Robotics and Autonomous Systems 60 (2012) 912–927

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

CPG modulation for navigation and omnidirectional quadruped locomotion

Cristina P. Santos *, Vítor Matos

Industrial Electronics Department, Algoritmi Center, University of Minho, Portugal

ARTICLE INFO

Article history:
Received 14 January 2011
Received in revised form
29 December 2011
Accepted 9 January 2012
Available online 30 January 2012

Keywords:
CPGs
Autonomous robots
Quadruped locomotion
Omnidirectional locomotion
Architecture bio-inspired

ABSTRACT

Navigation in biological mechanisms represents a set of skills needed for the survival of individuals, including target acquisition and obstacle avoidance.

In this article, we focus on the development of a quadruped locomotion controller able to generate omnidirectional locomotion and a path planning controller for heading direction. The heading direction controller is able to adapt to sensory-motor visual feedback, and online adapt its trajectory according to visual information that modifies the control parameters. This allows for integration of sensory-motor feedback and closed-loop control. This issue is crucial for autonomous and adaptive control, and has received little attention so far. This modeling is based on the concept of dynamical systems.

We present experiments performed on a real AIBO platform. The obtained results demonstrate both the adequacy of the proposed locomotor controller to generate the required trajectories and to generate the desired movement in terms of the walking velocity, orientation and angular velocity. Further, the controller is demonstrated on a simulated quadruped robot which walks towards a visually acquired target while avoiding online-visually detected obstacles in its path.

1. Introduction

Trajectory generation and modulation are two tightly coupled issues in robotics and animal motor control [1–4], that have not been completed solved in robotics. These notions are central if we consider that online trajectory generation is needed when facing dynamic and partially unknown environments, and there is a continuous online coupling to sensory information.

In this article we address some of the issues related to trajectory generation and modulation. As a main application, we focus the topical issue of robust, flexible quadruped locomotion generation and its modulation to achieve navigation, including omnidirectional locomotion and gait switching. We choose this task because it requires important features of movement control, notably timing, synchronization and behavior integration. Further, target acquisition and obstacle avoidance are important requirements for a legged robot that is expected to be able to navigate in its surrounding environment. Control of omnidirectional locomotion has not been fully explored but it is very relevant in terms of maneuverability.

We assume that despite the final desired behavior complexity, motor behaviors should be built upon small blocks, "motor primitives", which can be modulated according to small changes of parameter values that explicitly reflect the desired modulation in the final behaviors. Therefore, each motor primitive should have an associated set of parameters, that control the generation and achievement of the desired motor tasks. Further, selection and timing of the appropriate motor primitives, including their superposition, should be easily addressed, from signals that encode the required activity and modulation. This movement decomposition is supported by current neurological and human motor control findings [5–7]. It has been often used in robotics for generating movement [3,5,8–13].

Moreover, from a robotics point of view, the proposed controller has to present some relevant features. Generated trajectories have to be stable and smooth in order to resemble more human like, to reduce the risk of damage and that of loosing stability.

In order to tackle these challenges, we propose an architecture bio-inspired in the functional model of biological motor systems [14–17], on the use of dynamical systems to model Central Pattern Generators (CPGs) and the path planning controller. A similar architecture for other behaviors is presented in [18]. We also take some bio-inspiration in "motor primitives" and their use to build more complex biological movements [6,7,16], and propose simple dynamical systems that are able to generate discrete and rhythmic motor primitives, and more complex movements through their superposition. This extends related [3,5,10,19,20] and the teams current work [11–13,21,22].

In this contribution, omnidirectional locomotion in a rigid bodied robot is achieved by a combined use of the flap and swing hip joints, based on the ideas of the wheel model [23]. The proposed CPG network structure is well suited for this method since it allows to independently control the step movements of

* Corresponding author. Tel.: +351 914730808; fax: +351 253510189.
E-mail address: cristina@dei.uminho.pt (C.P. Santos).

0921-8890/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2012.01.004

the different joints while still keeping the intralimb and interlimb coordination.

Further, in this contribution we tackle gait switching according to the speed increase. Similarly to animal gaits, at all walking speeds the onset of swing in a foreleg should occur just after the onset of stance in the ipsilateral hind leg [14], i.e., both the duty factor and the interlimb phase relationships CPG parameters should be continuously changed according to the speed increase.

Gait switching has been previously tackled by Yoneda and Hirose [24], Sano and Furusho [25], Yasa and Ito [26], Inagaki and Kobayashi [27], Hardarson [28], Mcghee and Frank [29], using the wave gait rule, and also by the team [21]. Because the resultant motor patterns are modulated according to modulation of the CPG parameters, this is easily and straightforward achieved using our formulation.

Finally, the 2D motion of the robot is specified by a path planning controller based on previous work [12,30,31]. An attractor-based dynamics is formulated for the heading direction variable to generate paths that circumnavigate obstacles and find their way to a target. The velocity of the robot is governed by a simple dynamics, that takes into consideration the distance to the sensed target or to the sensed obstacles.

The system is implemented and tested in, both, a simulated environment and the real ERS-7 robot from Sony. The system is able to adapt to sensory-motor visual feedback, and online adapt its trajectory according to visual information that modifies the control parameters. This allows for integration of sensory-motor feedback and closed-loop control. This issue is crucial for autonomous and adaptive control, and has received little attention so far.

The system is implemented and tested in, both, a simulated environment and the real ERS-7 robot from Sony. The robot walks in an environment towards a visually acquired target that may be obstructed by some obstacles. This may oblige to object circumnavigation. The navigation system provides for the modulation of the CPG parameters and the joint movements are generated achieving the necessary motions for circumnavigation and target reaching.

The system is able to adapt to sensory-motor visual feedback, and online adapt its trajectory according to visual information that modifies the control parameters.

However, this is an open loop controller and the generated trajectories are bound to the predefined shape of the CPG solution. The next step is to increase locomotion flexibility by including feedback from sensory information, such that locomotion is adaptively generated according to the environment.

Main contributions are in the expansion of a previously developed CPG framework for the generation and switching among motor behaviors namely omnidirectional locomotion and autonomous obstacle avoidance and target reaching. Further, innovation results from the use of dynamical systems with superimposition of primitives to generate movement which is modulated by sensory information according to the environment, such that action is steered by perception. In addition, navigation was planned within the same framework and the different behaviors were successfully integrated.

This article is structured as follows. We first review recent work on quadruped locomotion using Central Pattern Generators and omnidirectional locomotion. Section 3 describes the mathematical systems used to model a CPG. Then, Section 4 briefly introduces the proposed architecture. The lower level of the proposed architecture is described in Section 5. The second level is presented in Section 6 together with gait transition, omnidirectional locomotion and the corresponding CPG modulation. Section 7 presents the obtained results on the real AIBO robot for several experiments relative to omnidirectional locomotion. Section 8 describes the heading direction control and the experimental evaluation of the overall system. We conclude by discussing the main results we obtained and possible system improvements and extension.

2. State-of-the-art

The study of robot locomotion has always found solutions and inspiration from nature. We believe that in order to devise flexible, adaptive, relevant locomotor models it is imperative to integrate concepts of the vertebrate locomotor generator structure organization, function, components and flexibility. However, our perspective is an engineering one and abstractions are done such that the proposed models are well suited for robots.

In quadruped locomotion the concept of Central Pattern Generators (CPG) has been widely used as a reliable alternative to traditional controllers. CPGs [32] are spinal–neural networks capable of autonomously producing coordinated rhythmic output signals.

There are successful implementations where the CPG has been integrated with sensory feedback [11,13], adaptive rules [33,34] and reflexes [9,19]. Some of these approaches have been developed using dynamical systems theory and coupled oscillators. For an in-depth review see [35].

2.1. Central pattern oscillators as dynamical oscillators

Control approaches based on CPGs and nonlinear dynamical systems are widely used in robotics to achieve tasks which involve rhythmic motions such as biped and quadruped autonomous adaptive locomotion over irregular terrain [9,36], juggling [37], drumming [3], playing with a slinky toy and basis field approaches for limb movements [6].

These systems offer multiple interesting features, which apply well to model CPGs for robotic controllers. All these properties were explored and applied in several works.

In [33] the adaptation of these oscillators was explored by developing an online learning system that attempts to minimize the necessary energy for the gait.

In [38] a mechanism that enables to control independently the duration of the swing and stance step phases has been presented, which is fundamental for generating correct locomotor movements for the hip. It also introduces a method for coupling the oscillators, based in the symmetry property of the dynamical oscillators [39]. The work is further extended in [19], where a method for designing CPGs using coupled dynamical oscillators and a systematic manner of adding sensory feedback is presented.

Degallier et al. [10] exhibit the possibility of joining the movements of crawling and feet placement movements, describing the behavior of hand placement for reaching marks on the ground.

In this contribution, we propose an architecture inspired on the ideas described on [5,32] that extends the team related [12,20,40,41] and current work on postural control [11]; gait switching [21] locomotion-induced head movement minimization [42] and on a drumming task [13]. Previously, the team developed some work in which sensory-motor information has been integrated in a dynamical architecture to generate timed trajectories [12,20]; coordinate robots behaviors [20,40,41]; and achieve steering and obstacle avoidance [12,40].

The proposed architecture integrates sensory-motor skills with obstacle circumnavigation and target reaching. The design of the architecture takes into account experimental knowledge about how the nervous system deals with the control problem in a robust and flexible way [5,7,16,32,43,44]. It is loosely inspired from the biological concepts of CPGs and by the concepts of force fields [6].

The architecture is organized onto hierarchical layers, similarly to the motor control systems involved in goal-directed locomotion in vertebrates. This modularity between the layers enables to achieve independency between them which is adequate for a real implementation of the architecture, from a computational perspective. Higher layers that require more computational power

but with larger time scales, can easily be implemented in external computers and communicate with the robot when needed.

Similar ideas have already been applied by Degallier et al. [10] for the generation of other behaviors, namely in drumming and in switching between crawling and reaching. Herein, we further explore this idea for the generation and switching among motor behaviors continuing our previous work [13]. Further, the system is able to autonomously detect and reach a target while avoiding obstacles in its path, modulating the generated trajectories as required.

However, to the best of our knowledge, omnidirectional locomotion has not been addressed in the framework of dynamical systems with superimposition of primitives. The purposed work tries to serve this purpose by creating systems that autonomously bifurcate from single point attractors for discrete movements (fixed points) to limit cycles for rhythmic movements, or simply by modulating the rhythmic movement, when sensory information is used to steer action. Small changes of the CPG parameters modulate the desired trajectories and the resultant motor patterns. This modulation is encoded in high level operational locomotion parameters, as the desired walking orientation, translational speed and angular velocity of the desired motion, similarly to signals derived from the brainstem of biological systems [17]. Further, these signals reflect the requirements of the surrounding environment. The current sensorial context specifies the required motor behavior in a very simplistic form, considering target reaching and obstacle avoidance, similarly to the biological systems. Additionally, the path is flexibly chosen and depends on the surrounding environment and robot state.

The obtained results demonstrate both the adequacy of the proposed locomotor controller (1) to generate the required coordinated trajectories for locomotion; (2) to modulate the generated motor patterns according to a velocity increase; (3) to achieve omnidirectional locomotion in terms of the walking velocity, given orientation and angular velocity; and (4) to navigate in the sensed environment. Further, the switch among the different motor programs was smooth and easily elicited according to visual sensory information.

This contribution represents a step forward in the attempt to achieve flexible, natural and adaptive goal-directed locomotion. It is simpler when compared to other solutions [9], enables us to elicit behavior switching, provides for a functional description, and enables the possibility of performing steps in any direction. The system navigates in its environment relying upon on sensory information visually acquired.

2.2. Omnidirectional locomotion

Quadruped omnidirectional locomotion has been achieved through several methods. The most usual method for achieving omnidirectional motion is planning the footholds and gaits through the use of inverse kinematics and body dynamics [45–47]. A parameterizable omnidirectional walk can also be found in [23], where optimization was used to find the best parameters for the leg motions [48,49]. These gaits are optimized for one type of motion, typically performing poorly on others.

Omnidirectional path following was successfully achieved in [50], by optimizing the parameters simultaneously for all directions of motion and turning rates. They implemented on the controller the learning of a very high-dimensional policy, through dimensionality reduction, constructing low-class policies in the space of the controllers.

There are a couple of implementations of CPG based controllers where quadruped steering is achieved, but not omnidirectional locomotion. Tsujita et al. [51] proposed a dynamic turning control system for a quadruped robot by using nonlinear oscillators. The suggested steering approach is not suitable for the AIBO robot which has a rigid body. Kimura et al. [52] designed a locomotor controller based on neural systems and integrated it with a new navigation controller in order to achieve the turning motion. However, their steering approach is specific for a quadruped robot with yaw joints on the legs.

