A new embodied motor-neuron architecture

Paolo Arena *Senior, IEEE*, Luca Patanè *Member, IEEE*, and Angelo G. Spinosa

***Abstract*—This paper introduces a novel bio-inspired motor neuron dynamical unit where the neural dynamics, responsible for the generation of the action potentials is embedded into the actuator dynamics, which here plays the role of, and substitutes, the recovery variable into the classical neuron equations. A recently introduced nullcline-based control strategy, over servomotors embedded into piecewise linearly approximated FitzHugh Nagumo neuron models, is here applied to the synchronization of two embodied motor-neuron units in the form of either a continuously active proportional gain or an event-driven strategy. In view of the application to such problems as the generation of adaptive control laws for distributed oscillatory networks, at the basis of bio-inspired walking machines, the advantages in terms of the reduced-order dynamical equations and ease of synchronization are presented both through simulations and with experiments devoted to control networked Dynamixel MX-28AT servomotors via an MCU board.**

***Index Terms*—Nonlinear oscillator, FHN neuron model, synchronization, motor control**

I. INTRODUCTION

Bio-inspired locomotion generation and control involve physical and functional relationships among specific neural circuits and the corresponding muscle activation, which generates leg periodic motion alternating between stance and swing phases. Gait generation is also strongly connected with the coordination among legs. Two main approaches followed in literature are often identified as Central Pattern Generator (CPG) and Decentralized Locomotion Control (DLC). The former considers coordinated locomotion activity as stereotyped and imposed by the neural circuits, which can act, in principle, independently of sensory feedback [1]. The latter presents the gait as the dynamical solution emerging from reflex-based leg motions joined to the definition of coordination rules: here sensory feedback is needed to generate locomotion [2]–[5]. **DLC strategies can also be developed based on multiple CPGs where gait coordination and walking stability is obtained using sensory feedback [6]. Further interactions will include the interplay between the robot body and limbs using cross-coupled sensory feedback [7], [8].** In the following, an approach between CPG and DLC will be presented, exploiting the advantages of both strategies. The main novelty of the approach presented in this paper is that, starting from considerations on the basics of the neural dynamical behaviour, a completely new motor-neuron configuration is introduced, where neuron and motor act symbiotically: the neuron dynamics lose one state variable, in details its recovery one, which is directly drawn by the actuator dynamics. In this way, the actuator is merged within the neural controller for the realization of a new conceptual type of adaptive neuron motor unit, called "embodied". As a consequence, from the one side, the neural controller needs the actuator feedback to work properly (like the DLC), whereas, from the side of the network perspective, the phase displacement is decided in a master-slave configuration, as within the CPG archetype. The new concept of embodied motor neuron unit differentiates from CPG and DLC, since both take into main account the neural controller side, independently of the actuation part. Another important aspect of our approach is the extremely reduced order of the whole motor neuron dynamical unit: only one additional state variable is added to the actuator dynamics (namely the neuron membrane potential) for control purposes. The advantage increases in view of the typical applications of neural control to bio-inspired robots which possess multiple legs with multiple actuators and neural units, often implementing a flexor-extensor control mode, which further increases the dynamics order [9], [10].

A further interesting characteristic of our strategy is the ease of synchronization control exploiting the modulation of the system nullclines directly into the neuron motor phase plane. In literature, a few approaches show some similarities with ours [11]. These, however, do not consider several key aspects of our approach, namely: the possibility to replace the motor dynamics with, in principle, any actuator and the implementation of a control strategy directly acting on the neuron motor phase plane. The suitability in controlling not only the single embedded actuator dynamics but also the phase among different embodied units will be achieved through a master-slave adaptive control rule implemented in the phase plane [12], where the dynamic evolution can be suitably and easily controlled. A relevant aspect is the control of the time spent by the leg in the swing or the stance phase, which affects the overall spiking frequency. Typically, models of biological CPGs endow spiking neurons with some additional state variables which slow the spiking frequency [10]. A biological neuron can easily reach a firing frequency of 10Hz [13], and even lower: for example, in the crustacean stomatogastric neural ganglion, neurons can reach very low firing frequencies (0.5-2Hz), to adapt to the actuation medium. In our model, we can also easily speed up or slow down the neural oscillation frequency. This property largely helps to adapt the neuron dynamics to the actuator one. Following the strategy presented in [12], the neuron dynamics adaptation is faced by adopting a Piecewise linear (PWL) version of the Fitz-Hugh Nagumo (FHN) neuron, together with a control action developed within the phase plane. The modulation of the nullcline position is

Paolo Arena is with the Department of Electrical, Electronic and Computer Engineering, University of Catania, 95100, Italy and CNR-IASI, Istituto di Analisi del Sistemi e Informatica "A. Ruberti", Rome, Italy (e-mail: paolo.arena@unict.it)

Luca Patanè is with the Department of Engineering, University of Messina, 98166, Italy (e-mail: lpatane@unime.it).

Angelo Giuseppe Spinosa is with the Department of Electrical, Electronic and Computer Engineering, University of Catania, 95100, Italy (e-mail: angelo.spinosa@unict.it)

used to directly control the time spent in such regions of the phase plane corresponding to the stance/swing phase of the leg motion. The embodiment and master-slave adaptive phase plane control will be demonstrated through a hardware setup, using a simple adaptation control law, and exploiting the model of a commercial servomotor.

