Central Pattern Generator Control of a Differential Wheeled Robot

Ionuţ Urziceanu, Fernando Herrero-Carrón, Juan González-Gómez, Mircea Niţulescu, *Member, IEEE*, Francisco de Borja Rodríguez, Pablo Varona

***Abstract*—Bio-inspired central pattern generator approaches have been used for robotic autonomous locomotion mainly in legged, serpentine and worm like robots. In this paper we propose a new bio-inspired central pattern generator (CPG) neural circuit to control a differential wheeled robot. The CPG uses bursting neurons coupled through time evolving synapses, which endows them with rich rhythm negotiation capabilities. The overall circuit activity produces robust yet flexible rhythms. We show that this CPG can produce sequential command signals to rhythmically control the motion of the differential wheeled robot in a situation where the wheels can only partially rotate.**

I. Introduction

CENTRAL pattern generators are neural circuits that autonomously generate signals that are used to produce rhythmic motion. Recent research has shown that their neurons have rich intrinsic oscillating dynamics and that their connection topology is built to negotiate their overall pattern of activity as a function of environmental conditions and external input from the central nervous system [1]. Inspiration from these biological neural circuits has been used to propose solutions for robot locomotion [2].

While CPG approaches have been used extensively to control legged, serpentine and worm like robots, it has not been widely explored in the context of wheeled robots. This is mainly because wheeled robots are usually thought to require a continuous control of the wheels [3] while CPG circuits produce rhythmic motion. In this paper we show how rhythmic motion can be an adequate strategy to control a wheeled robot in the case where the wheels can only partially rotate.

II. Differential Wheeled Robot

A differential wheeled robot is a mobile robot whose locomotion is based on two (or more) wheels being controlled independently. The relationship between the rotation of each wheel determines the type of locomotion. In our case we use the SkyBot, which is a differential drive platform built for educational purposes. The customizable Skybot [4] (Fig. 1) is a differential drive robot composed of printable wheels and two hobby servos. The robot uses two electronic control boards called SkyPic [5] and Sky293 [6].

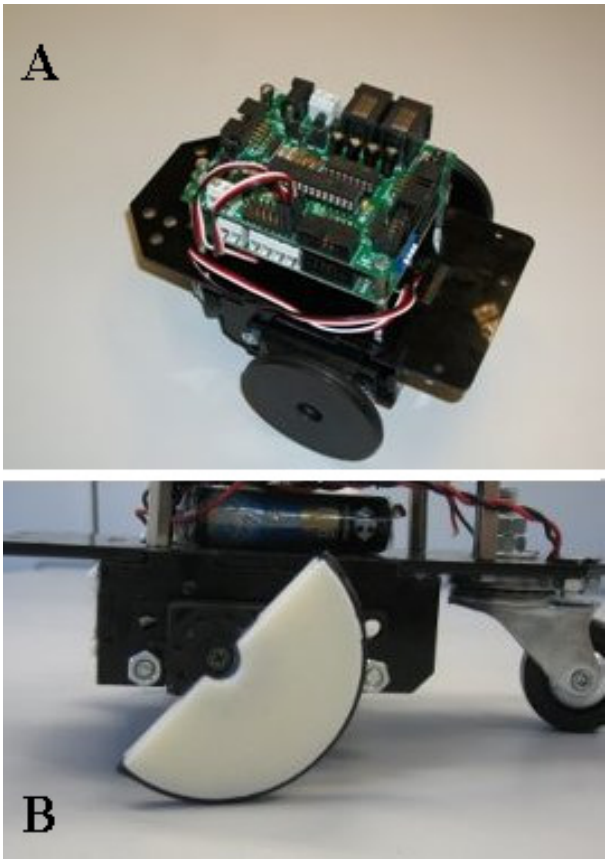

Fig. 1. A: Skybot: Overview of the differential drive robotic platform. B: SkyBot: Limited wheel detail (half-wheel).

The SkyPic is a minimal design with only the necessary components for controlling the robot. It includes an 8-bit PIC16F876A microcontroller, headers for connecting the servos, an I2C bus for additional communication, serial connection to the PC, a test LED and a switch for resetting the circuit. The Sky293 board [6] is an extension board that was used to accommodate the sensors. We have considered the case in which the wheels of the robot cannot perform complete rotations: this limitation can be attributed to the actuators that are used (linear motors, SMAs, broken gearboxes, etc), to the environment (the wheel does not have enough space to rotate) or to the actual shape of the wheel (the wheel may be broken). Their movements are thus confined in an angular interval, let us say $[-\varphi_{max}, \varphi_{max}]$. In this case, a new locomotion principle can be applied such that the robot can still navigate, but with reduced mobility.

