2nd International Workshop on Evolutionary and Reinforcement Learning for Autonomous Robot Systems (ERLARS 2009)
St. Louis, Missouri, USA, October 15 2009

Combining Central Pattern Generators with the Electromagnetism-like Algorithm for Head Motion Stabilization during Quadruped Robot Locomotion

Cristina P. Santos, Miguel Oliveira, Vitor Matos, Ana Maria A.C. Rocha and Lino Costa

***Abstract*— Visually-guided locomotion is important for autonomous robotics. However, there are several difficulties, for instance, the head shaking that results from the robot locomotion itself that constraints stable image acquisition and the possibility to rely on that information to act accordingly.**

In this article, we propose a controller architecture that is able to generate locomotion for a quadruped robot and to generate head motion able to minimize the head motion induced by locomotion itself. The movement controllers are biologically inspired in the concept of Central Pattern Generators (CPGs). CPGs are modelled based on nonlinear dynamical systems, coupled Hopf oscillators. This approach allows to explicitly specify parameters such as amplitude, offset and frequency of movement and to smoothly modulate the generated oscillations according to changes in these parameters. We take advantage of this particularity and propose a combined approach to generate head movement stabilization on a quadruped robot, using CPGs and a global optimization algorithm. The best set of parameters that generates the head movement are computed by the electromagnetism-like algorithm in order to reduce the head shaking caused by locomotion.

Experimental results on a simulated AIBO robot demonstrate that the proposed approach generates head movement that does not eliminate but reduces the one induced by locomotion.

I. INTRODUCTION

Robot locomotion is a challenging task that involves several relevant subtasks, not yet completely solved. The motion of quadruped, biped and snake-like robots, for instance, with cameras mounted in their heads, causes head shaking.This kind of disturbances, generated by locomotion itself, makes it difficult to keep the visual frame stable and, therefore, to act according to the visual information. Head stabilization is very important for achieving a visually-guided locomotion, a concept which has been suggested from a considerable number of neuroscientific findings in humans and animals [18].

As a basic research to realize visually-guided quadruped locomotion, we aim in this article at head stabilization of a quadruped robot that walks with a walking gait. In our research, we propose a motion stabilization system of an ers-7 AIBO quadruped robot, which performs its own head motion according to a feedforward controller. Several similar works have been proposed in literature [4], [7], [6], [5]. But these methods consider that the robot moves according to a scheduled robot motion plan, which imply that space and time constraints on robot motion must be known before hand as well as robot and environment models. As such, control is based on this scheduled plan. Other works have successfully achieved gaze stabilization [5], that consists on image stabilization during head movements in space. The overall of the gaze stabilization approaches can be divided into two types of techniques. One uses specific hardware, like accelerometers and gyroscope to estimate the 3D posture of the head, and complex control algorithms to compensate the oscillations. The use of inertial information was already proposed by several authors [5], [16], [17]. Typically this kind of techniques is used in binocular robot heads, where gaze is implemented through the coordination of the two eye movements. Most of the approaches are inspired in biological systems, specifically in the human Vestibular-Ocular Reflex (VOR). In robots with fixed eyes, the fixation point procedure is achieved by compensatory head or body movements, based on multisensory information of the head.

In this work, a combined approach to generate head movement stabilization on a quadruped robot, using Central Pattern Generators (CPGs) and the electromagnetism-like algorithm is proposed. We intend to use a head controller, based on Central Pattern Generators (CPGs), that generates trajectories for tilt, pan and nod head joints. CPGs are neural networks located in the spine of vertebrates, able to generate coordinated rhythmic movements, namely locomotion [11]. These CPGs are modelled as coupled oscillators and solved using numeric integration. These CPGs have been applied in drumming [1] and postural control [3]. This dynamical systems approach model for CPGs presents multiple interesting properties, including: low computation cost which is well-suited for real time; robustness against small perturbations; the smooth online modulation of trajectories through changes in the dynamical systems parameters and phase-locking between the different oscillators for different DOFs.

In order to achieve the desired head movement, opposed to the one induced by locomotion, it is necessary to appropriately tune the CPG parameters. This can be achieved by optimizing the CPG parameters using an optimization method. The optimization process is done offline according to the head movement induced by the locomotion when no stabilization procedure was performed.

Some algorithms for solving this type of problem require substantial gradient information and aim to improve the

Cristina Santos, Miguel Oliveira and Vitor Matos are with Industrial Electronics Department, School of Engineering, University of Minho, 4800-058 Guimaraes, Portugal cristina@dei.uminho.pt, mcampos@dei.uminho.pt, vmatos@dei.uminho.pt