II. Embodied neuron models: mathematical tools

An embodied neuron is here formalised as an augmented $N+1$-order system which embeds an N-order sub-system, corresponding to the "body" of a more articulated unit. Let $\mathbb{X}_{\mathcal{L}}$ and $\mathbb{X}_{\mathcal{N}}$ be the two-state spaces of the neural and embedded parts, respectively. Thereby, neural and embedded state functions are described as a scalar function $\mathcal{F}:\mathbb{X}_{\mathcal{L}}\times\mathbb{X}_{\mathcal{N}}\rightarrow\mathbb{X}_{\mathcal{L}}$, and a vector function $\mathcal{G}:\mathbb{X}_{\mathcal{L}}\times\mathbb{X}_{\mathcal{N}}\rightarrow\mathbb{X}_{\mathcal{N}}$, respectively. On the whole, an (augmented) embodied model can be described as follows:

$$\begin{cases}\varepsilon\dot{v} &= \mathcal{F}(v,\mathbf{y_f})\in\mathbb{X}_{\mathcal{L}}:dim(\mathbb{X}_{\mathcal{L}})=1\\ \dot{\mathbf{y}} &= \mathcal{G}(v,\mathbf{y})\in\mathbb{X}_{\mathcal{N}}:dim(\mathbb{X}_{\mathcal{N}})=N\\ \mathbf{y_f} &= \mathcal{H}(\mathbf{y})\in\mathbb{X}_{\mathcal{M}}:dim(\mathbb{X}_{\mathcal{M}})=M\leq N\end{cases} \tag{1}$$

where $\mathbf{y_f}$ are *feedback variables* outputted by the embedded system through a proper function $\mathcal{H}(\mathbf{y})$. In the following, we can suppose $\mathbb{X}_{\mathcal{L}}\subseteq\mathbb{R}$, $\mathbb{X}_{\mathcal{N}}\subseteq\mathbb{R}^N$ and $\mathbb{X}_{\mathcal{M}}\subseteq\mathbb{R}^M$ for simplicity. Moreover, $0<\varepsilon\ll 1$ accounts for the relative speed of the neuron (fast) dynamics with respect to the embedded (slow) one.

In our work, the second order dimensionless FHN [14], [15] neuron model has been considered:

$$\begin{cases}\varepsilon\dot{v} &= f(v)-\theta+I\\ \dot{\theta} &= v+a-b\theta\end{cases} \tag{2}$$

where ε plays the same role as in Eq. (1), determining the slow and fast dynamics associated with the θ and v state variables respectively. Moreover, $a,b\in\mathbb{R}$ are constant parameters, $I\in\mathbb{R}$ is an external bias and $f(v)$ is a cubic function of the membrane potential. This nonlinearity can be modelled through a suitable PWL approximation, consisting in breaking the cubic curve into joined segments each having its own slope. Therefore, if the number of segments is $n+1$ and the i-th slope is m_{i-1}, then $f(v)$ can be replaced with:

$$\Pi_n(v)\doteq a_0+a_1v+\sum_{i=1}^{n}b_i\left|v-e_i\right| \tag{3}$$

where

$$\begin{cases}a_1=0.5(m_0+m_n)\\ b_i=0.5(m_i-m_{i-1})\\ a_0=f(0)-\sum_{i=1}^{n}b_i|e_i|\\ f(0)\in\mathbb{R}\\ e_i\in\mathbb{R}\end{cases} \tag{4}$$

with $i\in\{1,...,n\}$. Parameters e_i in Eq. (3) denote those membrane potentials at which slopes change; they are ordered so that $e_{i-1}<e_i$, $\forall i\,\{1,...,n\}$. Additionally, m_0 (m_n) refers to the slope of the leftmost (rightmost) segment. To properly model an FHN neuron, an approximated cubic function composed of $n+1=3$ segments has been designed adopting the following constraints:

$$\begin{cases}e_1<0<e_2\\ m_0,m_2<0\\ m_1>0\end{cases} \tag{5}$$

In this way, all the original existing fixed points and their properties are preserved and thereby the original FHN and its PWL approximation (denoted in the following as P-FHN) are topologically equivalent.

Therefore, the original FHN neuron in Eq. (2) becomes a P-FHN if:

$$f(v)=\Pi_2(v) \tag{6}$$

and then also a particular case of embodied model in Eq. (1), if we fix:

$$\mathbf{y_f}=\mathcal{H}(\mathbf{y})\equiv\mathbf{y}\equiv\theta \tag{7}$$

and thereby:

$$\mathcal{F}(v,\theta)=\Pi_2(v)-\theta+I \tag{8}$$

and

$$\mathcal{G}(v,\theta)=v+a-b\theta \tag{9}$$

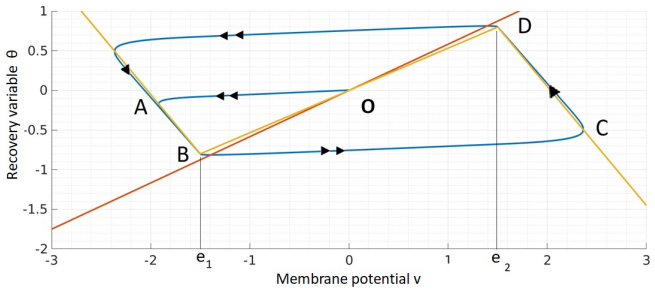

Fig. 1: Phase portrait of a P-FHN neuron. The abscissa of point B(D) corresponds to e_1 (e_2) in Eq.(5).

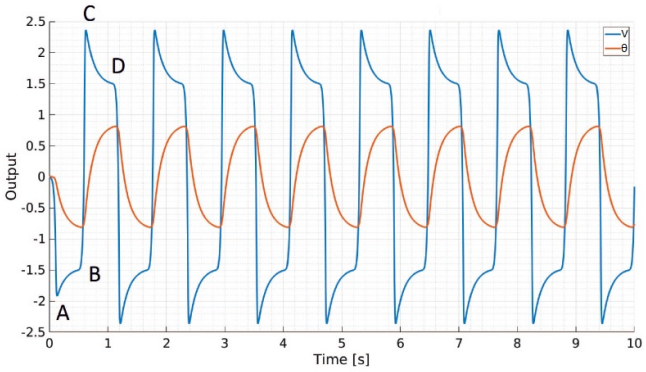

Fig. 2: Time evolution of the two P-FHN state variables.

