Locomotion gait optimization for a quadruped robot

Miguel Oliveira [1], **Cristina Santos** [2], **Manuel Ferreira**[3], **Lino Costa** [4] and **Ana Rocha** [5]

Abstract. This article describes the development of a gait optimization system that allows a fast but stable robot quadruped crawl gait.

We focus in the development of a quadruped robot walking gait locomotion that combine bio-inspired Central Patterns Generators (CPGs) and Genetic Algorithms (GA). The CPGs are modelled as autonomous differential equations, that generate the necessary limb movement to perform the walking gait, and the Genetic Algorithm perform the search of the CPGs parameters.

This approach allows to explicitly specify parameters such as amplitude, offset and frequency of movement and to smoothly modulate the generated trajectories according to changes in these parameters. It is therefore easy to combine the CPG with an optimization method. A genetic algorithm determines the best set of parameters that generates the limbs movements. We intend to obtain a walking gait locomotion that minimizes the vibration and maximizes the wide stability margin and the forward velocity.

The experimental results, performed on a simulated Aibo robot, demonstrated that our approach allows low vibration with a high velocity and wide stability margin for a quadruped walking gait locomotion.

1 Introduction

Robot locomotion is a challenging task that involves the control of a great number of degrees of freedom (DOF's). Several previous works, [11, 22, 15], proposed biologic approaches to modulate the gait locomotion of quadruped robots, combining biometric sensory information with motion oscillators such as CPGs.

The problem of finding the best possible locomotion is a problem currently addressed in the literature [2, 6, 14]. Usually optimization systems are applied to improve the performance of the Aibo quadruped robot locomotion. The competition in Robocup is one of the motivation engines, for these works. In the following, we briefly describe some relevant works in this domain.

In [2] it is presented a Genetic Algorithm robust to the noise in the parameters evolution and that also avoids premature local optima. The evaluation is made on a robot soccer field, and the robot communicates by wireless with an external computer where the learning algorithm is executed. The goal of the fitness is to maximize the robot velocity. As a result of this learning algorithm the robot moves with a velocity of 0.290 (m/s).

A comparison between several learning locomotion algorithms, including Genetic Algorithm and Policy Gradient Algorithm, is presented in [14]. This is also an online learning performance that uses three Aibo robots for decreasing the time spending on each test. The optimization goal is to determine the best 12 parameters of an elliptical locus scheme of locomotion, such that the robot takes less time to walk a certain distance. Each learning has a previous hand-tuned set of parameters and the best results were achieved the by the hill climbing and Policy Gradient Algorithm. The average speed achieved by the Policy Gradient Algorithm was 0.291 (m/s).

In [6] it is presented an evolutionary algorithm based on a genetic algorithm. The genetic operators are chosen by an adaptation mechanism. The locomotion is implemented in real time and it is evaluated by analyzing the forward-backward motion, the side-walk, the rotation motion and the vibration. For measuring the vibration they use accelerometers of the robot. For each sensor and during a test the standard deviation (std) of accelerometer measurements is calculated. The evaluation tests are performed in a robot soccer field, and the evaluation calculus is made in an external computer using an external camera for motoring the translation and rotational movements.

In [10] it is presented an evolutionary algorithm (EA) to optimize a vector of parameters for locomotion of an ERS110 robot. The EA uses a steady-state algorithm that applies the mutation and/or the recombination of operators to create new individuals from sets of parents. To avoid local minima an individual can be a parent during a predefined number of times. The solution obtained by the EA moves the robot with a velocity of 0.167 m/s.

In [13] it is presented an optimization system for the locomotion of Aibo 210 based on the Powell's method. It optimizes 12 parameters of a locus locomotion scheme. Optimization is made online. In each iteration the robot moves between two landmarks, and the goal is to maximize the forward velocity. They achieved an average speed of 0.2269 (m/s) for the rectangular locus and an average speed of 0.25 (m/s) for the trapezoid locus.

In this work, we propose an approach to optimize a walk gait locomotion, using Central Pattern Generators (CPGs) and a genetic algorithm.