Omnidirectional locomotion by neural oscillator networks is presented in [53] and demonstrated in 6- and 8-legged robots. It describes modular neural control structures, consisting of three different functional modules, utilizing discrete-time neurodynamics, capable of performing omnidirectional walking as well as reactive behavior. An extension of this work addressing sensory-motor control and movement generation is proposed in [54]. The work is quite similar to the one in this article: a CPG approach for locomotion generation, modulated through a descending signal p, which is changed based on sensory information.

On the contrary, the presented controller and omnidirectional method for locomotion is not exclusive for the AIBO platform, but further it is also applicable to any robot with similar limb configurations.

In the following we briefly discuss the used dynamical systems.

3. Dynamical system

Consider a nonlinear dynamical oscillator, containing a supercritical Hopf bifurcation, given by the following differential equations

$$\dot{x} = \alpha\left(\mu\nu - r^2\right)(x - O) - \omega\iota z, \tag{1}$$

$$\dot{z} = \alpha\left(\mu\nu - r^2\right)z + \omega\iota(x - O), \tag{2}$$

where $r = \sqrt{(x-O)^2 + z^2}$ and x, z are state variables.

This nonlinear oscillator contains a Hopf bifurcation such that the solution bifurcates to either a stable fixed point at $(x, z) = (O, 0)$ (for $\mu\nu < 0$), or to a structurally stable harmonic oscillation around $(x, z) = (O, 0)$ (for $\mu\nu > 0$). Herein, only parameter $\nu \in \{-1, 1\}$ is used to control these bifurcations, while $\mu > 0$ is guaranteed. Speed of convergence is given by $\left|\frac{1}{2\alpha\mu\nu}\right|$. These solutions are depicted in Fig. 1.

Oscillation frequency is specified by $\omega > 0$ (rad s^{-1}). Amplitude of the oscillations (limit cycle radius) is given by $\sqrt{\mu}$. Variable O is used to control the x solution offset.

The limit-cycle rotates clockwise or counterclockwise if $\iota = -1$ or $\iota = 1$, respectively (Fig. 1). This change in the limit-cycle's direction results in the inversion of the solutions in time.

Finally, this oscillator is able to modularly generate trajectories that can be summarized as follows

1. a discrete movement to an offset, if $\nu = -1$;
2. a rhythmic movement around a offset, if $\nu = 1$;
3. and the superimposition of both movements, resulting in a more complex movement, if $\nu = 1$ and the offset is defined as a time-changing variable such as the state variable of another dynamical system. These ideas were further explored in [10,13].

These different type of trajectories allow us to generate a myriad of movements, including those of locomotion.

3.1. Properties

This oscillator has many useful properties for CPG based controllers and for the online generation of trajectories on robotic applications. Trajectories are generated online and in real time with a low computational cost. Further, the intrinsic robustness of the oscillator to small perturbations allows the inclusion

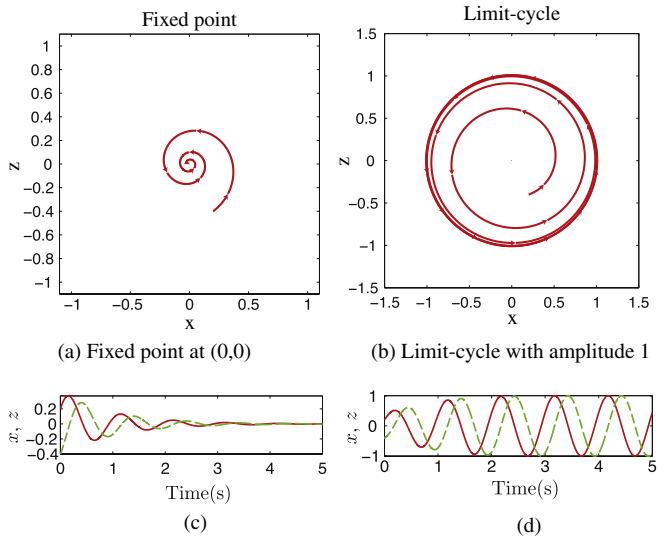


Fig. 1. Oscillator solutions for a system defined by Eqs. (1) and (2) when (a), (c) $\nu = -1$ and (b), (d) $\nu = 1$. $(x_0, z_0) = (0.2, -0.4)$, $O = 0$, $\mu = 1$, $\alpha = 1$, $\iota = 1$ and $\omega = 6.3$ rad s^{-1}. In (a) and (b) the vector field is presented in the background. In (c) and (d) x solution is the solid red line and z is the dashed green one. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

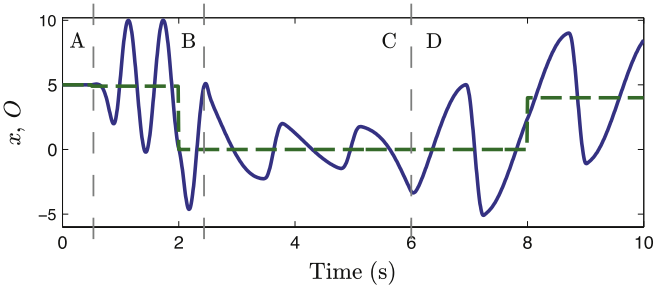


Fig. 2. Evolution of the generated trajectory (blue) through several changes of parameters. The dashed green line depicts offset O throughout time. On (A), since $\nu = -1$ the oscillator relaxes to the value of O. The rhythmic activity is activated in (B)–(D). In (B) and (D) we have both discrete and rhythmic movements. We decrease frequency from (B) to (C) and (D). In (D) the oscillator is inverted. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of feedback mechanisms to close the control loop, as shown in [19], and ensure robust control of the movements in time-varying environments [11,13]. Furthermore, the modulation of the generated trajectories with respect to their amplitude, frequency (for speed change), activation and offset, is carried out explicitly through the specification of a small set of parameters. The outcome of small changes in parameters results in straightforward and smooth modulation of the trajectories. It also allows for a distributed organization due to its entrainment properties.

Fig. 2 demonstrates the parameters' roles in the generated trajectories and in their modulation.

In (panel A), there is a discrete movement towards the offset O parameter (dashed green line). Since $\nu = -1$, the rhythmic movement is off. Rhythmic movement is turned on at $t = 0.8$ s (panel B). The O parameter is also changed does changing the offset of the rhythmic movement. This superposition of discrete and rhythmic movement is verified in both panels (B) and (D).

In panel (C), small changes of the ω parameter modulates the generated trajectories in frequency. Further, note the fast but smooth amplitude modulation of the generated trajectories according to small changes in the μ parameter. The oscillator promptly changes the frequency and amplitude of the generated solutions, resulting in smooth and responsive trajectories. In (D) the oscillator is inverted.

The system is able to generate motor patterns without sensory feedback and without any rhythmic inputs, when activated by simple commands that encode their rhythmic activation, frequency and amplitude. These features are similar to their biological counterparts.

This dynamical oscillator is well suited for distributed use. This aspect will be explored to create an organized network of oscillators that will result in a CPG, and further, a network of CPGs that results in the locomotor controller. By coupling or synchronizing several oscillators, it allows for a flexible coordination between the trajectories of the robot joints.

3.2. Independent control of oscillator phase durations

The oscillator described by Eqs. (1) and (2) generates an x oscillatory trajectory in which the ascending and descending parts have equal durations. In order to achieve an independent control of the duration of these parts, we employ the following equation proposed by Righetti and Ijspeert [19],

$$\omega = \frac{\frac{1-\beta}{\beta}\omega_{sw}}{e^{-\alpha z}+1} + \frac{\omega_{sw}}{e^{\alpha z}+1}, \tag{3}$$

where ω alternates between two different values, $\frac{1-\beta}{\beta}\omega_{sw}$ and ω_{sw}, depending on the value of the z variable. Variable $\beta \in [0, 1]$ is the duty factor, the proportion of support phase to the gait period.

For $\iota = 1$, if $z < 0$, x solution is in the descending part of its movement. If $z > 0$, x solution is in the ascending part of its movement. By changing the frequency value in each part of the x solution, we are able to independently control the duration of each part. The alternation speed between these two values is controlled by a. For further details see [19].

4. CPG architecture

The proposed architecture is bio-inspired in the vertebrate biological motor systems [14–17], and is structured in two functional hierarchical layers according to their level of abstraction [18], similar to the motor control systems involved in goal-directed locomotion in vertebrates. Fig. 3 presents a schematic of the proposed architecture.

The lower level addresses the role of the spinal cord and generates the motor patterns by networks of Central Pattern Generators (CPGs) [32,55]. The concept of biological CPG includes the idea of hierarchical organized unitary oscillators: the unit-CPGs. A single unit-CPG controls and activates the antagonistic muscle pairs, controlling the movements of a single joint. Movements of a limb are controlled by a limb-CPG, composed by a group of coordinated unit-CPGs within a limb. These are coordinated in a flexible way to provide many activation patterns in a single limb.

Based on previous work [11,13,21,22], we apply (oscillator-based) differential equations to model a network of four coupled limb-CPGs. These systems are solved using numerical integration and sent to the lower level PIDs of the joints.

The second layer models very basically the brainstem command centers for initiating, regulating and stopping CPGs activity and therefore initiate a walking gait, switch among gaits, control the direction of movement and stop the locomotion. This layer receives the desired robot angular velocity, the robot walking orientation and a modulatory signal which strength encodes the desired duty factor and outputs the set of CPG parameters: frequency, amplitude and relative phases. By sending these at the right timing to the lower level, it results in the modulation of the generated trajectories and thus in different motor behaviors. A smooth behavior switch and behavior modulation is therefore achieved which can be elicited according to sensory information.

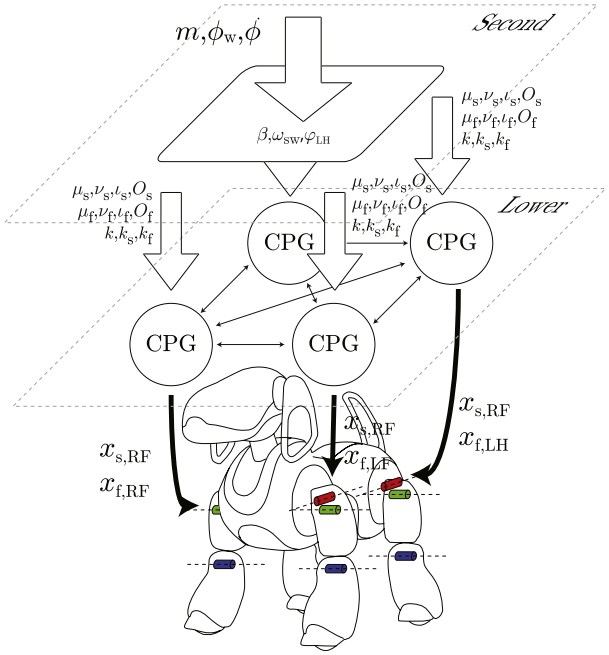


Fig. 3. Proposed architecture based on a CPG network. Each CPG has a set of parameters that specify the limb movement.

5. Lower level

5.1. Unit-CPG

Inspired by neurobiology we named of unit-CPG the Hopf oscillator given by Eqs. (1)–(3) (described in Section 3). The generated x solution of this nonlinear oscillator is used as the control trajectory for a hip joint of the robot limbs. These trajectories encode the values of the joint's angles (°) and are sent online to the lower level PID controllers of each hip joint.

Herein, we consider that the descending phase of the x trajectory when $\iota = 1$ corresponds to the stance step phase in which the limb moves backwards, and the hip swing joint value is decreasing, thus propelling the robot forward. The ascending phase is the movement that places the foot in a more advanced position, ready for the next step, and corresponds to the swing step phase. This is depicted in Fig. 4(a).

Changes in the limit cycle direction, through parameter ι, result in changing the step stance phase between descending ($\iota = 1$, counterclockwise limit cycle) and ascending ($\iota = -1$, clockwise limit cycle) phases of the x trajectory. The swing phase is descending when $\iota = -1$ and ascending for $\iota = 1$.

For all conditions, when $z < 0$ the limb is executing the swing step phase. When $z > 0$ the limb is executing the stance step phase (Fig. 4(b)).

In quadruped locomotion, an animal changes its velocity by increasing or decreasing the number of steps per second. The two main phases of the movement can be unequal: the swing phase duration (extension) keeps approximately constant, whereas the stance phase (flexion) varies in duration from a slow walk to fast running, a trot or a gallop [16].

By controlling the durations of the ascending and descending phases of the x trajectory, we are controlling the durations of the swing and stance step phases. This enables controlling the duty factor (β) of the generated trajectories by keeping the swing frequency constant ($\omega_{sw} = \frac{\pi}{T_{sw}}$) and changing only the stance duration, thus achieving different quadrupedal gaits.