III. The neuron-motor embodiment

The phase portrait of the P-FHN model is shown in Fig. 1 where the classical slow-fast limit cycle is depicted. Here, the role of each nullcline is evident. The intersection at the origin of the first (cubic-like PWL) and the second (linear)

nullcline, due to their relative slopes arrangement, assures the existence of an unstable fixed point [16]. Any trajectory starting nearby (e.g., point O in Fig.1) is repelled towards the outer sides of the phase portrait: the fast state variable (i.e., membrane potential v) imposes a nearly horizontal escaping direction to the trajectory until this one meets the left branch of the PWL nullcline (point A). From here the trajectory proceeds along this branch, obeying the decreasing flow of the recovery variable θ, meeting the extreme left position (point B). From here, trajectories are quickly pushed horizontally towards the right outer branch of the PWL nullcline (point C). Then they slowly proceed towards point D and so on. It is interesting to observe the corresponding trend of the two-state variables, depicted in Fig.2. This interplay between variables v and θ is at the basis of the onset of the spiking neural limit cycle dynamics in the P-FHN phase plane. From these considerations, it is herewith proposed to substitute the dynamics represented by the second nullcline with that one of an energy user, like an actuator. In this case, the second state variable θ could be the physical position assumed by the "user", leaving to the first variable v the role of the "supplier". This is congruent also with the linear and positive slope of the second nullcline in the phase plane (v, θ).

In the following, the proposed strategy is applied to a commercially available servo motor, although the whole approach can be adopted using other types of actuators, whose motion takes place at the expense of the energy released by the neuron.

A. Dynamixel MX-28AT modelling and embedding

Dynamixel motors are smart actuators, widely used in robotics, made up of an electromechanical part, constituted by a DC motor with a gear-based reduction system, and an MCU-based controller system. They are used in this paper both to demonstrate our embodiment concept and to indicate the most direct application through a phase-locking strategy on networks of embodied motor-neuron nodes.

Referring to Eq.(1), the scalar state variable and the associated PWL nonlinearity are associated with the v variable, and the function $\mathcal{F}$, respectively. A proper embedded system function $\mathcal{G}$, representing the actuator dynamics, has to be introduced.
A first attempt to obtain an accurate model for Dynamixel motors was proposed in [17], together with a list of key parameters reported in Table I, partially drawn from the motor data sheet and useful for our simulations [18]. In the following, we consider the electromechanical part and the PID controller embedded into the actuator, separately.

The former is given by a DC motor with reduction gear characterised by the following transfer function [17]:

$$\frac{\Theta(s)}{U(s)} = \frac{\tau \eta K_t}{s(as^2 + bs + c)} \tag{10}$$

where $a = L(J_l + J_m\tau^2\eta)$, $b = R(J_l + J_m\tau^2\eta) + Lb_m\tau^2\eta$ and $c = b_m\tau^2\eta R + \frac{K_t\tau^2\eta}{K_\omega}$.

In Eq. (10), $\Theta(s) = \mathcal{L}\{\mathbf{y_f}\}$, stands for the Laplace transform of $\mathbf{y_f}$ introduced in Eq.(1) and corresponds to the motor angular displacement at the load side of the motor gearbox, whereas $U(s)$ is the output of the PID controller output embedded into the servomotor.

A Dynamixel actuator, as a whole, can be controlled both in speed (wheel mode) and in position (servo mode). In our simulations and experiments, we employed the wheel mode control, whereas the information used in the neural sub-system is the motor position $\theta(t)$. The servomotor internal speed error function with respect to the speed reference signal $\omega_{\text{Ref}}(s)$ represents the input to the PID controller:

$$E(s) = \omega_{\text{Ref}}(s) - \omega(s) \tag{11}$$

Dynamixel motors cannot handle all possible input sources owing to the intrinsic limitations at the input stage. Let u_{min} and u_{max} be the minimum and maximum values for $u(t)$, respectively. Therefore, the PID processes the error $E(s)$ and provides a saturated output signal as reported below:

$$u(t) \doteq \min(\max(\mathcal{L}^{-1}\{Y_{\text{PID}}(s)\}, u_{\text{min}}), u_{\text{max}}) \tag{12}$$

In our case, the feedback variable corresponds to the motor position, $\mathbf{y_f}$ defined as:

$$\mathbf{y_f} = \mathcal{H}(\mathbf{y}) \equiv \theta \tag{13}$$

so that it holds:

$$\mathcal{F}(v, \theta) = \Pi_2(v) - \theta + \theta_0 \tag{14}$$

and

$$\omega_{\text{Ref}}(t) \doteq v(t) - b(\theta(t) - \theta_0) \tag{15}$$

where the parameter a in Eq.(2) corresponds to $a = b \cdot \theta_0$ and the position offset θ_0, corresponding to the bias I, reflects the average angular excursion:

$$\theta_0 = \frac{\theta_{\text{max}} + \theta_{\text{min}}}{2} \tag{16}$$

Furthermore, the middle PWL nullcline segment m_1 (BD in Fig.1) is bounded within the region delimited by the predefined break-points (e_1 and e_2 in Fig.1) and the corresponding angle limits θ_{max} and θ_{min}, resulting in:

$$m_1 = \frac{\theta_{\text{max}} - \theta_{\text{min}}}{e_2 - e_1} \tag{17}$$

In the following, the adopted PWL neuron model with an embedded Dynamixel motor will be referred to as *P-Dynamixel*. Fig.3 depicts a simple network of two embodied neurons, where the role played by the neural and motor sub-parts and their connections are depicted. The interaction between two motor-neuron units, based on a master-slave topology, is also reported and further detailed in Section III-B and V.

