



University of Southern Denmark

Ground Reaction Force Estimation in a Quadruped Robot via Liquid State Networks

Arena, Paolo; Cusimano, Maria Francesca Pia; Meli, Lorenzo Emmanuele; Taffara, Salvatore; Patane, Luca; Poramate, Manoopong; Manoonpong, Poramate

Published in:

2022 International Joint Conference on Neural Networks, IJCNN 2022 - Proceedings

DOI:

10.1109/IJCNN55064.2022.9892423

Publication date:

2022

Document version:

Accepted manuscript

Citation for published version (APA):

Arena, P., Cusimano, M. F. P., Meli, L. E., Taffara, S., Patane, L., Poramate, M., & Manoonpong, P. (2022). Ground Reaction Force Estimation in a Quadruped Robot via Liquid State Networks. In *2022 International Joint Conference on Neural Networks, IJCNN 2022 - Proceedings* IEEE.
<https://doi.org/10.1109/IJCNN55064.2022.9892423>

Go to publication entry in University of Southern Denmark's Research Portal

Terms of use

This work is brought to you by the University of Southern Denmark.

Unless otherwise specified it has been shared according to the terms for self-archiving.

If no other license is stated, these terms apply:

- You may download this work for personal use only.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying this open access version

If you believe that this document breaches copyright please contact us providing details and we will investigate your claim. Please direct all enquiries to puresupport@bib.sdu.dk

Ground Reaction Force Estimation in a Quadruped Robot via Liquid State Networks

Paolo Arena, Maria Pia Cusimano,
Lorenzo Emmanuele Meli, Salvatore Taffara
DIEEI - Università degli studi di Catania
Catania, Italy
paolo.arena@unict.it

Luca Patanè
Dipartimento di Ingegneria
Università degli studi di Messina
Messina, Italy
lpatane@unime.it

Manoopong Poramate^{1,2}

¹*Embody Artificial Intelligence and Neurorobotics Laboratory,*
SDU Biorobotics, The Mærsk Mc-Kinney Møller Institute University of Southern Denmark
Odense, Denmark

²*College of Mechanical and Electrical Engineering,*
Nanjing University of Aeronautics and Astronautics
Nanjing, China
poma@mmti.sdu.dk

Abstract—This paper aims to investigate the Liquid State Machines (LSMs) learning capability and robustness of a complex robot-environment interaction. The goal is to design an efficient robot state estimation method based on reservoir computing. The method maps local proprioceptive information acquired at the level of the leg joints of a simulated quadruped robot. The robot taken into account is the simulated version of Lilibot, a small-sized and reconfigurable bio-inspired robot with multiple real-time sensory feedback. Global information was provided from the ground reaction forces acquired on the tips of each leg. Simulation results are reported and compared, also in presence of faulty conditions in the sensory system.

I. INTRODUCTION

Navigation over complex and unstructured terrains represents a challenge for wheeled, legged, and hybrid robots that have to demonstrate high stability and adaptation capabilities [1]. Terrains subject to landslides, for example, are a difficult task for any robotic structure involved in terrain exploration since the ground conditions can change abruptly. Legged robots represent a good alternative to the classical wheeled robots, due to their adaptation capabilities to complex unstructured environments, but they pose serious challenges in terms of steering and attitude control. Nowadays, gait control is a problem for bipeds [2], hexapods [3], [4] and quadrupeds [5], and gait parameters such as the step length and walking speed should be optimized depending on the faced terrain.

In general, legged locomotion provides an important contribution in uneven terrain exploration due to the reduced contact area with the ground, if compared with wheeled vehicles. For example,

to assess locomotion stability, hexapods [6], [7] or even octopus robots [8] need to guarantee at least three contact points with the ground. The integration of effective control strategies allows reducing the number of legs needed, obtaining good performance, as in a mini cheetah-inspired quadruped robot [9]. In [10], a locomotion control strategy, based on foot sensory feedback, is implemented optimizing the force distribution between legs and terrain and, consequently, minimizing the risk of slipping. This technique can be used to stabilize and correct the robot gait when calculating the best route, once a starting and ending point has been determined. For all these reasons, quadrupedal locomotion guarantees a high level of robustness to disturbances permitting a relevant improvement of the robot's agility. However, locomotion reliability could be affected by such conditions as endogenous (e.g., electromechanical) faults, or exogenous faults depending on external disturbances, i.e., severe weather conditions, crashes in structure, or unstructured terrains. Several efficient control strategies, based on distributed sensing devices, have already been introduced in the literature [11]. However, a weakness in legged locomotion is the lack of efficient high-performance and low-cost sensing devices and processing techniques for identifying the terrain characteristics in real-time. It is well known that to achieve optimal locomotion stability in legged robots, haptic feedback is a primary information source, especially on unstructured terrains where real-time gait and posture control adaptation are needed [12]. The analysis of robot terrain-interaction is a critical requirement that can

be fulfilled using costly sensors. In our study, we provide an alternative efficient method for ground reaction force (GRF) estimation by exploring the use of liquid state machines (LSMs). The GRF estimation is based on proprioceptive local information acquired at the level of the leg joints of a quadruped robot. The dynamic model is defined by extracting the temporal dependencies between the input data during the learning process.