Manuscript received June 17, 2011. This work was supported by an Erasmus grant and MICINN BFU2009-08473, TIN-2010-19607.

FHC, FBR, PV. Authors are with Grupo de Neurocomputación Biológica (GNB), Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid. (email: pablo.varona@uam.es)

JGG is with the Robotics Lab, Universidad Carlos III de Madrid.

IU, MN are with the University of Craiova, Romania (email: urziceanuionut@yahoo.com)

To analyze the movement, we consider the kinematic model of a differential drive robot with a castor wheel [7]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \theta \end{bmatrix} = \begin{bmatrix} \frac{R}{2}(\dot{\varphi}_1 + \dot{\varphi}_2)\cos(\frac{R}{L}(\varphi_1 - \varphi_2)) \\ \frac{R}{2}(\dot{\varphi}_1 + \dot{\varphi}_2)\sin(\frac{R}{L}(\varphi_1 - \varphi_2)) \\ \frac{R}{L}(\varphi_1 - \varphi_2) \end{bmatrix} \quad (1)$$

where terms x, y and θ are the robot coordinates and orientation relative to the global reference frame, φ_1, φ_2 are wheel angles and, in our case, control input signals, R and L correspond to real robot parameters: R is the wheel radius and L is the distance between the two wheels (base width). The locomotion restriction consists in that each wheel can only perform an oscillatory movement between the above given angle values. By coordinating the oscillatory movements of the wheels, three possible movements can be found:

1. If the wheels oscillate in phase (phase difference is 0°), i.e., in the same direction, with the same frequency and at the same time, the robot will move forward and backward repeatedly.
2. If the wheels oscillate in anti-phase (phase difference is 180°), i.e. in opposite directions, with the same frequency and at the same time, the robot will pivot left and right around a vertical axis (Instantaneous Center of Rotation).
3. If the wheels oscillate with a phase difference between 0 and 180°, the movement will be a combination of four movements: turning to one side, go forward, turn to the other side, go backward. We call "step" the length in a straight line of the robot's trajectory during a sequence of these four movements. It has been shown that the largest step value is obtained for a phase difference of 90° [7].

Other parameters that describe the motion are:

1. Initial phase (φ_0): determines the initial robot orientation relative to the path. It has no effect on the locomotion when the robot is travelling along its path.
2. Amplitude: the rotation angle interval for each wheel, it has an effect on the step size.
3. Offset (O): positive or negative angular value that's added to the initial wheel angle such that the wheel's resting position is not φ, but φ+O. The maximum rotation angle is $\varphi_{max} - |O|$.
4. Direction (Γ) is the angle measured between the initial robot orientation and a new orientation, after changing the offset values. This depends on the physical robot parameters, wheel radius and base width. Let R and L be the wheel's radius and robot's base width, respectively. From the equations of the robot's kinematic model, the orientation angle can be calculated as:

$$\Gamma = \frac{R}{L} \cdot O \quad (2)$$

A full description of the locomotion principle can be found in [7]. A video demonstration of this principle can be found at http://goo.gl/2qv6V.

III. Central Pattern Generator

Given the locomotion principle, the first step in designing a central pattern generator is to select the building blocks and the topology that are to be used for the neural circuit. Moreover, one must specify the means in which the information is encoded in the activity of the CPG. Three elements are needed in order to build a complete description: neurons, synapses and motoneurons to drive the servos [8].

A. Neuron model

We use a neuron model developed by Rulkov et al. [9] that mimics the activity of living bursting neurons. The model is computationally efficient and the possible set of behaviors can be controlled depending on the selection of a few parameters [8]. Three stable regimes may be selected by combination of its parameters: silent, in which the potential of the neuron remains in a constant resting state; tonic spiking, in which the neuron produces spikes at a constant rate, and tonic bursting, in which bursts of spikes are produced at a constant rate, with a silent interval in between. Furthermore, in the boundaries of the parametric regions of those regimes, chaotic behavior may be found [10]. For this study we will set the parameters of the neurons to work in tonic bursting regime. See figure 2 for an overall idea of the model working in this regime.