Ana Rocha and Lino Costa are with Production Systems Department, School of Engineering, University of Minho, 4710-057 Braga, Portugal arocha@dps.uminho.pt, lac@dps.uminho.pt

solution in a neighborhood of a given initial approximation. When the problem has more than one local solution, the convergence to the global solution may depend on the provided initial approximation. Thus, searching for a global optimum is a difficult task that could be done by using stochastic-type algorithms. The stochastic methods can be classified in two main categories, namely, the point-to-point search strategies and the population-based search techniques. From the population-based techniques, we would like to emphasize three particular algorithms, the electromagnetism-like algorithm (EM) [12], the particle swarm optimization [13] and genetic algorithms (GA) [2] that despite employing different strategies, they are easy to implement and computationally inexpensive in terms of memory requirement. The GA is well suited and has already been applied to solve this optimization problem because it can handle both discrete and continuous variables, nonlinear objective and constrain functions without requiring gradient information [14]. Recently, EM algorithm appeared as a promising algorithm for handling optimization problems with simple bounds. This technique is finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to nondifferentiable objective functions [15]. In this paper, we are interested in the application of the EM algorithm, proposed in [12], to optimize the CPG parameters of amplitude, offset and frequency of each head oscillator to head motion stabilization during quadruped robot locomotion.

The remainder of this paper is organized as follows. In Section II, the system architecture and how to generate locomotion and head movement is described. The main ideas concerning the optimization system, namely the problem statement that evaluates the head movement, the EM mechanism to optimize the CPG parameters and some experimental results, are described in Section III. Simulated results are described in Section IV. Conclusions are made in Section V.

II. SYSTEM ARCHITECTURE

Our aim is to propose a control architecture that is able to generate locomotion for a quadruped robot and to generate head motion such as to minimize the head movement induced by the the locomotion itself.

The overall system architecture is depicted in Figure 1.

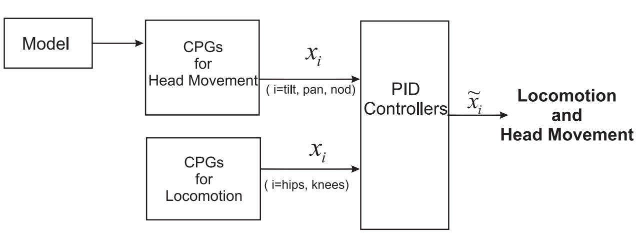

Fig. 1. Overall system architecture

The proposed movement controllers are biologically inspired in the concept of CPGs. A locomotion controller generates hip and knee trajectories. A head controller specifies the planned neck tilt, pan and nod joint values. These trajectories are used as input for the PID controllers of these joints.

The head controller parameters have to be tuned such that the resultant movement is as desired. Using our CPG approach allows us to assign explicit parameters for each of the nonlinear oscillators, independently controlling the amplitude, offset and frequency of the movement. We apply a stochastic optimization method, the EM algorithm, in order to determine the best set of CPG control parameters that results in, or close to the desired movement. This set of parameters constitute the Model module in Fig. 1.

A. Locomotion Generation

In this section we present the network of CPGs used to generate locomotion. A CPG for a given degree-of-freedom (DOF) is modelled as coupled Hopf oscillators, that generate a rhythmic movement.

1) Rhythmic Movement Generation: Rhythmic movements are generated by the following Hopf oscillator

$$\dot{x}_i = \beta\left(\mu_i - r_i^2\right)(x_i - O_i) - \omega z_i, \quad (1)$$
$$\dot{z}_i = \beta\left(\mu_i - r_i^2\right) z_i + \omega\,(x_i - O_i), \quad (2)$$

where $r_i = \sqrt{(x_i - O_i)^2 + z_i^2}$, ω specifies the oscillations frequency (in rad s^{-1}), peak-to-peak amplitude of the oscillations are given by $A_i = 2\sqrt{\mu_i}$ and relaxation to the limit cycle is given by $\frac{1}{2\beta\mu_i}$.

This Hopf oscillator contains a bifurcation from a stable fixed point at $x_i = O_i$ (when $\mu_i < 0$) to a structurally stable, harmonic limit cycle, for $\mu_i > 0$. The fixed point x_i has an offset given by O_i.

Thus, this Hopf oscillator exhibits limit cycle behaviour and describes a stable rhythmic motion where parameters A_i, ω and O_i control the desired amplitude, frequency and offset of the resultant oscillations.

2) Locomotion Controller Architecture: We have to couple the oscillators in order to ensure phase-locked synchronization between the hip and knee DOFs of the robot, and generate locomotion with a desired gait.

Fig. 2 depicts the network structure used to generate locomotion for a quadruped robot. Hopf oscillators of the

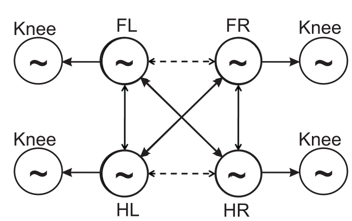

Fig. 2. Locomotion controller architecture depicting coupling structure among the CPGs for a walking gait. The footfall sequence is: HL-FL-HR-FR, with each foot lagging a quarter of a cycle from the previous.