CPGs are neural networks located in the spine of vertebrates, that generate coordinated rhythmic movements, namely locomotion [8]. In this work, a locomotion controller, based on CPGs, generates trajectories for hip robot joints [15]. These CPGs are modelled as coupled oscillators and solved using numeric integration. They have been previously applied in drumming [4] and postural control [1].

The proposed CPG is based on Hopf oscillators, and allows to explicitly and smoothly modulate the generated trajectories according to changes in the CPG parameters such as amplitude, offset and frequency. In order to achieve the desired walk gait movement, it is necessary to appropriately tune these parameters. In this work, these

[1] Industrial Electronics Department, School of Engineering,University of Minho, Portugal, email: mcampos@dei.uminho.pt
[2] Industrial Electronics Department, School of Engineering,University of Minho, Portugal, email: cristina@dei.uminho.pt
[3] Industrial Electronics Department, School of Engineering,University of Minho, Portugal, email: mjf@dei.uminho.pt
[4] Production Systems Department, School of Engineering, University of Minho, Portugal, email: lac@dps.uminho.pt
[5] Production Systems Department, School of Engineering, University of Minho, Portugal, email: arocha@dps.uminho.pt

parameters are optimized using a Genetic Algorithm. Optimization is done online in a simulated ers-7 AIBO robot using Webots [17].

This optimization is a non-linear problem where continuity and convexity conditions are not guaranteed. Thus, searching for a global optimum is a difficult task that requires approaches based on stochastic algorithms like evolutionary algorithms, in particular, genetic algorithms. These are search algorithms that mimic the process of natural selection [5]. Thus, unlike conventional algorithms, in general, only the information regarding the objective to optimize is required. Moreover, they are based on a population that evolves over time, possibly in the direction of the optimum.

This article is structured as follows. In Section 2, we explain how we generate locomotion . Section 3 presents the optimization system and it is discussed the objective function. Simulated results are described in Section 4. The paper ends with a discussion and conclusions in Section 5.

2 LOCOMOTION GENERATION

In this section we describe the system used to generate locomotion. Firstly, a brief description of gaits focusing in the generated gait is done. Next, a description of the modelled CPGs is done, including the network of Hopf oscillators.

2.1 Gait Description

During locomotion, quadruped walking animals have to usually move their legs in a manner that provides the suitable forward force at a minimal energy expenditure while maintaining their equilibrium. This coordinated cyclic manner of lifting and placing the legs on the ground, called a gait, is important for equilibrium stability and the step cycle sequence is typical for vertebrates: left forelimb (LF), right hindlimb (RH), right forelimb (RF), and left hindlimb (LH).

Quadrupedal gaits are classified according to the duration of their stance phases [16], i.e. their duty factor values, and their relative phases. In general, the duty factor β reduces as the speed increases.

In this work we will address a crawl gait. This is a symmetric gait, meaning that the two legs of the same girdle are 0.5 out of phase. This gait is singular (two or more legs are simultaneously lifted or placed during a stride) and regular (all the legs have the same duty factor).

In general, the number of step cycles per second increases as the speed of locomotion increases [7]. This corresponds to a reduction in the step cycle duration almost exclusively due to a shortening of the stance phase (limb in contact with the ground), whereas the swing phase (no ground contact) is kept nearly constant.

Herein, we assume that at all walking speeds the onset of swing in a foreleg occurs just after the onset of stance in the ipsilateral hind leg [7]. In order to achieve this, we use the wave gait rule: the gait phase (φ_{LH}) follows the value of the duty factor (β). The use of this rule improves the stability of the locomotion [9, 16, 12]. Stability is measured by calculating the stability margin [9] which decreases approximately linearly with the velocity increase (see results).

2.2 Rhythmic Movement Generation

The rhythmic movements of each hip joint of a limb, i, are generated by a Hopf oscillator, given by

$$\dot{x}_i = \alpha\left(\mu - r_i^2\right)(x_i - O_i) - \omega z_i, \quad (1)$$

$$\dot{z}_i = \alpha\left(\mu - r_i^2\right) z_i + \omega (x_i - O_i), \quad (2)$$

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

This oscillator contains an Hopf bifurcation from a stable fixed point at $(x_i, z_i) = (O_i, 0)$ (when $\mu < 0$) to a structurally stable, harmonic limit cycle, for $\mu > 0$.