The bursting regime of the model presents a slow wave (slow time scale) with fast spikes of activity sitting on top of it (fast time scale). We use the slow time scale (y_n in (3)) to encode movement duration, i.e., the temporal length of the burst defines the temporal length of the movement, and the fast time scale (x_n in (3)) to define angular velocity of the servo (higher frequency of the spikes corresponds to faster servo movements). The mathematical description of Rulkov's model as used in this work is as follows:

$$\begin{cases} x_{n+1} = f(x_n, y_n + \beta_e \cdot I_n) \\ y_{n+1} = y_n - \mu \cdot (x_n + 1) + \mu \cdot \sigma + \mu \cdot \sigma_e \cdot I_n \end{cases}$$

$$f(x_n, y_n) = \begin{cases} \frac{\alpha}{1 - x_n} + y_n, & x_n < 0 \\ \alpha + y_n, & 0 \le x_n < \alpha + y_n \\ -1, & otherwise \end{cases} \quad (3)$$

with μ = 0.001 in all experiments. This is a bi-dimensional model, where variable x_n repre-sents a neuron's membrane voltage and y_n is a slow dyna-mics variable with no direct biological resemblance, but with similar meaning as gating variables in biological models that represent the fraction of open ion-channels in the cell. While x_n oscillates on a fast time scale, representing individual spikes of the neuron, y_n keeps track of the bursting cycle, a sort of context memory. Units are dimensionless, and can be rescaled to match the requirements of the robot. The combination of σ and α selects the working regime of the model: silent, tonic spiking or tonic bursting.

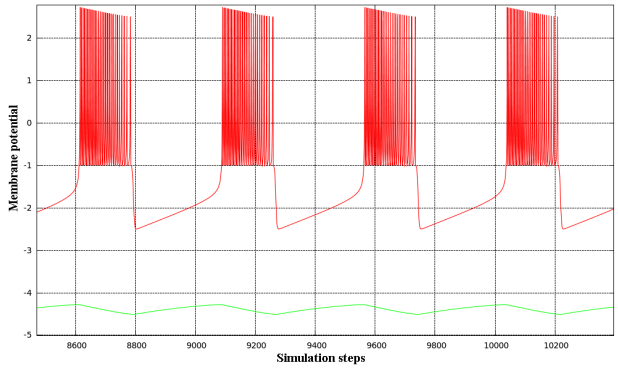

Fig. 2. Tonic bursting regime of a Rulkov neuron model. X_n is plotted in red, representing the fast subsystem; y_n is plotted in green, representing the slow subsystem. Parameters used in this simulation: $\alpha = 7$, $\beta_e = 0$,$\mu = 0.001$, $\sigma = -0.1$, $\sigma_e = 1$.

In the bursting regime, these parameters also control several properties of neural activity. Finally, external input is modeled through I_n. Depending on this value, a neuron will modify its behavior. For instance, an external stimulus, e.g. a sensory signal, may be input using this parameter. This property is essential for autonomous organization: processing units in the CPG must be able to negotiate the rhythm among them. Also, entrainment between the CPG and the physical robot can be achieved through I_n by adding an error term as external input to a neuron [11]. The total effect of this parameter will depend upon past history of events, the exact value of I_n and the phase within the burst cycle at which the neuron finds itself. In our work, I_n is the current flowing from one neuron to another: a periodic sampling of the continuous function described below in (5).

B. *Kynetic synapse model*

A key property of CPGs is that they are autonomous and the different units in the circuit talk to each other to negotiate the overall function. Here we present the model we have chosen to implement synapses, the communication channel of neurons. In this work we use a chemical synapse model [12]. Chemical synapses are unidirectional. When a potential spike arrives from the presynaptic neuron, the synapse releases a certain amount of neurotransmitter molecules that bind to the postsynaptic neuron's receptors. With time, neurotransmitter molecules begin to unbind. If a succession of spikes arrives within a short time, the synaptic response to each of them may overlap. Therefore the state of the synapse is dependent upon past events, a mechanism of context memory [8]. The additional time-scale provided by kinetic synapses in a CPG enriches synchronization between bursting neurons. For instance, we may choose to synchronize two bursting neurons upon the spike (fast) time scale or the burst (slow) time scale. We have selected the kinetics of the binding and unbinding processes such that synapses act as filters of the fast time scale and synchronization occurs at the slow time scale. That is, the basic unit of synchronization will be the burst as a whole, not every individual spike. Beyond this, synapses may introduce delays for finer control of phase difference between neurons. The mathematical description of the model follows:

$$\dot{r} = \begin{cases} \lambda[T]\cdot(1-r) - \beta\cdot r, & t_f < t < t_f + t_r \\ -\beta\cdot r, & otherwise \end{cases} \quad (4)$$

