators.

In order to conduct the experiments, a 14 node computer cluster has been used. Each node has two Xeon E5320 processors at 1.86 GHz, with four nuclei and 4 GBs of RAM at each node.

3. Results

This Results section is focused in how to validate this new proposed methodology. Therefore, this section is structured in different steps showing how this methodology should be applied when it is particularized for a certain experiment.

Towards this aim, first, a predictable trajectory has been used, that is, a rectilinear trajectory (Fig. 4). If the results of the evolved set of receptive fields, obtained with this simplified problem, are suitable and consistent, it will be possible to extrapolate the followed methodology to an extension of the problem over different trajectories.

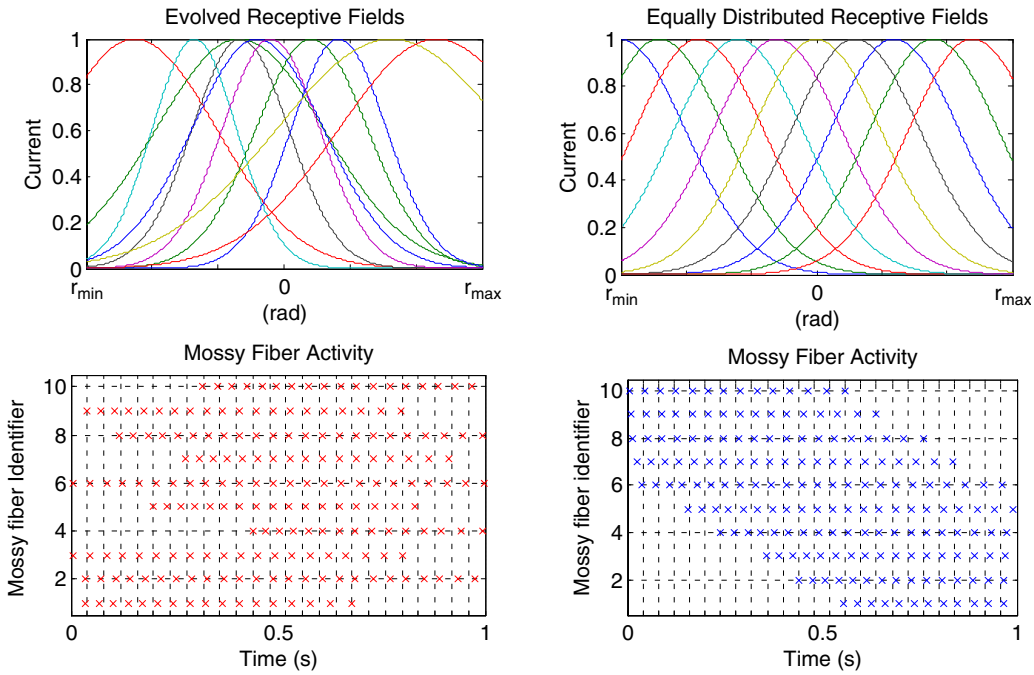
At this point, it is important to define specific metrics to evaluate how good a solution is. The Metrical Discrimination Analysis plays a fundamental role in the interpretation of the results (in terms of spike generation) and therefore, the evaluation

of the obtained solutions. We have included a subsection with the Metrical Discrimination Analysis within the Results section to make the reading more understandable.

3.1. *Metrical analysis*

In order to abstract the strategy behind the evolved values of the receptive fields, we work in a scenario in which D_{vr} (r_i, r_j) is linear. In this case, if D_{vr} (r_i, r_j) were linear, the way in which spikes would be distributed to maximize the distance between each other should be equidistant. An optimal distribution under linear assumption of the "cost" function D_{vr} in the mossy fiber number 1 of Fig. 6(a) would be a single spike in each time window (0.04 s) with a relative separation of 0.0016 s (time window/number of windows) from the previous time window spike time. The optimal relative distance between spikes in mossy fiber 1 would be 0.0016 s. This process should be repeated trough the other mossy fibers obtaining a value of $0.0016 \cdot$number of mossy fibers (in our case $0.0016 \cdot 10$). Translating this value into van Rossum distance ($t_c = TimeWindow$); the obtained intra-stimulus-value is $D_{\mathrm{vr}}(s_0, s_{0.0016}) = 0.0392 \cdot$ number of mossy fibers. As it was previously established, this result would correspond to a linear cost function, but D_{vr} is essentially an exponential cost function, which means that the obtained result cannot be used as an accurate optimum but, at least, it can be used as a nonfeasible upper bound for the fitness value the evolutionary algorithm could achieve.

The trajectory shown in Fig. 4(a) is used as the only input that feeds the evolvable receptive fields. The evolutionary algorithm, after 1500 epochs/generations of evolution using a population of 100 individuals, obtains a feasible near-optimal distribution (Fig. 6(a)). As it is shown, receptive fields which are placed near range extremes



(a) (b)

Fig. 6. Ten Evolved Receptive Fields versus Ten Equally distributed Receptive Fields. The trajectory represented in Fig. 4(a) is used as an input of the evolvable receptive fields whose final configuration is driven by the genetic algorithm (GA). (a) Evolved receptive fields generate a population coding that ensures a maximal fitness of Eq. (12). (Fitness 0.0434 instead of 0.0 in the equidistant solution.) A nonzero D_{vr} means that we have a set of spikes that represent the coding of the executed trajectory in unambiguous way (each time window has its own unique spike train representation that implies that any spike train can be distinguished from any other). The higher the fitness is, the more separated representation we have (a spike train is distanced from any other spike train of the whole set as much as possible). This involves that we can distinguish a spike train from any other sooner and more robustly in the presence of noise.

$[r_{\max}–r_{\min}]$ have a wider tail in comparison to equally distributed receptive fields (general solution) in which receptive fields around range extremes are under-utilized. The evolutionary algorithm optimizes the width of the receptive field at the extremes with two purposes; to cover a wider area and to provide a spike contribution in a more extended range. The evolutionary algorithm also distributes central receptive fields in an equally distributed way but with different widths; equally distributed central fields ensure proper range coverage in a rectilinear trajectory, different widths make distances between different stimuli not linearly related with the estimation being encoded.