The methodology adopted is called reservoir computing (RC), which is a class of recurrent neural networks (RNNs) characterised by hidden random connections, where only the readout connections are trained. **RCs are generally used for designing data-driven models involving nonlinear dynamic behaviors. Respect to other neural architectures, RNN can process inputs of any length and, even if the input size is larger, the model size does not increase. For these reasons, we decided to use RNNs respect to other type of neural configurations [13], [14] for our application. REV 2,** n1 Liquid State Machines (LSMs, spiking neurons) and Echo State Networks (ESNs, continuous output neurons) are the two neural architectures belonging to the RC framework [15].

In literature, starting from a mathematical formulation, RCs were extended to include physical devices of the complex neural model [16], [17]. Nakajima and colleagues in [18] demonstrated the real-time computation capabilities of a soft silicon arm where nonlinearities, memory effects, and multiple degrees of freedom were enrolled in the computing mechanism. In [19], a reservoir compliant tensegrity robot was presented as an integrated part of the computational system used to generate a set of target oscillatory motor signals. In [20], an embodied control system has been formulated in a quadruped robot using its own dynamics.

In previous studies ([21], [22]) the application of ESN to develop a robot state estimation system for robot-terrain interaction analysis was shown. The performances were really interesting and outlined the efficiency of RC networks in this kind of application. Based on these results, in this work we further explore the application of LSMs to the same problem, i.e., to estimate the ground reaction forces generated at the tip of each touching leg in a quadruped robot, using proprioceptive local information acquired at the level of its leg joints. The model dynamics with embedded short-term/temporal memory is defined by extracting the temporal dependencies between the input data during the learning process. A comparative analysis between the ESN and LSM approaches is performed referring to the same task. Results show that using the same performance metrics, spiking recurrent networks (LSMs) can reach similar results

using much fewer neurons and learnable parameters than continuous echo state networks (ESNs), leading to greater computational efficiency. Thanks to the embedded memory of the LSM, we can achieve robust robot state estimation despite sensor failure. In other words, it can also serve as sensory memory. This outcome can open new perspectives into the field of neuromorphic bio-inspired robotics, where biological inspiration is not only related to structural mimicry but covers all the other aspects: from locomotion generation and control to the neuro-inspired way of efficient and robust signal processing for proprioception.

The remaining of this work is structured as follows: Section 2 shows an LSM overview and the experiment issues with the setup related to the proposed problem. In Section 3, the methodology used to set the LSM configuration is presented, while the simulation results are reported in Section 4 where the robustness also in presence of sensory faults is evaluated. Finally, the conclusions are drawn in Section 5.

II. LSM OVERVIEW

The LSM theory is strongly related to the RNN field. RNNs are powerful architectures typically used to compute sequential or time-series data such as in language translation, natural language processing, and speech recognition [13]. These types of networks, due to their recurrent nature, are characterized by a high computational effort and are subject to learning problems such as gradient vanishing [14], [23].

To solve the previous issues, Maass and colleagues in [15] introduced LSMs as an alternative approach to RNNs. LSMs have a randomly-generated, fixed "reservoir" of spiking neurons (no training), with a trainable output layer. The output layer is trained using either offline algorithm based on the Moore–Penrose inverse matrix or online methods based on the recursive least square algorithms. RC is computationally universal for continuous-time, continuous-value real-time systems modelled with bounded resources. There are several analogies between RC principles and architectural-dynamical properties of mammals. Another interesting aspect of the RC approach consists in the possibility to extend previously learned models without weakening or destroying previously learned representations. RC-based solutions, thanks to their limited number of trainable weights, have an intrinsically reduced computational complexity and training cost, exploiting random, fixed synaptic strengths [24], [25]. LSMs represent the third generation of neural networks, in which biological inspiration involves also the massive use of recurrent spiking neurons with randomly connected fixed synapses. Moreover, multiple readout maps are allowed to realise multiple parallel processing tasks

exploiting the same recurrent spiking lattice [26], [27]. Each neuron within the lattice (i.e., the liquid layer) is characterised by its state whose value is based on the number of spikes emitted in a given time window.

A. LSM mathematical formalisation

An LSM maps input streams $u(\cdot)$ onto output streams $y(\cdot)$ exploiting the dynamics of the internal liquid state $x(t)$ (see Fig.1). Input signals are translated into spiking sequences that are injected into the neurons belonging to the liquid layer. The different readout maps extract the liquid state information, modulated by trainable weights. Of course, the target value $y(t)$, i.e. the desired output at time t , may depend on the values $u(s)$ of the input streams at previous time steps s . However, there is no need for explicitly introducing the input or output delayed data since memory is built internally.

The liquid layer L maps input functions $u(\cdot)$ onto a state function:

$$x(t) = L(u(t)) \quad (1)$$

Moreover, it is important to specify that the readout map is a memory-less function f that transforms, at every time t , the current liquid state $x(t)$ into the output:

$$y(t) = f(x(t)) \quad (2)$$

f is generally chosen in a task-specific way. The memory-less property refers to the fact that f does not need to retain any memory of previous states of the liquid, but as a result of learning, the readout map will contribute to the system memory [27].