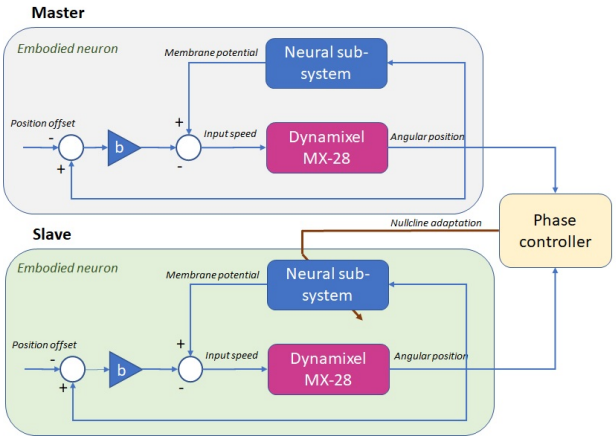

Fig. 3: Block scheme describing the connections between two embodied neurons when a master-slave topology is adopted. The position offset (θ_0) is a desired angular reference around which the motor oscillates.

Dynamixel MX-28 AT	
Parameter	*Value*
Reduction ratio τ	193
Gear efficiency η	0.836
Resistance R	8.3 Ω
Inductance L	2.03e-3 H
Inertia J_m	8.68e-8 kgm^2
Friction b_m	8.87e-7 Nm/(rad/s)
Torque constant K_t	0.0107 Nm/A
Motor back-electromotive constant K_ω	93.1 (rad/s)/V
$[u_{\min}, u_{\max}]$	$[-12, 12]$ V
Proportional gain K_p	4
Integral gain K_i	0
Derivative gain K_d	0

TABLE I: List of the parameters employed for simulations.

B. Networks of embodied neurons

We have introduced isolated, single embodied models in the previous section. However, the emergence of more interesting and complex phenomena arises when multiple units are networked together, as in the case of CPGs. Although many networking strategies are possible, in this context we assume that nodes are linked together in a master-slave topology, where the slave nodes are compelled to track the activity of the master node. Because of the master-slave setting, the resulting topology is a tree graph with directed edges from the master to every slave. Weights among nodes represent phase displacements with respect to the master unit.
Let Q be the number of uncoupled embodied units and suppose $\exists! \bar{m} \in \mathbb{I} \doteq \{1, ..., Q\}$ which corresponds to the index of the master node. Conversely, $\exists \mathbb{S} \doteq \{i \in \mathbb{I} | i \neq \bar{m}\}$ which is the set of all the indices referred to slave nodes. Thus, a concise Q-dimensional vector can be introduced to represent how the oscillators should be coupled in phase:

$$\mathbf{h} \doteq [\phi_1, \cdots, \phi_Q] \in [0, 2\pi]^Q \tag{18}$$

$\phi_{\bar{m}} = 0$ by definition, whereas all the other values correspond to phase displacements imposed to every slave node.

The problem of synchronization will be faced mainly from the experimental side. However, such important issues like stability and robustness are worth being investigated and are currently an active theme of research, above all given the embedded nature of the whole neuron motor system. In literature, among the different existing methods, the authors are currently investigating the application of partial contraction theory, as a suitable tool to formally and rigorously address stability in this family of embedded networked motor neuron architectures [19], [20].

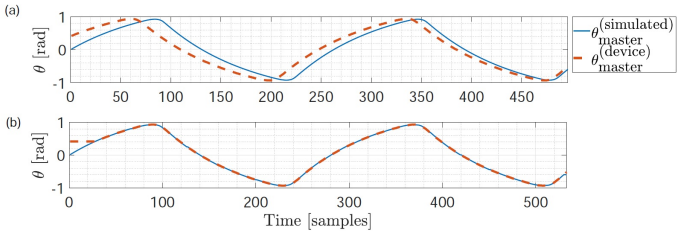

Fig. 4: Comparison between the angular positions of a simulated and a real master servomotor. Results before (a) and after (b) the parameter optimization phase, performed acting on the nominal reduction ratio, are reported.

IV. Description of the Experimental Setup

The embodiment framework illustrated so far was extensively assessed through several simulation campaigns. Subsequently, the corresponding experimental setup was designed and built. A preliminary identification phase was performed to assess if the motor model parameters duly matched those ones effectively embedded into the real motors. The results, reported in Fig.4 (a), show that the model output (Eq.10) and the real motor output showed the same dynamics, but a slightly different steady-state oscillation period. This discrepancy was compensated by adding a multiplicative factor $\Delta\tau = 1.813$ to the nominal reduction ratio τ. This optimised value was obtained through the Dynamic Time Warping (DTW) algorithm [21]. Results obtained with this correction are shown in Fig.4 (b).

Our experimental setup, depicted in Fig.5, comprises two nodes, each one consisting of a servomotor endowed with a rod (the longer one attached to the master), used as a visual reference for the motor oscillations. The motors are connected to an Arduino MEGA 2560 through a tri-state buffer DM74LS241N. Communication from and to the Arduino board is serial with a baud rate set to $1Mbps$. Because of the number of nodes, in this particular setup, Eq. (18) reduces to $\mathbf{h} = [0, \phi]$ if the first entry is related to the master.
In our experiments, the sampling time for the neural equation within the board was set to $t_s = 8ms$, by taking into account the time needed to integrate the motor dynamics. Moreover, the parameter ε in Eq. (1) was set to $\varepsilon = 0.025$.