Each unit-CPG takes a set of parameters for the modulation of the generated trajectories for the specified joint, as follows:

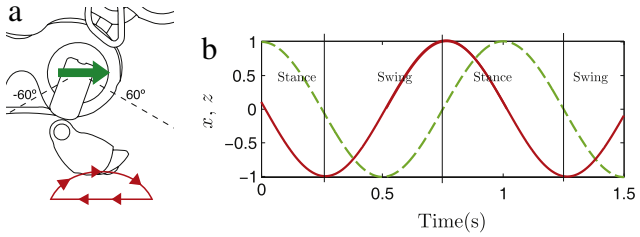


Fig. 4. Stance and swing phases for $\iota = 1$. (a) The robot is pushed forward. The corresponding movement of the descending trajectory is the movement of pushing the robot forward. (b) x (solid red line) and z (dashed green line) trajectories. x trajectory (solid red line) is the control policy for the hip swing joint of the robot. When $z < 0$ ($z > 0$), the robot is performing the swing (stance) phase movement. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- $\nu \in \{-1, 1\}$, switches on/off the rhythmic trajectory;
- $\mu > 0$, modulates the amplitude of oscillations ($\sqrt{\mu}$);
- $\iota \in \{-1, 1\}$, modulates the direction of the trajectories;
- $\beta \in]0, 1[$, changes the walking velocity since it controls the stance duration of the generated movement.
- O, sets the value of the oscillation's offset or the goal-discrete movement.

These parameters are not hand tuned but rather their values are specified by the mechanism presented in the next sections, controlling the parameters for the different motor programs.

The parameters α, ω_{sw} and a are set *a priori*. Parameter ω_{sw} specifies the swing phase duration, which is kept constant. Its value depends on the desired speed of movements and on the robotic platform.

5.2. Limb-CPG

To ensure a proper synergy of movements in all joints within a limb, it is required that these movements are expressed correctly, with coordinated step phases in all limb joints in order to result in a step.

This intralimb coordination is achieved by a limb-CPG[1] that controls and coordinates the different joint movements in a single limb. It is composed by an activation unit for the knee joint and a pair of unit-CPGs: one controlling the hip swing joint (denoted by subscript s) and the other controlling the hip flap joint (denoted by subscript f). Each of these generate the trajectories for the knee, hip swing and hip flap joints, respectively (Fig. 5).

An important property of dynamical oscillators is that they are suitable for distributed organizations. Considering that a unit-CPG is controlling a joint, coordination of the DOFs within a limb is achieved by coupling, in a given manner, the dynamics of each unit-CPG within the same limb. These couplings ensure that the joint within a limb stays synchronized.

Specifically, when a swing joint is in the limb swing phase, the corresponding flap joint should also be in the limb swing phase; independently of their individual limit cycle directions. Thus, each flap unit-CPG is unilaterally coupled to the corresponding limb swing unit-CPG. This is achieved by changing the flap's z differential equation (1) as follows,

$$\dot{z}_f = \cdots + \kappa \frac{z_s}{r_s}, \tag{4}$$

where κ is the coupling strength. The unit-CPGs are coupled through z because this allows the simplest coupling method, where

[1] CPG and limb-CPG are interchangeable terms throughout this article. Both terms refer to the locomotor CPG that controls the movement of a limb in a quadruped mammal.

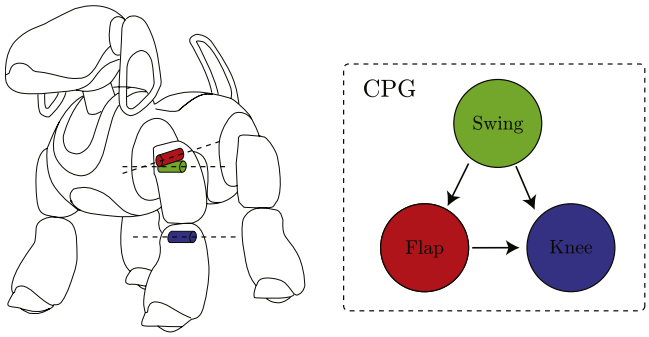


Fig. 5. To each hip joint, swing (light green) and flap (red), is assigned a unit-CPG. The knee (dark blue) is controlled by the other kind of activation unit. All the units are coordinated in order to generate the correct limb movement during locomotion. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

it is possible to independently control the directions of oscillations on both unit-CPGs.

This unidirectional coupling also makes sense when considering the findings and ideas from the distribution of the spinal locomotor CPGs [55]. Leading oscillators (hip swing unit-CPGs), entrain less excitable oscillators (hip flap unit-CPGs).

5.3. Knee control

The knee joints are controlled as simply as possible, according to the corresponding hip swing joint. When the limb performs the swing phase, the knee flexes to a fixed angle θ_{sw}. When performing the stance phase, the knee extends to another fixed angle θ_{st}.

This discrete motion is generated by

$$\dot{y} = v \tag{5}$$

$$\dot{v} = -\frac{b^2}{4}(y - g) - bv \tag{6}$$

whose stable solutions converge asymptotically and monotonically to a goal fixed point g ($= \theta_{st}, \theta_{sw}$). Relaxation is controlled by parameter b.

The state variable y is the control trajectory for the knee and follows the value of g. Because the knee movement is modeled by a differential equation, the solution of the system smoothly adapts to the g parameter variations. Herein, this is specially important, since g corresponds to the angles that the knee joints should have. These are not fixed but change between two fixed values, accordingly to the current limb step phase, as follows:

$$g = \frac{\theta_{st}}{e^{-az_s} + 1} + \frac{\theta_{sw}}{e^{az_s} + 1}. \tag{7}$$

If $z_s < 0$, the limb is in the swing phase and the knee flexes to θ_{sw}. If $z_s > 0$, the limb is in the stance phase and the knee joint extends to θ_{st}. As g is a stable, globally attractive point, no stability problem will occur for any value of g. Thus, each time g is changed to any of these values according to the current limb step phase, the system is attracted to the new goal g and modifies the resulting position y, generating a discrete movement towards g.

We achieve a continuous velocity since we apply a second order system to generate the knees' movement. This control of the knee movement allows a greater clearance when the limb describes the swing phase.

5.4. CPG network

In order to generate the adequate stepping sequence, the four CPGs must be coordinated. This interlimb coordination is achieved by bilaterally coupling among each other the swing and flap unit-CPGs, ensuring a correct coordination between the limbs (Fig. 6).

Table 1
Phase relationships between the oscillators.

i	j	θ_i^j
LF	RF	$-\pi$
LF	LH	$-\varphi_{LH} 2\pi$
LF	RH	$(-\varphi_{LH} + 0.5)\, 2\pi$
RF	LH	$(-\varphi_{LH} + 0.5)\, 2\pi$
RF	RH	$(-\varphi_{LH} + 1)\, 2\pi$
LH	RH	π

Unit-CPGs are coupled as follows

$$\begin{bmatrix} \dot{x}_{p,i} \\ \dot{z}_{p,i} \end{bmatrix} = \cdots + \kappa_{p,i} \sum_{j \neq i} \mathbf{R}(\iota_{p,j}\, \theta_i^j) \begin{bmatrix} \dfrac{(x_{p,j} - O_{p,j})}{r_{p,j}} \\ \dfrac{z_{p,j}}{r_{p,j}} \end{bmatrix}, \tag{8}$$

where $p \in \{s, f\}$, represents the respective joint's unit-CPG, and i and j represent the limb $\in$ {LF, RF, LH, RH}.

Parameter $\kappa_{p,i}$ specifies the coupling strengths from p to i unit-CPGs. This is useful when it is necessary to detach a specific limb from the others, and perform tasks other than walking. We normalize the coupling contributions, minimizing the effects of the different amplitudes of the other unit-CPGs. The parameter $\iota_{p,j}$ is used to enable a correct coupling between unit-CPGs with different directions.

The rotation matrix $\mathbf{R}(\theta_i^j)$ rotates the linear terms onto each other, where θ_i^j is the required relative phase between i and j oscillators to perform a certain gait.

These relative phase relationships between the oscillators can also be calculated according to the gaits' relative phases. The gait relative phase of leg i, φ_i, is the time elapsed from the setting down of an arbitrary reference limb (herein chosen as the left forelimb), until the foot of leg i is set down, given as the fraction of the cycle time. Relative phase relationships are given by,

$$\theta_i^j = (\varphi_i - \varphi_j)\, 2\pi. \tag{9}$$

In this work, we only address symmetric gaits, which always have $\varphi_{LF} = 0$, $\varphi_{RF} = 0.5$, and $\varphi_{RH} = \varphi_{LH} - 0.5$. By substituting these values onto (Eq. (9)), it is possible to express the oscillators relative phases θ_i^j in terms of the gait phase φ_{LH}, as presented in Table 1 ($\theta_i^j = -\theta_j^i$). Interlimb coordination can then be achieved by specifying only the gait phase φ_{LH}. Typically, $\varphi_{LH} = 0.5$ and $\varphi_{LH} = 0.75$, for trot and walk, respectively.

The final network of CPGs (Fig. 6) has controlled phase relationships and is able to generate complex, synchronized rhythmic patterns; discrete movements and a combination of both in a stable and flexible way. Due to the properties of this type of coupling among oscillators, the generated trajectories are smooth, stable and robust to perturbations; and thus ideally suited for trajectory generation in a robot.

This network constitutes the layer one of the proposed architecture (Fig. 3). It receives from the upper level the required parameters that specify and modulate in a simple and straightforward manner the generated trajectories. Each CPG receives two sets of ν, μ, ι and O parameters (one set for each unit-CPG). Also, the coupling among CPGs is specified by parameters κ_s, κ_f and κ for swing and flap unit-CPGs and knee activation units, respectively. Parameters β and φ_{LH} are the same for all unit-CPGs.

This structure also allows omnidirectional locomotion because it is possible to independently control the step movements of the different joints and still maintain the intralimb and interlimb coordination.

6. Second level

The second layer models very basically the brainstem command centers for initiating, regulating and stopping the CPGs activity of

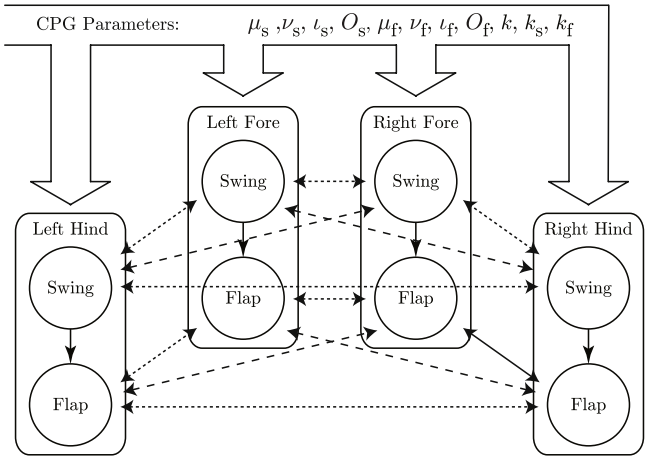


Fig. 6. Structural view of the CPG network. The swing unit-CPGs are bilaterally coupled among each other, as well as the flap unit-CPGs (dashed arrows). The swing unit-CPGs are unilaterally coupled to the corresponding flap unit-CPGs (solid arrows). This level receives the required parameters from the upper level.

the lower layer. It is responsible for selecting a motor program (MP) and for determining and sending the corresponding MP parameters for the lower level at regular time intervals such that the desired task is achieved.

The MPs' parameters are the inputs of the four limb-CPGs, and are the control parameters required for generating movements. As they may be defined relatively to the environment, according to the time-varying sensory, movements are generated adapted to the environment and allow for coordination and timing as well as behavior selection.

In order to perform omnidirectional locomotion, the robot must be able to move to any point of interest, with a given translation speed, a turning rate $\dot{\phi}$ and walking orientation ϕ_w (Fig. 8). But the increase in the locomotion speed represents a change in the type of gaits from walk to trot, with a corresponding variation in the required interlimb relationship. We have tackled separately these issues, considering each issue as an individual MP. Thus, we define four MPs: initiating locomotion, gait transition, stop locomotion and omnidirectional locomotion (steering).

Modulation happens at the level of the CPG parameters: ν for stopping/initiating locomotion; μ, that modulates the amplitude of the unit-CPG solution; duty factor β for adjusting frequency; gait phase φ_{LH} for adjusting interlimb coordination; and ι, to modulate the direction of the unit-CPG solution.

These different MPs and the corresponding modulation are explained in detail bellow.

6.1. Gait transition

Different sets of CPG parameters correspond to different motor behaviors. However, bio-inspiration suggests that single tonic signals, from supraspinal regions, should somehow encode the required activity and/or modulation; providing a mapping from the tonic signals to the set of CPG parameters. Such mapping reduces the dimensionality of the control problem to just one excitatory signal. For instance, increasing activation of the brainstem locomotor center commands leads to an increase in quadruped locomotion speed, and to a gait switch from walk to trot.