3.1.1. *Metrical discrimination analysis*

In order to validate the previously presented EA optimization process, a metrical analysis is needed to have a proper tool for discerning a good solution from a bad one.

A metrical discrimination analysis allows us to numerically measure the main features of a given solution. As was established in Sec. 2.4, the desirable objective to achieve, broadly speaking is to generate a spike population set over time that represents the current sensorial state in an unambiguous way. That is, each sensorial state (trajectory state) should have a sole spike train representation which differentiates it from others. Consequently, it is necessary to prove that the evolved solution presents this discrimination feature between spike trains. Numerically, the EA maximizes the fitness function to enhance inter-stimulus distance. As a result, a given number representing this fitness is obtained, but, how does this inter-stimulus distance behave over time? How long does the discrimination between spike trains take? How can we evaluate that we have obtained an unambiguous spike train representation?

The metrical discrimination analysis of Eq. (13) gives answer to these questions.

$$[\Phi_{\max|\min}]_{i_{(1,1)},j_{(i+1,1)}}^{i_{(n\ Time Window-1,n\ Step)},j_{(n\ Time\ Window,n\ Step)}}, \tag{13}$$

where j and i are the sub-indexes that indicate the pair of selected spike trains to calculate the van Rossum distance $D_{\mathrm{vr}}(r_i, r_j)$. *n Step* is the integration step number within a time window. For instance, a time window of 40 ms, assuming an integration step of 1 ms, has 40 steps, i.e. *n Step* runs from 1 to 40. Finally, *n TimeWindow* is the index within the number of time windows into which a certain trajectory can be divided. For instance, a 1 s trajectory can be divided in 25 time windows of 40 ms each (thus, *n TimeWindow* runs from 1 to 25). As an example, $r_i = (1, 1)$ value corresponds to the set of spikes belonging to the first spike train of the first time window (0 s to 0.04 s) that are located in the first integration step (0 s to 0.001 s) (Fig. 7).

Since the stimulus changes along the trajectory, we can measure the inter-stimulus-distance in each time window. The first spike train belonging to the first 0.04 s time window is compared with the second spike train belonging to the second 0.04 s time window. The first spike train is then compared consecutively with the third spike train, then with the fourth, and so on. After this, the second spike train is compared successively with the third, the fourth, the fifth spike train, and so on. This is repeated for each spike train, thus making an exhaustive comparison process.

As a result, a *minimal-inter-stimulus-distance* curve is obtained. As it is shown in Fig. 8(a), Eq. (13) is applied to the EA solution (Fig. 6(a)). We can see

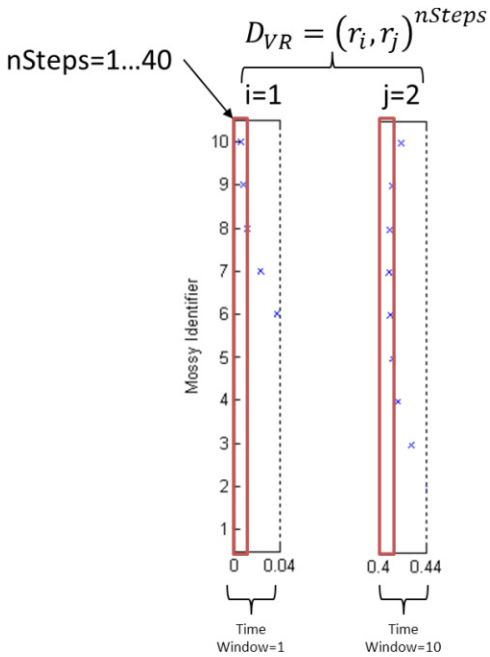


Fig. 7. Metrical discrimination analysis. Van Rossum distance is calculated between two spike trains (r_i, r_j) of different time windows (i, j) along n integration Steps within their corresponding time windows.