hips are bilaterally coupled, these couplings being illustrated by right-left arrows, and hip Hopf oscillators are unilaterally coupled to the corresponding knee Hopf oscillators. For the hip joints, this is achieved by modifying (1) and (2) as follows:

$$\begin{aligned}\begin{bmatrix}\dot{x}_{i[1]}\\ \dot{z}_{i[1]}\end{bmatrix} &= \begin{bmatrix}\beta\mu_i & \omega\\ -\omega & \beta\mu i\end{bmatrix}\begin{bmatrix}x_{i[1]}-O_{i[1]}\\ z_{i[1]}\end{bmatrix} - \beta r_{i[1]}^2\begin{bmatrix}x_{i[1]}-O_{i[1]}\\ z_{i[1]}\end{bmatrix}\\ &+ \sum_{j\neq i}\mathbf{R}(\theta_{i[1]}^{j[1]})\begin{bmatrix}x_{j[1]}-O_{j[1]}\\ z_{j[1]}\end{bmatrix}\end{aligned}$$

For the knee joints, we modify (1) and (2) as follows:

$$\begin{aligned}\begin{bmatrix}\dot{x}_{i[3]}\\ \dot{z}_{i[3]}\end{bmatrix} &= \begin{bmatrix}\beta\mu i & \omega\\ -\omega & \beta\mu i\end{bmatrix}\begin{bmatrix}x_{i[3]}-O_{i[3]}\\ z_{i[3]}\end{bmatrix} - \beta r_{i[3]}^2\begin{bmatrix}x_{i[3]}-O_{i[3]}\\ z_{i[3]}\end{bmatrix}\\ &+ \frac{1}{2}\mathbf{R}(\psi_{i[3]}^{j[1]})\begin{bmatrix}x_{j[1]}-O_{j[1]}\\ z_{j[1]}\end{bmatrix}\end{aligned}$$

where $r_i[k]$ is the norm of vector $(x_i[k]-O_{i[k]}, z_i[k])^T$ (k = 1,3, that is hip and knee joints) and i, j = Fore Left (FL), Fore Right (FR), Hind Left (HL) and Hind Right (HR) limbs. The linear terms are rotated onto each other by the rotation matrices $\mathbf{R}(\theta_{i[1]}^{j[1]})$ and $\mathbf{R}(\psi_{i[3]}^{j[1]})$, where $\theta_{i[1]}^{j[1]}$ is the relative phase among the i[1]'s and j[1]'s hip oscillators and represents bidirectional couplings between these oscillators such that $\theta_{i[1]}^{j[1]} = -\theta_{j[1]}^{i[1]}$ and $\psi_{i[3]}^{j[1]}$ is the required relative phase among the i[3]'s and j[1]'s oscillators (see Fig. 2). We assure that closed-loop interoscillator couplings have phase biases that sum to a multiple of 2 π.

Each hip oscillator lags a quarter of a cycle from the previous. The relative phases between hips and knees, $\psi_{i[3]}^{j[1]}$, were all set to 180.

Due to the properties of these coupled Hopf oscillators, the generated trajectories are always smooth and thus potentially useful for trajectory generation in a robot.

This network structure constitutes the locomotion controller that generates desired trajectories, x_i, obtained by integrating the CPGs dynamical systems. These are sent online for the PID controllers of each hip and knee joints and result in the actual trajectories $\tilde{x}_i$.

3) Generating a walking gait: A gait event sequence is specified using the duty factors and the relative phases, where the first event, and the start of the stride, is chosen as the event when the fore left leg (reference leg) is set down. We have set a non-singular, regular and symmetric gait with a FL-HR-FR-HL gait even sequence $\{\varphi_{FL}, \varphi_{HR}, \psi_{FR}, \psi_{HL}, \varphi_{FR}, \varphi_{HL}, \psi_{FL}, \psi_{HR}\}$), a duty factor of 0.73 and a velocity of 19mms^{-1} (measured in the Z direction, see Fig. 3).

We have implemented in webots [8] this locomotion controller (simulation results and the experiment description is detailed explained in section).

B. Head Movement Generation

Head movement is generated similarly to locomotion, but a CPG for a given DOF is modelled as an Hopf oscillator, not coupled to any other oscillator. Each CPG, therefore, generates a rhythmic movement according to

$$\begin{bmatrix}\dot{x}_i\\ \dot{z}_i\end{bmatrix} = \begin{bmatrix}\beta\mu_i & \omega\\ -\omega & \beta\mu i\end{bmatrix}\begin{bmatrix}x_i-O_i\\ z_i\end{bmatrix} - \beta r_i^2\begin{bmatrix}x_i-O_i\\ z_i\end{bmatrix}, \quad (3)$$

where i =tilt,pan,nod.

The control policy is the x_i variable, obtained by integrating the CPGs dynamical systems, and represents tilt, pan and nod joint angles in our experiments. These are sent online for the corresponding PID controllers.

Note that the final movement for each of these joints is a rhythmic motion which amplitude of movement is specified by μ_i, offset by O_i and its frequency by ω.

The differential equations for locomotion and head movement are solved using Euler integration with a fixed time step of 1ms. The x_i trajectories represent angular positions and are directly sent to the PID controllers of the joint servomotors.

III. OPTIMIZATION SYSTEM

In this section, we explain how the head CPGs are optimized in order to reduce the camera (head) movement induced by locomotion itself. We will optimize the distance between the generated head movement for a set of head CPG control parameters and the one induced by locomotion.