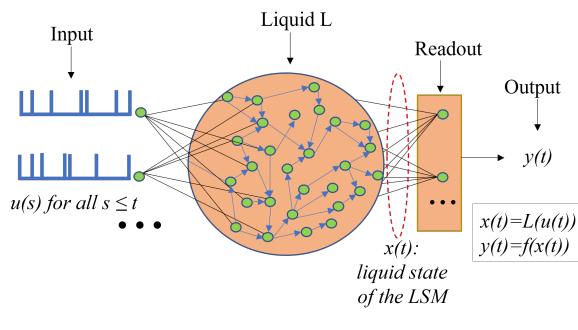


Figure 1. Structure of an LSM. L transforms input streams $u(\cdot)$ into output streams $y(\cdot)$. The liquid state $x(t) \in R^k$ is the input to the readout at time t .

B. LSM topology and parameters

The LSM structure was evaluated using the NEST simulation tool where a large collection of neuron and synaptic plasticity models is available [28]. The input data provided to the LSM is timely organized

in a stream as described in the following. The stimulus interval ($Stim_i$) is the time interval between two consecutive input-samples presentations, fixed to $60ms$. The stimulus interval consists of two time windows that identify the stimulus length and the readout delay. The stimulus length ($Stim_l$) is the time length in which the input is active and it is set to $50ms$, whereas the readout delay (R_d), fixed to $10ms$, represents the time left for the network evolution after the input injection and before the recording instant. The whole simulation time (Sim_t) is fixed to $3 \cdot 10^5 ms$.

After defining the relevant parameters and selecting the number of excitatory and inhibitory neurons in the liquid lattice (e.g., based on the chosen balancing between performance and execution time), the network architecture can be created as described in the following procedure:

- **Input layer:** defining the connections between the input data (e.g., 12-channel data related to the 12 actuated joints distributed in the quadruped robot legs, three joints per leg in the proposed application) and the step current generators following a one-to-one connection rule (i.e., one current generator for each input channel). The model synapses used are the static ones with an unitary weight. Using the step current generator, each input data is maintained for a given time window to be processed by the network. These steps have time periods set in the $Stim_i$ and are normalised in amplitude scaling the signals in a range $[0, 50] pA$. Each current generator is connected to one spike generator that, in the proposed network, is implemented using a class I Izhikevich's neuron [29], which generates the spike train to be injected into the liquid lattice. The equation defining the Izhikevich's neuron is the following:

$$\begin{cases} \frac{dV_m}{dt} = 0.04V_m^2 + 5V_m + 140 - u + I \\ \frac{du}{dt} = a \cdot (b \cdot V_m - u) \end{cases} \quad (3)$$

If $V_m \geq V_{th}$:

$$\begin{cases} V_m \leftarrow c \\ u \leftarrow u + d \end{cases} \quad (4)$$

where V_m is the membrane potential; u is the recovery variable that provides negative feedback to V_m ; V_{th} is the spike threshold; I is the input current; the parameter a describes the time scale of the recovery variable; the parameter b describes the sensitivity of the recovery variable to the subthreshold oscillations of V_m ; c represents the after-spike reset value of V_m ; finally the parameter d is the after-spike reset

value of the recovery variable. In our work, we used the typical default parameters: $a = 0.02$, $b = 0.2$, $c = -65$ and $d = 2$ to implement a Class I behaviour where the spiking frequency is proportional to the amplitude of the input current [29]. Spike generators are randomly connected with a subset of lattice neurons, called neuron targets. The fixed-outdegree connection synaptic rule is adopted: the nodes of an input layer are randomly connected with the nodes of the following layer such that each node in the first layer has a fixed out-degree N . This parameter is fixed to the 10% of the number of neuron targets. The synaptic model used is the static one with uniform weight distribution, setting the minimum weight value ($W_{low} = 125$) and the maximum weight value ($W_{high} = 375$). The delay for the connections is obtained using a normal-clipped distribution that is a Gaussian distribution which takes in input the two boundary parameters low limit delay (D_{low}) and high limit delay (D_{high}) beyond the standard deviation ν and the mean σ . In our work, we set: $\nu=10$, $\sigma=5$, $D_{low} = 3$ and $D_{high} = 200$. A flow chart representing the complete encoding process is reported in Fig.2.

- **Liquid layer:** the neurons in the liquid layer are implemented with the Izhikevich's model previously introduced. The synaptic connections follow the previously adopted fixed in-degree rule with $N = 2$ for excitatory-excitatory and excitatory-inhibitory connections and $N = 1$ for the inhibitory-excitatory and inhibitory-inhibitory ones. The adopted synaptic model, introduced by Tsodyks, implements synaptic short-term depression and short-term facilitation [30]. A Gaussian distribution is used for the weight setting with $\nu = 50$ and $\sigma = 0.7 \cdot |\nu|$. A Poisson generator is used to inject noise inside the excitatory and inhibitory neurons. The synaptic connections are static and a normal weight distribution is adopted with $\sigma = 1$ and $\nu = 0.7$. The delay distribution is obtained using a normal-clipped structure with $\sigma = 10$, $\nu = 20$, $W_{low} = 3$ and $W_{high} = 200$.
- **Output layer:** To evaluate the liquid lattice activity, the exponential time decay amplitude in a window of $60ms$ for each stimulus is collected, according to the readout delay chosen. This index can be expressed with the following equation [31]:

$$f_i(t) = e^{(-\frac{t-t_i^{spike}}{\tau})} \quad (5)$$

where $t_i^{spike} \in [0, t]$ is the last spike time of the liquid neuron i and τ is a global fading term fixed to $20ms$ in the following simulations. The

function f_i maps a spike train of a neuron i to a continuous signal. The state evolution of the LSM is then expressed as the sum, for each time window, of the exponential time decay of each readout neuron. The supervised learning rule adopted to determine the readout map is based on the Moore–Penrose inverse method as reported in the following:

$$W = (X^T X + kI_p)^{-1} X^T y_t \quad (6)$$

where W is the readout weight matrix, X correspond to the state matrix that includes the state of each readout neuron for each input pattern, k is a small constant gain introduced in presence of ill-conditioned matrices, I_p is the $p \times p$ identity matrix and y_t is the target signal.

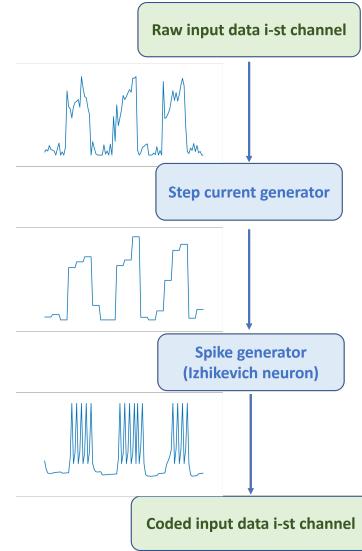


Figure 2. Flow chart of the encoding process for a generic i -th input channel. The raw input data are processed by the step current generator from which a train of step signals is obtained. This signal is used as input for the spike generator (i.e., Izhikevich's neuron) whose output represents the coded version of the input data.

III. THE SIMULATED ROBOT

The CoppeliaSim framework [32] provides an accurate and realistic dynamic simulation environment where performance can be evaluated before implementation on the actual robot prototype. This simulation approach becomes essential when a large amount of data has to be acquired, i.e., when data-based learning is involved.

The quadruped involved in the simulation experiments is a small size, lightweight, and open-sourced robot [33]. During the simulations, Lilibot walks at a fixed speed showing a trot gait, i.e., each leg is in phase with its diagonal and 180° out of phase with the other two legs.

The aim is to create a map between torques acquired at the leg joints and ground reaction forces acquired at the leg tips. The quadruped is shown in Fig.3. The robot legs are defined as: *FR* (front right leg), *HR* (hind right leg), *FL* (front left leg), *HL* (hind left leg). The involved joints are H_1 (hip 1 joint), H_2 (hip 2 joint), K (knee joint). The target signals consist of the vertical component of the ground reaction forces at the tip of each leg. The terrain configurations considered in the simulation environment are three: flat, downhill (-5°), and uphill ($+5^\circ$) from which three datasets were generated.

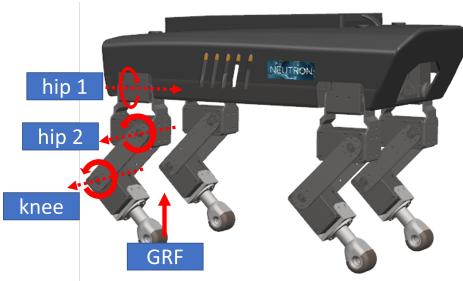


Figure 3. Lilibot robot used in the dynamic simulation environment. Each leg is characterized by three actuated joints for a total of 12 degrees of freedom.

IV. SIMULATION RESULTS

Experiments were carried out analyzing the data obtained simulating the Lilibot robot. In particular, the input patterns containing the torque signals of all leg joint was provided to the LSM. A total of 5000 patterns (80% for learning and 20% for testing) was acquired with a sampling time of 50ms. The estimation capability of the proposed architecture for each terrain configuration was evaluated separately. Two different input datasets were analyzed: the first one showing a perfectly working acquisition system and the other in presence of faulty sensors at the joint level, hence, associating a “zero torque” signal to a broken joint or leg, seen as a collection of joints, as needed.

To choose the best LSM topology configuration, a grid search was performed by changing the number of excitatory and inhibitory neurons and evaluating the corresponding MSE (Mean Square Error) index (see Fig.4). The MSE index is defined as follows:

$$MSE = \frac{\sum_{i=1}^n (y_i - y_{ti})^2}{n} \quad (7)$$

where y_i is the prediction, y_{ti} is the true value and n is the number of considered patterns.

The best network configuration was obtained for an LSM with a size of in total only 18 neurons having a 14-4 configuration (i.e., 14 excitatory and 4 inhibitory neurons). In this setup, we use all the excitatory neurons as readout neurons. This network

is considerably small if compared with the ESN network presented in [21] where 100 reservoir neurons are considered. Inputs are coded as step currents eliciting 12 Izhikevich’s class I neurons (i.e., one for each input) [29] whose operative input range varies in $[0 \text{ } 50] \text{ pA}$. This last interval was assumed as a normalization range for the input encoding and, as consequence, the “zero torques” was set equal to 25 pA.