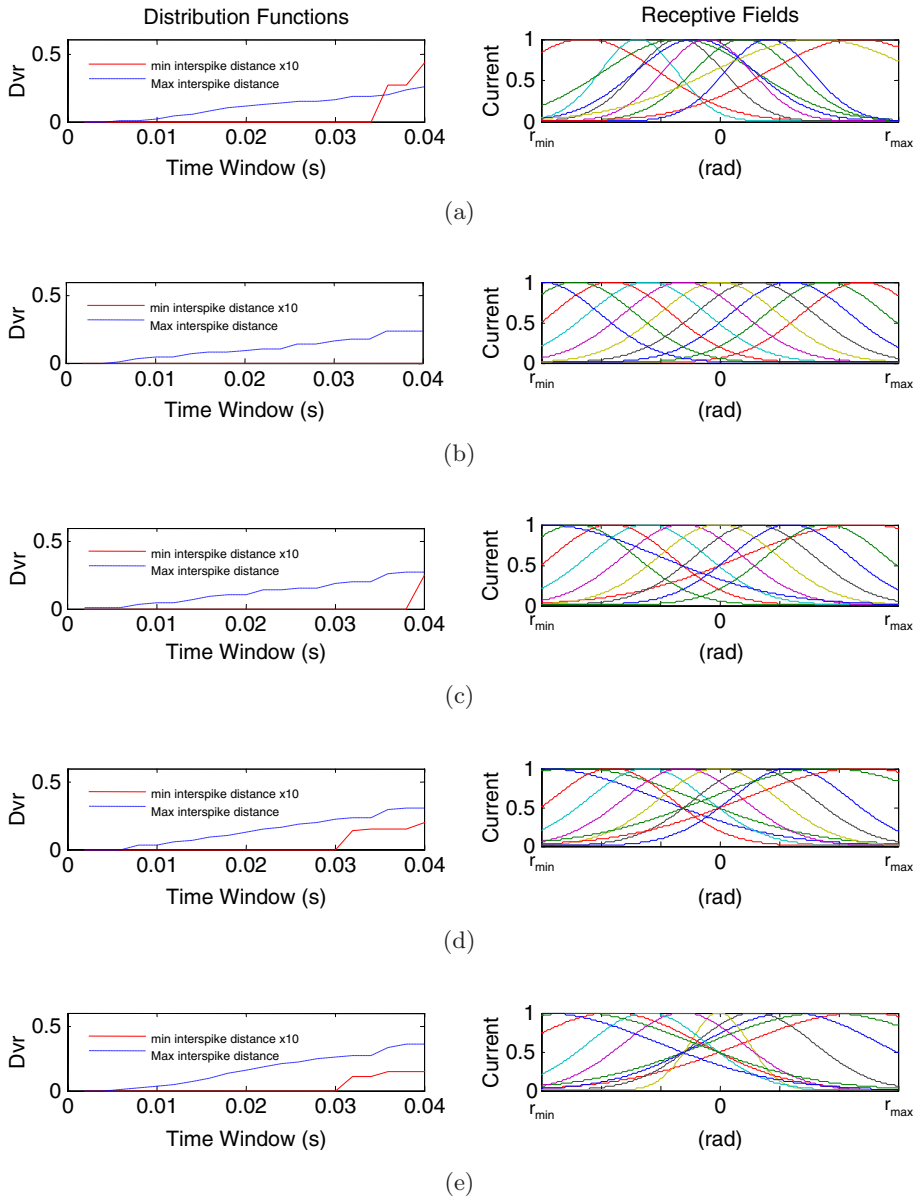


Fig. 8. (Color online). Inter-stimulus distribution function for different receptive fields illustrates the discrimination capability of the system (a 10 scale factor has been used to better plot the *minimal-inter-stimulus-distance* in the left panels). (a) Evolved receptive fields. (b) Equally distributed receptive fields. (c) and (d) Equally distributed receptive fields with a better coverage at both ends of the range. (e) Equally distributed receptive fields with a decreasing coverage of the range from the ends to the center.

the *minimal-inter-stimulus-distance* behavior. Considering a 40 ms time window, we can ensure that it takes 34 ms (first nonzero value for minimal-inter-stimulus-distance) to distinguish any two spike trains (i.e. the actual state — joint angle — in the trajectory) of the generated spike set when a rectilinear trajectory is performed and the state variables (joint angle in these experiments) are translated into spikes through the evolved receptive fields given by the EA. A nonzero value for minimal-inter-stimulus-distance means a perfect discrimination. The earlier this nonzero value is obtained, the sooner it is possible to distinguish a spike train from any other. That means that we can ensure a perfect discrimination

between any two spike trains over any 40 ms time window because discrimination actually takes time before 34 ms.

We can also verify that the *maximum minimal-inter-stimulus-distance* is 0.0434 and it is achieved around 40 ms. This maximum value represents an estimate of how easy the discrimination between different stimuli becomes in different time windows (sensory states).

To establish a comparison, this solution has been compared against other four solutions; an equally distributed receptive field solution (Fig. 8(b)), two different equally distributed receptive field solutions with a better coverage at both ends of the range (Figs. 8(c) and 8(d)), and finally, an equally distributed receptive field solution with a decreasing coverage of the range from the ends of the range to the center. The second solution is a hand-calibrated solution; the third and fourth solutions try to emulate the behavior of the solution obtained by the EA. Receptive fields placed near both ends of the range are modified in order to ensure a better response to initial and final trajectory segments and the fifth solution not only tries to mimic the behavior of the EA solution at both ends of the range but also the behavior at central range positions, the receptive field width is modified from the range of extreme positions to the center position; the higher the distance of the receptive field center from the range center value, the wider receptive field is used. Different widths covering center range values ensure quite different responses to slightly different entries.

As is shown in Fig. 8(a), in order to distinguish any two spike trains (microstates) of the generated set, it takes over 0.034 s. A maximum of 0.434 ($0.0434 \cdot$ numMossyFibers) in the *minimal-intra-stimulus-distance* is achieved (this value is consistent with the result obtained by the EA solution). Figure 8(b) shows a constant zero *minimal-inter-spike-distance*; equally, distributed receptive fields are not able to properly discriminate two spike trains of the generated spike set.

On the other hand, although solutions of Figs. 8(c) and 8(d) really do a discrimination between spike trains (*minimal-inter-stimulus-distance* does not remain constantly zero) even sooner than the evolved solution (Fig. 8(d) at 0.03 s), neither of them achieves a maximum value of the *minimal-inter-stimulus-distance* near 0.434. The obtained maximum values are around 50% lower, so a better coverage of the whole range is implemented by the evolved solution.

Finally, the last solution (Fig. 8(e)) shows the same problem as the previous one, whereas a discrimination is possible even sooner than the evolved solution and the *maximal-inter-stimulus-distance* is better than Figs. 8(c) and 8(d) (blue dashed curve), the maximum value of the *minimal-inter-stimulus-distance* is 66% less than the evolved solution. The evolved solution still represents a better way to cover the whole range of joint angles; the evolved solution not only ensures a perfect discrimination between spike stimuli but also ensures a maximal distance between their spike representations (Fig. 9).

3.2. *Analysis of evolved receptive fields in multiple trajectories*

Once the proposed methodology has been shown to succeed in a single-trajectory scenario, now the EA-based optimization methodology is generalized to different trajectories. All the trajectories illustrated in Fig. 1 are used.