In order to implement the head motion it is necessary one or several optimal combinations of amplitude, offset and frequency of each head oscillator. This is possible because we can easily modulate amplitude, offset and frequency of the generated trajectories according to changes in the A_i, O_i and ω CPG parameters and these are represented in an explicit way by our CPG. Therefore, we have to tune the head CPG parameters: amplitude A_i, offset O_i and common frequency ω. In order to optimize the combinations of the different head CPG control parameters the EM algorithm is used.

The multitude of parameter combinations is large, and it is difficult to derive an accurate model for the tested quadruped robot and for the environment. Besides, such a model based approach would also require some post-adaptation of results (because of backlash, friction, etc).

In this study, the search of parameters suitable for the implementation of the required head motion was carried out based on the data from a simulated quadruped robot. The (X,Y,Z) head coordinates, in a world coordinate system (Fig. 3), are recorded when a simulated robot walks during 30s and no head stabilization is performed. We are interested in the opposite of this movement around the (X,Y,Z) coordinates. This data was mathematically treated such as to keep only the oscillations in the movement and remove the drift that the robot has in the X coordinate and also the forward movement in the Z coordinate. From now on, this data is referred to as $(X,Y,Z)_{\text{observed}}$.

In the simulation, we have set a cycle time of 30ms, that is, the time needed to perform sensory acquisitions, calculate the planned trajectories (integrating the differential equations) and send this data to the servomotors. The $(X,Y,Z)_{\text{observed}}$ data is sampled with a sample time of 30ms, meaning we have a total of 1000 samples. A simulated time of 30s corresponds to 10 strides of locomotion. This time is arbitrary and could have been chosen differently but seems well suited to find a model representative of the head movement induced by the locomotion controller.

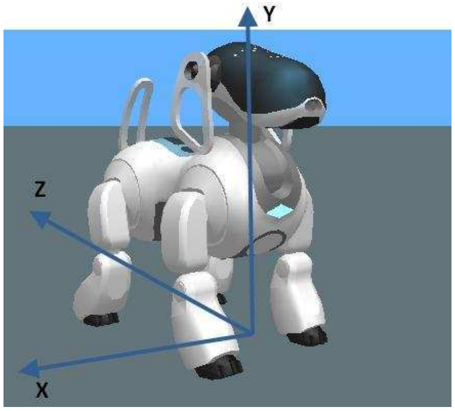

Fig. 3. World coordinate system.

The basic idea is to combine the CPG model for head movement generation with the optimization algorithm. Fig. 4 illustrates a schematics of the overall optimization system.

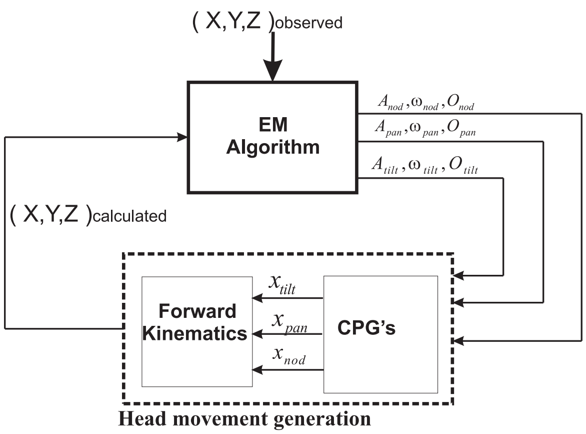

Fig. 4. Schematics of the optimization system.

Three head CPGs (3) generate during 30s rhythmic motions for the tilt, pan and nod joints. By applying forward kinematics, we calculate the resultant set of 1000 samples of $(X,Y,Z)_{\text{calculated}}$ head coordinates in the world coordinate system.

A. Problem Definition

The sum of the distances between each sample of the observed and calculated head coordinates is used as fitness function in order to evaluate the resulting head movement. Thus, the fitness of the ith point is given by

$$f_{\text{i}} = \sum_{j=1}^{n} \sqrt{\left(X_{\text{j}} - X'_{\text{j}}\right)^2 + \left(Y_{\text{j}} - Y'_{\text{j}}\right)^2 + \left(Z_{\text{j}} - Z'_{\text{j}}\right)^2} \quad (4)$$

where j is an head position sample (because the points are generated and acquired in a discrete manner); n is the total number of samples originated during the evaluation time; (X', Y', Z') represent the calculated head coordinates with the CPG parameters and (X,Y,Z) represent the offline observed head coordinates. Only head position errors are computed in the fitness function, because we only control three DOFs and as such cannot control head orientation.

In the optimization process each point is evaluated according to its fitness function value. Since we have a population of points the one with the smallest distance is denoted as the best point. Then, in the EM algorithm, each point is directed for a better position, inside of the allowed limits. The search ranges of the head CPG control parameters were set beforehand as shown in Table I for the purpose of efficient learning and according to the limits of the tilt, pan and nod DOFs. Search for optimal parameters is carried out by performing the overall optimization system over a preset number of iterations.

