


Spring 3-23-2016

Information Representation and Computation of Spike Trains in Reservoir Computing Systems with Spiking Neurons and Analog Neurons

Amin Almassian
Portland State University

Let us know how access to this document benefits you.

Follow this and additional works at: http://pdxscholar.library.pdx.edu/open_access_etds

 Part of the [Bioinformatics Commons](#), and the [Computer Engineering Commons](#)

Recommended Citation

Almassian, Amin, "Information Representation and Computation of Spike Trains in Reservoir Computing Systems with Spiking Neurons and Analog Neurons" (2016). *Dissertations and Theses*. Paper 2724.

10.15760/etd.2720

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

Information Representation and Computation of Spike Trains in Reservoir
Computing Systems with Spiking Neurons and Analog Neurons

by

Amin Almassian

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

Thesis Committee:
Christof Teuscher, Chair
Melanie Mitchell
Bart Massey

Portland State University
2016

Abstract

Real-time processing of space-and-time-variant signals is imperative for perception and real-world problem-solving. In the brain, spatio-temporal stimuli are converted into spike trains by sensory neurons and projected to the neurons in subcortical and cortical layers for further processing.

Reservoir Computing (RC) is a neural computation paradigm that is inspired by cortical *Neural Networks* (NN). It is promising for real-time, on-line computation of spatio-temporal signals. An RC system incorporates a *Recurrent Neural Network* (RNN) called reservoir, the state of which is changed by a trajectory of perturbations caused by a spatio-temporal input sequence. A trained, non-recurrent, linear readout-layer interprets the dynamics of the reservoir over time. *Echo-State Network* (ESN) [1] and *Liquid-State Machine* (LSM) [2] are two popular and canonical types of RC system. The former uses non-spiking analog sigmoidal neurons – and, more recently, *Leaky Integrator* (LI) neurons – and a normalized random connectivity matrix in the reservoir. Whereas, the reservoir in the latter is composed of *Leaky Integrate-and-Fire* (LIF) neurons, distributed in a 3-D space, which are connected with dynamic synapses through a probability function.

The major difference between analog neurons and spiking neurons is in their neuron model dynamics and their inter-neuron communication mechanism. However, RC systems share a mysterious common property: they exhibit the best performance when reservoir dynamics undergo a criticality [1–6] – governed by the reservoirs’ connectivity parameters, $|\lambda_{max}| \approx 1$ in ESN, $\lambda \approx 2$ and w in LSM – which is referred to as the edge of chaos in [3–5]. In this study, we are interested in exploring the possible reasons for this commonality, despite the differences imposed by different neuron types in the reservoir dynamics.

We address this concern from the perspective of the information representation in both spiking and non-spiking reservoirs. We measure the *Mutual Information* (MI) between the state of the reservoir and a spatio-temporal spike-trains input, as well as that, between the reservoir and a linearly inseparable function of the input, temporal parity. In addition, we derive *Mean Cumulative Mutual Information* (MCMI) quantity from MI to measure the amount of stable memory in the reservoir and its correlation with the temporal parity task performance. We complement our investigation by conducting isolated spoken-digit recognition and spoken-digit sequence-recognition tasks. We hypothesize that a performance analysis of these two tasks will agree with our MI and MCMI results with regard to the impact of stable memory in task performance.

It turns out that, in all reservoir types and in all the tasks conducted, reservoir performance peaks when the amount of stable memory in the reservoir is maximized. Likewise, in the chaotic regime (when the network connectivity parameter is greater than a critical value), the absence of stable memory in the reservoir seems to be an evident cause for performance decrease in all conducted tasks. Our results also show that the reservoir with LIF neurons possess a higher stable memory of the input (quantified by input-reservoir MCMI) and outperforms the reservoirs with analog sigmoidal and LI neurons in processing the temporal parity and spoken-digit recognition tasks. From an efficiency stand point, the reservoir with 100 LIF neurons outperforms the reservoir with 500 LI neurons in spoken-digit recognition tasks. The sigmoidal reservoir falls short of solving this task. The optimum input-reservoir MCMI's and output-reservoir MCMI's we obtained for the reservoirs with LIF, LI, and sigmoidal neurons are 4.21, 3.79, 3.71, and 2.92, 2.51, and 2.47 respectively. In our isolated spoken-digits recognition experiments,

the maximum achieved mean-performance by the reservoirs with $N = 500$ LIF, LI, and sigmoidal neurons are 97%, 79% and 2% respectively. The reservoirs with $N = 100$ neurons could solve the task with 80%, 68%, and 0.9% respectively.