The following expression for ω allows an independent control of speed of the ascending and descending phases of the rhythmic signal [18], meaning an independent control of the stance ω_{st} and the swing durations ω_{sw},

$$\omega = \frac{\omega_{st}}{1 + e^{-qz_i}} + \frac{\omega_{sw}}{1 + e^{-qz_i}}. \quad (3)$$

The stance phase frequency, ω_{st}, is determined based on the constant swing frequency, ω_{sw}, and on the desired duty factor, β as follows:

$$\omega_{st} = \frac{1 - \beta}{\beta}\omega_{sw} \quad (4)$$

Each CPG takes a set of parameters for the modulation of the generated trajectories for the specified joint. These are:

- μ, switches on/off the rhythmic solution, and for $\mu > 0$ modulates the amplitude of oscillations;
- β, changes the walking velocity since it controls the stance duration of the generated movement.

All these parameters will be tuned by the optimization system described in section 3, controlling the parameters for a locomotion that maximizes a fitness. 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 capabilities of the robotic platform.

2.3 Interlimb Coordination

We have four CPGs, one for each Hip joint. These four CPGs are coupled in order to achieve the limb coordination required in a walking gait pattern. The applied coupling scheme is depicted in fig 1 and is given by

$$\begin{bmatrix} \dot{x}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} \alpha\left(\mu - r_i^2\right) & -\omega \\ \omega & \alpha\left(\mu - r_i^2\right) \end{bmatrix} \begin{bmatrix} x_i - O_i \\ z_i \end{bmatrix} + \sum_{j \neq i} \mathbf{R}(\theta_i^j) \begin{bmatrix} x_j - O_j \\ z_j \end{bmatrix}. \quad (5)$$

The linear terms are rotated onto each other by the rotation matrix $\mathbf{R}(\theta_i^j)$, where θ_i^j is the required relative phase among the i and j hip oscillators to perform the gait (we exploit the fact that $\mathbf{R}(\theta) = \mathbf{R}^{-1}(-\theta)$).

The final result is a network of oscillators with controlled phase relationships, able to generate more complex, synchronized behavior such as locomotion. Due to the properties of this type of coupling among oscillators, the generated trajectories are stable and smooth and thus potentially useful for trajectory generation in a robot.

The generated x_i 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 for the lower level PID controllers of each limb joint. The knee joints are controlled as simple as possible. When the limb performs the swing phase, the knee flexes to a fixed angle. When performing the stance phase, the knee extends to other angle.

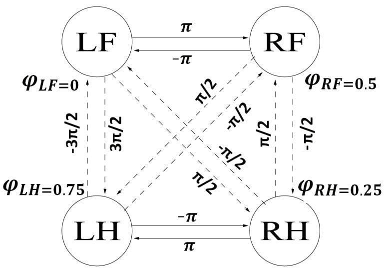

Figure 1. Each leg lags one quarter of a cycle, in the sequence: Left Fore, Right Hind, Right Fore, Left Hind

3 OPTIMIZATION SYSTEM

In this section, we explain how the limbs trajectories are optimized, in order to obtain the best walking pattern locomotion. We intend to maximize the velocity and wide stability margin, and to minimize the vibration of the robot. A scheme of the optimization system is depicted in fig 2.

CPGs generate trajectories for the robot limbs, modulated according to a set of parameters. A chromosome is constituted by the required set of these parameters such that the robot performs the desired locomotion gait.

Initially, a random initial population of chromosomes is generated. After the evaluation of all chromosomes of the population, a genetic algorithm generates a new population to be tested. The stopping criterion of the optimization system is the performed number of iterations.