TABLE I
SEARCH RANGES OF CPG PARAMETERS

Parameter	Range	Unit
A_{tilt}	$[0,75]$	$(^o)$
ω_{tilt}	$[1,12]$	(rads^{-1})
O_{tilt}	$\left[-75+\frac{A_{\text{tilt}}}{2}, 0-\frac{A_{\text{tilt}}}{2}\right]$	$(^o)$
A_{pan}	$[0,(88+88)]$	$(^o)$
ω_{pan}	$[1,12]$	(rads^{-1})
O_{pan}	$\left[-88+\frac{A_{\text{tilt}}}{2}, 88-\frac{A_{\text{tilt}}}{2}\right]$	$(^o)$
A_{nod}	$[0,(45+15)]$	$(^o)$
ω_{nod}	$[1,12]$	(rads^{-1})
O_{nod}	$\left[-15+\frac{A_{\text{tilt}}}{2}, 45-\frac{A_{\text{tilt}}}{2}\right]$	$(^o)$

The combinations of amplitude, offset and frequency of each tilt, pan and nod oscillators, that are necessary to generate the desired head movement, form each point of the population. Each coordinate of the point consists in 9 CPG free parameters that span our vector x^i for the optimization, as follows

x^i_1	x^i_2	x^i_3	x^i_4	x^i_5	x^i_6	x^i_7	x^i_8	x^i_9
A_{tilt}	ω_{tilt}	O_{tilt}	A_{pan}	ω_{pan}	O_{pan}	A_{nod}	ω_{nod}	O_{nod}

B. Electromagnetism Algorithm

The EM algorithm starts with a population of randomly generated points from the feasible region. Analogous to electromagnetism, each point is a charged particle that is released to the space. The charge of each point is related to the fitness function value and determines the magnitude of attraction of the point over the population. The better the fitness function value, the higher the magnitude of attraction. The charges are used to find a direction for each point to move in subsequent iterations. The regions that have higher attraction will signal other points to move towards them. In addition, a repulsion mechanism is also introduced to explore new regions for even better solutions. Thus, the EM algorithm comprises 3 procedures: *Initialize* that will run only once in the start of the EM algorithm, *CalcF* and *Move*, these latter running sequentially every iteration. A more detailed explanation of the EM algorithm follows.

Initialize is a procedure that aims to randomly generate a population of points, x^i, from the feasible region, where each coordinate of a point is assumed to be uniformly distributed between the corresponding upper and lower bounds. Note that in order to guarantee the feasibility of the initial points and all points generated during the search a repair mechanism was implemented. Thus, an infeasible solution is repaired exploring the relations among variables expressed by the box constraints.

Then to compute the fitness function value for all the points in the population, they will be the input of the head movement generation process (see Fig. 4) and by applying forward kinematics the resultant $(X,Y,Z)_{\text{calculated}}$ head coordinates are computed. With them the fitness function value for all the points is calculated and the best point, which is the point with the best fitness function value, is identified.

For the *CalcF* procedure, the Coulomb's law of the electromagnetism theory is used. Thus, the force exerted on a point via other points is inversely proportional to the square of the distance between the points and directly proportional to the product of their charges. Then, we compute the charges of the points according to their fitness function values. The charge of each point determines the power of attraction or repulsion for that point. In this way the points that have better fitness function values possess higher charges. The total force vector exerted on each point is then calculated by adding the individual component forces between any pair of points.

The *Move* procedure uses the total force vector to move the point in the direction of the force by a random step length. The best point is not moved and is carried out to the subsequent iterations. To maintain feasibility, the force exerted on each point is normalized and scaled by the allowed range of movement towards the lower or the upper bound, for each coordinate. To ensure feasibility in this movement algorithm we define the projection of each coordinate of the point to the feasible region, according to the range presented in Table I.

After the EM algorithm, each point should be evaluated in terms of fitness function value, so they should go to the head movement generation process. Then this algorithm is repeated.

C. Experimental Results

The optimization system was implemented in Matlab (Version 6.5) running in an AMD Athlom XP 2400+ 2.00Gz (512 MB of RAM) PC. The system of equations was integrated using the Euler method with 1ms fixed integration steps (similarly to the simulated robotic experiments). The evaluation time for head movement generation is 30s.

In our implementation, the optimization system ends when the number of iterations exceeds 2000 iterations. In this study the number of points in the population was set to 20. When stochastic methods are used to solve problems, the impact of the random number seeds has to be taken into consideration and each optimization process should be run a certain number of times. In this experience we set it to 10.

Table II contains the Best, Mean and standard deviation (SD) values of the solutions found (in terms of fitness function and time) over the 10 runs. We can see that the SD

TABLE II
PERFORMANCE OF EM ALGORITHM IN THE OPTIMIZATION SYSTEM

Best fitness (mm)	Mean fitness (mm)	SD fitness (mm)	Best time (hours)	Mean time (hours)	SD time (hours)
4261	5325.53	870.6349	6.1047	6.5089	0.4120

value, in terms of fitness function, is a large value. It denotes that fitness values obtained in each run are not similar. It can be seen by Fig. 5 that shows the evolution of the best (solid line) and mean (dashed line) fitness function value over the 2000 iterations. The best point has a fitness value of 4261 that was achieved at iteration 1150. The best run took 6h18min (CPU time) and each iteration took in average 11.16 seconds.