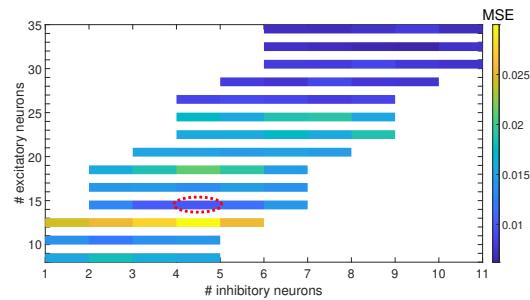


Figure 4. MSE values for different network topologies. The selected network is composed of 14 excitatory neurons and 4 inhibitory ones.

Based on the previous considerations, in the following analysis, an LSM with 14 excitatory neurons and 4 inhibitory neurons in the liquid layer was considered (see Fig.5).

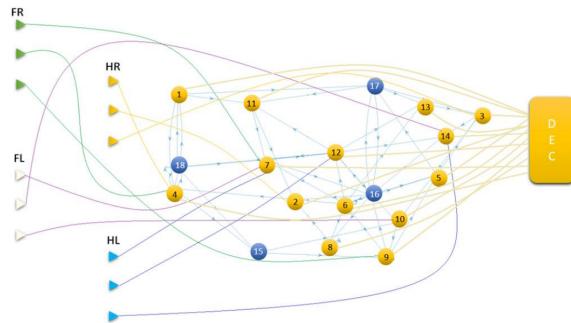
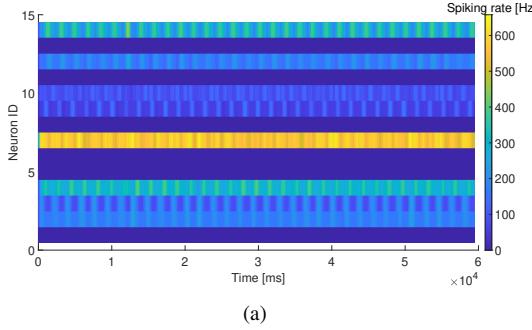


Figure 5. LSM 14-4 structure graph where all 14 excitatory neurons are also used as readout neurons. The inputs are represented by four groups of three triangles indicating the three joints (i.e., hip 1, hip 2, and knee) of each leg. The 14 yellow nodes represent the excitatory neurons and the 4 blue nodes are the inhibitory ones of the liquid layer. The decoder (i.e., DEC) represents the readout map which finally provides the four estimated GRFs.

A first analysis was focused on the excitatory spiking rate in order to investigate the dynamic range for each Izhikevich’s spiking neuron involved. The reservoir shows multiple distinct dynamic behaviours that underline the LSM activity in response to the injected stimulus. Some of the liquid neurons achieve a relevant spiking frequency, some others remain silent, failing to reach the activation threshold as can be seen in Fig.6 where the spiking rate map related to the excitatory neurons is shown.



(a)

Figure 6. Spiking rate map showing the spiking rate for the 14 excitatory neurons. Some of the neurons (ID 1-5-6-8-11-13, dark blue) are silent during the reported simulation.

The richness of the spiking activity for each time window is illustrated in Fig. 7 where the spike events and the state temporal evolution related to the neuron ID 7, taken as an example, are reported. In particular, Fig. 7(a) shows the exponential time decay amplitude in a window of 60ms for each stimulus whereas in Fig. 7(b) the state temporal evolution of the LSM is represented as the sum, for each time window, of the exponential time decays.

Considering flat terrain locomotion, as shown in Fig. 8, all legs follow the estimated GRF in a reasonable accuracy range. Fig. 9 reports the test results related to the FL leg, taken as an example, also for the downhill and uphill cases where similar performances are obtained.

In Table I, the MSE and its standard deviation (STD) are reported together with the Pearson correlation coefficient (R) for the three considered terrains, both in learning and test. The performance obtained with the use of LSM, in terms of MSE and R is similar to those one reported in [21]. It has to be noticed, however, that in this work, the LSM network is made up of only 18 neurons, in front of 100 neurons in the ESN proposed in [21]; the number of trainable weights, therefore, is much lower in the LSM case.

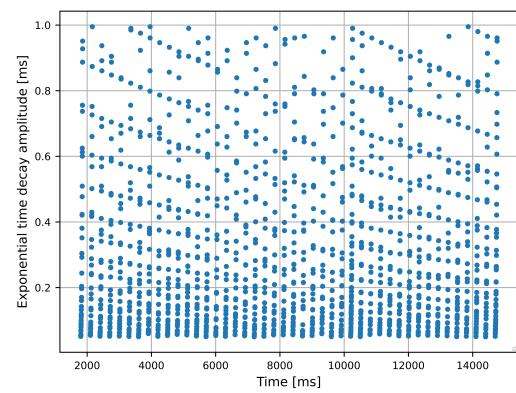
Table I

PERFORMANCE INDEXES: MSE WITH STANDARD DEVIATION AND CORRELATION COEFFICIENT R FOR THE LEARNING (L) AND TEST (T) PHASES.