In our model [21], a given modulatory drive signal, m, models the mechanisms that underlie gait transitions induced by simple electrical stimulation of the brain stem, and regulates the activity of the unit-CPGs network. Different values of the drive lead to different MPs, namely: locomotion initiation, speed change with adjustment of interlimb coordination and consequent gait transition, similarly to the biological counterparts [17]. In order to increase the robot stability we apply the wave gait rule [27,29], meaning a gradually shift in interlimb coordination, from walk to trot. These MPs correspond to different specifications of the set of CPG parameters $\{\nu, \beta, \varphi_{\text{LH}}\}$: the ν parameter for stopping/initiating locomotion; the duty factor β for adjusting frequency and gait phase φ_{LH} for adjusting interlimb coordination.

Below a low threshold, m_{low}, the robot ceases stepping. Above this threshold, the robot starts with a slow walk (non-singular crawl), gradually increasing speed without adjusting the phase relationships. Above $m = 1$, locomotion speed is increased with adjustment of interlimb coordination. At $m = 2.5$, the robot is in a trot. Herein, both the range and the thresholds for the modulatory drive were chosen arbitrarily.

6.2. Omnidirectional locomotion

In order to achieve omnidirectional locomotion on our robotic platform, we base ourselves on the *wheel model* presented in [23]. Basically, it assumes that each foot performs a step with a specified direction and length, and that the overall propulsion of the steps results in the desired robot motion. Hence, the feet move on a certain locus, considering the limb as a small wheel, describing a circular movement during a step in a vertical plane with a specific direction. By controlling the orientation of this plane it is possible to control the step orientation and therefore the robot movement.

For performing a step with a specified orientation and length it is necessary to combine the feet movements on the sagittal and transverse plane. If feet movements are only in the robot sagittal plane, the robot moves straight: forward or backward. When feet movements are only in the robot transverse plane, it moves sidewards. Movement in any direction is achieved by superimposing the movements of a foot on these two planes.

On the AIBO robot, the movements on the sagittal and transverse planes are controlled by the hip swing and hip flap joints, respectively, 7.

Fig. 7 shows some possible combinations of step directions and the resulting robot movement.

The robot rotates in the spot in (a). All flap joints rotate the robot to the right, with fore and hind flaps moving in opposite directions, and the swing joints maintaining the robot in place. The robot steers right while walking forward in (b). The fore flap joints make the robot front move to the right, while the hind flap joints move the robot back to the left. All the swing joints propel the robot forward. This superposition of movements steers the robot right while walking forward.

When the flap joints make the robot move to the left and the swing joints propel it forward, the robot moves straight diagonally to the right (c).

The robot moves to the right in (d), because only the flap joints are employed to propel the robot.

By independently generating movements for the hip swing and flap joints, we are able to generate the desired movements in the sagittal and transversal planes, and thus a step with a certain direction and velocity.

Velocity of a step can be changed by either adjusting the stance phase duration or the step length. However, stance phase duration must be the same for all the limbs for proper expression of the gait. We must then change the step length in sagittal and transversal planes for each leg, enabling achieving the desired step direction and velocity: steps with greater length propel the robot with greater velocity, and it is the opposite for smaller step lengths.

The wheel method is well suited for the proposed CPG network since both the step direction and length in the sagittal and transverse planes can be controlled independently for each

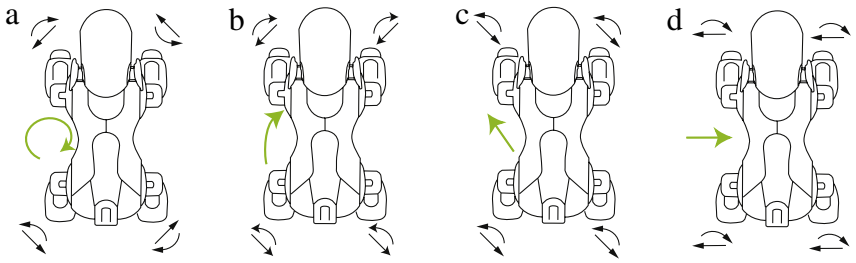


Fig. 7. Some examples of possible robot movements. Resultant movement is indicated by light gray arrow. (a) Rotation in spot; (b) walk forward and turn; (c) walk straight diagonally to the left; and (d) walk straight sideways to the right. Swing and stance phase directions are indicated by curved and straight arrows near the legs, respectively.

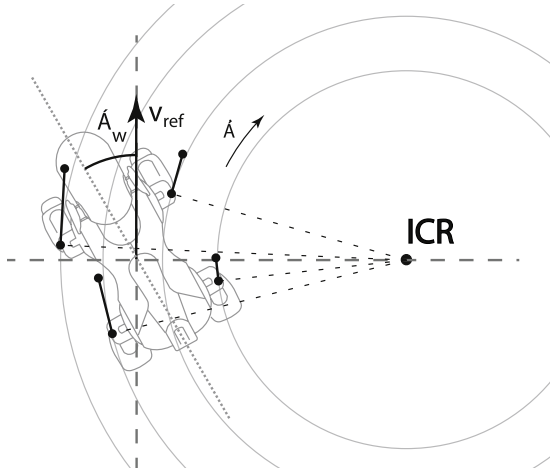


Fig. 8. Robot motion for $\dot{\phi} \neq 0$, $v > 0$ and ϕ_w. The robot walks in a circle path centered around the instantaneous center of rotation (ICR). The figure also depicts the steps direction and length for achieving such walking motion.

leg, while maintaining the required intralimb and interlimb coordination.

Parameterizing the CPGs requires finding the correct movement amplitudes, activations and directions of each step in the sagittal and in the transverse plane.

Because it is possible to describe the rotation of any point of the robot around a certain common point, where the translation speed is zero, i.e., the instantaneous center of rotation (ICR), it is possible to find the trajectory of any point of the robot when performing such rotation. For the robot to move around a certain ICR, each limb should perform a step of proportional size into a suitable direction, tangential to the circle with radius $r = \frac{v_{ref}}{\dot{\phi}}$, around the rotation center (Fig. 8).

When moving forward ($\dot{\phi} = 0$) the ICR is located at infinity.

From the ICR we are able to ascertain the movement for each limb. These trajectories will not describe ideal circles having the ICR as common centers, but instead approximate straight tangential trajectories.

By applying trigonometry, the amplitudes for hip swing (s) and hip flap joints (f) are given as follows, according to the walking velocity (v_{ref}), desired angular velocity, $\dot{\phi}$ and the desired walking orientation ϕ_w:

$$A_{s,i} = A_{\text{ref}} \frac{\dot{\phi} Y_i - v_{\text{ref}} \cos(\phi_w)}{v_{\text{ref}}}, \tag{10}$$

$$A_{f,i} = A_{\text{ref}} \frac{-\dot{\phi} X_i + v_{\text{ref}} \sin(\phi_w)}{v_{\text{ref}}}, \tag{11}$$

where (X_i, Y_i) are the limb i coordinates in the robot reference frame. v_{ref} is an approximate obtained velocity when using a reference amplitude A_{ref} and the desired duty factor β (encoded in m). v_{ref} can be calculated from the duration of the propulsion phase (T_{st}) and the length of the step (l_{st}),

$$v_{\text{ref}} = \frac{l_{\text{st}}}{T_{\text{st}}}. \tag{12}$$

The length of the step can be roughly deduced from the length of the limb (l_{limb}) during the stance phase and the full amplitude of the locomotor movements ($2A_{\text{ref}}(\text{deg})$), as $l_{\text{st}} = 2 l_{\text{limb}} A_{\text{ref}}$.

A_{ref} is an amplitude value set *a priori*, when tuning the parameters for locomotion. Since it is the value from where the chosen amplitude will revolve, it must be suitable for the robotic platform, with the possibility to be increased and decreased without impairing locomotion.

CPG parameters have to be set such that small parameter changes modulate the generated trajectories. Therefore, for each of these unit-CPGs, a mechanism must determine their parameter values according to their roles in the final modulation.

$A_{s,i}$ and $A_{f,i}$ encode all the information needed to parameterize the CPGs, as demonstrated next.

6.3. CPG modulation

6.3.1. Initiating/stopping locomotion

Qualitatively, by modifying on the fly the ν parameter, the system (Eqs. (1) and (2)) switches between a stable fixed point at $x = 0$ (for $\nu = -1$) and a purely rhythmic movement (for $\nu = 1$). Hence, the ν parameter controls whether or not there are oscillations generated by the unit-CPG and thus, locomotion generation. We consider that for $\nu = -1$ the generated movement (that is relaxation to the fixed point) is in fact a discrete movement. The fixed point could be an offset that changes if it becomes the state variable of another dynamical system [13].

The $\nu_{p,i}$ parameters are set according to the modulatory drive, m, and to $A_{p,i}$, as follows:

$$\nu_{p,i} = \begin{cases} -1, & m < m_{\text{low}} \ \vee \ |A_{p,i}| \leq 0.5 \\ 1, & m \geq m_{\text{low}} \ \wedge \ |A_{p,i}| > 0.5 \end{cases}, \tag{13}$$

where $p \in \{s, p\}$, and i represents the limb $\in$ {LF, RF, LH, RH}. A dead zone is defined such that when the amplitude of the movement is negligible, the unit-CPG oscillatory behavior is turned off. Below a lower threshold, $m_{\text{low}} = 0.2$, the oscillators are shut down and the robot rests.

For example, consider a situation in which the robot walks straight forward ($\phi_w = 0°$) during 2.5 s and then a command of $\phi_w = 90°$ is given for the robot to propel to the left. Such a trajectory corresponds to the following sets of values of the CPG parameters.

Fig. 9 shows forelimb trajectories. During the first 2.5 s, the swing joints perform the locomotor movements, pushing the robot forward ($\nu_{s,\text{LF}}, \nu_{s,\text{RF}} = 1$). The flap joints do not move because they are not needed for propelling the robot straight forward ($\nu_{f,\text{LF}}, \nu_{f,\text{RF}} = -1$). After $t = 2.5$ s, the swing joints' rhythmic movements stop ($\nu_{s,\text{LF}}, \nu_{s,\text{RF}} = -1$) and the flap joints start to move ($\nu_{f,\text{LF}}, \nu_{f,\text{RF}} = 1$), indicated by a large vertical line.

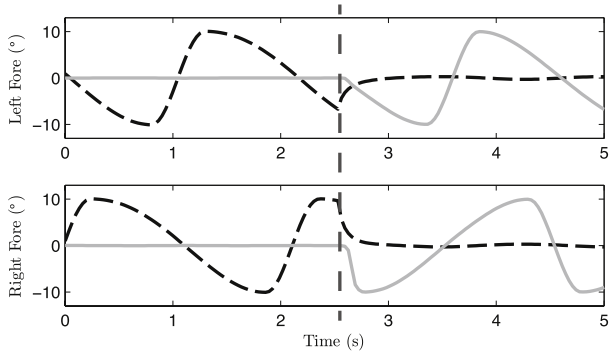


Fig. 9. Generated trajectories for the swing (dashed black) and flap (solid gray) joints in fore limbs. In the first half the robot walks forward, then at 2.5 s the rhythmic locomotor movements in the swing joints stop and start those of the flap joints, making the robot walk sideways. The moment at which the change occurs is indicated by a large vertical line.

6.3.2. Robot velocity modulation

The robot velocity is changed by changing the duty factor β, which results in controlling the stance phase duration: a smaller stance phase duration results in a higher velocity.

As the modulatory drive increases in strength, the duty factor, β, linearly decreases from 0.89 (for the crawl gait) to $\beta = 0.5$ (for the trot gait). The duty factor is mathematically defined as a piecewise linear function of the modulatory drive

$$\beta = \begin{cases} 0.89, & m < m_{\text{low}} \\ -0.1667m + 0.9167, & m_{\text{low}} \leq m < 2.5 \\ 0.5, & m \geq 2.5. \end{cases} \quad (14)$$

This function presents a saturation for $\beta = 0.5$ because the robotic platform cannot perform faster gaits.

6.3.3. Interlimb coordination modulation

In this work, we want that in between $m_{\text{low}} \leq m < 1$, the robot gradually increases its speed from a slow walk but without adjusting the phase relationships. For a β ranging between 0.89 and 0.76, the robot presents a non-singular crawl with a constant gait phase $\varphi_{\text{LH}} = 0.75$.

For $1 \leq m \leq 2.5$, the crawl slowly transfers into a trot but adjusting the phase relationship accordingly. For that, we apply the wave gait rule [21,27,29] for a quadruped: $\varphi_{\text{LH}} = \beta$. In the resulting gait, subsequent legs are lifted closely after previous ones are set down, such that the time difference between the two events is equal to zero. Hence, m modulates the gait phases by specifying the gait phase φ_{LH} (Eq. (15)) as follows:

$$\varphi_{\text{LH}} = \begin{cases} 0.75, & m < 1 \\ -0.1667m + 0.9167, & 1 \leq m < 2.5 \\ 0.5, & m \geq 2.5. \end{cases} \quad (15)$$

The gait phase remains in $\varphi_{\text{LH}} = 0.5$ for values of the modulatory drive greater than 2.5, that correspond to a $\beta = 0.5$. The robot is then with a trot gait.