In this scenario, the solutions will not be easily compared to manually handcrafted ones capturing the essence of the evolved solutions. The EA has been set to manage up to 30 possible receptive fields to be conjointly evolved. After the EA evolves a population of 100 individuals over 2000 epochs, a final

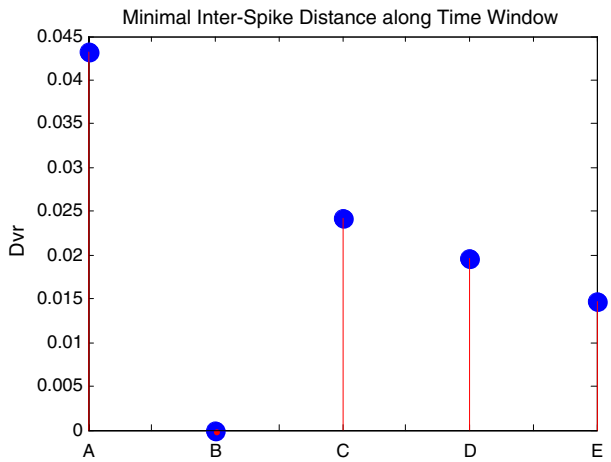


Fig. 9. Minimal-inter-stimulus-distances (D_{vr}) achieved at time 0.04 s by different by different receptive field distribution solutions. Cases A, B, C, D, and E corresponding to the respective (a), (b), (c), (d), and (e) solutions illustrated in Fig. 8.

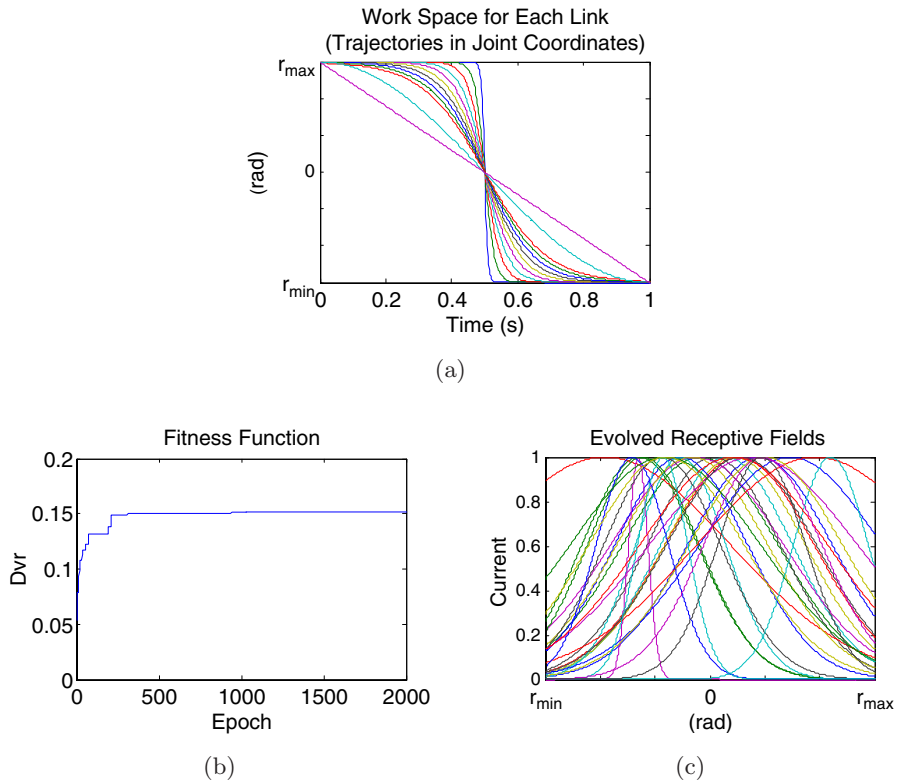


Fig. 10. Evolved receptive field solution. (a) The EA uses the multiple trajectory benchmark to cover the defined workspace [$r_{\max}$–$r_{\min}$]. Through evolution, the EA obtains a receptive field distribution that ensures discrimination between any spike-train mossy fiber produces (Fig. 2). (b) Fitness evolution. Fitness curve converges properly after 1000 epochs. (c) Final evolved distribution of the receptive fields.

evolved solution is obtained as is shown in Fig. 10. Figure 10(c) shows the resulting receptive fields after the optimization process.

We can see that the EA has concentrated the receptive field distribution in the [$2/3r_{\max}$–$2/3r_{\min}$] range. In this range region, the majority of the trajectories have sharp changes in their values; having a pretty concentrate centroid distribution right in [$2/3r_{\max} - 2/3r_{\min}$] values of the range ensures a proper population of sensitized neurons (their base current forces neurons to be closer to their firing state) to fast changes in the input value. That is, fast changes in trajectory values involve very different generated spike trains, which is what we are looking for in this area.

On the other hand, at the end of the range values, the EA has increased the width of the receptive fields providing a sparse distribution of them. Placing those wide receptive fields at the ends of the range is a way to distinguish the extreme areas in the spiking code. This involves that, at least, immediately, one neuron is firing in this area, being accompanied by the rest of firing neurons with certain delays no longer than 40 ms (time windows) due to the width of the central receptive fields. Central receptive fields are wide enough to be sensitive to input values belonging to both ends of the range areas.

3.2.1. *Metrical discrimination analysis*

The metrical discrimination analysis extended to the multiple trajectories benchmark is given by Eq. (14).

$$\frac{1}{n}\sum_{1}^{n}[\Phi_{\max|\min}]_{i_{(1,1)},j_{(i+1,1)}}^{i_{(n\,TimeWindow-1,n\,Step)},j_{(nTimeWindow,nStep)}}, \tag{14}$$

where j and i are the sub-indexes that indicate the pair of selected spike trains to calculate the van Rossum distance $D_{\mathrm{vr}}(r_i, r_j)$. As previously described, *nStep* is the integration step number within a time window. *nTimeWindow* is an index within the number of time windows in which a

certain trajectory can be divided into, and finally, n is the number of the trajectories of the benchmark. This equation is computed once in every trajectory obtaining a set of curves (a curve per trajectory). Each curve represents the behavior of the *minimal-inter-stimulus-distance* over each 0.04 ms time window along the trajectory (using the receptive field solution given by the EA). As it was done in Sec. 3.1, the spike train set generated by a trajectory is computed according to Eq. (13). As a result, a *minimal-inter-stimulus-distance* curve is obtained per trajectory.