This equation defines the ratio of bound chemical receptors in the postsynaptic neuron, where r is the fraction of bound receptors, and are the forward and backward rate constants for transmitter binding and $[T]$ is neurotransmitter concentration. The equation is defined piecewise, depending on the specific times when the presynaptic neuron fires (t_f): during t_r units of time, the synapse is considered to be releasing neurotransmitters that bind to the postsynaptic neuron. After the release period, no more neurotransmitter is released and the only active process is that of unbinding, as described by the second part of the equation. Times t_f are determined as the times when the presynaptic neuron's membrane potential crosses a given threshold. Synaptic current is then calculated as follows:

$$I(t) = g_{syn}\cdot r(t)\cdot(x_{post}(t) - E_{syn}) \quad (5)$$

where $I(t)$ is postsynaptic current at time t, g_{syn} is synaptic conductance, $r(t)$ is the fraction of bound receptors at time t, $x_{post}(t)$ is the postsynaptic neuron's membrane potential and E_{syn} its reversal potential, the potential at which the net ionic flow through the membrane is zero. When coupling two Rulkov map neurons we will need to use a discrete synaptic function. We will build a sequence, let us call it I_n, by simulating $I(t)$ as a continuous function and then taking samples every 0.001 time units. We say that a synapse is excitatory when the probability of the postsynaptic neuron firing a spike increases after the presynaptic neuron has fired. If the probability decreases, the synapse is inhibitory. If the postsynaptic neuron rhythmically emits spikes, an excitatory synapse will generally increase its frequency while an inhibitory one will generally decrease it.

C. *Motoneurons*

Movement information is robustly encoded in the neurons' bursting episodes. A neuron called motoneuron is then responsible of decoding this information and translating it into the signal that will finally be sent to the servo controller. This signal tells the angle at which the servo should be positioned, in degrees. With a slightly modified version of the motoneuron model provided in [8], we have tried to mimic the real transformation occurring between living motoneurons and muscles. Motoneurons read the activity of other neurons using a simple threshold function that equals 1 if the membrane potentials of their corresponding neurons exceed the threshold values; otherwise the function equals 0. The role of this function is to detect individual spikes of neurons. By setting the threshold to, for example, $\upsilon = -1.5$ a.u., this function applied to the potential trace of one neuron will have value 1 during individual spikes and 0 otherwise. In this way, communication between neurons is event-based. That is, the actual shape of neural activity is not so important, only their timing is. This can be a mechanism that the

nervous system employs to lower the impact of noise [1], [13].

The role of motoneurons is now to integrate the individual events emitted by each one of the neurons. If a neuron emits a spike, the motoneuron will move the servo a little bit in a positive or a negative angle, depending on the promotor or remotor effect that is intended (correspondently). If it emits a second spike close enough to the first one, the servo will be positioned a little bit further. If the neurons that are connected to the motoneuron are silent, the motoneuron will slowly drive the servo to a resting position of angle 0. This is accomplished through the following equation governing motoneurons in our CPG:

$$\dot{\varphi}_{1,2} = \dot{m}_{1,2} = C(t) - m_{1,2}(t) + O \qquad (6)$$

where $m_{1,2}(t)$ represents the output signal of the motoneurons, O is an offset value and $C(t)$ is a function described by:

$$C(t) = \gamma \cdot (a_1 \cdot s(t,\nu) + a_2 \cdot s(t,\nu) + \ldots + a_n \cdot s(t,\nu)) \qquad (7)$$

where γ is the amplitude of the signal, $s(t,\upsilon)$ is the threshold function described above and terms $a_1, a_2, \ldots a_n$ describe the effect of the neurons over the motoneuron: if $a_i = 1$, neuron i has a promotor effect on the corresponding motoneuron, if $a_i = -1$, neuron i will have a remotor effect on the corresponding motoneuron. Given the characteristics for this type of locomotion, the motion can be controlled by means of three parameters: amplitude - to modify the step, offset - to modify the direction of movement and frequency - to modify the speed. Because the locomotion is composed of coupled oscillatory movements, it is possible to design a CPG that can generate the appropriate motor commands to drive the robot.