Our study sheds light on the impact of the information representation and memory of the reservoir on the performance of RC systems. The results of our experiments reveal the advantage of using LIF neurons in RC systems for computing spike-trains to solve memory demanding, real-world, spatio-temporal problems. Our findings have applications in engineering nano-electronic RC systems that can be used to solve real-world spatio-temporal problems.

Acknowledgements

I would like to sincerely thank my advisor, Dr. Christof Teuscher, for his support, guidance, and patience. This work would not have been possible without his support and supervision. Many thanks to Dr. Patrick Roberts for his great insights in models and simulations. Thanks also to thesis committee members Prof. Melanie Mitchell and Dr. Bart Massey. I learned invaluable lessons from their classes, which I have applied in this work. I also would like to thank Alireza Goudarzi and Alex Lamb for their valuable comments on the text. To my lovely wife Fatemeh, who always inspired me and for her constant encouragement and support during my study. Special recognition goes out to my family, for their patience and encouragement throughout my study abroad. This study was supported in part by the National Science Foundation under grant number 1028378.

Contents

Abstract	i
Acknowledgements	iv
List of Tables	viii
List of Figures	x
1 Introduction	1
1.1 Goals and Motivations	2
1.2 Contributions	3
2 Background and Related Works	7
2.1 Reservoir Computing	7
2.1.1 Prior Attempts	7
2.1.2 Reservoir Computing Systems	8
2.2 Echo State Network and the Critical Connectivity	11
2.3 Liquid State Machine and the Critical Connectivity	11
2.4 Relation of the Work with the Edge of Chaos	12
2.5 Neurons	14
2.6 Synapses	17
3 Model	18
3.1 The Reservoir with Non-Spiking Analog Neurons	18
3.1.1 sigmoidal Neuron	19

3.1.2	Leaky Integrator Neuron	19
3.2	The Reservoir with Spiking Neurons	20
3.2.1	The Leaky Integrate-and-Fire Neuron Model	21
3.2.2	The Synaptic Connection	22
4	Methodology	23
4.1	Measuring the Mutual Information between the Reservoir and Spatio- Temporal Inputs	25
4.1.1	Measuring MI between the Reservoir and Input	26
4.1.2	Measuring the MI between the Reservoir and PAR_n of the Input	32
4.2	Measuring Memory and the Stability of the Reservoir	34
4.3	Benchmarking the Reservoirs' Performance in Recognizing the Iso- lated Spoken-Digits	40
4.4	Testing the Impact of Memory and Stability of the Reservoirs in Recognizing Spoken-Digit Sequences	42
5	Results	44
5.1	Mutual Information Results for Analog Non-Spiking and Spiking Reservoirs	45
5.1.1	MI Results for the Reservoir with Sigmoidal Neurons	45
5.1.2	MI Results for the Reservoir with Leaky Integrator Neurons	60
5.1.3	MI Results for the Reservoir with Leaky Integrate-and-Fire Neurons	75
5.1.4	Discussion	89

5.2	Recognizing the Isolated Spoken-Digits with Analog Non-Spiking and Spiking Reservoirs	91
5.2.1	Discussion	96
5.3	Recognizing the Spoken-Digits Sequences with Analog Non-Spiking and Spiking Reservoirs	97
5.3.1	Discussion	101
6	Conclusion	103
	References	108

List of Tables

5.1	Comparison of the reservoir types in terms of the maximum obtained MCMi for the input and PAR_3 of the input. Max. I_μ input and Max. $I_\mu PAR_3$ refer to the maximum achieved $I_\mu(u; g(R))$ and $I_\mu(PAR_3(u); g(R))$ respectively. The reservoir with LIF neurons (shown in bold face) has the highest input and PAR_3 MCMi's (4.21 and 2.92).	90
-----	--	----

5.2 Comparison of the reservoir types with respect to the maximum obtained mean-performance in recognizing isolated-spoken-digit recognition (I. S-D) and a sequence of 7 spoken-digits (Seq. S-D 7) when $N = 100$ and $N = 500$ neurons are used in the reservoir. The standard deviations of the performances are shown in parentheses. The LIF reservoir is shown in boldface. Our reservoir with $N = 500$ LIF neurons could recognize the isolated spoken-digits with 97% mean-performance. Compared to a 95% mean-performance obtained from LSM in [10], it showed a slight improvement. Comparing that to the mean-performance of 79% obtained from the reservoir with LI neurons, it has a considerable performance increase. The proposed LIF reservoir also showed higher performance than the LI reservoir even with $N = 100$ neurons. Performance results for isolated spoken-digits with $N = 100$ is not available in [10], and the sequence-recognition tasks have not been performed in [10]. The sigmoidal reservoir failed to solve the problem. All the numbers are rounded values. 102