A final mean curve is calculated using this set of minimal-inter-stimulus-distance curves applying Eq. (14). As it is shown in Fig. 11, the evolved solution has been compared against a designed solution which consists of equally distributed receptive fields (other equally distributed solutions with different RF widths were tested but these experiments did not provide any new further information) and also, to a solution manually implemented which tries to emulate the EA solution. This illustrates how the EA solution itself can be used, or how it is also possible to try to emulate it (after interpreting it) towards designing efficient hand-crafted solutions based on the EA guidance.

Figure 11(a) shows the evolved solution and its performance. The discrimination condition between any pair of spike trains from the generated spike set using evolved receptive fields is possible after 0.014 s in average with a maximum at the *minimal-inter-stimulus-distance* of $D_{\text{vr}} = 0.0476$. In contrast,

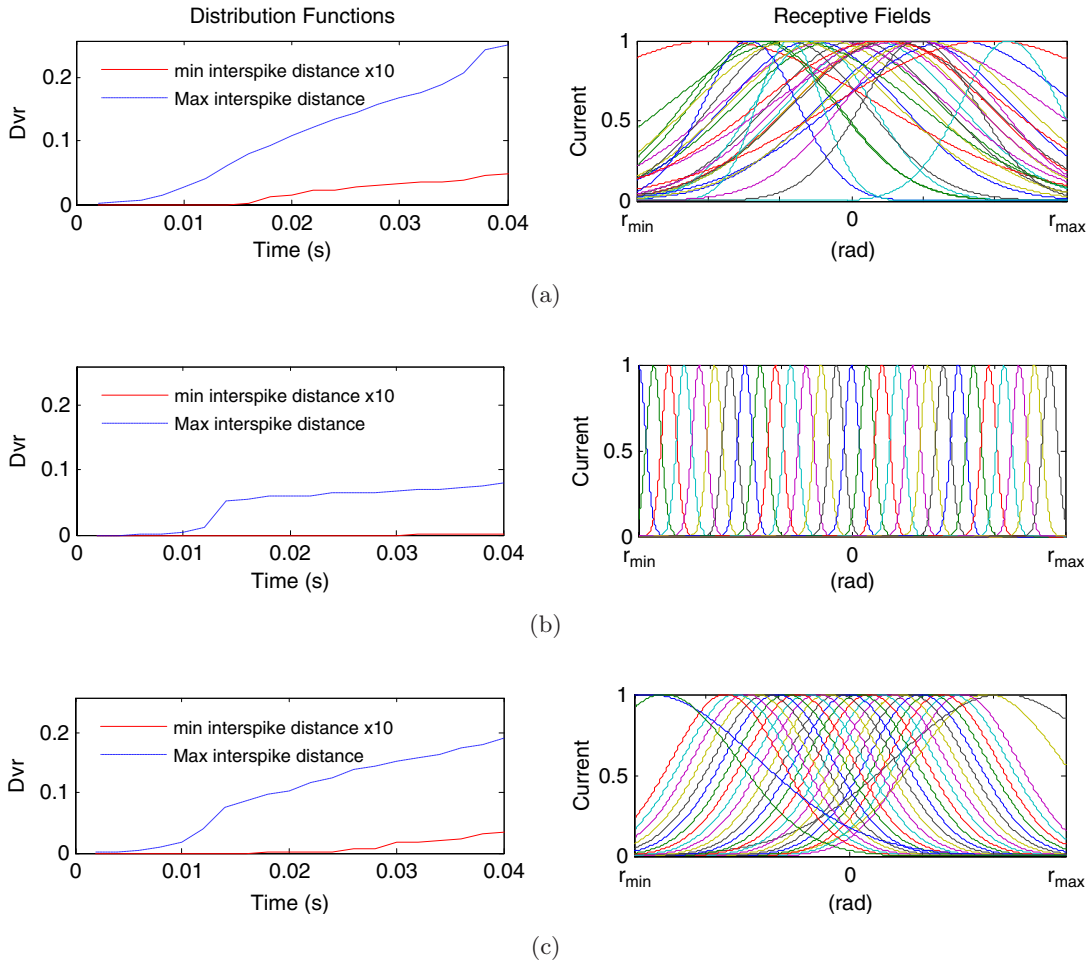


Fig. 11. Inter-stimulus distribution obtained by the EA using the whole set of benchmark trajectories. (a) Evolved receptive fields. (b) Equally distributed receptive fields. (c) Equally distributed receptive fields with a better coverage at both ends and in interval $[2/3r_{\max}–2/3r_{\min}]$ of the range.

the equally distributed receptive field standard solution presents a maximum value at *minimal-inter-stimulus-distance* of $D_{\mathrm{vr}} = 0.00138$. The discrimination between any pair of spike trains is possible after 0.032 s on average. These values are clearly improved by the evolved solution. Finally, equally distributed receptive fields with a better coverage at both ends and in interval $[2/3r_{\max}–2/3r_{\min}]$ of the range are able to discriminate any pair of spike trains from the generated spike set after 0.018 s on average. The maximum value of the *minimal-inter-stimulus-distance* at each time window is $D_{\mathrm{vr}} = 0.0356$. This maximum value of the *minimal-inter-stimulus-distance* is larger in the evolved solution than in the others; therefore, the discrimination process is not only executed sooner, but also with higher *inter-stimulus-distance* values (which represent a larger margin that is useful in the case of noise in the sensory signal estimation). The EA solution not only ensures an earlier discrimination between spike trains, but also increases the distance between any pair of spike trains.

4. Conclusion

A methodology for efficiently representing incoming encoder signals from different links in terms of spikes in a plausible robot scenario is presented. Several approaches in controlling robots with cerebellum-like networks have been proposed in the literature[26–29] (all of them keeping classical control theory in mind). Little attention has been given to efficient sensory representation in these approaches. To that aim, an evolutionary algorithm as an optimization engine has been proposed in this contribution. In this way, a goal function that captures how sensory information can be efficiently represented in terms of spike trains was defined, maximizing the *minimal-inter-stimulus-distance* when performing movements (benchmark trajectories). In the framework of experiments with cerebellar-based robot control[1–3,13,72,73] or other bio-inspired experiments,[74,75] the presented contribution will allow at initial stages of the adaptation mechanisms of the cerebellum to distinguish more accurately specific instants along the trajectories in which potential corrections or actions need to be performed.