The motion is split in four sequences:

1. The wheels start by rotating in opposite directions with the same speed, offset and amplitude;
2. Both wheels rotate forward;
3. Both wheels rotate in opposite directions, but contrary to sequence 1;
4. Both wheels rotate backward.

The next design step is to identify what information needs to be encoded in the activity of the CPG. After that, a dynamical invariant must be defined, i.e. a stable, reproducible, repeatable neural pattern that is based on an activation sequence. As mentioned above, offsets, amplitude and speed of movement must be encoded. Also, a constant phase difference between 0 and 180° must be maintained in order to achieve a steady locomotion.

The proposed CPG model comprised of a group of four neurons is considered (see figure 3). To drive the wheels, we need two motoneurons. The main idea of the model is that all four neurons will trigger in a specific non-overlapping sequence. The dynamical properties of neurons and synapses together with the topology of the circuit produces a coordinated alternating rhythm. Each of them will have a promotor or remotor effect on the motoneurons in such way

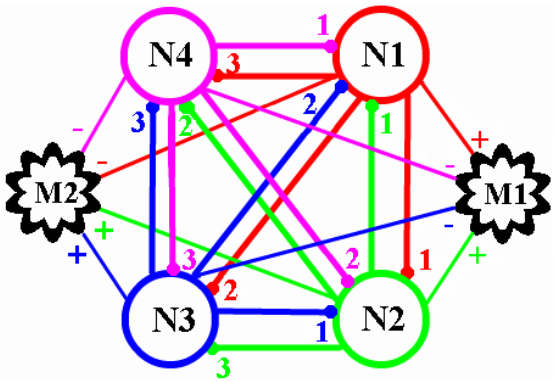

Fig. 3. CPG topology: the neural circuit is made of four neurons: N1, N2, N3, N4, two motoneurons: M1, M2 and twelve chemical synapses, each of them being identified by color and number. The arithmetic signs indicate the promotor or remotor effects that neurons have on the motoneurons. For forward locomotion, synapses red-1, green-3, blue-3, magenta-1 must have a smaller coupling strength compared to the others (See figure 5 for specific values).

that the motion sequences are obtained. The motoneurons are connected to all neurons of the model. Finally, the CPG will self-organize so that at any moment, each motoneuron will receive a signal from only one neuron to drive the wheels.

The angle values for the wheels will be encoded in the spiking activity of each neuron. Each spike represents a small increment in the wheel angle. The motoneurons then integrate the spikes and output the actual angle value, which is represented by m(t) in (6). Therefore, amplitude can be controlled by changing the number of spikes per burst period. The larger the number, the greater the wheel angle.

Amplitude can also be tweeked by changing the γ parameter in (7). The speed of the movement is controlled by altering the burst periods. A longer burst will translate into a slower evolution of the motoneuron signal.

Offsets are also encoded in the motoneurons, using the "*O*" parameter in (6), such that the starting angle values are not zero, but *O*. By adjusting the offset values, one can change the trajectory of the robot. It is worth noting that in all cases, the robot will move in a straight line, regardless of the values of the offsets and the amplitudes. By changing the offset values, we change only the robot orientation.

Finally, the most important parameter encoded in the CPG is the phase difference that establishes the coordination between the wheels. In a more intuitive approach, phase difference is related with the time delay between the moment at which one wheel starts moving and the moment at which the second wheel starts moving. By measuring the angle difference between the wheels in this time interval, we can obtain the phase difference. In this CPG model, the desired phase difference is achieved through the activation sequence of the neurons which, in term, is obtained using inhibition and asymmetric coupling. Inhibition is closely related to the synaptic conductance (g_{syn} in (5)).

Depending on the sign of the synaptic current, the slow subsystem (see equation (3)) is either increased (case of an excitatoy synapse), causing a burst, or decreased (case of an inhibitory synapse), triggering an early resting state of the neuron.