6.3.4. Amplitude

Each unit-CPG amplitude is modulated such that the corresponding controlled foot describes a step with the correct step length, and thus the correct velocity. The μ parameter modulates the amplitude of each hip swing and flap unit-CPG, according to the $A_{s,i}$ and $A_{f,i}$ values, as follows ($p = \{s, f\}$):

$$\mu_{p,i} = A_{p,i}^2. \quad (16)$$

6.3.5. Step direction

The signs of $A_{s,i}$ and $A_{f,i}$ hold the information on the direction for the step movements of each unit-CPG.

Table 2
ι values for the hip swing (s) and hip flap (f) unit-CPGs.

	$\iota_{s,\text{LF}}$	$\iota_{s,\text{RF}}$	$\iota_{s,\text{LH}}$	$\iota_{s,\text{RH}}$
$A_{s,i} > 0$	1	1	1	1
$A_{s,i} < 0$	−1	−1	−1	−1
	$\iota_{f,\text{LF}}$	$\iota_{f,\text{RF}}$	$\iota_{f,\text{LH}}$	$\iota_{f,\text{RH}}$
$A_{f,i} > 0$	1	−1	−1	1
$A_{f,i} < 0$	−1	1	1	−1

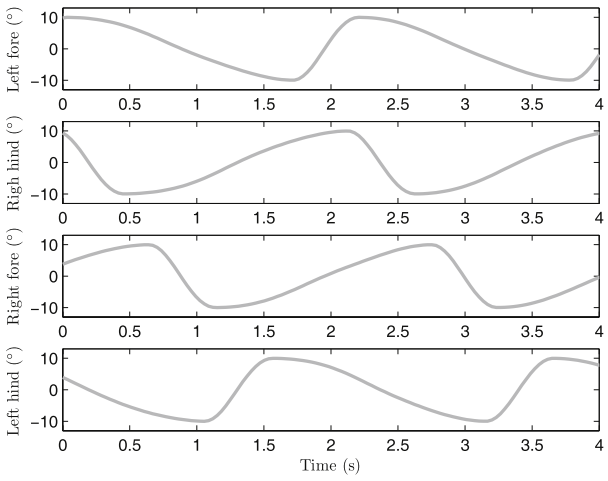


Fig. 10. Generated trajectories for flap joints. Despite the symmetric trajectories in the contralateral limbs due to the inversion of the oscillators solution, the coordination among the limbs is maintained. The sequence and timing of the trajectories are those of a walk gait.

Limit cycle direction is specified by the value of ι. Basically, changing the limit cycle direction is changing whether the step stance phase is controlled by the ascending ($\iota = -1$) or descending phase ($\iota = 1$) of the x trajectory.

When the robot walks forward ($A_s > 0$), all the swing unit-CPGs have $\iota_s = 1$, performing the stance phase during the descending trajectory of the movement. When the robot walks backwards ($A_s < 0$), the opposite happens, and the unit-CPGs have $\iota_s = -1$, performing the stance phase during the ascending trajectory.

Due to the robot's flap joints configurations other rules are required. When walking right ($A_f > 0$) and during the stance phase, the left joints perform the ascending trajectory and the right joints perform the descending trajectory. The opposite happens when walking left ($A_f < 0$).

Table 2 shows the assigned values of ι according to this set of rules, depending on the swing and flap amplitudes.

Consider that the robot is walking left ($\phi_w = 90°$). Only the flap unit-CPGs generate the rhythmic trajectories with the directions given in Table 2. The contralateral flap joints perform trajectories with inverted directions (Fig. 10).

Results show that by online modifying the CPG parameters we are able to modulate in real time the generated trajectories which end up in a smooth adaptation of the robot behavior. Further, despite fast parameter changes, the generated trajectories remain smooth.

7. Omnidirectional walking Experiments

Several experiments were performed to verify the adequacy of the locomotor CPG network both to movement generation, velocity change, to achieve omnidirectional locomotion and to verify if the resulting robot motion matches the desired, specified high level commands, i.e., walking velocity, walking orientation and angular velocity. Experiments were performed both on the webots robotics simulator and on the real AIBO robotic platform. Many possible

motions were experimented over several runs by specifying different angular velocities and walking directions. Experiments were carried with a walking direction $\phi_w \in [0, 360](^\circ)$ in steps of 5°, and angular velocity $\dot{\phi} \in [0, 0.5]$ $\left(\text{rad s}^{-1}\right)$ in steps of 0.03 rad s^{-1}.

Due to space constraints, we only depict four experiment results obtained with the real platform. Firstly, the robot walks forward and steers with a given angular velocity. Secondly, the robot walks diagonally with a given walking orientation. Next, the robot walks sideways. Fourthly, the robot moves diagonally while steering with a given angular velocity.

The AIBO dog robot is an 18 DOFs quadruped robot made by Sony. Unlike its natural counterpart this robot has three joints per limb, with different configurations of a real dog limbs. Besides, the robot body and limbs are rigid with non-compliant servo joints. The joints are stiff, without any elasticity, and their position is specified by an angle value.

We set the frequency to $\omega_{\text{sw}} = 6.28$ rad s^{-1} in regards with the motor limitations. Further, the dynamical parameters controlling the speed of convergence of unit-CPGs were set to $\frac{1}{2\,\alpha_{\text{i}}\,\mu_{\text{i}}} = 0.01$ s, in regard to stability during the integration process and to feasibility of the desired trajectories.

At each sensorial cycle, dynamic equations are calculated and numerically integrated using the Euler method with a fixed time step of 1 ms, thus specifying servo positions. The robot control loop is measured and has 8 ms.

Experiments with the robot were carried on a flat environment with a grid of markers, spaced 20 cm apart. This grid enabled us to visually measure the performed path, comparing it with the expected path, to measure the achieved translational speeds, angular velocities and walking directions.

We do not expect precise and exact motions, since the CPG approach is not intended for such a goal. We expect that the overall motion of the robot respects the specified motion commands within an acceptable but marginal error margin, specially because in this work the CPG based controller is open-loop and disregards physical effects and other disturbances. These should be used to improve the locomotion's performance.

7.1. Steering

We start by verifying the steering behavior, with a desired angular velocity. In this experiment, the robot walks forward, $\phi_w = 0^\circ$, with an angular velocity of $\dot{\phi} = -0.21$ rad s^{-1}. The modulatory drive is set to $m = 2.5$, performing a trot gait ($\beta = 0.5, \varphi_{\text{LF}} = 0.5$), where the diagonal limbs are in-phase and the stance and swing phases have equal durations. For the given β, the robot velocity is approximately $v_{\text{ref}} \approx 5.7$ cm s^{-1}. The robot is expected to perform a circle with a radius of $r_{\text{ICR}} \approx 27$ cm.

Fig. 11 presents an image composition of the experiment. It is possible to verify that the robot performs a full circle with an approximate radius of 25 cm, a marginal error when compared to the overall setup dimensions. This error is both due to the physical effects, error measurements and the open loop nature of the proposed controller.

Fig. 12 depicts actual trajectories recorded from the joints encoders and planned trajectories x_{i} of all limb joints. Trajectories were generated as expected. Because the robot is steering right, amplitudes for the right swing joints (green and red lines) are smaller than those for the left swing joints (yellow and blue lines).

In-phase trajectories are generated for the flap joints due to the robot's configuration. Left flap joints are symmetrical to the right flap joints, and because the diagonal limbs move out of phase, the generated trajectories must be in phase (middle panel of Fig. 12).

The knee joints flex and extend at the correct times, reducing the limb length while on swing and extending the limb during the stance phase (bottom panel of Fig. 12).

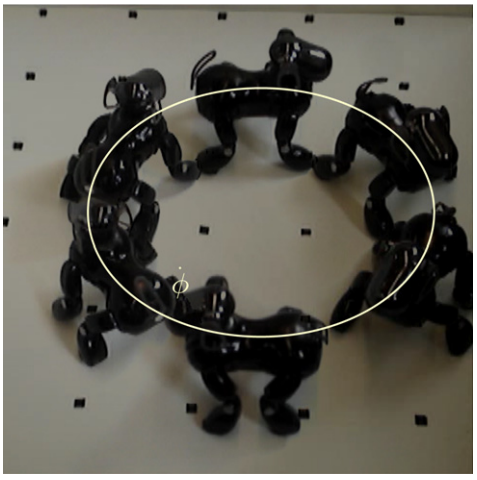


Fig. 11. Image composition when the robot steers right when $\dot{\phi} = -0.21$ rad s^{-1} and $\phi_w = 0^\circ$. The resulting radius is ≈25 cm.

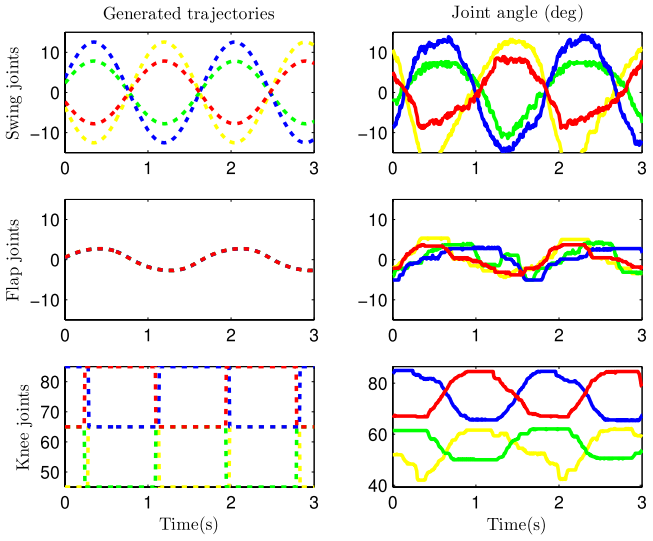


Fig. 12. Planned trajectories (left) and actual performed movements (right) for hip swing, hip flap and knee joints for the steering motion depicted in Fig. 11. Left fore; right fore; left hind; and right hind are depicted in yellow, green, blue and red lines, respectively. All the generated trajectories respect the coordination constraints imposed by the couplings. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The platform's weight and the resultant dynamics from forces on the limbs influence the performed joint angles. For instance, right fore knee does not extend to the planned value. Considering forelimbs swing joints, during the stance phase, the weight of the robot induces a further extension of the joint.

Highest discrepancies between generated and performed trajectories can be seen in flap joints. While generated trajectories overlap (left panel), this does not happen in performed movements (right panel). This difference is mainly due to the forces exerted over the limbs and to the fact that flap joints have less torque than swing joints.

Despite these discrepancies, main features of the generated trajectories are still present on the performed joint trajectories, and the desired robot motion was achieved.

7.2. Diagonal walk

In this experiment, we verify the robot ability to walk with a given walking direction. The robot performs a slow walking gait

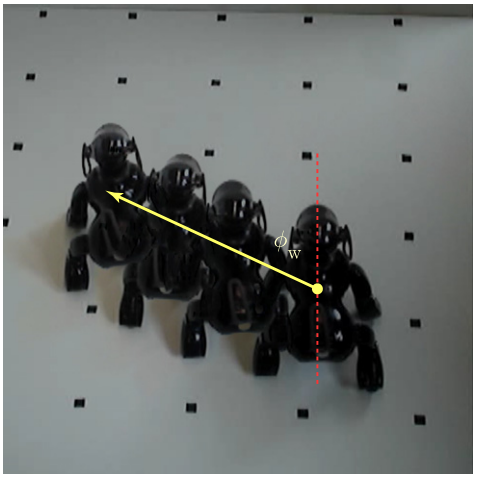


Fig. 13. Image composition of four individual snapshots when the robot walks slowly ($m = 0.4$) forward diagonally to the left. It describes the path with $\dot{\phi} = 0$ rad s^{-1} and $\phi_w = -45°$.

given by $m = 0.4$, yielding a $\beta = 0.8$ and $\varphi_{\text{LF}} = 0.75$, meaning $v_{\text{ref}} \approx 1.4$ cm s^{-1}.

An angular velocity $\dot{\phi} = 0°$ and a walking orientation $\phi_w = -45°$ are specified, such that the robot is expected to walk diagonally, forward to the left, with equal forward and lateral velocities and no angular motion.

We set $v_{s,i} = 1$ and $v_{f,i} = 1$, $i \in \{\text{LF, RF, LH, RH}\}$, such that hip swing and flap joints have rhythmic movements. Forward motion is achieved by choosing the oscillators direction in the hip swings, $\iota_{s,i} = 1$. Right lateral motion is achieved by setting the direction of the hip flap oscillators to $\iota_{f,\text{LF}} = \iota_{f,\text{RH}} = -1$ and $\iota_{f,\text{RF}} = \iota_{f,\text{LH}} = 1$.