The receptive fields in our sensory input layer have been evolved. We focus on the way in which these receptive fields have to be distributed both to encode each sensorial state in an unambiguous way and to enhance information transfer (in terms of entropy) between mechanoreceptor signals and their spike representations. The receptive field configuration task is carried out by the aforementioned Evolutionary Algorithm. Such an algorithm evolves receptive fields along the robot-link work space according to a goal function that takes into account the metrical properties of the spike train space.

Beyond this specific contribution, this work also presents a general methodology of using EAs for optimization purposes when addressing reverse engineering of biological systems. In this scenario, it is important to implement a goal function that captures the essence of attributed properties of the system which is being optimized. In our case, the goal function is the optimization of sensory representation in terms of spikes with inter-spike discrimination capability along movement trajectories. This required the definition of a metric to allow the evaluation of the different candidate solutions, in order to derive a final fitness function for the EA. The definition of a fitness function that allows convergence through an EA is not straight forward; it required a preliminary experimental stage in which preliminary simulations where done with a single trajectory in which the results (and obtained solutions in terms of receptive field configuration) were easy to interpret. The searching space in this kind of problems and the computational cost of spike train distances may require the parallelization of the EA, as it has been done in this work.

This technique will be included into robotic experiments with cerebellar-like modules as corrective engines to evaluate how an optimal sensory representation facilitates an effective adaptation at the cerebellum. Thus, it will be applied to object manipulation experiments with an adaptive cerebellar-like module. In previous experimental studies,[76–78] the translation from analog robotic sensory signals to spike trains has been done manually (through a manually designed receptive filter configuration) to facilitate an easy discrimination when performing different trajectories.

We will also apply the presented technique to tactile sensors[79–82] to maximize information

transmission as discrimination between microstates in the framework of sensing tasks.

We will also study the possibility of introducing an STDP[83] law that increases the performance of the evolved system in such sensing task frameworks. Furthermore, we will apply other parallel optimization schemes[84,85] in order to scale up the complexity of the representations that can be studied.

Acknowledgments

This work has been supported by the EU grant REALNET (FP7-IST-270434) and national projects ARC-VISION (TEC2010-15396), MULTIVISION (TIC-3873), and ITREBA (TIC-5060).

References

1. R. R. Carrillo, E. Ros, C. Boucheny and O. J.-M. D. Coenen, A real-time spiking cerebellum model for learning robot control, *Biosystems* **94**(1–2) (2008) 18–27.
2. R. R. Carrillo, E. Ros, B. Barbour, C. Boucheny and O. J.-M. D. Coenen, Event-driven simulation of neural population synchronization facilitated by electrical coupling, *Biosystems* **87**(2–3) (2007) 275–280.
3. C. Boucheny, R. R. Carrillo, E. Ros and O. J.-M. D. Coenen, Real-time spiking neural network: An adaptive cerebellar model, LNCS, Vol. 3512 (2005), pp. 136–144.
4. E. Nichols, L. J. McDaid and N. H. Siddique, Case study on self-organizing spiking neural networks for robot navigation, *Int. J. Neural Syst.* **20**(6) (2010) 501–508.
5. S. P. Johnston, G. Prasad, L. Maguire and T. M. McGinnity, An FPGA hardware/software codesign methodology towards evolvable spiking networks for robotics application, *Int. J. Neural Syst.* **20**(6) (2010) 447–461.
6. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *Int. J. Neural Syst.* **19**(4) (2009) 295–308.
7. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for eeg classification and epilepsy and seizure detection, *Integr. Comput-Aided Eng.* **14**(3) (2007) 187–212.
8. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Neural Networks* **22**(10) (2009) 1419–1431.
9. J. Iglesias and A. E. P. Villa, Emergence of preferred firing sequences in large spiking neural networks during simulated neuronal development, *Int. J. Neural Syst.* **18**(4) (2008) 267–277.
10. S. Soltic and N. Kasabov, Knowledge extraction from evolving spiking neural networks with rank order population coding, *Int. J. Neural Syst.* **20**(6) (2010) 437–445.
11. A. Vidybida, Testing of information condensation in a model reverberating spiking neural network, *Int. J. Neural Syst.* **21**(3) (2011) 187–198.
12. A. F. Jahangiri and D. M. Durand, Phase resetting of spiking epileptiform activity by electrical stimulation in the CA3 region of the Rat Hippocampus, *Int. J. Neural Syst.* **21**(2) (2011) 127–138.
13. J. B. Passot, N. R. Luque and A. Arleo, Internal models in the cerebellum: A coupling scheme, LNAI, Vol. 6226 (2010), pp. 435–446.
14. M. Ito, *The Cerebellum and Neural Control* (Raven Press, New York, 1984).
15. M. Kawato, K. Furukawa and R. Suzuki, A hierarchical neural-network model for control and learning of voluntary movement, *Biol. Cybern.* **57**(3) (1987) 169–185.
16. D. M. Wolpert, R. C. Miall and M. Kawato, Internal models in the cerebellum, *Trends Cognit. Sci.* **2**(9) (1998) 338–347.
17. R. M. C. Spencer, R. B. Ivry and H. N. Zelaznik, Role of the cerebellum in movements: Control of timing or movement transitions? *Exp. Brain Res.* **161** (2005) 383–396.
18. M. Desmurget and S. Grafton, Forward modeling allows feedback control for fast reaching movements, *Trends Cognit. Sci.* **4**(11) (2000) 423–431.
19. M. R. Delong and P. L. Strick, Relation of basal ganglia, cerebellum, and motor cortex, *Brain Res.* **71** (1974) 327–335.
20. G. Hinton, Parallel computations for controlling an arm, *J. Mot. Behav.* **16** (1984) 171–194.
21. J. R. Flanagan *et al.*, Control of trajectory modifications in target-directed reaching, *J. Mot. Behav.* **25** (1993) 140–152.
22. J. Butterfaß, M. Grebenstein, H. Liu and G. Hirzinger, *DLR Hand II*: *Next Generation of a Dextrous Robot Hand* (ICRA, 2001).
23. A. Albu-Schäffer *et al.*, The DLR lightweight robot: Design and control concepts for robots in human environments, *Int. J. Ind. Robot* **34**(5) (2007) 376–385.
24. A. De Luca and B. Siciliano, Closed-form dynamic model of planar multilink lightweight robots, *IEEE Trans. Syst., Man, Cybern.* **21**(4) (1991) 826–839.
25. P. van der Smagt, F. Groen and K. Schulten, Analysis and Control of a Rubbertuator Arm, *Biol. Cybern.* **75**(5) (1996) 433–440.
26. J. S. Albus, Data storage in the cerebellar model articulation controller (CMAC), *J. Dyn. Syst. Meas. Contr.* **3** (1975) 228–233.
27. J. C. Houk, J. T. Buckingham and A. G. Barto, Models of the cerebellum and motor learning, *Behav. Brain. Sci.* **19**(3) (1996) 368–383.
28. N. Schweighofer, Computational models of the cerebellum in the adaptive control of movements, Ph.D. thesis (1995).