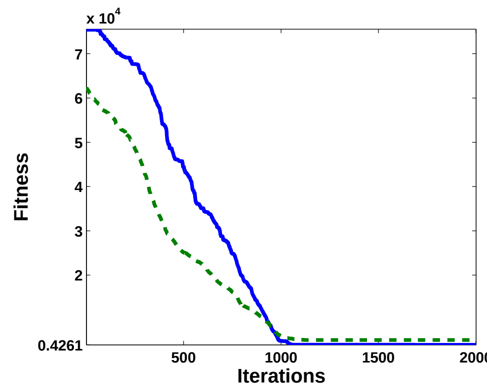

Fig. 5. Best (solid) and mean (dashed) fitness evolution.

Table III shows the tuned CPG parameters representing the best point found, over 2000 iterations, in the 10 runs.

TABLE III
BEST POINT CPG PARAMETERS

Parameter	Value	Unit
A_{tilt}	0.0001	$(^o)$
y_{tilt}	-6×10^{-5}	$(^o)$
w_{tilt}	6.707	(rads^{-1})
A_{pan}	7.77	$(^o)$
y_{pan}	0.072	$(^o)$
w_{pan}	2.12	(rads^{-1})
A_{nod}	0.0001	$(^o)$
y_{nod}	-1.18	$(^o)$
w_{nod}	1	(rads^{-1})

A better understanding of the evolution of the fitness function can be seen in Fig. 6 where the distance between observed and calculated values of the head movement at the beginning and at the end of the optimization system is displayed. We can observe that this distance, in each sample time for time ranging between $t = 5$ and 15s, is smaller at the end of the process. In average, we can also conclude that after 2000 iterations of the optimization system, a reduction

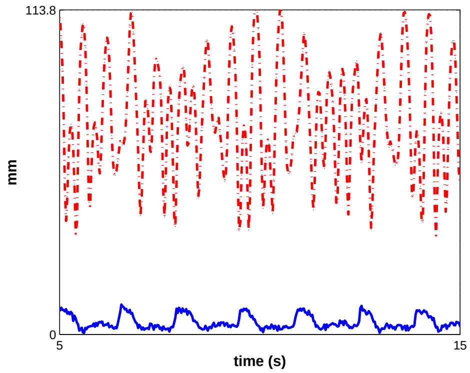

Fig. 6. Distance between observed and calculated values of the head movement at the beginning (dotted line) and at the end (solid line) of the optimization system, for time ranging from 5 to 15 seconds.

of 22,17% of the head movement is verified.

Fig. 7 depicts the time courses of the (X,Y,Z) calculated (solid line) head movement according to the head CPG control parameters of the best solution found. The observed (dotted line) head movement is also illustrated. Table IV gives the maximal movement variation in the (X,Y,Z) co-ordinates for the calculated and observed movements. We conclude that the generated movements are quite similar in the X coordinate. The calculated movement is quite different in the Y and Z coordinate. This results from the fact that only the pan joint controls movement in the X coordinate, while both the tilt and nod joints control the Y and Z coordinates.

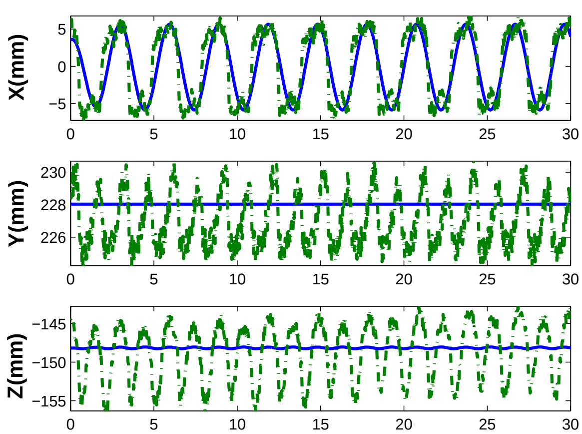

Fig. 7. (X,Y,Z) calculated (solid line) and observed (dotted line) head movement, during 30s, according to the head CPG control parameters from best point on the final of optimization system.

Fig. 8 depicts 3D calculated (solid line) and observed (dotted line) head movement for the best point.

We have also made another experiment, where we have changed the size of the population to 50 points, maintaining the number of 2000 iterations to terminate the process. Running the optimization system we obtained a best fitness function value of 3991 at iteration 1760.