Both swing and flap joints should perform stepping movements with the same amplitudes, propelling the robot equally forward and left.

Fig. 13 shows an image composition of the resulting robot behavior, a motion close to $-45°$ diagonally.

Fig. 14 depicts planned (left panel) and performed (right panel) trajectories for hip swing, hip flap and knee joints. As expected, swing and flap trajectories have equal amplitudes. Further, intralimb and interlimb joints are coordinated as required. Hip swings and hip flaps are synchronized because swing and stance step phases are in phase. Directions in left fore and hind limbs are inverted as specified in the parameters.

Generated and performed joint movements are much more similar than on the steering experiment because since the frequency of movements is lower and the used gait is much more stable, the resulting dynamics of the body exert less force on limb joints.

These results show that the proposed controller is well-suited for the generation of synchronized movements between the joints in a limb (intralimb coordination) and between the joints of different limbs (interlimb coordination).

7.3. Walking laterally

In this experiment we demonstrate the robot capacity to walk laterally with a walking orientation $\phi_w = 90°$. The robot performs a slow walking gait according to $m = 1$, yielding a $\beta = 0.8$ and $\varphi_{\text{LF}} = 0.75$. Thus $v_{\text{ref}} \approx 1.35$ cm s^{-1}.

This requires the hip swing joints to be at rest, $v_{s,i} = -1$, $i \in \{\text{LF, RF, LH, RH}\}$, while the hip flap joints perform the rhythmic locomotor movement, $v_{f,i} = 1$. In order for the robot to move to the left, hip flap joint directions are set as $\iota_{f,\text{RF}} = \iota_{f,\text{RH}} = 1$ and $\iota_{f,\text{LF}} = \iota_{f,\text{LH}} = -1$.

Fig. 15 shows snapshots of the robot walking laterally.

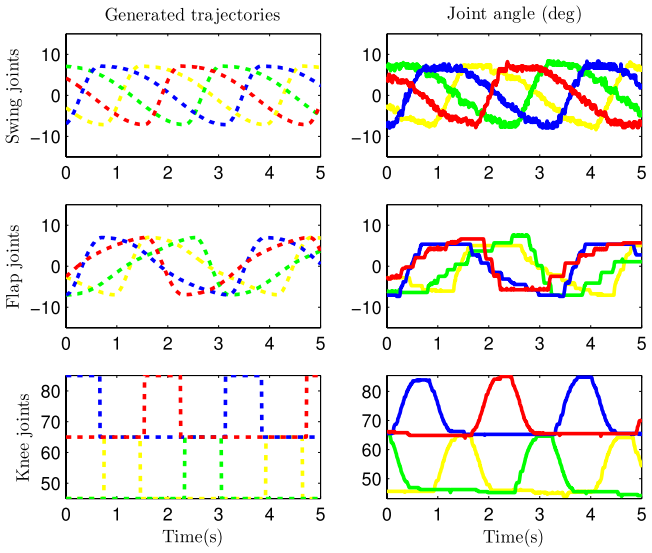


Fig. 14. Planned trajectories (left) and actual performed movements (right) for hip swing, hip flap and knee joints for the diagonal walk motion depicted in Fig. 13. Left fore; right fore; left hind; and right hind are depicted in yellow (lightest), green, blue (darkest) and red lines, respectively. The robot closely performs the planned trajectories. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

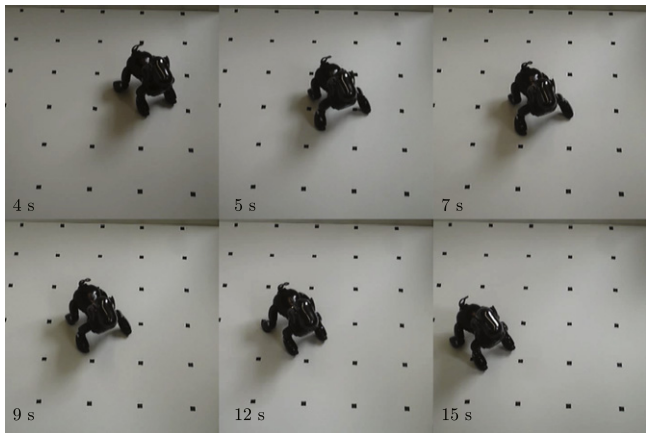


Fig. 15. Snapshots depicting the robot trotting sideways to its right with an approximate velocity of 1.35 cm s^{-1}. We specify a walking orientation of $\phi_w = 90°$.

7.4. Steering diagonally

If the robot moves with a certain walking orientation and angular velocity, the robot will perform a circular path while not heading straight forward, i.e., the body orientation will not be tangential to the circle specified by the angular velocity.

In this experiment, the robot moves with a trot gait, specified by $\beta = 0.5$ and $\varphi_{\text{LF}} = 0.5$ ($m = 2.5$), achieving a $v_{\text{ref}} \approx 5.7$ cm s^{-1}. The walking orientation is set to $\phi_w = 65°$ and angular velocity to $\dot{\phi} = 0.27$ rad s^{-1}. The robot is expected to walk while turning, heading about $\approx 65°$ to the center of its circular path with 22 cm radius. The obtained robot path is shown in Fig. 16.

Similarly to previous experiments, flap joints have more difficulty in following the planned trajectories. However, the performed trajectories do present the features of the planned trajectories, even though the forces exerted on the limbs, due to the body dynamics, influence the angles of the joints (Fig. 17).

Despite the discrepancies between generated and performed trajectories, the robot does perform the desired path with an heading orientation that approximates the desired one.

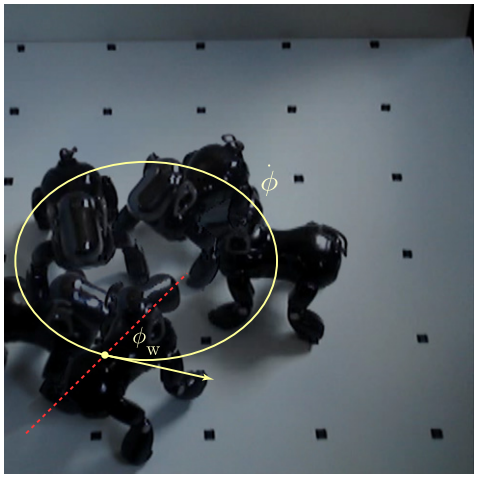


Fig. 16. Image composition of four snapshots when the robot steers while heading to the center of the path, with $\dot{\phi} = 0.27$ rad s^{-1} and $\phi_w = -65°$. It walks diagonally to the right while steering right, in a path with a radius of ≈22 cm.

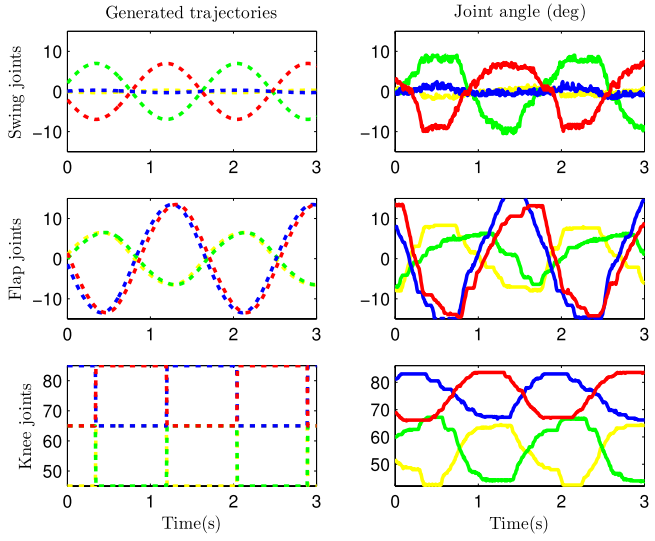


Fig. 17. Planned trajectories (left) and actual performed movements (right) for hip swing, hip flap and knee joints for the robot steering diagonally motion depicted in Fig. 16. Left fore – yellow; right fore – green; left hind – blue; and right hind – red. The joints do not correctly follow the planned trajectories. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

7.5. Discussion

These experiments have shown that by specifying m; the walking orientation ϕ_w; and the angular velocity $\dot{\phi}$; the required movements for limbs in order to achieve omnidirectional locomotion and velocity change can be fully characterized.

Further, these experiments suggest that the calculation of the movement amplitudes and directions of each limb, despite being based on an approximate velocity value, can successfully be used to modulate the CPG parameters in terms of the desired translational speed, angular velocity and walking orientation. The generated trajectories are modulated as required and the robot is able to perform omnidirectional locomotion.

However, the performed locomotion showed some inconsistencies in performance. For instance, sometimes a foot did not lift from the ground, or the robot fall over a foot during its swing phase. These performance inconsistencies happen when generated limb movements are not appropriate for the body dynamics at a given instant, given the linear and angular momentum. Therefore, performance is largely dependent on movements amplitude and direction during omnidirectional locomotion, which are specified according to the desired gait, walking orientation and angular velocity.

Note, however, that the proposed controller generates limb trajectories in an open-loop fashion, and does not take into account neither the body dynamics or any type of sensory information. In the future, these should be included onto the model, by closing the control loop. Nonetheless, the robot performs the desired locomotion and the obtained results have been quite satisfactory, especially considering the simplicity of the system. The underlying idea is that such an architecture is functionally organized in separate layers of abstraction. Higher levels specify the desired motor patterns which are then translated to correct sets of CPG parameters and sent to the lower level.

Herein we do not consider the important problem of selecting the best omnidirectional movement. For instance, in order to walk diagonally the robot could either walk diagonally or instead rotate in spot and then walk forward in the desired direction. This selection and performing this successive gait transition will be addressed in future work. Herein we consider the robot is able to perform movement in any direction and address the gait generation required to implement this omnidirectional movement in an AIBO quadruped robot. We also address the robot navigation ability such that the robot is capable of reaching a target within a cluttered environment.

8. Navigation application

If the omnidirectional locomotion motor program (MP) is expected to be useful in a behavioral context, it has to be more adapted to the environment, and a more complex form of sensory motor integration is required. In this section, we address the robot navigation ability such that the robot is capable of reaching a target within a cluttered environment.

Visual information is acquired through a fixed camera mounted on the head of the robot, facing. Target and obstacles are represented in the world by green and red colored balls, respectively, with known dimensions. The location of the target and obstacles is continuously extracted from visual segmented information.

Since this work is focused on real-time visual control rather than scene understanding, image processing has been simplified by working in a structured environment, returning to the system the object distance and its angle to the robot.

Steering is based on a dynamical system that determines the heading direction angle (steering angle) and paths that enable the robot to circumnavigate obstacles and find their way to a target, while at all times the dynamical variable sits in a fixed point attractor. Extensive studies about this system can be found in [12,30,40,56].

This dynamical system provides the robot's desired heading direction ϕ_h.

The robot's angular velocity $\dot{\phi}$, one of the commands that specify the desired omnidirectional locomotion, is the output of another dynamical system based on the discrepancy of the robot's desired and current heading directions.

The desired tonic drive (m) is herein given by another, very simple, dynamical system, that assures that in case obstacles are present, the robot's velocity is reduced as required.

The third and last command, the walking orientation (ϕ_w) is kept constant, with the robot always facing forward.

8.1. Heading direction dynamics

The robot's heading direction, ϕ_h, in angular space and in an allocentric coordinate, is controlled by a nonlinear vector field in which task constraints contribute independently. The task of

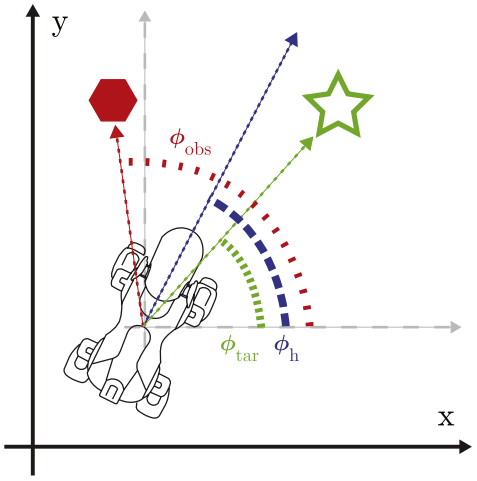


Fig. 18. The angles ϕ_h, ϕ_{tar} and ϕ_{obs} are measured in the allocentric coordinate system (X, Y).

reaching the target, $F_{tar}(\phi_h)$, attracts ϕ_h towards the direction in which the target lies. The task of avoiding obstacles, $F_{obs}(\phi_h)$, repels ϕ_h from the direction in which obstacles are perceived. These variables are depicted in Fig. 18.

