Development of a neuromorphic control system for a lightweight humanoid robot

Michele Folgheraiter, Amina Keldibek, Bauyrzhan Aubakir, Shyngys Salakchinov

School of Science and Technology, Robotics and Mechatronics Department, Nazarbayev University, Kazakhstan

E-mail: michele.folgheraiter@nu.edu.kz, b.aubakir@nu.edu.kz, akeldibek@nu.edu.kz, shyngys.salakchinov@nu.edu.kz

Giuseppina Gini, Alessio Mauro Franchi, Matteo Bana

DEIB, Politecnico di Milano, piazza L. da Vinci 32, Milano, Italy

E-mail: giuseppina.gini@polimi.it, alessiomauro.franchi@polimi.it, matteo.bana@mail.polimi.it

Abstract. A neuromorphic control system for a lightweight middle size humanoid biped robot built using 3D printing techniques is proposed. The control architecture consists of different modules capable to learn and autonomously reproduce complex periodic trajectories. Each module is represented by a chaotic Recurrent Neural Network (RNN) with a core of dynamic neurons randomly and sparsely connected with fixed synapses. A set of read-out units with adaptable synapses realize a linear combination of the neurons output in order to reproduce the target signals. Different experiments were conducted to find out the optimal initialization for the RNN's parameters. From simulation results, using normalized signals obtained from the robot model, it was proven that all the instances of the control module can learn and reproduce the target trajectories with an average RMS error of 1.63 and variance 0.74.

1. Introduction

For humanoid robots to be able to operate in household and public environments, they should be physically safe when sharing the workspace with humans and be autonomous and adaptable while performing locomotion, object recognition, and grasping.

The conventional approach towards ensuring safe human robot interaction is to prevent collisions, which is usually achieved by setting mechanical barriers, using special actuators, or internal force and torque sensors [1]. This cannot be implemented for humanoid robots as it leads to limited workspace, which is not acceptable in operating environments shared with humans where physical contact is inevitable. Therefore, the system should be designed such that forces applied on human are below the limit that can injure or inhibit the human operator. Our solution is to decrease the weight and the inertia of the links by employing light-weight materials and high power-to-weight ratio actuation systems. This has the effect of reducing the potential and kinetic energy of the robot limbs while operating and therefore, makes human robot interaction safer. Following these guidelines, the humanoid we designed has a total of 28 Degrees of Freedom

(DOF) and a weight of $3kg$. In particular, as depicted in figure 1, there are 6 DOF in each leg, 6 DOF in each arm, 2 DOF in the pelvis and 2 DOF in the neck. This will allow the robot to perform 3D dynamic walking and to reach different objects in its workspace by performing human-like motion.

Autonomy and the capability to adapt to new environments are very important features for a humanoid robot. The learning and execution phases occur at the same time in biological organisms as they are forced to exist and survive in a changing environment while simultaneously learning and improving their sensory-motor strategies. Therefore, a challenging goal for humanoid robotics is to develop control systems that can be executed and adapted while the robot is operating. By integrating Artificial Neural Networks (ANN) in the control architecture the robot can elaborate input signals in real time [2] and respond to stimuli through a set of coordinated motor actions [3], [4], [5]. RNN based on leak-integrate-and-fire neurons were proven to reproduce the behavior of microcircuits located in cortex of the human brain. In particular, they are capable to predict sensory inputs [6], compute tasks in real-time such as speech recognition and analysis of visual information [7] [8] [9] and generate reaching trajectories [10]. In [11] it was demonstrated that the presence in the circuit of feedback signals from read-out units with adaptable synapses allow the RNN to learn complex periodic signals without the excitation of additional inputs. The synapses can be adapted on the base of an error signal calculated as the difference between the target signal and the output of the neural circuit or by a reward-modulated Hebbian learning rule [12]. This represents a more biologically plausible mechanism that can be used in all the situations where the current performance of the robot depends on motor actions performed in the past [13].