V. Simulation and Experimental Tests

This section provides the experimental results mostly carried out with the hardware-based setup. In particular, four main scenarios were considered, referring to two different kinds of control actions, (taking place continuously or at specific time

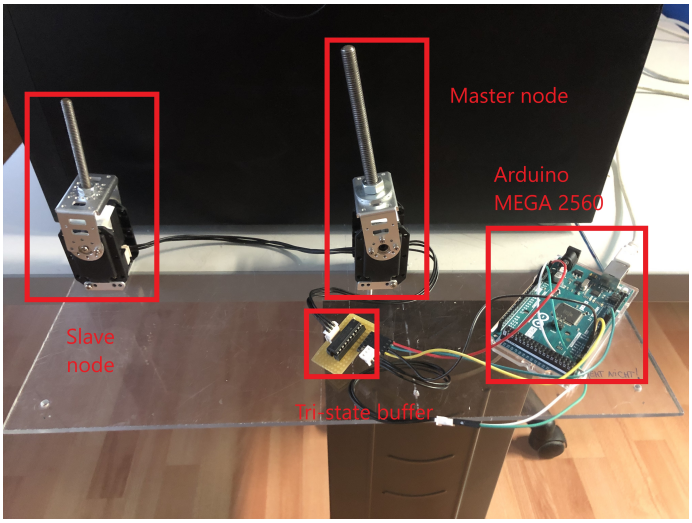

Fig. 5: Hardware-based setup for experimental tests.

instants) and two parametric configurations (i.e., identical and different) between the master and slave system. For the sake of simplicity, the zero-phase and anti-phase synchronization tasks were considered, even if the strategy can be applied to any desired phase shift between master and slave. A simple proportional rule over the master-slave phase error was found suitable in most cases to guarantee synchronization. However, in case of failure, a proportional-integral law was successfully adopted.

To address the synchronization issue even in presence of an appreciable difference in the oscillation period between master and slave, the strategy has consisted of changing the shape of the phase portrait and in particular, the most external slopes of the PWL nullcline, as in [12]. Since the slopes change the period in an affine form, the relationship slope-period is easily invertible; Slopes of nullclines have been modified in several manners. The continuous control action has taken the form of a proportional gain α applied over a pairwise phase error. In particular, slopes have incrementally changed according to the following updating rule:

$$m_{i,j\text{-th slave}} \leftarrow m_{i,\text{master}} + \alpha \times phase\ error_j \qquad (19)$$

where $m_{i,j\text{-th slave}}$ ($m_{i,\text{master}}$) is either the leftmost or rightmost slope of the j-th slave (master), thereby $i \in \{0, 2\}$ only and $j \in \mathbb{S}$, and:

$$phase\ error_j \doteq \arctan\left(\frac{\theta_{\text{master}}}{v_{\text{master}}}\right) - \arctan\left(\frac{\theta_{j\text{-th slave}}}{v_{j\text{-th slave}}}\right) - \phi_j \quad (20)$$

This equation defines the error as the vector difference between the positions of the master and the slave over the v-θ phase portrait, whilst ϕ_j is one of the entries in $\mathbf{h}$ given by Eq. (18). Eq. (19) can be applied either continuously or when specific events are triggered, but the update rule does not change formally. Although the relationship between nullclines and the resulting period of oscillation is a quite known topic [22], their precise manipulation in certain regions can be difficult: through the PWL approximation, nonlinear branches are linearised so that the resulting slopes, which are still related to the period of oscillation, are easily modifiable.

Table II lists the parameters employed for those experiments with networked P-Dynamixels with the same phase portrait (scenarios 1 and 3), whereas Table III reports those ones for P-Dynamixels with different phase portraits (scenarios 2 and 4).

Parameters for scenarios 1 and 3: master and slave with equal phase portraits	
Parameter	*Value*
Excursion $[\theta_{\min}, \theta_{\max}]$ [rad]	$[-0.8, 0.8]$
Break-points $[e_1, e_2]$	$[-1.6, 1.6]$
Initial slopes $[m_0, m_2]$	$[-2, -2]$
Gain b	0.8
ε	0.025
Measured period of the master T [s]	2.1956
Gain α	4

TABLE II: List of the parameters employed for both scenario 1 and scenario 3.

Parameters for scenario 2 and 4: master and slave with different phase portraits	
Parameter	*Value*
Excursion $[\theta_{\min}, \theta_{\max}]$ [rad]	$[-0.8, 0.8]$
Break-points $[e_1, e_2]$	$[-1.6, 1.6]$
Slopes of the master $[m_0, m_2]$	$[-4, -4]$
Initial slopes of the slave $[m_0, m_2]$	$[-1, -1]$
Gain b	0.8
ε	0.025
Measured period of the master T [s]	2.2931
Gain α	7.5

TABLE III: List of the parameters employed for both scenario 2 and scenario 4.

All the following scenarios will take into account one master and one slave, therefore, the subscript representing the phase of the slave system will be omitted.

A. Scenario 1: Continuous control action with equal phase portraits

Here, even if the master ad slave systems have the same parameters, phases may differ due to different initial conditions. Eq. (19) can be applied to adjust either one or both the external slopes of the slave node. For zero-phase synchronisation task ($\phi = 2\pi$), an example of real signals produced by two networked P-Dynamixels is shown in Fig.6. The effects of the proportional control action are sufficient to lead the slave activity to track the master because of the continuously changing of the leftmost slope (according to Eq. (19)). The lowest panel in Fig.6 shows the phase error which becomes negligible almost everywhere, except during the "jumps", occurring around the PWL corners.

Fig. 7 displays the acquired variables from our experimental setup in the anti-phase synchronization task ($\phi = \pi$). A longer transient is here needed before reaching the anti-phase condition. Despite being effectively always active and therefore capable of correcting errors at every time, energy costs can be high and even numerical problems may arise while updating the slopes.

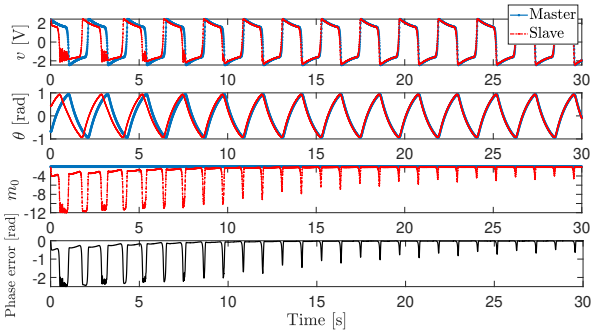

Fig. 6: Zero-phase synchronisation in real networked P-Dynamixels with equal phase portraits undergoing a continuous control action.