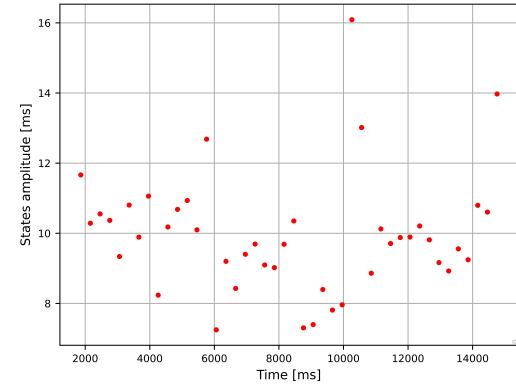
Terrain	MSE		STD		R	
	L	T	L	T	L	T
Flat	0.008	0.008	6.81%	6.75%	0.96	0.96
Downh.	0.007	0.008	5.97%	6.93%	0.97	0.96
Uph.	0.01	0.01	8.11%	7.97%	0.95	0.94

A. LSM robustness in presence of sensory faults

The robustness of the proposed network in presence of unexpected faults was also investigated. The subsequent simulations focus on the introduction of faults at the level of the joint torque sensors. It has

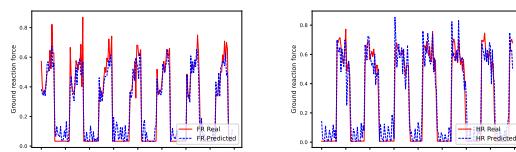


(a)

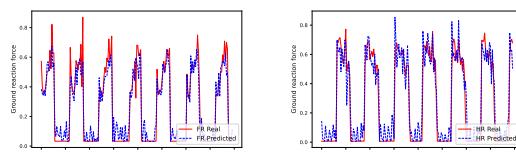


(b)

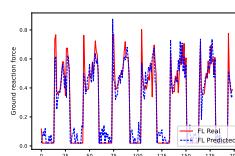
Figure 7. Temporal spiking evolution for the flat terrain scenario when neuron ID 7 is considered: (a) exponential time decays related to the readout neurons; (b) state temporal evolution.



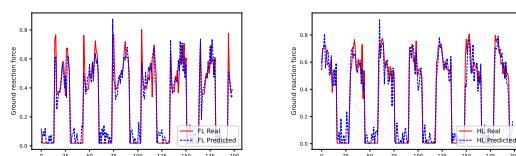
(a)



(b)



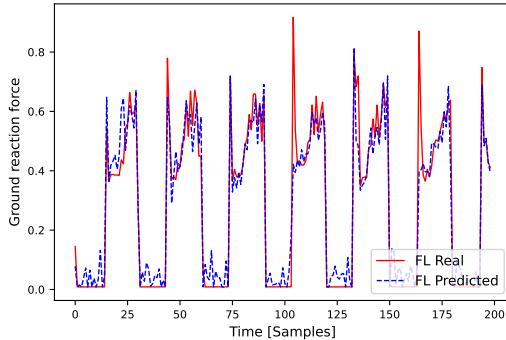
(c)



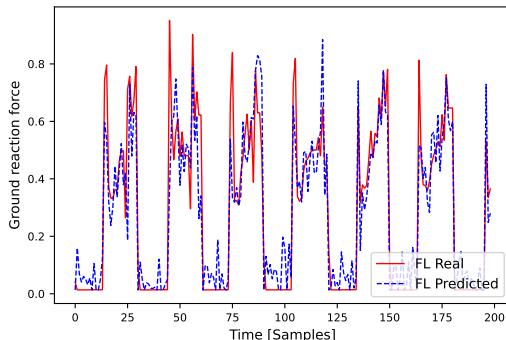
(d)

Figure 8. Test results for the flat scenario predicted vs target GRFs. (a) FR leg, (b) HR leg, (c) FL leg, (d) HL leg.

to be underlined that the LSM here evaluated have been trained without any fault in the dataset. So, the



(a)



(b)

Figure 9. Test results in the case of downhill (a) and uphill (b) for the FL leg.

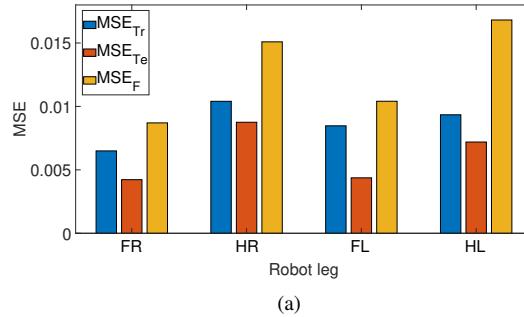
network is asked to reconstruct the missing information in one or multiple joints from the data coming from all the others. Each experiment aims to exploit the dynamical response of the network in presence of unexpected failures, for different configurations. The purpose is to assess the robustness of the LSM and its capability to work as a fading memory.