29. P. van der Smagt, Cerebellar control of robot arms, *Connect. Sci.* **10** (1998) 301–320.
30. P. van der Smagt, Benchmarking cerebellar control, *Robot. Auton. Syst.* **32** (2000) 237–251.
31. J. R. Flanagan and A. M. Wing, The role of internal models in motion planning and control: Evidence from grip force adjustments, *J. Neurosci.* **17** (1997) 1519–1528.
32. S. Keele, R. Ivry and R. Pokorny, Force control and its relation to timing, *J. Mot. Behav.* **19** (1987) 96–114.
33. R. C. Miall, D. J. Weir, D. M. Wolpert and J. F. Stein, Is the cerebellum a smith predictor? *J. Mot. Behav.* **25** (1993) 203–216.
34. M. Ito, Neurophysiological aspects of the cerebellar motor control system, *Int. J. Neurol.* **7** (1970) 162–176.
35. M. Kawato and H. Gomi, A computational model of four regions of the cerebellum based on feedback-error learning, *Biol. Cybern.* **68** (1992) 95–103.
36. N. Schweighofer, M. A. Arbib and M. Kawato, Role of the cerebellum in reaching movements in human. I. Distributed Inverse dynamics control, *Eur. J. Neurosci.* **10** (1998) 86–94.
37. M. Glickstein, *The Cerebellum*: *From Structure to Control, Mossy-fibre Sensory Input to the Cerebellum* (Elsevier, Progress in Brain Research, 1997).
38. J. R. Bloedel and J. Courville, *Handbook of Physiology, The Nervous System, Motor Control. Cerebellar Afferent Systems* (John Wiley and Sons, Inc., 2011).
39. M. Glickstein *et al.*, Visual pontocerebellar projections in the macaque, *J. Comp. Neurol.* **349**(1) (1994) 51–72.
40. C. R. Legg, B. Mercier and M. Glickstein, Corticopontine projection in the rat: The distribution of labelled cortical cells after large injections of horseradish peroxidase in the pontine nuclei, *J. Comp. Neurol.* **286**(4) (1989) 427–441.
41. P. Brodal, The corticopontine projection from the visual cortex in the cat. I. The total projection and the projection from area 17, *Brain Res.* **39**(2) (1972) 297–317.
42. P. Brodal, The corticopontine projection from the visual cortex in the cat. II. The projection from areas 18 and 19, *Brain Res.* **39**(2) (1972) 319–335.
43. M. Glickstein, J. G. May 3rd and B. E. Mercier, Corticopontine projection in the macaque: The distribution of labelled cortical cells after large injections of horseradish peroxidase in the pontine nuclei, *J. Comp. Neurol.* **235**(3) (1985) 343–359.
44. G. Nyberg and A. Blomqvist, The central projection of muscle afferent fibres to the lower medulla an upper spinal cord: An anatomical study in the cat with the transganglionic transport method, *J. Comp. Neurol.* **230**(1) (1987) 99–109.
45. P. L. van Kan, A. R. Gibson and J. C. Houk, Movement-related inputs to intermediate cerebellum of the monkey, *J. Neurophysiol.* **69**(1) (1993) 74–94.
46. P. D. Mackie, J. W. Morley and M. J. Rowe, Signalling of static and dynamic features of muscle spindle input by external cuneate neurons in the cat, *J. Physiol.* **519**(2) (1999) 559–569.
47. J. Baker, A. Gibson, M. Glickstein and J. Stein, Visual cells in the pontine nuclei of the cat, *J. Physiol.* **255**(2) (1976) 415–433.
48. S. Wu, S. Amari and H. Nakahara, Population coding and decoding in a neural field: A computational study, *Neural Comp.* **14** (2002) 999–1026.
49. A. Pouget, P. Dayan and R. Zemel, Information processing with population codes, *Nature Rev. Neurosci.* **1** (2000) 125–132.
50. T. Flash and T. J. Sejnowski, Computational approaches to motor control, *Curr. Opin. Neurobiol.* **11**(6) (2001) 655–662.
51. B. Amirikian and A. P. Georgopoulos, Directional tuning profiles of motor cortical cells, *Neuroscience* **36** (2000) 73–79.
52. R. S. Johansson and I. Birznieks, First spikes in ensembles of human tactile afferents code complex spatial fingertip events, *Nat. Neurosci.* **7** (2004) 170–177.
53. R. S. Johansson and J. R. Flanagan, Coding and use of tactile signals from the fingertips in object manipulation tasks, *Nat. Rev. Neurosci.* **10** (2009) 345–359.
54. S. Schliebs, N. Kasabv and M. Defoin-Platel, On the probabilistic optimization of spiking neural networks, *Int. J. Neural Syst.* **20**(6) (2010) 481–500.
55. J. J. Craig, *Introduction to Robotics, Mechanics and Control*, 3rd edn. (Pearson Education, Inc., New Jersey, 2005).
56. D. M. Wolpert, Z. Ghahramani and M. I. Jordan, An internal model for sensorimotor integration, *Science* **269** (1995) 1880–1882.
57. S. J. Goodbody and D. M. Wolpert, Temporal and amplitude generalization in motor learning, *J. Neurophysiol.* **79** (1998) 1825–1838.
58. D. R. Boff and J. E. Lincoln, *Engineering Data Compendium*: *Human Perception and Performance* (AAMRL, 1988).
59. N. V. Swindale, Orientation tuning curves: Empirical description and estimation of parameters, *Biol. Cybern.* **78**(1) (1998) 45–56.
60. W. Gerstner and W. Kistler, *Spiking Neuron Models* (Cambridge University, Cambridge, 2002).
61. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* (Springer Verlag, 2003).
62. E. Alfaro-Cid *et al.*, Comparing multiobjective evolutionary ensembles for minimizing type I and II errors for bankruptcy prediction, *IEEE Congress on Evol. Comp.* (2008).