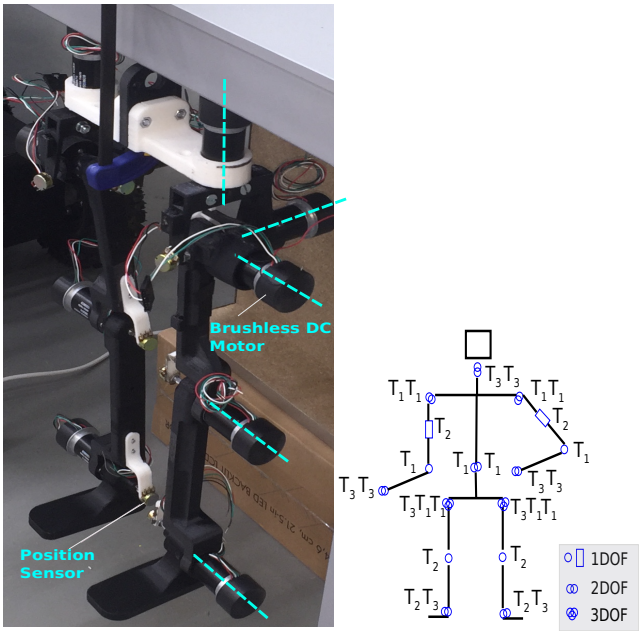

Figure 1: 28 DOF humanoid robot's kinematics architecture.

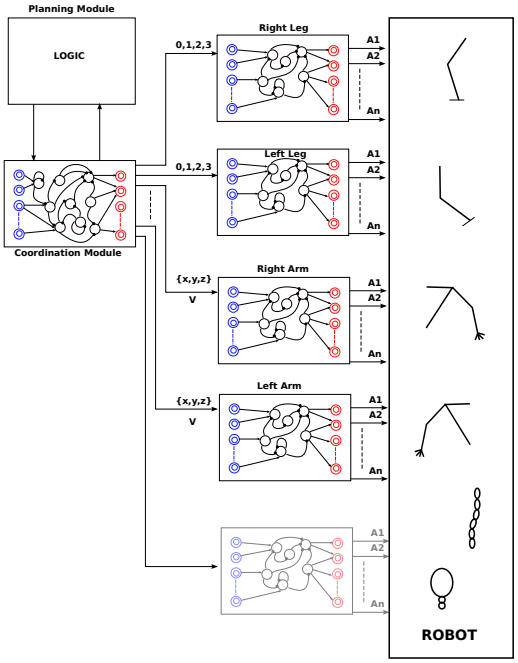

Figure 2: Control architecture.

The rest of the document is organized as follows: section 2 describes the control system and details a single RNN model, section 3 reports the preliminary results on a single control module capable to learn on training data obtained with a robotics simulator and to generate desired periodic joint trajectories. Finally, last section draws the conclusions and outlines future work.

2. Neuromorphic control architecture

The control architecture of the robot is organized in a hierarchical manner as depicted in figure 2. Neural modules that regulate the motion of each limb are at the lowest level. These modules have the same structure; in particular, each module is a RNN with fixed connections. The inputs and outputs of the module are routed towards an input and output layer of read-in and read-out units with tunable synapses. However, the number of read-in and read-out units can be adapted during the robot operation. This brings the advantage to introduce new inputs and outputs when required to perform a new task. Furthermore, if the read-in or read-out unit is wired to a specific piece of hardware (e.g. a joint motor or sensor) the architecture allows creating a new instance of the unit, adapting the correspondent synapses and passing smoothly the control of the hardware from previous to the new state. Low level modules that control leg motion have gait frequency, walking step size height as inputs whereas modules controlling arms receive target position, velocity, and instantaneous impedance as inputs.

Quantity	Value
Number of neurons N	100
Number of external inputs M	0
Number of read-out units L	5
Computational time, s	0.0001
Neuron time constant τ	0.013
Learning constant η	0.002
Decay learning constant dl	200
Chaos-modulation constant C	1.2

Table 1: The neural network parameters

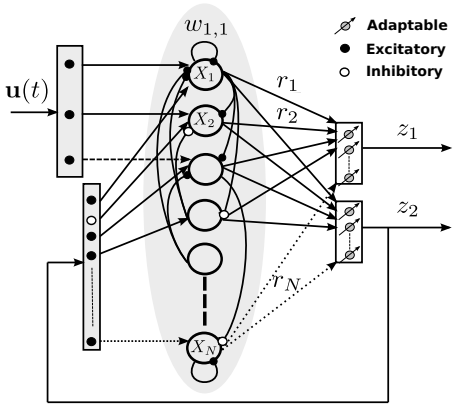

Figure 3: The single control module consists of a RNN where the outputs Z_i represent the read-out units.