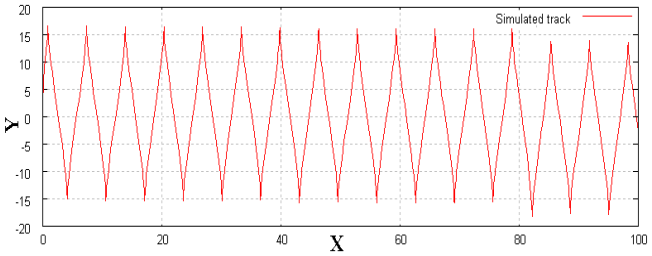

Fig. 4. Simulated trajectory of the differential drive using motoneuron-alike motor signals (sine signals) that are phase-shifted by 90°. The simulation has been carried in Matlab/Simulink, with a signal amplitude of 90°. Wheel radius is 55 mm, base width is 103 mm.

For a clearer example of how phase difference is obtained, let us consider this activation sequence: N1→N2→N3→N4. In figure 3, the network is completely connected, i.e., each neuron is connected to all others. Also, each neuron is connected to both motoneurons and can simultaneously drive them. If all synapses have equal coupling strengths, the neurons will fire depending on which of them gets the chance to fire first. What we can do is encourage some neurons to fire before others in order to obtain the desired sequence and we do this by adjusting coupling strengths. For the above chain of activation, synapses from the forward path (red-1, green-3, blue-3, magenta-1 in figure 3) must have a weaker inhibitory effect than all other synapses. Correspondingly, neuron N2 will be favored to burst before others (N1, N4), then N3 and so on. Therefore, we will end up with synapses that have a powerful inhibitory effect and synapses that have a weaker inhibitory effect. If all neurons share the same parameters (except for initial starting conditions), all strong synapses share a certain coupling strength and all weak ones share another coupling strength, the period of the whole CPG will be composed of four equal regions in which each one is a burst period of a corresponding neuron.

As it can be seen in figure 5, during the activation period of N1, the motoneuron will be driven in order to increase the output signal (wheel angle), whereas an activation of N3 will decrease the motoneuron's output signal. Considering promotor and remotor neuron effects on motoneurons and given the fact that a complete wheel oscillation takes one CPG period and the delay between wheel movements are one burst region wide, a phase difference of 90° between the wheels is obtained. More intuitively, when one wheel has reached a quarter of its oscillation amplitude, the second wheel will begin to rotate. With this model, the phase difference cannot be changed due to the non-overlapping activation sequence; it is fixed to approximately 90°.

To change the direction of movement between moving forward and moving backwards, the activation sequence is changed by changing coupling strengths. For moving forward, the sequence is N1→N2→N3→N4. For moving backward, the sequence is N1→N4→N3→N2. To steer the robot, the wheel offsets and the amplitude have to be changed, so that $O + A \leq |\varphi_{max}|$, where φ_{max} is the maximum rotation angle (clockwise or anti-clockwise) that the joint can achieve.

IV. Driving the Robot

Due to the relatively modest hardware resources of the PIC16F876A, the CPG software model has been implemented on a personal computer. The simulation of the CPG is done on this computer and the outputs (i.e. angle values of the wheels) are sent via serial interface to the robot which commands the two servos. No real entrainment has been implemented between the CPG and the actuators. In an ideal case, the CPG would wait for the actuators to position to the specified positions; in our case, due to the fact that we did not have feedback from the actuators (we used standard hobby servos), the CPG simulation is halted; by taking into consideration the number of iterations between two motor commands that are sent to the robot and the type of the servo, we estimate the servo response time (let us call it *ts*). Therefore, the computer will wait *ts* units of time before sending a new motor command to the robot.

A simplified control interface has been implemented to drive the robot forward, backwards, left and right and stop it. Forward and backward locomotion are obtained just by changing the coupling strengths of particular synapses which changes the activation sequence of the neurons as described above. Left and right locomotion is achieved by altering the offset values of the motoneurons according to (7). Due to the fact that the offset and the amplitude of the movement are related, such that $\varphi_{max} \leq \gamma + O$, one must adjust the amplitude value. The main control program that is running on the PC receives commands from the user and changes the CPG parameters accordingly.