TABLE IV

MAXIMAL MOVEMENT VARIATION IN (X,Y,Z)

	Max ΔX (mm)	Max ΔY (mm)	Max ΔZ (mm)
Calculated Movement	11.47	0	0.2
Observed Movement	13.42	5.9	11.3

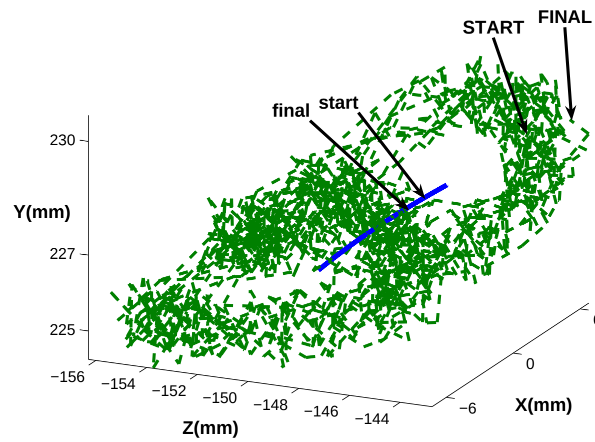

Fig. 8. 3D calculated (solid line) and observed (dotted line) head movement according to the CPG parameters of the 1150th iteration best point. START (FINAL) and start (final) indicate where the observed and calculated movement started (ended), respectively.

IV. SIMULATION RESULTS

Our aim was to build a system able to eliminate or reduce the head motion of a robot that walks in the environment. For that, we set a dynamical controller generating trajectories for the head joints such that the final head movement is opposite to the one induced by locomotion.

In this section, we describe the experiment done in a simulated ers-7 AIBO robot using Webots [8]. Webots is a software for the physic simulation of robots based on ODE, an open source physics engine for simulating 3D rigid body dynamics. The model of the AIBO is as close to the real robot as the simulation enable us to be. Thus, we simulate the exact number of DOFs, mass distributions and the visual system.

The ers-7 AIBO dog robot is a 18 DOFs quadruped robot made by Sony. The locomotion controller generates the joint angles of the hip and knee joints in the sagittal plane, that is 8 DOFs of the robot, 2 DOFs in each leg. Only walk gait is generated and tested.

The head controller generates the joint angles of the 3 DOFs: tilt, pan and nod. The other DOFs are not used for the moment, and remain fixed to an appropriately chosen value during the experiments.

The AIBO has a camera built into its head.

At each sensorial cycle (30ms), sensory information is acquired. The dynamics of the CPGs are numerically integrated using the Euler method with a fixed time step of 1ms thus specifying servo positions. Parameters were chosen in order to respect feasibility of the experiment and are given in Table V and VI.

TABLE V
PARAMETER VALUES FOR GENERATING LOCOMOTION

	β	ω (rad s^{-1})	μ_i	$\frac{1}{2\beta\mu_i}$ (s)
Front Limbs	0.1	2.044	6.25	0.8
Hind Limbs	0.025	2.044	25	0.8

TABLE VI
PARAMETER VALUES FOR GENERATING HEAD MOTION

	β	ω (rad s^{-1})	μ_i	$\frac{1}{2\beta\mu_i}$ (s)
tilt	1.25×10^9	4.19	2.5×10^{-9}	0.8
pan	0.041	2.09	15.13	0.8
nod	1.25×10^9	4.19	2.5×10^{-9}	0.8

Because we are working in a simulated environment, we are able to build a GPS into the AIBO camera, that enable us to verify how the head effectively moves in an external coordinate system. Two simulations are performed: the robot walks during 30s with and without the feedforward solution and its GPS coordinates are recorded. Results are compared for these two simulations. Fig. 9 shows the GPS coordinates for the experiments with (solid line) and without the feedforward solution (dotted line). The overall experiment can be seen in the attached video.

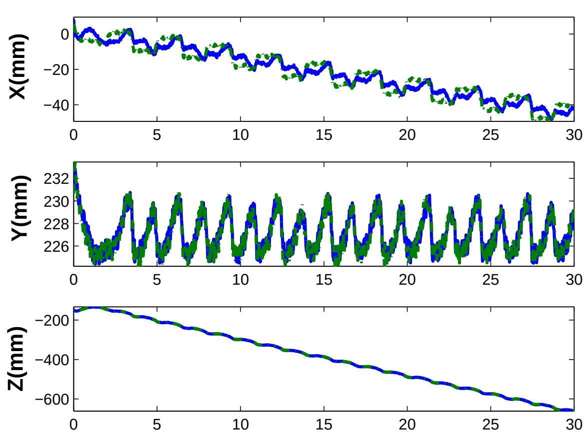

Fig. 9. (X,Y,Z) coordinates of the GPS positioned in the AIBO head when the robot walks during 30s. Solid and dotted lines indicate the experiment in which the feedforward solution is and is not implemented, respectively.

We expect that the proposed feedforward solution minimizes the variation of the GPS coordinates, meaning that the head remains near the same position during the experiment. We observe that the X coordinates of the marker position oscillate less. Note that there is some drift in the X coordinates, meaning the robot slightly deviates towards its side while walking. The observed peaks in the Y coordinate reflect the final stage of the swing phase and the begin of the stance phases of the fore legs, corresponding to an accentuated movement of the robot center of mass. This problem will be addressed in current work, by improving the locomotion controller and take into account balance control [3].