Integration of these tasks is achieved by adding each of them to the vector field that governs heading direction dynamics.

$$\dot{\phi_h} = F_{obs}(\phi_h) + F_{tar}(\phi_h) + F_{stoch}. \tag{17}$$

A stochastic component, F_{stoch}, is added to ensure an escape from unstable states. For a full discussion see [12,30,56] for examples.

This approach differs from the potential field approach basically in the aspect that the state of the behavioral system must be in or near an attractor state of the dynamical system during operation.

8.1.1. Target reaching

The following dynamics are formulated for the task of target reaching

$$F_{tar}(\phi_h) = -\lambda_{tar} \sin(\phi_h - \phi_{tar}). \tag{18}$$

This dynamical system erects an attractive forcelet at the direction $\phi_h = \phi_{tar}$, specifying the position of an attractor in the heading direction dynamics.

The parameter λ_{tar} (> 0) controls speed of convergence of the target attractor in the heading direction dynamics.

8.1.2. Obstacle avoidance

In this case, we wish to steer away from the detected obstacles. A repulsive-force, $f_{obs,i}(\phi_h)$, centered at $\phi_{obs,i}$ is erected for each obstacle i detected, and summed up for the overall obstacle avoidance dynamics

$$F_{obs}(\phi_h) = \sum f_{obs,i}(\phi_h) = \sum \lambda_{obs,i}(\phi_h - \phi_{obs,i}) e^{-\frac{(\phi_h - \phi_{obs,i})^2}{2\sigma_i^2}}. \tag{19}$$

Parameter $\lambda_{obs,i}$ controls speed of convergence of each erected repulsive-force and decays exponentially with the distance between obstacles and the sensors.

Parameter σ_i defines the angular range over which each obstacle forcelet exerts its repulsive effect.

Precedence of obstacle avoidance is accomplished making the strength of the obstacles contribution stronger than the target contribution.

The steering dynamics is given by

$$\dot{\omega} = -\alpha_\omega(\omega - \Delta_\phi)\left[1 + (\omega - \Delta_\phi)^2\right], \tag{20}$$

that formulates a simple attractor $(\alpha_\omega > 0)$ at Δ_ϕ, specifying the robot's angular rate with the goal to steer towards the desired heading direction. $\Delta_\phi = \phi_h - \phi_r$, where ϕ_r is the robot's actual heading direction.

8.2. Velocity dynamics

The robot's gait, and its velocity, is controlled by the tonic drive command $m \in [0, 2.5]$, which is afterwards mapped onto the required CPG parameters. The robot has to slow down when coming close to an obstacle, changing from the trot gait $(m = 2.5)$, used when the path is clear, to a walk, or even crawl when it approaches an obstacle. The following dynamics for the m variable

$$\dot{m} = -\alpha_m\left(m - \left(\frac{2}{1 + e^{-m_a(\min(d_{c,i}) - m_c)}}\right)\right), \tag{21}$$

shifts the fixed points depending on the current tonic drive and the distance to obstacles within a m_c range. By setting m_a and m_c parameters, we specify the curve response for m on the minimum distance over the all the detected obstacles $(\min(d_{c,i}))$. Parameter α_m sets the convergence of m to the solution.

8.3. Experiment

We are particularly interested in the architecture ability to switch between and combine MPs. This results on switching or sequencing between discrete and rhythmic movements (stopping/initiating locomotion) and superimposing both behaviors (gait transition and omidirectional locomotion).

In this section we present the simulation results for an autonomous omnidirectional locomotion with a navigation dynamical system. These simulations were performed in Webots.

Experiments are delineated such that the selection between the MPs is triggered according to external stimuli. The robot walks in an environment towards a visually acquired target. During its path, obstacles may appear and force the robot to adjust its motion in order to successfully reach the target. This may oblige to object circumnavigation. The navigation system provides for the robot angular velocity and the required translational speed. These together with the robot walking direction, modulate the amplitudes of the flap and swing oscillators. The CPG parameters are modulated accordingly and the joint's movements are generated such that the robot reaches the target, as required.

At each sensorial cycle, dynamic equations of heading direction and motor control are calculated and numerically integrated using the Euler method with a fixed time step of 1 ms, thus specifying servo positions. The robot control loop is measured and has 8 ms.

The built-in camera in the AIBO's head is used to detect online both obstacles and the goal. These are balls with known size, with distinct colors for the obstacles (red) and for the goal (green). Simple image processing is used to gather the distance and the angle of visible objects. The goal is always located higher such that obstacles do not block its visibility.

Consider the experiment depicted in Fig. 19, in which while the robot walks towards the target, two obstacles block its most direct route. The video can be seen in [57]. Fig. 20 depicts the distance to obstacles (top) and the m tonic drive (bottom). ϕ_h, ϕ_{tar}, and two different ϕ_{obs} (top) and ω (bottom) are shown in Fig. 21.

Initially, at $t = 0$ s, the robot is stopped $(m = 0)$. The robot faces both the target and two obstacles. The tonic drive increases towards $m = 2.5$, to reach the nominal trot gait (Fig. 20, left). As the robot approaches the obstacles $(d_c < 1$ m), it adjusts the gait

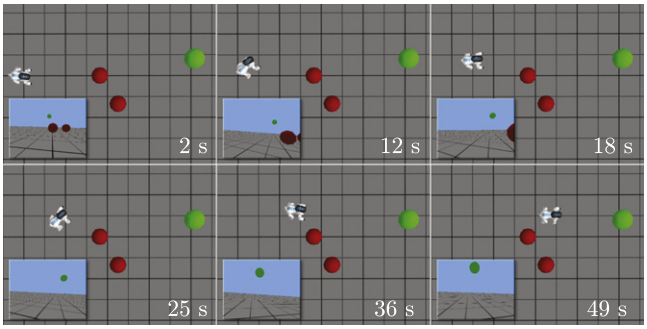


Fig. 19. Six top-view relevant snapshots of the simulation. Time increases from left to right, top to bottom. The robot avoids the two red obstacles, and walks towards the green target. Snapshots of the camera view at those instances are presented at the left down corner of each snapshot.

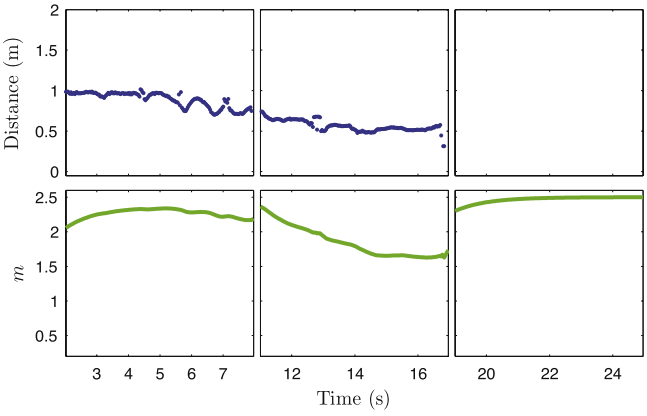


Fig. 20. Distance to obstacles (top) and the m tonic drive (bottom) for the simulation depicted in Fig. 19. CENTER: as the robot approaches an obstacle, it adapts its gait to slow down the locomotion. RIGHT: when no obstacle is detected, it walks with its nominal trot gait.

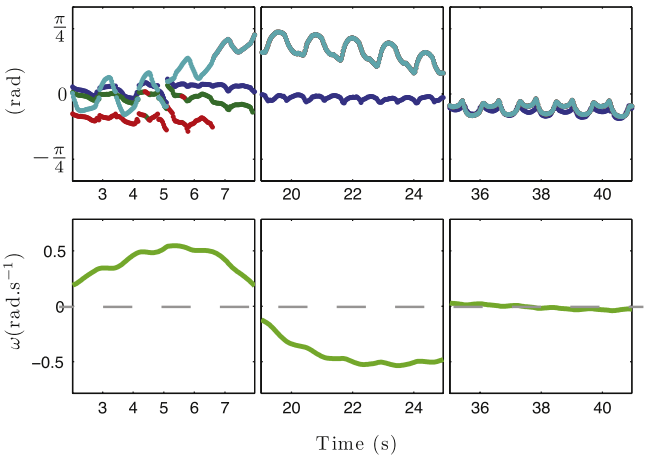


Fig. 21. ϕ_h, ϕ_{tar}, and two different ϕ_{obs} (top) and ω (bottom) variables for several intervals of time along the simulation depicted in Fig. 19. LEFT: the robot steers left ($\omega > 0$) to avoid the obstacles on its right. MIDDLE: after clearing the obstacles, it steers back to the target. RIGHT: when facing the target, it walks straight forward ($\omega > 0$).

in order to slow down and safely steer away (Fig. 20, middle). The gait is only slightly adjusted, never reaching the walk gait ($m < 1$).

The robot walks towards the target, but steers away from the obstacle (snapshot at 12 s). Obstacles are to the left of the robot (red and green lines), so it steers left ($\omega > 0$). However, while it steers away from the obstacles, it also steers away from the target (dark blue line) (Fig. 21, left).

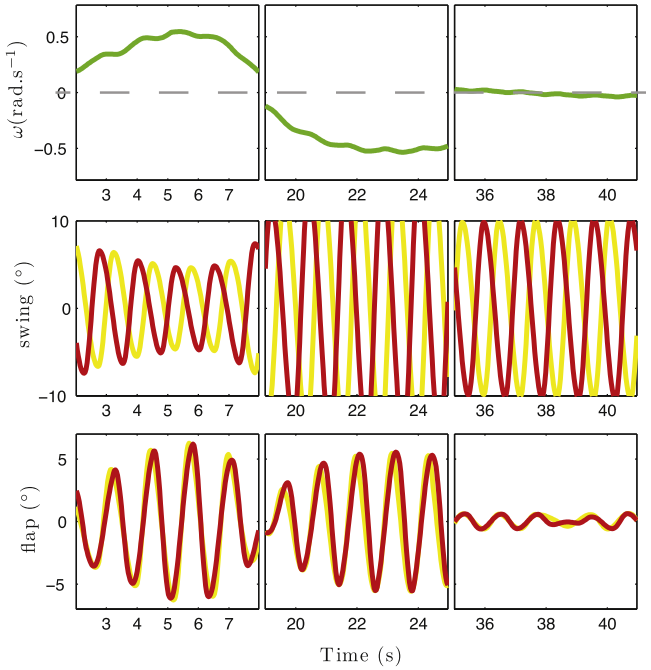


Fig. 22. ω variable (TOP) and trajectories of the hip swing (MIDDLE) and flap joints (BOTTOM) of the fore left (yellow) and hind left (red) legs for several intervals of time along the simulation depicted in Fig. 19. Left swing movements decrease when steering left ($\omega > 0$), and increase when steering right ($\omega < 0$). Flap rhythmic movements are performed when steering, with opposite step directions for steering left and right. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

After clearing the obstacles, the robot steers back to the target's direction (Fig. 21, middle). This situation can also be visualized in snapshots at $t = 18, 25$ s in Fig. 19.

In the final moments of the simulation, $t \approx 49$ s, there are no more obstacles, and the robot walks with $\omega = 0$ rad s^{-1} (Fig. 21, right) towards the target. The tonic drive is increased towards 2, performing a trot (Fig. 20, right).

When the robot steers to the left ($\omega > 0$) the amplitudes of hip swing on the left decreases, while on right side increases, and the hip flap swing starts to move (Fig. 22, left). All these coordinated movements of swing and flap joints steer the robot left.

When steering right ($\omega > 0$), the robot increases the left hip swing amplitudes, while the flap joints invert their step direction (Fig. 22, middle).

After facing the target, the robot walks forward. Hip swing movements have the same amplitudes on both sides, and the hip flap joints stop their movement (Fig. 22, right).

Finally, at $t \approx 60$ s, the robot autonomously reaches the target, after switching among the required motor programs.

9. Conclusion

In this article, we have proposed a bio-inspired architecture to: (1) generate the required coordinated trajectories for locomotion; (2) to modulate the generated motor patterns according to a velocity increase; (3) to achieve omnidirectional locomotion in terms of the walking velocity, given orientation and angular velocity; and (4) to navigate in the sensed environment (steering). Further, the switch among the different motor programs (MPs) was smooth and easily elicited according to visual sensory information.

We have achieved omnidirectional locomotion in a non-spine robot, by a combined use of the flap and swing hip joints. The movement generated by a limb-CPG results from the coupling of swing and flap oscillators. The proposed CPG was designed using

an Hopf nonlinear oscillator, which presents several desirable advantages in robotic applications. These include the possibility of integrating sensory feedback mechanisms already presented in other works [11,19] and the possibility to explore new ones. Its application also allows the coordination of all the limbs in all the omnidirectional motions.