At the highest level of architecture lies the coordination module that governs each neural module according to the specific tasks. This module receives commands from the planning module that implements the task; then, it sends feedback on the current state back. This layer of the control architecture is not based on RNN but rather on logic submodules.

2.1. Joints trajectory generation module

Each module of the control architecture depicted in figure 2 is capable to learn and generate different elementary motor paths that represent the building blocks to implement more complex locomotion and manipulation primitives.

The RNN we implemented (figure 3) consists of a reservoir of neurons which are randomly and sparsely connected through excitatory and inhibitory synapses of different strengths which are set randomly as well [10]. In particular, a single-layer architecture is used; i.e. it was demonstrated that addition of hidden layers does not provide any significant improvements [16]. Feedback loops that send the output of linear read-out units back to the network should be included to make learning possible without any external input. Results of experiments conducted by Sussillo and Abbott [14] showed that an initially chaotic network is faster to train and generates more accurate and robust output signals; therefore, the initial activity of our model is set to chaotic. In this RNN architecture only read-out unit weights are modified whereas other connections are not altered since their random initialization.

The RNN neuron is modeled by a differential equation (1), where x_i is the membrane potential, τ_i a time constant, C a parameter that modulates the chaoticity level of the RNN, r_j and u_j, the output and input of the j^{th} neuron respectively, and z_j the signal generated by the read-out units. A nonlinearity (2) is added by a sigmoidal activation function that has also the function to limit the output of the neurons between $[-1\ 1]$.

$$\tau_i \dot{x}_i(t) = -x_i(t) + C\sum_{j=1}^{N} w_{ij}^{c} r_j(t) + \sum_{j=1}^{L} w_{ij}^{fb} z_j(t) + \sum_{j=1}^{M} w_{ij}^{in} u_i(t) \tag{1}$$

$$\Phi(x) = \frac{1 - e^{-kx}}{1 + e^{-kx}} \tag{2}$$

To explore new dynamic regimes [15] a Gaussian noise with zero average and variance equal to 0.005 is added to the neurons output and the read-out units (equations 3 and 4 respectively).

$$r_j(t) = \Phi(x_j(t)) + GNoise_j(t) \tag{3}$$

The probability of a connection between two neurons in the reservoir is $P = 0.1$ and the input and feedback weights are set randomly according to the uniform distribution $[-11]$. The reservoir comprises a total of 50% of inhibitory synapses and 50% of excitatory one. Although the circuit is able to generate periodic trajectories without additional signals, external inputs $\mathbf{u}(t)$ can be added to influence the neuron potentials according to the weights vector $\mathbf{W^{in}} \in \mathbb{R}^N$. Using this we can eliminate the presence of phase shift in the generated trajectories with time. Furthermore, an additional signal scaled by the L by N matrix $\mathbf{W^{fb}}$ forms a feedback from the outputs of read-out neurons.

The RNN's output $\mathbf{z}(t)$ is computed by a linear combination (4) of the outputs of the read-out units $\mathbf{r}(t)$ of a L by N matrix $\mathbf{W^{Ad}}$.

$$\mathbf{z}(t) = W^{Ad}\mathbf{r}(t) + GNoise(t) \tag{4}$$

The adaption of the matrix $\mathbf{W^{Ad}}$ is performed by a simple learning rule based on the error calculated as the difference between the target vector and the filtered version $\mathbf{z}(t)$ of the read-out unit vector $\bar{\mathbf{z}}(t)$ at the instant t as in (5).