In Fig.10 the MSE and correlation coefficient between the estimated and actual GRFs are considered for each of the four legs when all the FR torque sensors are in fault. The degradation of the performance is particularly evident in the estimation of the GRF signal for the faulty leg, however, the still high level of correlation, with a maximum degradation of about 5.7%, guarantees a good following of the stepping sequence. The actual and estimated signals, when a fault in the time window between 200 and 800 samples is applied, are shown for the two legs FR (Fig.11(a)) and HL (Fig.11(b)). Also in this case it is evident that the sensory fault in FR is properly handled by the network through its fading memory and capability to exploit the available information coming from the other joints.

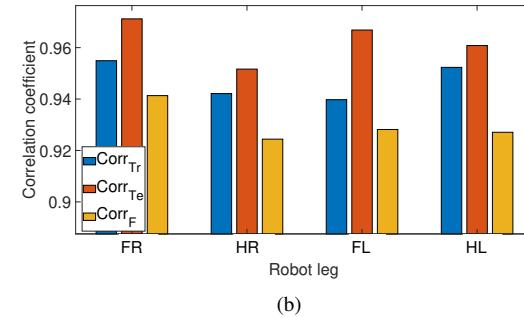
Estimation performances in terms of MSE between the actual and estimated GRF are reported in In

TableII where the outcome when each leg sensory system is in fault, is reported.

The obtained results demonstrate that the LSM is fairly robust to the presence of missing information at the level of the joint torque sensors. This is another outcome of the proposed approach if compared with ESN solutions as reported in [21] where the performance degradation was relevant if the faults were not presented during the learning phase.



(a)



(b)

Figure 10. FR sensory system in fault: (a) MSE between the estimated and actual GRF signals and (b) correlation coefficient in training (Tr), test (Te) and in presence of faults (F).

Table II
GRF MSE RESULTS IN TEST WHEN ALL THE JOINT OF ONE LEG ARE IN FAULT.

	MSE			
	FR	HR	FL	HL
FR in fault	0.015	0.011	0.011	0.007
HR in fault	0.014	0.040	0.008	0.028
FL in fault	0.014	0.016	0.024	0.013
HL in fault	0.008	0.015	0.010	0.010

V. CONCLUSIONS

This paper proposes a new architecture based on reservoir computing to be applied as an efficient and robust robot state estimation method in a quadrupedal robot-environment interaction scenario. The LSMs, as biologically inspired neural networks, showed reliable results in terms of learning capability and robustness to unexpected faults, exploiting the presence of an internal temporal memory through the recurrent connections and the potentiality of spiking

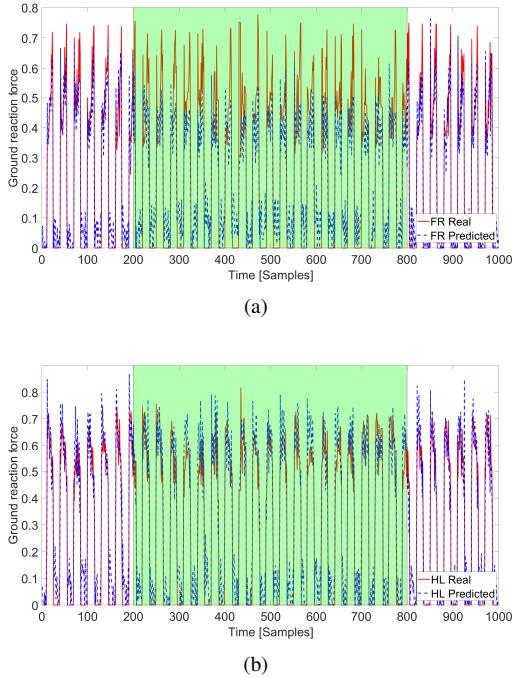


Figure 11. Test with a fault in the sensors of all the three joints in the FR leg, occurring in the time window between 200 and 800 samples (i.e., green area). Comparison between the target and estimated GRF signals for (a) FR and (b) the HL.

neurons processing. The proposed network is adopted as an efficient robot state estimation dealing with proprioceptive and exteroceptive information, able to estimate the GRF at the tip of the leg using joint torque information acquired on a simulated quadruped robot walking on different terrain slopes.

This study also suggests that spiking neurons in the LSM (used here) can encode temporal information more effectively than non-spiking neurons in the ESN [21]. The spiking neural technique can be also transferred to neuromorphic hardware toward the next generation of low power computation [22] for robot state estimation and motor control.

Several future applications and studies could involve the application of the developed system on the real robot on the basis of the promising results here reported.