In this work it was also presented a method for modulating all the CPG's parameters, requiring hand tuning for just a small number of parameters. A steering command modulates activity of the steering MP through modulation of the flap and swing amplitudes. The generated movements are modulated through higher level commands that encode the desired walking motion in terms of the translational speed, the walking orientation and the angular velocity; therefore reducing the dimensionality of the control problem.

We demonstrate the system working on a goal-oriented locomotion task. The robot walks towards a given target while avoiding obstacles, by steering and changing its velocity. The obtained results demonstrated that the proposed controller is capable of generating the required limb movements navigation including omnidirectional locomotion and gait switching.

The presented work takes part of a larger project which aims at developing a closed loop control architecture based on dynamical systems for the autonomous generation of robust, adaptive goal-directed quadruped locomotion in unknown, rough terrain.

For generating adaptive locomotion we are currently addressing accurate feet placement; predictive adjustments of locomotion including speed and/or step length control in advance and head stabilization for image acquisition. This work should also be integrated with our work for posture and balance control [11].

Acknowledgments

The authors work at the Algoritmi Center of the University of Minho. This research was supported by the Foundation for Science and Technology(FCT) under the project PTDC/EEA-CRO/100655/2008, and PhD grant SFRH/ BD/62047/2009.

References

[1] M. Vukobratovic, B. Borovac, Zero-moment point – thirty five years of its life, International Journal of Humanoid Robots 1 (1) (2004) 157–173.
[2] J. Pratt, C. Chew, A. Torres, P. Dilworth, G. Pratt, Virtual model control: an intuitive approach for bipedal locomotion, The International Journal of Robotics Research 20 (2) (2001) 129–143.
[3] A.J. Ijspeert, J. Nakanishi, S. Schaal, Learning Attractor Landscapes for Learning Motor Primitives, in: Advances in Neural Information Processing Systems, Vol. 15, MIT Press, 2002, pp. 1547–1554.
[4] R. Blickhan, The spring-mass model for running and hopping, Journal of Biomechanics 22 (11–12) (1989) 1217–1227.
[5] S. Schaal, S. Kotosaka, D. Sternad, Nonlinear dynamical systems as movement primitives, in: International Conference on Humanoid Robotics, Springer, Cambridge, MA, 2001, pp. 117–124.
[6] E. Bizzi, A. D'Avella, P. Saltiel, M. Tresch, Modular organization of spinal motor systems, Neuroscientist 8 (5) (2002) 437–442.
[7] M. Tresch, P. Saltiel, E. Bizzi, The construction of movement by the spinal cord, Nature Neuroscience 2 (2) (1999) 162–167.
[8] J. Tani, M. Ito, Y. Sugita, Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using rnnpb, Neural Network 17 (2004) 1273–1289.
[9] H. Kimura, Y. Fukuoka, A.H. Cohen, Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts, International Journal of Robotics Research 26 (2007) 475–490.
[10] S. Degallier, L. Righetti, A. Ijspeert, Hand placement during quadruped locomotion in a humanoid robot: a dynamical system approach, IEEE-RAS International Conference on Intelligent Robots and Systems (IROS07), 2007.
[11] J. Sousa, V. Matos, C. Santos, A bio-inspired postural control for a quadruped robot: an attractor-based dynamics, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei International Convention Center, Taipei, Taiwan, 2010.
[12] C.P. Santos, Generating timed trajectories for an autonomous vehicle: a non-linear dynamical systems approach, ICRA, IEEE (2004) 3741–3746.
[13] S. Degallier, C. Santos, L. Righetti, A. Ijspeert, Movement generation using dynamical systems: a humanoid robot performing a drumming task, IEEE-RAS International Conference on Humanoid Robots, 2006.
[14] S. Grillner, Locomotion in vertebrates: central mechanisms and reflex interaction, Physiological Reviews 55 (1975) 247–304.
[15] M. MacKay-Lyons, Central pattern generation of locomotion: a review of the evidence, Physical Therapy 82 (1) (2002) 69–83.
[16] S. Grillner, P. Walléna, K. Saitoha, A. Kozlova, B. Robertsona, Neural bases of goal-directed locomotion in vertebrates: an overview, Brain Research Reviews 57 (1) (2008) 2–12.
[17] S. Grillner, J. Hellgren, A. Ménard, K. Saitoh, M.A. Wikström, Mechanisms for selection of basic motor programs – roles for the striatum and pallidum, Trends Neurosciences 28 (7) (2005) 364–370.
[18] S. Degallier, L. Righetti, L. Natale, F. Nori, G. Metta, A. Ijspeert, A modular bio-inspired architecture for movement generation for the infant-like robot iCub, in: Proceedings of the 2nd IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2008.
[19] L. Righetti, A.J. Ijspeert, Pattern generators with sensory feedback for the control of quadruped locomotion, 2008 IEEE International Conference on Robotics and Automation, 2008.
[20] G. Schöner, C. Santos, Control of movement time and sequential action through attractor dynamics: a simulation study demonstrating object interception and coordination, SIRS 2001 (2001).
[21] V. Matos, C.P. Santos, C.M.A. Pinto, A brainstem-like modulation approach for gait transition in a quadruped robot, in: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 2665–2670.
[22] V. Matos, C. Santos, Omnidirectional locomotion in a quadruped robot: a cpg-based approach, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei International Convention Center, Taipei, Taiwan, 2010.
[23] B. Hengst, D. Ibbotson, S.B. Pham, C. Sammut, Omnidirectional locomotion for quadruped robots, in: RoboCup 2001: Robot Soccer World Cup V, Springer-Verlag, London, UK, 2002, pp. 368–373.
[24] K. Yoneda, S. Hirose, Dynamic and static fusion gait of a quadruped walking vehicle, Brain Research Reviews 9 (3) (1991) 7–15.
[25] A. Sano, J. Furusho, Static-dynamic transitional gait from crawl to pace, ROBOMEC'92 B (1992) 239–246.
[26] H. Yasa, M. Ito, An autonomous decentralized system with application to a gait pattern generator, Transactions of the Society of Instrument and Control Engineers 26 (12) (1989) 180–187.
[27] K. Inagaki, H. Kobayashi, A gait transition for quadruped walking machine, in: Proceeding of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Sytems.
[28] F. Hardarson, Stability analysis and synthesis of statically balanced walking for quadruped robots, Ph.D. Thesis, KTH 2002.
[29] R.B. Mcghee, A.A. Frank, On the stability properties of quadruped creeping gaits, Mathematical Biosciences 3 (1–2) (1968) 331–351.
[30] E. Bicho, P. Mallet, G. Schoner, Target representation on an autonomous vehicle with low-level sensors, The International Journal of Robotics Research 19 (5) (2000) 424–447.
[31] A. Steinhage, G. Schöner, Dynamic approach to autonomous robot navigation, the, in: ISIE"97, IEEE International Symposium On Industrial Electronics, 1997.
[32] F. Delcomyn, Neural basis for rhythmic behaviour in animals, Science (210) (1980) 492–498.
[33] G. Brambilla, J. Buchli, A. Ijspeert, Adaptive four legged locomotion control based on nonlinear dynamical systems, in: From Animals to Animats 9, Proceedings of the Ninth International Conference on the Simulation of Adaptive Behavior (SAB'06), in: Lecture Notes in Computer Science, vol. 4095, Springer Verlag, 2006, pp. 138–149.
[34] J. Buchli, A.J. Ijspeert, Self-organized adaptive legged locomotion in a compliant quadruped robot, Autonomous Robots 25 (4) (2008) 331–347. doi:10.1007/s10514-008-9099-2.
[35] A.J. Ijspeert, Central pattern generators for locomotion control in animals and robots: a review, Neural Networks 21 (4) (2008) 642–653.
[36] G. Taga, Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment, in: Proceedings of the NATO Advanced Research Workshop and EGS Topical Workshop on Chaotic Advection, Tracer Dynamics and Turbulent Dispersion, Elsevier North-Holland, Inc., New York, NY, USA, 1994, pp. 190–208.
[37] A. Rizzi, D. Koditschek, Further progress in robot juggling: solvable mirror laws, IEEE International Conference on Robotics and Automation (1994) 2935–2940.
[38] L. Righetti, A. Ijspeert, Design methodologies for central pattern generators: towards 'intelligent' locomotion in robots, Proceedings of 50th Anniversary Summit of Artificial Intelligence (2006).
[39] M. Golubitsky, I. Stewart, P. Buono, J. Collins, A modular network for legged locomotion, Physica D 115 (1–2) (1998) 56–72.
[40] C.P. Santos, M. Ferreira, Two vision-guided vehicles: temporal coordination using nonlinear dynamical systems, ICRA, IEEE (2007) 14–19.
[41] C. Santos, M. Ferreira, Timed trajectory generation using dynamical systems: application to a puma arm, Robotics and Autonomous Systems 57 (2) 2009 182–193, selected papers from 9th International Conference on Intelligent Autonomous Systems (IAS-9), 9th International Conference on Intelligent Autonomous Systems.

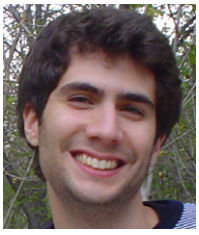
[42] C. Santos, M. Oliveira, A.M.A. Rocha, L. Costa, Head motion stabilization during quadruped robot locomotion: combining dynamical systems and a genetic algorithm, IEEE International Conference on Robotics and Automation (2009).
[43] C. Liu, J. Su, Basic behavior acquisition based on multisensor integration of a robot head, ICRA (2008) 3094–3099.
[44] N. Kohl, P. Stone, Machine learning for fast quadrupedal locomotion, in: AAAI'04: Proceedings of the 19th National Conference on Artificial Intelligence, AAAI Press/The MIT Press, 2004, pp. 611–616.
[45] T. Fukuda, Y. Adachi, H. Hoshino, E. Muro, K. Kurashige, Omnidirectional walking mechanism : redundancy and trajectory control, JSME International Journal, Series C, Mechanical Systems, Machine Elements and Manufacturing, 40 (4) (19971215) 694–701.
[46] L. Zhang, S. Ma, K. Inoue, Several insights into omnidirectional static walking of a quadruped robot on a slope, IROS (2006) 5249–5254.
[47] X. Chen, K. Watanabe, K. Kiguchi, K. Izumi, Implementation of omnidirectional crawl for a quadruped robot, Advanced Robotics 15 (2) (2001) 169–190.
[48] S. Chernova, M. Veloso, An evolutionary approach to gait learning for four-legged robots, In Proceedings of IROS'04 (2004).
[49] G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, M. Fujita, Evolving Robust Gaits with Aibo (2000).
[50] Z.J. Kolter, A. Ng, Learning omnidirectional path following using dimensionality reduction, in: Proceedings of Robotics: Science and Systems, Atlanta, GA, USA, 2007.
[51] K. Tsujita, H. Toui, K. Tsuchiya, Dynamic turning control of a quadruped locomotion robot using oscillators, Advanced Robotics 19 (19) (2005) 1115–1133.
[52] H. Kimura, Y. Fukuoka, H. Katabuti, Mechanical design of a quadruped "tekken3 & 4" and navigation system using laser range sensor, in: Proceedings of International Symposium of Robotics, Tokyo, 2005.
[53] P. Manoonpong, F. Pasemann, F. Wörgötter, Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines, Robotics and Autonomous Systems 56 (3) (2008) 265–288.
[54] S. Steingrube, M. Timme, F. Woergoetter, P. Manoonpong, Self-organized adaptation of simple neural circuits enables complex robot behavior, Nature Physics 6 (2010) 224–230. doi:10.1038/nphys1508.
[55] O. Kiehn, Locomotor circuits in the mammalian spinal cord, Annual Review of Neuroscience 29 (2006) 279–306.
[56] G. Schöner, M. Dose, A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion, Robotics and Autonomous Systems 10 (4) (1992) 253–267.
[57] C.P. Santos, V. Matos, Simulation experiment video, http://intranet.dei.uminho.pt/asbgroup/uploads/omnidirectional_locomotion_journal.avi (12 2010).

Cristina P Santos: received her B.S. degree in Industrial Electronics, her M.Sc degree in Robotics, and her Ph.D. degree in Robotics in the field of Nonlinear dynamics, all from the University of Minho, Guimaraes, Portugal, in 1994, 1998 and 2003 respectively.

The Ph.D was also in collaboration with the CNRS-CNRC Marseille, France.

She is currently working as an Auxiliar Professor at the University of Minho, Industrial Electronics Department.

Her research focuses on the extension of the use of the dynamical systems theory to the achievement of more complex behavior for robots: generating locomotion for multi-dof robots; achieving cooperativity among multi-robots and learning.



Vitor Matos: is a Ph.D student in Robotics at University of Minho. He holds an M.Sc. degree in Industrial Electronics from the University of Minho since 2009.

His current research includes robot locomotion and dynamical systems theory for achieving generation of rhythmic and discrete motions and behaviors.