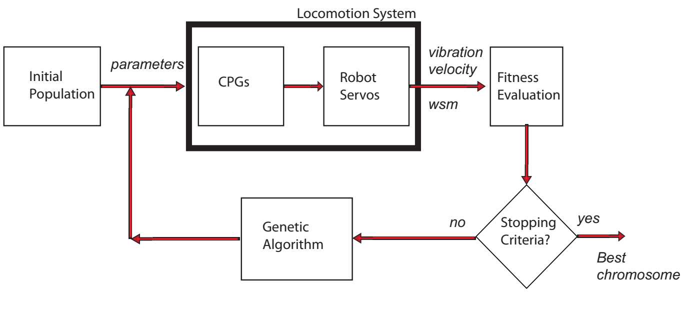

Figure 2. Optimization Locomotion System

3.1 Optimization of Parameters

As previously described, trajectories are generated and modulated by the proposed network of CPGs, by explicitly changing the CPGs parameters. These are the Amplitude (μ), the Offset(O), and the stance knee value, for each limb. Further, there is the swing frequency (ω_{sw}) for the overall network. Meaning a total of 13 parameters.

Fore and hind limb trajectories (LF, RF) (LH, RH) have the same amplitude, offset and frequency but a different relative phase. they have a relative phase of π among them.

Taking in regard these considerations, we can minimize the number of parameters it is required to optimize. The set of parameters is given by: amplitude of the front limbs (μ_{FL}), amplitude of the hind limbs (μ_{HL}), front limbs knee angle (K_{FL}),hind limbs knee angle (K_{HL}), front limbs offset (O_{FL}), hind limbs offset (O_{HL}) and the frequency of Swing (ω_{sw}). This yields a total of 7 parameters to tune.The limits of each parameters are defined in tab 1.

3.2 Limits of the parameters

The range of each parameter is defined in tab 1. These boundaries directly depend on the physics limits of the Aibo Ers-7 robot. The values of μ_{FL} and μ_{HL} are limited by the maximum range that the AIBO Hip joints may have. Noteh that amplitude is given by $\sqrt{\mu}$. Offset values O_{FL} and O_{HL} for the hips are limited by the same ranges and the calculated amplitude values, μ_{FL} and μ_{HL}, respectively .

We calculate the maximum and minimum values for each knee stance angle, such that leg collision does not occur during locomotion. This is given by

$$K_{FLmax1} = -(O_{FL} + \sqrt{\mu_{FL}}/2) + 50 \quad (6)$$
$$K_{FLmax2} = -(O_{FL} - \sqrt{\mu_{FL}}/2) + 50 \quad (7)$$
$$K_{FLmin1} = -(O_{FL} + \sqrt{\mu_{FL}}/2) + 20 \quad (8)$$
$$K_{FLmin2} = -(O_{FL} - \sqrt{\mu_{FL}}/2) + 20 \quad (9)$$

$$K_{HLmax1} = -(O_{HL} + \sqrt{\mu_{HL}}/2) + 40 \quad (10)$$
$$K_{HLmax2} = -(O_{HL} - \sqrt{\mu_{HL}}/2) + 40 \quad (11)$$
$$K_{HLmin1} = -(O_{HL} + \sqrt{\mu_{HL}}/2) - 5 \quad (12)$$
$$K_{HLmin2} = (O_{HL} - \sqrt{\mu_{HL}}/2) - 5 \quad (13)$$

where O_{FL} and O_{HL} are the offsets of the fore and hind hip joints, respectively. Finally, knee angles are given by $[max(min1, min2)min(max1, max2)]$.

Table 1. Parameter Limits

Parameters	Lower	Upper
μ_{FL}	0.0001	3600
$O_{FL}(°)$	$-1600+\mu_{FL}/2$	$400-\mu_{FL}/2$
μ_{HL}	0.0001	3600
$O_{HL}(°)$	$-400+\mu_{FL}/2$	$1600-\mu_{HL}/2$
ω_{sw}(rad/s)	1	12
$K_{FL}(°)$	$max(K_{FLmin1}, K_{FLmin2})$	$min(K_{FLmax1}, K_{FLmax2})$
$K_{HL}(°)$	$max(K_{FLmin1}, K_{FLmin2})$	$min(K_{HLmax1}, K_{HLmax2})$

3.3 Genetic Algorithm

Genetic Algorithms (GA) start from a pool of points, usually referred to as chromosomes. Chromosomes represent potential optimal solutions of the problem being solved. In order to implement a GA, it is necessary to define the representation of the search space and a fitness function which permits the comparison between the different chromosomes. Furthermore, genetic operators and the selection mechanism must also be defined.