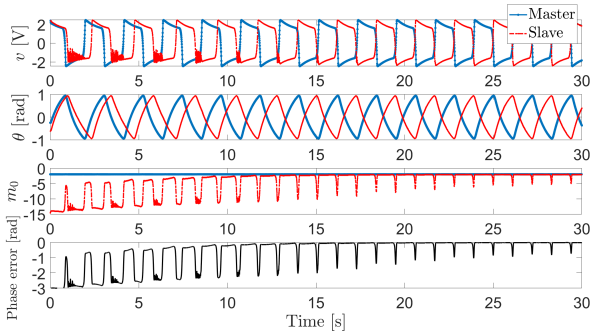

Fig. 7: Anti-phase synchronisation in real networked P-Dynamixels with equal phase portraits undergoing a continuous control action.

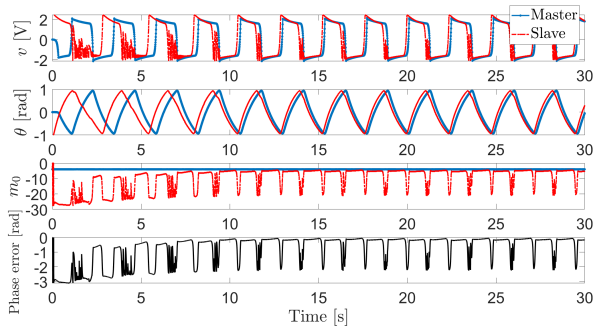

Fig. 8: Zero-phase synchronisation in real networked P-Dynamixels with different phase portraits undergoing a continuous control action.

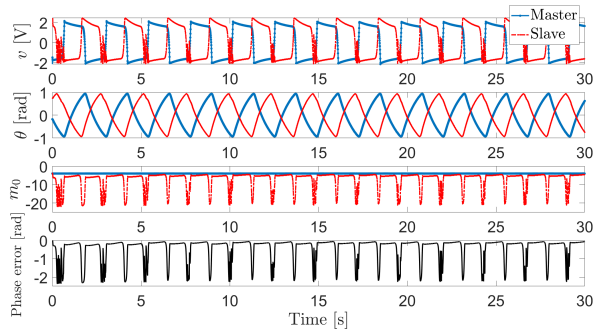

Fig. 9: Anti-phase synchronisation in real networked P-Dynamixels with different phase portraits undergoing a continuous control action.

B. Scenario 2: Continuous control action with different phase portraits

When the master and slave phase portraits differ, the control action ought to be sufficiently robust for proper compensation. To reproduce this disturbance in the node dynamics we modified some parameters with respect to the test performed before.

The signals acquired from the experimental setup for zero-phase synchronization are shown in Fig.8. Notwithstanding the discrepancy between the models, the control law action in Eq.(19) succeeds in reaching the master-slave synchronisation condition.

Fig.9 is an example test carried out to achieve anti-phase synchronisation. The largest error is present during the jumping interval. The effects of two distinct sets of PWL parameters are mostly clear over the membrane potential-like state variable.

C. Scenario 3: Event-driven control action with equal phase portrait

Differently from the continuous control case, to implement an event-driven strategy the control law takes place at specific time instants (events) where the neuron state variable changes its sign. In our experiments, the event was triggered by the falling edge of the neuron state variable, i.e. the time when the master state variable becomes negative [12]. In terms of computational costs and energy efforts, this corresponds to an energy-saving technique, which aims at performing phase-locking control only when needed, revealing less impactful than a continuously operative control as well as the previously presented strategy. Energy and time efficiency are sensible problems addressed in literature also with the introduction of memristive nanodevices for controlling a humanoid robot [23], [24].

Moreover, the event-driven control can contribute to saving computational cost and bandwidth in typical edge computing architectures. Here, when a distributed network of position-controlled actuators has to be synchronised with a unique master system, the event-driven control requires a master-slave communication only at specific time instants, whereas the continuous controller would implement a continuous master-slave communication.

However, the zero-phase synchronisation task, in this case, has required more time to be effective. Fig.10 shows the main variables in detail drawn from our real experimental setup. What is really remarkable is the way the controlled slope, and therefore the phase error, changes in time: in the case of an event-driven control, both these variables have behaved as step-like time-varying functions. The phase error is detected and compensated only at specific times.

We have tested the event-driven strategy with two networked systems in the anti-phase synchronisation regime too, obtain-

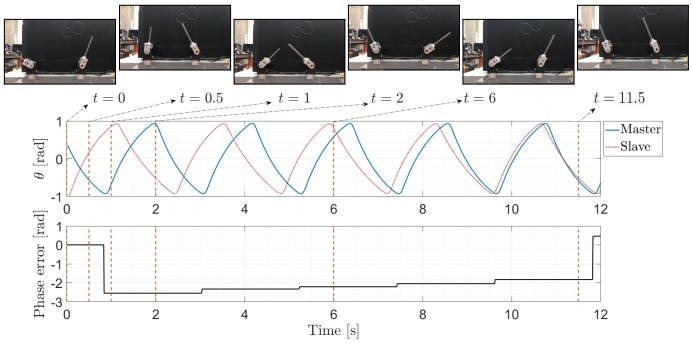

Fig. 10: Zero-phase synchronisation with an event-driven control through the real experimental setup, using two nodes with the same phase portrait. Snapshots show the real hardware setup: one master (longer rod) and one slave (shorter rod). A video is provided in the supplementary material.

