

# Generation of dynamic gaits in real Aibo using distributed neural networks

## Overview

Typically, gaits for autonomous robots are developed either manually or by developing walking algorithms, been in both cases necessary the implementation by hand of the walking pattern. Recently, some researchers have started to look for inspiration on biology and try to imitate the mechanisms that allow animals to walk. In this research we followed that path and used our distributed neural architecture to implement Central Pattern Generators and the coordination mechanisms between them, in order to make the Aibo robot walk.

## Motivation

We have two main motivations for this research:

- To test our distributed neural architecture and prove its validity when used on a dynamic task
- To create an autonomous walking mechanism for Aibo, 100% controlled by neural networks (no programmer code inside)

## Method

We apply neuro-evolutionary methods over our distributed architecture. Even using the systematic architecture, the walking behavior required to perform the evolution in three stages:

1. First the CPGs are evolved
2. Then CPGs of a same type are interconnected, evolving the coordination weights. This leads to the obtention of three different layers of CPGs
3. Last, the three layers are interconnected and the coordination weights evolved.

Experiments were run using [Webots](#) simulator and its Aibo 3D model. Once evolved on the simulator, the neural nets obtained were transfered to the real robot. The transference feature of Webots (from Aibo simulator to real Aibo robot) was a developing collaboration between our group, Cyberbotics and the EPFL, and it is at present included in the comercial version of the simulator.

## Results

Using the proposed method we achieved a quick and dynamic gait for Aibo. You can check yourself the results by looking at the following videos or trying the code below

### **Several simulation videos:**

[Video-1](#) | [Video-2](#) | [Video-3](#)

### **Several real robot videos:**

[Walking behavior-1](#) | [Walking behavior-2](#) | [Detail of legs oscillations](#)

### **Real robot controller used:**

You can test our results on your own Aibo robot, by downloading from here the OPEN-R neural controller and the neural nets evolved. Just download the OPEN-R program, unpack it and install it in the `/MS/OPENR` directory of an already prepared OPEN-R memory stick (if you don't know how to do it, please read the '[Aibo Quickstart Manual](#)'). Then download the neural nets, unpack them and install them into the

`/MS/OPENR/MW/DATA/P` directory of your memory stick.

By running the program on Aibo, the robot will perform a series of walking steps, governed by the neural nets, and then it will stop for some seconds for recording the joints stored values in memory (MS writing is very slow). A file called `MSV.csv` will be created in the `/MS/OPENR/MW/DATA/P` directory containing all the joint positions recorded during walking. After it has been written, the robot will continue walking again for some other steps. If you have any problem do not hesitate to [contact us](#).

OPEN-R neural program [here](#) | Neural nets [here](#)

## Results using libURBI

The real robot results presented above run onboard the robot by using the cross-compilation feature of Webots. However, it is possible to use the [URBI programming interface](#) to run the neural controller in C++ on a computer, controlling the real robot wirelessly from the computer. This is possible using libUrbi in two different modes, synchronous and asynchronous. Our tests show that synchronous mode is not suitable for this highly dynamic task and that asynchronous mode is more appropriate and close to the onboard control.

### **libURBI implementation videos of the walking behavior:**

[Synchronous mode](#) | [Asynchronous mode](#)

### **libURBI robot controllers used:**

In order to test our results on your own Aibo robot, you need to download and install first the URBI server on a memory stick, and the libURBI libraries on your computer. Then compile and execute the code provided below. Libraries required for the linkage between libUrbi and the neural net functions are provided within the same files.

If you have any problem do not hesitate to [contact us](#).

[Synchronous control program](#) | [Asynchronous control program](#)

## Related publications

R. Téllez, D. Pardo and C. Angulo, [Comparison of synchronous and asynchronous control modes on dynamic control](#) (abstract), *First URBI Workshop 2006*, Paris, France, 2006. ([slides](#))

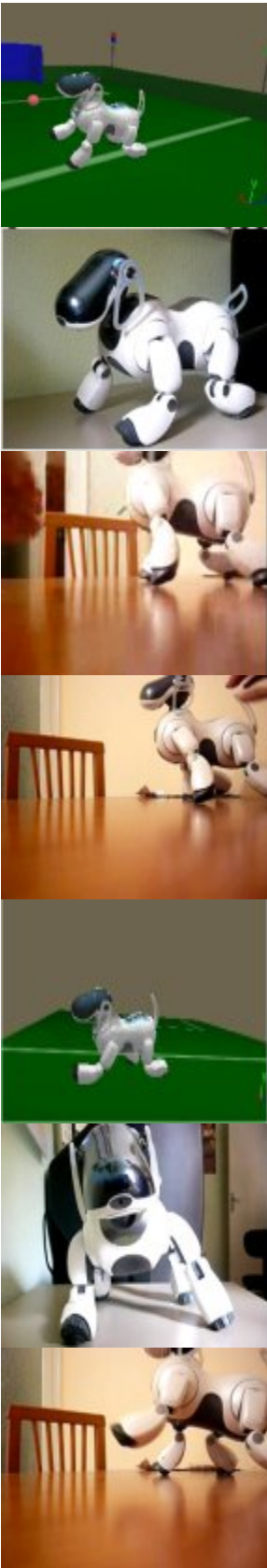
R. Téllez, C. Angulo and D. Pardo, [Evolving the walking behaviour of a 12 DOF quadruped using a distributed neural architecture](#), *2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT'2006)*, Osaka, Japan, 2006. Published in *Lecture Notes in Computer Science*, Volume 3853, p. 5 - 19, 2006

C. Angulo, R. Téllez, and D. Pardo, [Emergent Walking Behaviour in an Aibo Robot](#), *ERCIM News 64*, pp. 38-39. ERCIM EEIG, 2006

L. Holh, R. Téllez, O. Michel and A. Ijspeert, [Aibo and Webots: simulation, wireless remote control and controller transfer](#), *Robotics and autonomous systems*, Volume 54, Issue 6, pp 472-485, 2006

## The making off

Soon to be added!



Web page by R. Téllez using rubric css by Hadley Wickham

*Don't undertake a project unless it is manifestly important and nearly impossible (Edwin Land)*