One or several optimal combinations of amplitude and offset for the hip oscillators, offset for the knees and swing frequency are necessary in order to generate the desired forward locomotion movement, as explained before. Each chromosome consists in 7 CPG free parameters, as shown in fig. 3, that span our vector space for the optimization.

Chromosome

μ_{FL}	O_{FL}	μ_{HL}	O_{HL}	ω_{sw}	K_{FL}	K_{HL}

Figure 3. A chromosome is made of seven free parameters.

In our optimization system, we begin the GA search by randomly generating an initial population of chromosomes.

The GA selection operator assures that chromosomes are copied to the next generation with a probability associated to their fitness values. Therefore, this operator mimics the survival of the fittest in the natural world. Although selection assures that in the next generation the best chromosomes will be present with a higher probability, it does not search the space, because it just copies the previous chromosomes. The search results from the creation of new chromosomes from old ones by the application of genetic operators. The crossover operator, takes two randomly selected chromosomes; one point along their common length is randomly selected, and the characters of the two parent strings are swapped, thus generating two new chromosomes. The mutation operator, randomly selects a position in the chromosome and, with a given probability, changes the corresponding value. This operator does assure that new parts of the search space are explored, which selection and crossover could not fully guarantee.

In this work, real representation of the variables was considered. So, each vector consists of a vector of real values representing the decision variables of the problem. Genetic operators were chosen taking into account this representation. In order to recombine and mutate chromosomes, the Simulated Binary Crossover (SBX) and Polynomial Mutation were considered, respectively. These operators simulate the working of the traditional binary operators [3]. In order to select chromosomes for the application of genetic operators, a tournament selection was implemented.

3.4 Fitness Specification

The performance of each chromosome is evaluated according to the robot body vibration (f_a), the forward velocity (v) and the Wide Stability Margin WSM.

3.4.1 Vibration

We consider that a good gait should have less vibration, because the robot is subjected to less strain. In order to calculate the total vibration we sum the standard deviation of the measures of the (a_x, a_y, a_z) accelerometers built-in onto the robot, similarly to [19, 6, 20], as follows:

$$f_a = std(a_x) + std(a_y) + std(a_z) \quad (14)$$

3.4.2 Wide Stability Margin

For stability, we calculate the wide stability margin [21] (WSM). This is a measure of the locomotion stability that provides the shortest distance between the projection of the center of mass in the ground and the polygon formed by the vertical projection in the ground of robot feet contact points. A gait is considered better when has a higher WSM.

3.4.3 Velocity

We calculate the forward velocity using the traveled distance of the robot during the evaluation of each chromosome of the population, *i.e.* during 12 seconds. A gait is considered better if it achieves higher velocities.

We intend to determine the best gait considering minimization of the body vibration and maximization of the velocity and wide stability margin. The normalize fitness is given by:

$$\text{fitness}_{total} = W_{\text{a}} * \frac{f_a}{f_{\text{a,max}}} + W_v * \frac{1}{v} * v_{\text{min}} + W_{\text{wsm}} * e^{-\frac{wsm}{wsm_{\text{max}}}}, \quad (15)$$

where W_a,W_v and W_{wsm} are the vibration, velocity and WSM weights, respectively.

For each fitness component to have the same significance, we normalize the values of the fitness components. We have determined that $v_{\text{min}} = 10(mm/s)$, $f_{\text{a,max}} = 0.4$ and $wsm_{\text{max}} = 65$. These are the maximum values that the fitness components may achieve.