A first approximation of the robot trajectory, using the above described locomotion principle has been done by implementing an idealized kinematic model of a differential drive robot. By numerically integrating (1), we can obtain the robot positions and orientation relative to a global reference frame (see figure 4). Figure 5.C illustrates the actual trajectory of the robot that has been extracted from the video tracking of the robot performing a forward locomotion. A video demonstration of the robot controlled by the proposed central pattern generator is available at http://youtu.be/WhfcUNx3IQY.

V. Conclusion

In this paper we have shown how a differential wheeled robot with limited wheels can be controlled with a bio-inspired CPG. The CPG provides a command signal for the rhythmic control of the limited wheels that overcomes their restriction and produces an effective locomotion. The CPG approach can also account for sensory feedback in the form of additional stimulus to all or some neurons in the circuit. Sensory feedback can also be implemented in the form of a modulation of the CPG connectivity. The wide range of rhythm negotiation properties of the proposed model leads to a rich variety of self-organized locomotion.

Advances in CPG research in recent years provide new bio-inspiration for robotic design. We believe that novel CPG control strategies can lead to more autonomous robot locomotion.

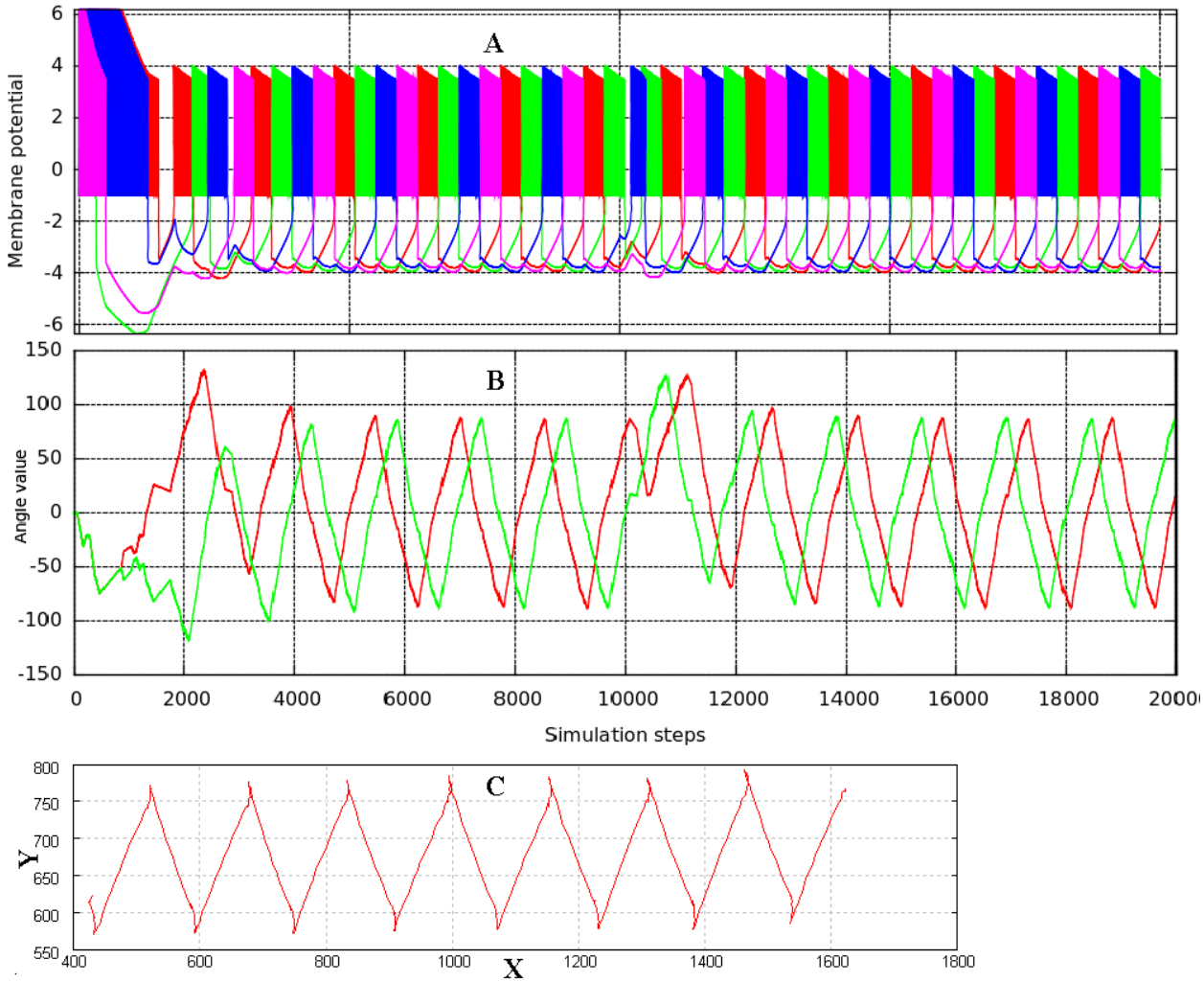

Fig. 5. A: Plot representing the activity of the CPG neurons (activation sequence). 20000 iterations have been simulated: 10000 representing a forward movement, 10000 representing a backward movement. B: the motoneurons output values (wheel angles). After an initial unstable transient period of negotiation, the CPG autonomously achieves a stable oscillation, with constant frequency and amplitude. C: Real robot trajectory extracted from video tracking of a forward movement (arbitrary pixel coordinates are used for Y axis, and number of frames is used for the X axis). Parameters used: neurons: $\alpha = 9$, $\sigma = 0.5$, $\sigma_e = 1$; synapses: $\alpha = 0.5$, $\beta =$, Esyn = 9, $g_{syn1,2} = 25$, T = 1, release_time = 0.01; motoneurons: $\gamma = 900$, $\upsilon = -1.5$, O = 0.

REFERENCES

[1] M. Rabinovich, P. Varona, A. Selverston, and H. Abarbanel, "Dynamical principles in neuroscience". *Reviews of Modern Physics*, *78*(4), 1213-1265, 2006. APS. doi: 10.1103/RevModPhys.78.1213.