REFERENCES

- [1] Luca Patan . Bio-inspired robotic solutions for landslide monitoring. *Energies*, 12(7), 2019.
- [2] Guilherme Franco, A. Coimbra, Manuel Cris stomo, and Jo o Ferreira. The nao robot in slippery scenarios: A strategy. *Journal of Information Systems Engineering and Management*, 6:em0133, 01 2021.
- [3] Rudolf Szadkowski, Milo  Pr gr, and Jan Faigl. Self-learning event mistiming detector based on central pattern generator. *Frontiers in Neurorobotics*, page 5, 2021.
- [4] Paolo Arena, Luigi Fortuna, and Mattia Frasca. Attitude control in walking hexapod robots: an analogic spatio-temporal approach. *International Journal of Circuit Theory and Applications*, 30(2-3):349–362, 2002.
- [5] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.
- [6] Eleonora Arena, Paolo Arena, and Luca Patan . Efficient hexapodal locomotion control based on flow-invariant subspaces. *IFAC Proceedings Volumes*, 44:13758–13763, 2011.
- [7] Jettanan Homchanthanakul and Poramate Manoonpong. Continuous online adaptation of bioinspired adaptive neuroendocrine control for autonomous walking robots. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [8] Christine Huffard, Farnis Boneka, and Robert Full. Underwater bipedal locomotion by octopuses in disguise. *Science (New York, N.Y.)*, 307:1927, 04 2005.
- [9] Paolo Arena, Fabio Di Pietro, Alessia Li Noce, Salvatore Taffara, and Luca Patan . Assessment of navigation capabilities of mini cheetah robot for monitoring of landslide terrains. In *2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI)*, pages 540–545, 2021.
- [10] Giorgio Valsecchi, Ruben Grandia, and Marco Hutter. Quadrupedal locomotion on uneven terrain with sensorized feet. *IEEE Robotics and Automation Letters*, 5(2):1548–1555, 2020.
- [11] JingYe He, JunPeng Shao, GuiTao Sun, and Xuan Shao. Survey of quadruped robots coping strategies in complex situations. *Electronics*, 8(12), 2019.
- [12] Tao Sun, Zhendong Dai, and Poramate Manoonpong. Distributed-force-feedback-based reflex with online learning for adaptive quadruped motor control. *Neural Networks*, 142, 06 2021.
- [13] Alejandro Chinea Manrique de Lara. Understanding the principles of recursive neural networks: A generative approach to tackle model complexity. 11 2009.
- [14] Abdel-Nasser Sharkawy. Principle of neural network and its main types: Review. *Journal of Advances in Applied & Computational Mathematics*, 7:8–19, 08 2020.
- [15] Wolfgang Maass, Thomas Natsch ager, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14:2531–2560, 2002.
- [16] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit H roux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 07 2019.
- [17] Kohei Nakajima. Physical reservoir computing—an introductory perspective. *Japanese Journal of Applied Physics*, 59, 04 2020.
- [18] Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. Information processing via physical soft body. *Scientific reports*, 5:10487, 05 2015.
- [19] Ken Caluwaerts, J r mie Despraz, Atil Iscen, Andrew Sabelhaus, Jonathan Bruce, Benjamin Schrauwen, and Vytaus Sunspiral. Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of the Royal Society, Interface / the Royal Society*, 11, 09 2014.
- [20] Jonas Degrave, Ken Caluwaerts, Joni Dambre, and Francis Wyffels. Developing an embodied gait on a compliant quadrupedal robot. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4486–4491, 2015.
- [21] Mario Calandra, Luca Patan , Tao Sun, Paolo Arena, and Poramate Manoonpong. Echo state networks for estimating exteroceptive conditions from proprioceptive states in quadruped robots. *Frontiers in Neurorobotics*, 15:118, 2021.
- [22] Sakyasingha Dasgupta, Dennis Goldschmidt, Florentin Worgotter, and Poramate Manoonpong. Distributed recurrent neural forward models with synaptic adaptation and cpg-based control for complex behaviors of walking robots. *Frontiers in Neurorobotics*, 9, 10 2015.
- [23] Barak A Pearlmuter. Gradient calculations for dynamic recurrent neural networks: A survey. *IEEE Transactions on Neural networks*, 6(5):1212–1228, 1995.

- [24] Nicholas Soures and Dhireesha Kudithipudi. Spiking reservoir networks: Brain-inspired recurrent algorithms that use random, fixed synaptic strengths. *IEEE Signal Processing Magazine*, 36(6):78–87, 2019.
- [25] R. de Azambuja, F. B. Klein, S. V. Adams, M. F. Stoelen, and A. Cangelosi. Short-term plasticity in a liquid state machine biomimetic robot arm controller. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3399–3408, 2017.
- [26] Gideon Gbenga Oladipupo. Research on the concept of liquid state machine, 2019.
- [27] Wolfgang Maass. Chapter 1 liquid state machines : Motivation , theory , and applications. 2010.
- [28] M.-O. Gewaltig and M. Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2:1430, 2007.
- [29] E.M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.
- [30] M. Tsodyks, A. Uziel, and H. Markram. Synchrony generation in recurrent networks with frequency-dependent synapses. *Journal of Neuroscience*, 20(1):RC50, 2000.
- [31] Jacques Kaiser, Rainer Stal, Anand Subramoney, Arne Roennau, and Rüdiger Dillmann. Scaling up liquid state machines to predict over address events from dynamic vision sensors. *Bioinspiration & Biomimetics*, 12, 06 2017.
- [32] E. Rohmer, S. P. N. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, 2013.
- [33] Tao Sun, Xiaofeng Xiong, Zhendong Dai, and Poramate Manoonpong. Small-sized reconfigurable quadruped robot with multiple sensory feedback for studying adaptive and versatile behaviors. *Frontiers in Neurorobotics*, 14:14, 2020.