3.5 Weights of the fitness

We apply weights to each component of the fitness function, similarly to [20]. We have three weights, one for each fitness component: W_{a} (vibration weight), W_{v} (velocity weight) and W_{wsm} (WSM weight). The sum of the three components is always 1, as follows:

$$W_{\text{a}} + W_{\text{v}} + W_{\text{wsm}} = 1. \quad (16)$$

We implemented a method for the computation of the weights such that the component weights change depending on the value of the components. Lower velocities give higher weights for the velocity component but higher velocities give lower weights for the corresponding component. This method is shown in fig 4. It is based on a negative exponential, such that the higher the velocity, the lower the velocity weight (W_v), as follows:

$$W_{\text{v}} = 0.7 \times \exp^{-(v \times 0.01)} \quad (17)$$

We want to minimize the overall vibration but to maximize the velocity. Then, we want lower vibrations for high velocities. This is achieved by setting

$$W_{\text{a}} = 0.7 - W_{\text{v}} \quad (18)$$

Finally,

$$W_{\text{wsm}} = 0.3 \quad (19)$$

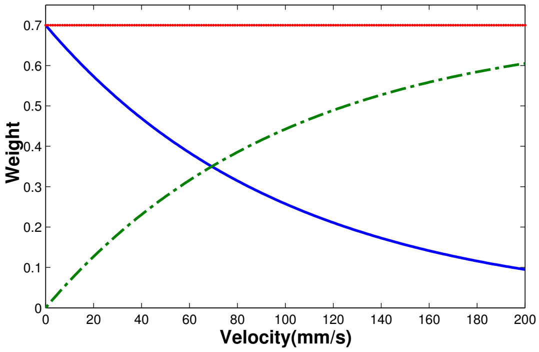

Figure 4. Weights W_v (solid line) and W_a (dashed line) for different velocity values. The sum between W_v and W_a is the dotted line.

4 SIMULATION RESULTS

In this section, we describe the experiment done in a simulated ers-7 AIBO robot using Webots [17]. The working scneraio is shown in fig 5. 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. We simulate the exact number of DOFs and mass distributions.

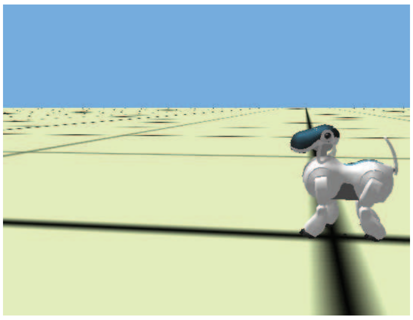

Figure 5. Simulation experimental setup.

The ers-7 AIBO dog robot is a 18 DOFs quadruped robot made by Sony. The locomotion controller generates trajectories for 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.

At each sensorial cycle (30 ms), sensory information is acquired.Each chromosome is evaluated during 12 seconds. We apply the Euler method with 1ms fixed integration step, to integrate the system of equations. At the end of each chromosome evaluation the robot is set to its initial position and rotation.

In our implementation, the optimization system ends when the number of generations exceeds 50 generations. We depict results when a population was established with 50 chromosomes and a preset number of 50 generations was set.

The generated gaits have a fixed duty factor $\beta = 0.75$ and a relative phase $\varphi_{\mathrm{LH}} = 0.75$.

Table 2 contains the Best, Mean and standard deviation (SD) values of the solutions found (in terms of fitness function and time) over 10 runs.

Table 2. Performance of GA algorithm in the optimization system

Fitness			Time (hours)		
Best	Mean	SD	Best	Mean	SD
0.2727	0.2973	0.0212	3.5350	4.0843	2.0975

Fig. 6 shows the evolution of all the 10 runs (lighter lines), best (solid line) and mean (dashed line) fitness function value over 50 generations. The best individual has a fitness value of 0.2727 that was achieved at generation 50. The best run took 3h53 min (CPU time) and each generation took in average 294.07 seconds.