$$\mathbf{Err(t)} = \bar{\mathbf{z}}(t) - F(\mathbf{z}(t)) \tag{5}$$

The read-out synapses are updated according to (6),

$$\mathbf{W^{Ad}}(t+1) = \mathbf{W^{Ad}}(t) - \eta(t)F(\mathbf{r}(t))\mathbf{Err(t)} \tag{6}$$

the learning constant decays according to the rule in (7).

$$\eta(t+1) = \frac{\eta(t)}{1 + t/dl}. \tag{7}$$

3. Simulation results

Training data was obtained from a simulation of robot static walking. We used the open source V-REP simulator [16] that has such functionalities as robot kinematics and dynamics, collision detection, virtual sensors and actuators, and particle dynamics. We acquired the position of the right leg joints during the execution of a static walk with a sampling time of $10ms$ for $220s$ and a total of 22000 samples.

The RNN we implemented is not computationally expensive [17] and can run on power-efficient computational units. It consists of 100 neurons and 5 read-out units receiving as inputs all the neurons outputs. The parameters of the circuit are reported in the table 1.

Particularly critical are the initialization of the weight matrix $W^C(i,j)$ for reservoir neurons, the chaos-modulation constant, the time constant, and the learning constant. The first three parameters regulate the RNN dynamics that transforms from an initially chaotic to a stable with the neuron outputs as periodic signals of constant amplitude. This allows to reproduce the target signals with a specific linear combination of the neurons outputs. The learning constant value should be higher at the beginning of the learning phase to force the RNN to reach a limit cycle (see figure 5) and then should be reduced.

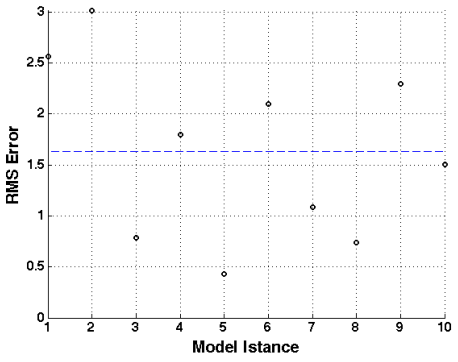

Figure 4: Overall RMS error for each of the 10 instances calculated on the validation set. The dashed line represents the mean of the RMS error.

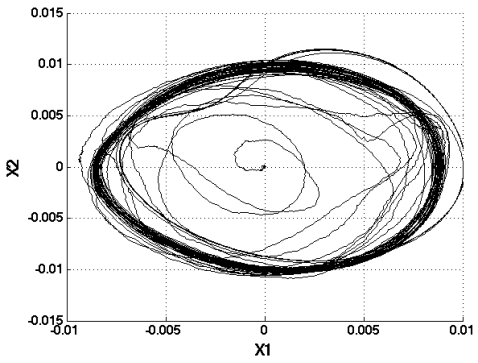

Figure 5: First two principal components of the neurons potential signals demonstrating the convergence ability of RNN circuit.

In total the adaptation phase lasts for $200s$. To find an optimal value of the matrix $W^C(i,j)$ we performed different experiments with different parameters kept constant and the matrix $W^C(i,j)$ randomly initialized for each model instance. The RNN was trained and the overall RMS error calculated as the sum of the RMS errors for each read-out unit. From figure 4 the average RMS error is 1.63 and variance is 0.74.

In figures 6 and 7 the outputs of the first two read-out unit are reported. The target trajectory is represented with a black dashed line and the RNN output with a continuous red line. From this measurements we can see that after the learning phase the RNN is able to reproduce periodic signals autonomously.

We also observed that after $100s$ the RNN outputs start to experience a phase shift. This can be eliminated by introducing an additional RNN input with a specific frequency that can serve as clock reference in the system.