V. CONCLUSIONS AND FUTURE WORKS

In this article, we have addressed head stabilization of a quadruped robot that walks with a walking gait. A locomotion controller based on dynamical systems, CPGs, generates quadruped locomotion. The required head motion needed to eliminate or reduce the head shaking induced by locomotion, is generated by CPGs built-in in the tilt, pan and nod joints. These CPG parameters are tuned by an optimization system. This optimization system combines CPGs and the EM algorithm. As a result, set of parameters obtained by the EM allows to reduce the head movement induced by the locomotion.

Currently, we are using other optimization methods, like the particle swarm optimization, and testing other fitness functions. We will extend this optimization work to address other locomotion related problems, such as: the generation and switch among different gaits according to the sensorial information and the control of locomotion direction. We further plan to extend our current work to online learning of the head movement similarly to [9].

REFERENCES

[1] S Degallier, C P. Santos, L Righetti and A Ijspeert,*Movement Generation using Dynamical Systems: a Drumming Humanoid Robot.* In Humanoids06 IEEE-RAS International Conference on Humanoids Robots, Genova, Italy, December 4-6 2006

[2] Goldberg, D.,*Genetic Algorithms in Search*, Optimization,and Machine Learning, Addison-Wesley, (1989).

[3] Castro, Luiz, Santos, C P; Oliveira, M and Ijspeert, A, *Postural Control on a Quadruped Robot Using Lateral Titl: a Dynamical System Approach*, European Robotics Symposium EUROS 2008, Prague, 2008.

[4] Takizawa, S, Ushida, S, Okatani, T, Deguchi, K.,*2DOF Motion Stabilization of Biped Robot by Gaze Control Strategy.* 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005). Aug. 2005:3809-3814

[5] Ravi Kaushik, Marek Marcinkiewicz, Jizhong Xiao, Simon Parsons, and Theodore Raphan*Implementation of Bio-Inspired Vestibulo-Ocular Reflex in a Quadrupedal Robot* 2007 IEEE International Conference on Robotics and Automation (ICRA 2007), Roma, Italy, 10-14 April 2007: 4861-4866

[6] Yamada, H.; Mori, M.; Hirose, S.,*Stabilization of the head of an undulating snake-like robot*, it 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, CA, USA, Oct 29 - Nov 2, 2007 : 3566-3571

[7] Yurak Son, Takuya Kamano, Takashi Yasuno, Takayuki Suzuki,and Hironobu Harada,*Generation of Adaptive Gait Patterns for Quadruped Robot with CPG Network Including Motor Dynamic Model*, Electrical Engineering in Japan, Vol. 155, No. 1, 2006

[8] O. Michel, “Webots: Professional mobile robot simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[9] A. Sproewitz, R. Moeckel, J. Maye, M. Asadpour, and A.J. Ijspeert. *Adaptive locomotion control in modular robotics.* In Workshop on Self-Reconfigurable Robots/Systems and Applications IROS07, pages 81-84, November 2007.

[10] Ludovic Righetti and Auke Jan Ijspeert.*Pattern generators with sensory feedback for the control of quadruped locomotion.* IEEE International Conference on Robotics and Automation ICRA 2008, California, 2008.

[11] S. Grillner. *Neurobiological bases of rhythmic motor acts in vertebrates.* Science, Vol. 228, No. 4696, pp. 143-149, 1985

[12] S.I. Birbil and S. Fang, *An electromagnetism-like mechanism for global optimization*, Journal of Global Optimization 25 (2003), pp. 263–282.

[13] J. Kennedy and R.C. Eberhart, *Particle swarm optimization*, in *IEEE International Conference on Neural Network*, 1995, pp. 1942–1948.

[14] Cristina Santos, Miguel Oliveira, Ana Maria A.C. Rocha and Lino Costa, *Head Motion Stabilization During Quadruped Robot Locomotion: Combining Dynamical Systems and a Genetic Algorithm*, IEEE International Conference on Robotics and Automation (ICRA 2009), May 12-17, Kobe, Japan 2009.

[15] Rocha, Ana Maria A.C. and Fernandes, Edite M.G.P., *Performance profile assessment of electromagnetism-like algorithms for global optimization*, AIP Conference Proceedings, Vol. 1060 (1), Springer-Verlag (2008), (15-18).

[16] F. Patane, C. Laschi, H. Miwa, E. Guglielmelli, P. Dario, and A. Takanishi, *Design and development of a biologically-inspired artificial vestibular system for robot heads*, in IEEE Conference on Intelligent Robots and Systems, IROS'04, Sendai, Japan, September 28 - Ocotber 2, 1317- 1322, 2004.

[17] G. Asuni, G. Teti, C. Laschi, E. Guglielmelli, and P. Dario, *A robotic head neuro-controller based on biologically-inspired neural methods*, in IEEE Conference on Robotics and Automation, ICRA'05, Barcelona, Spain, April 18-22, 2005.

[18] R. Cromwell, J. Schurter, Scott Shelton, S. Vora, *Head stabilization strategies in the sagittal plane during locomotor tasks*, Physiotherapy Research International, Whurr Publishers Ltd, 9(1), pp.3342, 2004