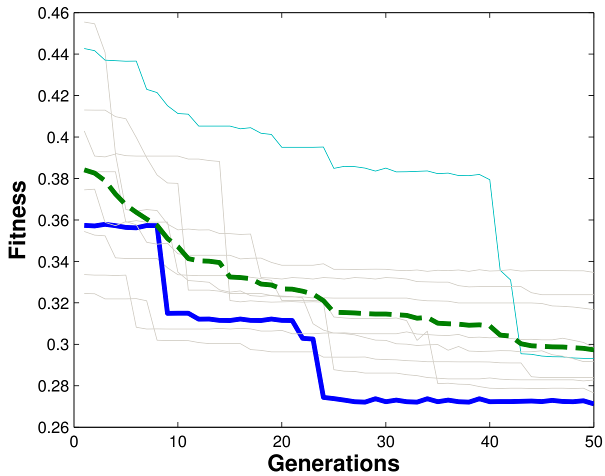

Figure 6. Fitness evolution during 50 generations.

Table 3, shows the parameters values of the best chromosome of the first and last generation.

Table 3. Optimization Parameters Results

Parameters	**1st Generation**	**50th Generation**
μ_{FL}	2692.6	66.26
$O_{FL}(°)$	-12.77	8.19
μ_{HL}	1954.5	252.81
$O_{HL}(°)$	9.21	15.87
$\omega_{sw}(rad/s)$	11.87	10.62
$K_{FL}(°)$	81.64	52.65
$K_{HL}(°)$	30.34	5.00
Fitness	0.3573	0.2727

Fig. 7 depicts the evolution of the measurements of sensor data, vibration, velocity and WSM of the best chromosome of each generation.

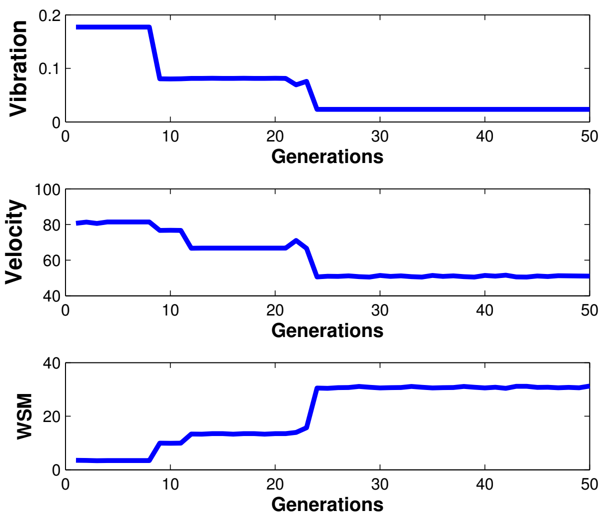

Figure 7. Evolution of the sensor measurements during the 50 Generations.

Table 4 lists the values of the vibration, velocity and WSM for the first and last generation. Velocity is $0.08623(ms^-1)$ and $0.05098(ms^-1)$ for the first and 50th generation, respectively. These values may seem much worst then those achieved by previous works, specially in the RobotCup domain. However, the robot configuration was the required to achieve higher velocities: the robot knees were completely folded. In our work, we just try to achieve a higher velocity for a crawl gait. In fact, it is a slow gait since three legs are kept in ground contact. But gait specification, duty factor and relative phase, were maintained as expected.

Table 4. Optimization Sensor Results

Generation	$Vibration$	$Velocity(mm/s)$	$wsm(mm)$	Fitness
1st Generation	0.177	80.623	3.508	0.357
50th Generation	0.0233	50.980020	31.253	0.271

5 Conclusion

In this article, we have addressed the locomotion optimization of a quadruped robot that walks with a walking gait.

A locomotion controller based on dynamical systems, CPGs, generates quadruped locomotion. These CPG parameters are tuned by an optimization system. This optimization system combines CPGs and a genetic algorithm. As a result, sets of parameters obtained by the genetic algorithm were adequate for the implementation of a locomotion walking gait with a velocity of 50.98 (mm/s), low vibration and a high wide stability margin.

Currently, we are using other optimization methods such as evolutionary strategies and electromagnetism algorithm. 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 implementation on the Aibo ers7 the locomotion optimization similarly to [2].

6 Acknowledgments

Work supported by the Portuguese Science Foundation (grant PTDC/EEA-CRO/100655/2008).