[2] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. Neural Networks, 21(4), pp. 642-653. Elsevier, 2008.

[3] M. Nițulescu, Theoretical aspects in wheeled mobile robot control, Proc. of International Conference on Automation, Quality and Testing, Robotics –AQTR 2008, Cluj-Napoca, Romania, ISBN 978-1-4244-2576-1.

[4] WikiRobotics: Skybot repository. Available at next web address: http://www.iearobotics.com/wiki/index.php?title=Skybot

[5] WikiRobotics: SkyPic repository. Available at next web address: http://www.iearobotics.com/wiki/index.php?title=Skypic

[6] WikiRobotics: Sky293 repository. Available at next web address: http://www.iearobotics.com/proyectos/sky293/sky293.html

[7] J. Gonzalez-Gomez, J. G. Victores, A. Valero-Gomez, M. Abderrahim , "Motion Control of Differential Wheeled Robots with Joint Limit Constraints", in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, (to appear).

[8] F. Herrero-Carrón, F. B. Rodríguez, and P. Varona (2011). Bio-inspired design strategies for central pattern generator control in modular robotics. *Bioinspir Biomim*, *6*(1), 16006. doi: 10.1088/1748-3182/6/1/016006.

[9] N. F. Rulkov, "Modeling of spiking-bursting neural behavior using two-dimensional map", *Physical Review*, vol. 65, 2002, pp. 1-9.

[10] A. L. Shilnikov and N. F. Rulkov, "Origin of chaos in a two-dimensional map modeling spike-bursting neural activity", Int. J. Bif. and Chaos, 13:3325–3340, 2003.

[11] F. Herrero-Carron, F.B. Rodriguez, P.Varona. Flexible entrainment in a bio-inspired modular oscillator for modular robot locomotion. Lect Notes Comput. Sc 6692: pp. 532-539.

[12] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski, "An Efficient Method for Computing Synaptic Conductances Based on a Kinetic Model of Receptor Binding". *Neural Computation*, *6*(1), 14-18, 1994, MIT Press. doi: 10.1162/neco.1994.6.1.14.

[13] F. Herrero-Carrón, F.B. Rodriguez, P. Varona. Studying robustness against noise in oscillators for robot control. Proceedings of the ARCS Conference, pp. 58-64, Orlando, 2011.

[14] J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno, M. Abderrahim, "A New Open Source 3D-printable Mobile Robotic Platform for Education". Proc. of the 6th Int. Symposium on Autonomous Minirobots for Research and Edutainment, 2011, May, 23-25. Bielefeld. Germany.