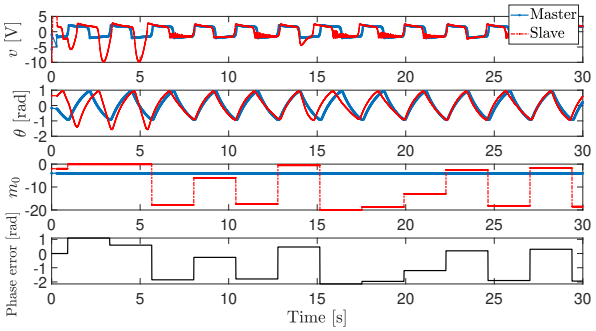

Fig. 11: Zero-phase synchronisation in real networked P-Dynamixels with different phase portraits undergoing an event-driven control.

ing similar results to those reported for the other cases.

D. Scenario 4: Event-driven control action with different phase portraits

To differentiate the nodes within the network, we have changed the external slopes of the master, as performed in Scenario 2.

Despite being a critical case, the event-driven approach has been capable of handling zero-phase synchronisation tasks easily, even though PWL functions have been supposed to differ. Fig. 11 graphically reports these results.

Anti-phase synchronisation tasks have shown to be quite affordable, no matter the approach. However, when having different PWL functions an event-driven control sometimes fails to guarantee the phase requirements. Fig.12 shows the results drawn from real experiments, which show difficulties in reaching and maintaining acceptable performance.

This suggests looking for more sophisticated control laws with respect to Eq.(19), but which would require a more powerful hardware setup. At this aim, we employed the simulated model, identified in Section IV.

Therefore we extended Eq.(19) to include an integral part, so that slopes are changed as follows:

$$m_{i,j\text{-th slave}} \leftarrow m_{i,j\text{-th slave}} + \alpha \times phase\ error_j + \varsigma \times time\ error_j \quad (21)$$

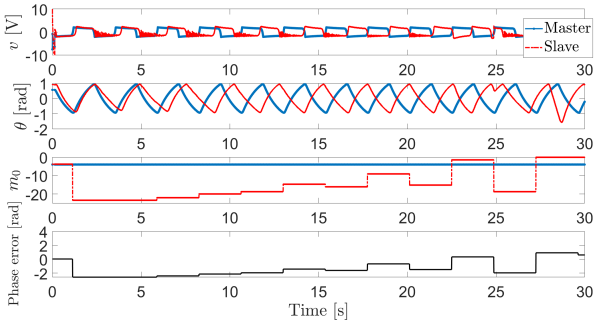

Fig. 12: Anti-phase synchronisation in real networked P-Dynamixels with different phase portraits undergoing an event-driven control.

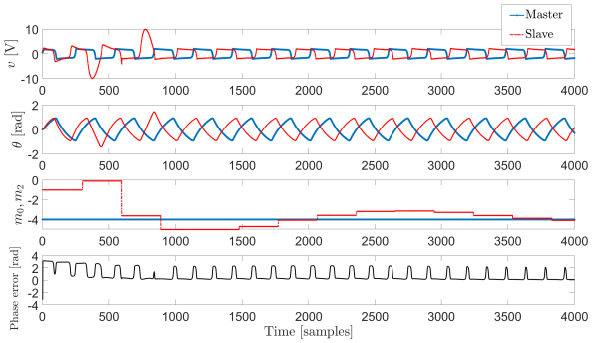

Fig. 13: Example of anti-phase synchronisation using the simulated model with an event-driven control strategy through the employment of Eq. (21).

where $time\ error_j \doteq \hat{T}_{\text{master}} - \hat{T}_{j\text{-th slave}}$ is the error between the period of oscillation computed when simulating Eq.(10) for the master and the slave systems. Two main differences are clear: instead of updating m_i of the j-th slave by applying a proportional action referred to the slope m_i of the master, Eq. (21) takes into consideration the previous value of m_i of the j-th slave itself so that the effect is integral. Additionally, another weighted term takes into account the relative difference of periods between the master and the j-th slave. Here we have chosen $\alpha = 0.8$ and $\varsigma = 10$. Moreover, control action has taken place twice, one for each external slope. In this way, even large phase differences were compensated.

Fig.13 depicts the results obtained applying Eq. (21) when handling an anti-phase synchronisation task with two simulated networked P-Dynamixels with different PWL functions. Even though the phase error has kept oscillating throughout the whole simulation, the additional effect due to the time error (Eq. (21)) was able to lead both the controlled slopes of the slave towards the reference ones given by the master.

It has to be remarked that the event-driven control strategy is less energy demanding but typically requires a longer time to achieve synchronization than the continuous control law.

VI. CONCLUSIONS

In this paper, a new bio-inspired motor neuron embodied system oriented to networked systems synchronization and, in particular, to locomotion control has been introduced. The original idea is to host the actuator dynamics directly within, and as a part of, the neuron dynamics. At this aim, a PWL representation of an FHN neuron was adopted, where the dynamics can be directly and easily controlled in the phase plane, through the modulation of the whole embodied system nullclines. The embodiment of the motor dynamics is realised through the inclusion of the dynamics of the actuator within the phase plane of the neuron, substituting part of the neural dynamics with part of the actuator one. This led to a reduction in the overall dynamical order of the neuron-motor control system. The actuator taken into account in this paper refers to the well known Dynamixel servomotor family, widely used in robotics. They were here experimentally employed to generate oscillatory motion in an embodiment framework, under the direct control of PWL approximated nonlinear oscillators. Results confirm that, like our case of actuators, it is possible to include multiple, not strictly neural, dynamics within a neural sub-system to enhance it and provide more functionalities. The strategy was demonstrated both via a simple experimental setup and using a model of the actuator, embedded into the neuron dynamics for synchronization purposes. The model was suitably tuned to match the experimental outcome using an optimization algorithm. Being the whole work oriented to the locomotion control of legged machines, the results obtained open the way to novel, reduced-order and efficient ways for adaptive, light and actuator feedback-based neural pattern generators for bio-inspired robots. In virtue of the results achieved it has been outlined that the strategy adopted is quite general and can be extended to different families of actuators or, in general of embedded dynamics. Moreover, these achievements can open new methods for embodied control in different application fields. The direct one regards highly adaptive locomotion controllers for legged machines. Also, the PWL embodied neuron can be easily integrated for large scale microactuator devices which could be able to act as adaptive space distributed smart structures (adaptive MEMS). In addition, learning strategies could be added to the proposed strategy to enhance the adaptive capabilities beyond a simple proportional/integral control. The capability to react to even high parametric changes can lead to new families of compliant devices able to adapt to the dynamics of the actuator.