63. L. Trujillo and G. Olague, Automated design of image operators that detect interest points, *Evol. Comput.* **16** (2008) 483–507.
64. R. Brasselet, R. S. Johansson and A. Arleo, Optimal context separation of spiking haptic signals by second-order somatosensory neurons, *Adv. Neural Inform. Process. Syst.* **22** (2009) 180–188.
65. R. Brasselet, R. S. Johansson and A. Arleo, Quantifying neurotransmission reliability through metrics based information analysis, *Neural Comput.* **23**(4) (2011) 852–881.
66. M. C. W. van Rossum, A novel spike distance, *Neural Comput.* **13** (2001) 751–763.
67. J. D. Victor and K. P. Purpura, Nature and precision of temporal coding in visual cortex: A metric-space analysis, *J. Neurophysiol.* **76** (1996) 1310–1326.
68. J. D. Victor and K. P. Purpura, Metric-space analysis of spike trains: Theory, algorithms, and application, *Network* **8** (1997) 127–164.
69. E. Alba and M. Tomassini, Parallelism and evolutionary algorithms, *IEEE Trans. Evol. Comput.* **6**(5) (2002) 443–462.
70. M., Tomassini, *Spatially Structured Evolutionary Algorithms* (Springer, 2005).
71. E. Cantu-Paz, A survey of parallel genetic algorithms, calculateurs paralleles, *Reseaux et Systems Repartis* **10**(2) (1998) 141–171.
72. J. Porrill, P. Dean and J. V. Stone, Recurrent cerebellar architecture solves the motor-error problem, *Proc. Biol. Sci.* **271**(1541) (2004) 789–796.
73. M. Fujita, Adaptive filter model of the cerebellum, *Biol. Cybern.* **45**(3) (1982) 195–206.
74. T. Yamazaki and S. Tanaka, The cerebellum as a liquid state machine, *Neural Networks* **20** (2007) 290–297.
75. T. Yamazaki and S. Tanaka, Neural modeling of an internal clock, *Neural Comput.* **17**(5) (2005) 1032–1058.
76. N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J.-M. D. Coenen and E. Ros, Cerebellar input configuration toward object model abstraction in manipulation tasks, *IEEE Trans. Neural Netw.* **22**(8) (2011) 1321–1328.
77. N. R. Luque, J. A. Garrido, R. R. Carrillo, O. J.-M. D. Coenen and E. Ros, Cerebellar-like corrective-model abstraction engine for robot movement control, *IEEE Trans. Syst. Man Cybern B Cybern.* **41**(5) (2011) 1299–1312.
78. N. R. Luque, J. A. Garrido, R. R. Carrillo, S. Tolu and E. Ros, Adaptive cerebellar spiking model embedded in the control loop: Context switching and robustness against noise, *Int. J. Neural Syst.* **21**(5) (2011) 385–401.
79. F. Vidal-Verdú, M. J. Barquero, J. Castellanos-Ramos, R. Navas-González, J. A. Sánchez, J. Serón and A. García-Cerezo, A large area tactile sensor patch based on commercial force sensors, *Sensors* **11** (2011) 5489–5507.
80. F. Vidal-Verdú, O. Oballe-Peinado, J. A. Sánchez-Durán, J. Castellanos-Ramos and R. Navas-González, Three realizations and comparison of hardware for piezoresistive tactile sensors, *Sensors* **11**(3) (2011) 3249–3266.
81. J. Castellanos-Ramos, B. Navas-González, H. Macicior, T. Sikora, E. Ochoteco and F. Vidal-Verdú, Tactile sensors based on conductive polymers, *Microsyst. Technol.* **5** (2010) 765–776.
82. R. Maldonado-López, F. Vidal-Verdú, G. Liñán and A. Rodríguez-Vázquez, Integrated ciruitry to detect slippage inspired by human skin and artificial retinas, *IEEE Trans. Circuits Syst. I, Reg. Papers* **56** (2009) 1554–1565.
83. T. J. Strain, L. J. McDaid, L. P. Maguire and T. M. McGinnity, An STDP training algorithm for a spiking neural network with dynamic threshold neurons, *Int. J. Neural Syst.* **20**(6) (2010) 463–480.
84. J. L. Redondo, I. García and P. M. Ortigosa, Parallel evolutionary algorithms based on shared memory programming approaches, *J. Supercomput.* **1** (2009) 1–10.
85. J. L. Redondo, B. Pelegrín, P. Fernández, I. García and P. M. Ortigosa, Finding multiple global optima for unconstrained discrete location problems, *Optim. Meth. Software* **26**(2) (2011) 207–224.