4. Conclusion

In this paper we presented a lightweight humanoid robot equipped with a neuromorphic control architecture. The joints trajectories for a static walking are learned and reproduced by different modules organized into a hierarchical architecture. Each module is represented by a RNN consisting of 100 neurons modeled by a first-order differential equation. The neurons are coupled with fixed connections randomly initialized while special read-out units with adaptive synapses realize a linear combination of the neurons outputs in order to reproduce complex periodic signals. To optimize the weight matrix, ten experiments were performed where the training and validation phases were accomplished and the RMS error calculated. Preliminary simulation results show

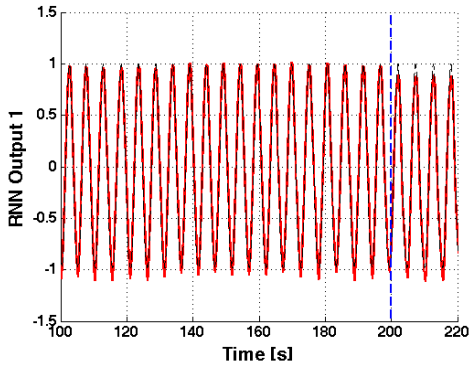

Figure 6: Read-out 1 reference signal with dashed line and RNN signal with red line.

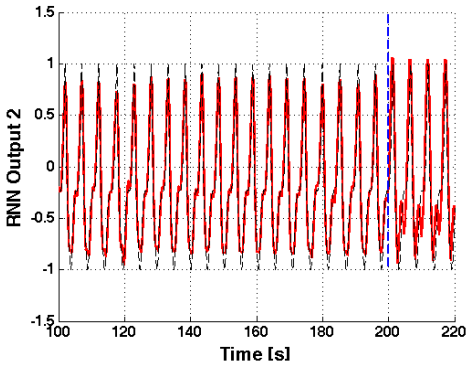

Figure 7: Read-out 2 reference signal with dashed line and RNN signal with red line..

that despite the small dimension of the network the module is able to reproduce the required trajectories. As a future work, we intend to implement the algorithm on a small computational unit e.g. a Raspberry Pi 3 or a BeagleBone and test the trajectories on the real prototype. It will be interesting also to integrate in the control architecture additional input signals from the sensory system of the robot. This will allow to produce more complex motor behaviors, e.g. alter the step size and height to avoid obstacles in its path.

Acknowledgment

This work was supported by the Ministry of Education and Science of the Republic of Kazakhstan under the grant and target funding scheme agreement #(348)226 – 2016.

Reference

[1] Lasota P A, Rossano G F and Shah J A *IEEE International Conf. on Automation Science and Engineering (Taipei, Taiwan, August 2014)* (IEEE) pp 339–344
[2] Folgheraiter M, Gini G, Nava A and Mottola N 2006 *Proc. of IEEE Robio06* (Kunming: IEEE) pp 1646–1651
[3] Chow T W S and Fang Y 1998 *IEEE Trans. on Industrial Electronics* **45** 151–161
[4] Carrillo R R, Ros E, Boucheny C and Coenen O J M 2008 *Biosystems* **94** 18–27
[5] Webb B and Consi T R (eds) 2001 *Biorobotics* (MIT Press)
[6] Markram H, Wang Y and Tsodyks M 1998 *Proc. Natl. Acad. Sci.*
[7] Legenstein R, Markram H and Maass W 2003 *Reviews in the Neurosciences* **14** 5–19
[8] Buonomano D V and Merzenich M M 1995 *Science* **267** 1028–1030
[9] Maass W, Natschlaeger T and Markram H 2002 *Neural Computation* **14** 2531–2560
[10] Joshi P and Maass W 2005 *Neural Computation* **17** 1715–1738
[11] Maass W, Joshi P and Sontag E D 2007 *PLoS Comput. Biol.* **3** e165
[12] Hoerzer G M, Legenstein R and Maass W 2014 *Cereb. Cortex* **24** 677–690
[13] Sutton R 1984 Ph.D. thesis University of Massachusetts Amherst
[14] Sussillo D and Abbott L F 2009 *Neuron* **63** 544–557
[15] Maass W 2014 *Proc. of the IEEE* **102** 860–880
[16] E Rohmer S P N Singh M F *Proc. of the International Conf. on Intelligent Robots and Systems (Tokyo, Japan, 2013)* (IEEE) pp 1321–1326
[17] Hoerzer G M, Legenstein R and Maass W 2012 *Cerebral Cortex* **24** 677–690