ACKNOWLEDGMENT

The authors acknowledge Eng. Andrea Bonanzinga contribution in the earliest implementation of the experimental setup.

REFERENCES

[1] E. Arena, P. Arena, and L. Patanè, "CPG-based locomotion generation in a drosophila inspired legged robot," in *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, June 2012, pp. 1341–1346.

[2] Y. Fukuoka, Y. Habu, and T. Fukui, "A simple rule for quadrupedal gait generation determined by leg loading feedback: a modeling study." *Scientific Report*, vol. 5, p. 8169, 2015.

[3] Z. Gan, T. Wiestner, M. A. Weishaupt, N. M. Waldern, and C. David Remy, "Passive dynamics explain quadrupedal walking, trotting, and tölting," *Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 2, 08 2015.

[4] M. Schilling and H. Cruse, "Decentralized control of insect walking: A simple neural network explains a wide range of behavioral and neurophysiological results," *PLoS Comput Biol*, vol. 16, no. 4, p. e1007804, 2020.

[5] P. Arena, A. Bonanzinga, and L. Patanè, "Role of feedback and local coupling in cnns for locomotion control of a quadruped robot," in *CNNA 2018; The 16th International Workshop on Cellular Nanoscale Networks and their Applications*, 2018, pp. 1–4.

[6] D. Owaki, M. Goda, S. Miyazawa, and A. Ishiguro, "A minimal model describing hexapedal interlimb coordination: The tegotae-based approach," *Frontiers in Neurorobotics*, vol. 11, p. 29, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2017.00029

[7] S. Suzuki, T. Kano, A. J. Ijspeert, and A. Ishiguro, "Sprawling quadruped robot driven by decentralized control with cross-coupled sensory feedback between legs and trunk," *Frontiers in Neurorobotics*, vol. 14, p. 116, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2020.607455

[8] A. Fukuhara, Y. Koizumi, S. Suzuki, T. Kano, and A. Ishiguro, "Decentralized control mechanism for bodyâ€"limb coordination in quadruped running," *Adaptive Behavior*, vol. 28, no. 3, pp. 151–164, 2020. [Online]. Available: https://doi.org/10.1177/1059712319865180

[9] S. Aoi, P. Manoonpong, Y. Ambe, F. Matsuno, and F. Wörgötter, "Adaptive control strategies for interlimb coordination in legged robots: A review." *Front Neurorobot*, vol. 11, 2017.

[10] S. Danner, S. Wilshin, S. N.A., and R. I.A., "Central control of interlimb coordination and speed-dependent gait expression in quadrupeds." *J Physiol.*, vol. 594, no. 23, pp. 6947–6967, 2016.

[11] E. del Rio and M. G. Velarde, "A prototype 2n-legged (insect-like) robot. A non-linear dynamical system approach," in *Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots II, An Insect Brain Computational Model*, 2014, pp. 123–149.

[12] P. Arena, L. Patané, and A. G. Spinosa, "A nullcline-based control strategy for pwl-shaped oscillators," *Nonlinear Dynamics*, Jun 2019.

[13] A. Spanne, P. Geborek, F. Bengtsson, and H. Jarntell, "Spike generation estimated from stationary spike trains in a variety of neurons in vivo," *Front. in Cellular Neurosci.*, vol. 8, p. 199, 2014.

[14] R. FitzHugh, "Mathematical models of threshold phenomena in the nerve membrane," *The bulletin of mathematical biophysics*, vol. 17, no. 4, pp. 257–278, Dec 1955.

[15] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2061–2070, Oct 1962.

[16] S. Coombes, "Neuronal networks with gap junctions: A study of piecewise linear planar neuron models," *SIAM Journal on Applied Dynamical Systems*, vol. 7, pp. 1101 – 1129, 2008.

[17] M. R. Maximo, C. H. Ribeiro, and R. J. M. Afonso, "Modeling of a position servo used in robotics applications," in *XIII Simpósio Brasileiro de Automação Inteligente*, 2017, pp. 2032–2038.

[18] Robotis. (2021) E-manual for dynamixel servomotor. [Online]. Available: https://emanual.robotis.com/docs/en/dxl/mx/mx-28/

[19] W. Wang and J. Slotine, "On partial contraction analysis for coupled nonlinear oscillators," *Biol Cybern*, vol. 92, p. 38–53, 2005.

[20] E. Arena, P. Arena, and L. Patanè, "Efficient hexapodal locomotion control based on flow-invariant subspaces," in *18th IFAC World Congress*, vol. 44, no. 1, 2011, pp. 13 758–13 763. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016458350

[21] K. Paliwal, A. Agarwal, and S. Sinha, "A modification over sakoe and chiba's dynamic time warping algorithm for isolated word recognition," *Signal Processing*, vol. 4, pp. 329–333, 1982.

[22] J. Keener and J. Sneyd, *Mathematical Physiology*. Springer, 2009.

[23] A. Ascoli, D. Baumann, R. Tetzlaff, L. O. Chua, and M. Hild, "Memristor-enhanced humanoid robot control system - Part I: Theory behind the novel memcomputing paradigm," *International Journal of Circuit Theory and Applications*, vol. 46, no. 1, pp. 155–183, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cta.2431

[24] D. Baumann, A. Ascoli, R. Tetzlaff, L. Chua, and M. Hild, "Memristor-enhanced humanoid robot control system - Part II: Circuit theoretic model and performance analysis," *International Journal of Circuit Theory and Applications*, vol. 46, no. 1, pp. 184–220, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cta.2430