REFERENCES

[1] Luiz Castro, Cristina Santos, Miguel Oliveira, and Auke Ijspeert, 'Postural control on a quadruped robot using lateral tilt: A dynamical system approach.', in *EUROS*, volume 44 of *Springer Tracts in Advanced Robotics*, pp. 205–214. Springer, (2008).
[2] Sonia Chernova and Manuela Veloso, 'An evolutionary approach to gait learning for four-legged robots', in *In Proceedings of IROS'04*, (September 2004).
[3] R.B. Deb, K. Agrawal, 'Simulated binary crossover for continuous search space.', *Complex Systems*, **9(2)**, 115–149, (1995).
[4] Sarah Degallier, Cristina Santos, Ludovic Righetti, and Auke Ijspeert, 'Movement generation using dynamical systems: a humanoid robot performing a drumming task', in *IEEE-RAS International Conference on Humanoid Robots*, (2006).
[5] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
[6] Dragos Golubovic and Huosheng Hu, 'Evolving locomotion gaits for quadruped walking robots', *Industrial Robot: An International Journal*, **32**, 259 – 267, (2005).
[7] S. Grillner, 'Locomotion in vertebrates: central mechanisms and reflex interaction.', *Physiological Reviews*, **55**, 247–304, (1975).
[8] S. Grillner, 'Neurobiological bases of rhythmic motor acts in vertebrates', *Science*, **228**(4696), 143–149, (1985).
[9] Freyr Hardarson, *Stability analysis and synthesis of statically balanced walking for quadruped robots*, Ph.D. dissertation, KTH, 2002.
[10] Gregory S. Hornby, Seichi Takamura, Takashi Yamamoto, and Masahiro Fujita, 'Autonomous evolution of dynamic gaits with two quadruped robots', *IEEE Transactions on Robotics*, **21**, 402–410, (2005).
[11] Gia Loc Vo Tran Duc Trong Young Kuk Song Ig Mo Koo, Tae Hun Kang and Hyouk Ryeol Choi, 'Biologically inspired control of quadruped walking robot', *International Journal of Control, Automation and Systems*, **7**(4), 577–584, (2009).
[12] K. Inagaki and H. Kobayashi, 'A gait transition for quadruped walking machine', *Proceeding of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Ssytems*, (1993).
[13] Min Sub Kim and William Uther, 'Automatic gait optimisation for quadruped robots', in *In Australasian Conference on Robotics and Automation*, (2003).
[14] Nate Kohl and Peter Stone, 'Machine learning for fast quadrupedal locomotion', in *in The Nineteenth National Conference on Artificial Intelligence*, pp. 611–616, (2004).
[15] Vítor Matos, Cristina P. Santos, and Carla M. A. Pinto, 'A brainstem-like modulation approach for gait transition in a quadruped robot', in *IROS*, pp. 2665–2670, (2009).
[16] R. B. Mcghee and A. A. Frank, 'On the stability properties of quadruped creeping gaits', *Mathematical Biosciences*, **3**(1-2), 331–351, (August 1968).
[17] O. Michel, 'Webots: Professional mobile robot simulation', *Journal of Advanced Robotics Systems*, **1**(1), 39–42, (2004).
[18] Ludovic Righetti and Auke Jan Ijspeert, 'Pattern generators with sensory feedback for the control of quadruped locomotion', *IEEE International Conference on Robotics and Automation*, (2008).
[19] Thomas Röfer, 'Evolutionary gait-optimization using a fitness function based on proprioception', in *RobuCup*, pp. 310–322, (2004).
[20] Manish Saggar, Nate Kohl, and Peter Stone, 'Autonomous learning of stable quadruped locomotion', in *RoboCup2006: Robot Soccer World Cup X*. Springer Verlag, (2007).
[21] S. Song and K. Waldron, *Machines that Walk: The Adaptive Suspension Vehicle*, MIT Press, Cambridge, 1989.
[22] Kobayashi Toshiya Inoura Takashi Masuda Tatsuya Tsujita, Katsuyoshi, 'A study on adaptive gait transition of quadruped locomotion', in *none*, pp. 2489 – 2494, Tokyo, 113, Japan, (2008).