List of Figures

- 2.1 A schematic view of a generic RC system. Reservoir is the main component of the RC system and is comprised of recurrently connected computing units such as spiking neurons or non-spiking analog nodes. In this specific example, an input signal is injected into three reservoir nodes to perturb the reservoir. The temporal input causes a trajectory in the reservoir’s network dynamics. The three white nodes compose a readout layer. Each trained readout node receives inputs from all or a subset of the reservoir nodes and interprets the perturbations in the reservoir in a distinct way. To solve a particular task, one or more readout nodes may be required, depending on the task. For instance, the readouts for tasks 1 and 2 have analog outputs. These readout nodes are trained by a linear-regression algorithm over the reservoir node states. The readout for task 3 is a binary classifier. This readout is trained by a linear classification algorithm, yielding two classes of 0 and 1. 10
- 2.2 Neurons and synaptic transmission: an action potential is generated and travels down the axon to the axon terminal, where it provokes a neurotransmitter release that acts on the post-synaptic end. Reproduced with permission from the National Institute on Aging - US National Institutes of Health. 16

4.1	A) Template 0 (in red) and template 1 (in blue) consist of 4 sections s_i of spike patterns. Each section s_i is comprised of 5 randomly generated $200ms$ spike trains: i.e., 5 channels over $200ms$, from a Poisson distribution with the frequency of $20Hz$. B) An input instance, comprised of a random selection of the sections s_i , chosen from template 0 ($s_i = 0$) or from template 1 ($s_i = 1$). In this particular example, $s_1 = 0, s_2 = 1, s_3 = 0, s_4 = 1$	28
4.2	A schematic view of reading out the state of a reservoir of size $N = 8$ at time $720ms$ when measuring the mutual information. A) The readout node is receiving inputs from all nodes in the reservoir. B) A randomly selected subset of reservoir components is selected.	31
4.3	A schematic view of training the readout layer over trials with a reservoir of size $N = 8$ at time $720ms$ and input with $n = 4$ sections to obtain the g function. Later at the MI measurement phase, the mutual information $I(u, g(R))$ is measured between the input u in sections $s_i = 1, 2, 3, 4$ and the state of the reservoir at time $t = 720ms$	32

- 4.4 Three examples of $I(u; g(R))$ taken from ESN with size $N = 100$ and input with 16 sections using the method described in section 4.1. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) When $\lambda_{max} = 0.9$. Notice that at any time t_i , the information about the past 2 sections of the input is present in the state of of the reservoir, denoted by random variable R . B) When $\lambda_{max} = 1$. Notice that at any time t_i , the information about the past 4 sections of the input is present in the reservoir state, denoted by random variable R , which is much more than when $\lambda_{max} = 0.9$ (shown in A). C) When $\lambda_{max} = 1.5$, the reservoir is only able to represent the past sections at time t in the beginning of the input; then the reservoir no longer has any memory of the past input sections. (A) and (B) show the stable reservoir dynamics, as the MI consistently remains at a certain level. (C), however, shows the information dynamics of an unstable (chaotic) reservoir. 35
- 4.5 I_μ of the reservoir with analog sigmoidal neurons (ESN) obtained from $I(u; g(R))$ shown in figure 4.4. Notice that I_μ at $|\lambda_{max}| = 1.5$ is less than I_μ at $|\lambda_{max}| = 0.9$ because we ignore the memory of the reservoir before section 7. The memory about past input sections is decreased after section 7 as $|\lambda_{max}|$ is increased (and $|\lambda_{max}| > 1$) in ESN. 40

5.1	Mutual information of the reservoir state and the input with 16 sections. Reservoir with analog sigmoidal neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.5$, E) $w = 2.0$. By increasing w up to $w = 1.0$, MI is increased. By further increase in w , the MI starts dropping from $t = 1200ms$ on. This means that the system does not represent the input anymore after the critical $w = 1$	46
5.2	Mutual information of the reservoir state and PAR_3 of the input with 16 sections. Reservoir with analog sigmoidal neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and PAR_3 of section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.5$, E) $w = 2.0$. By increasing w up to $w = 1.0$, MI is increased. By further increase in w , the MI starts dropping from $t = 1200ms$ on. The system does not represent the input anymore; therefore PAR_3 cannot be extracted from the state of the reservoir after the critical $w = 1$. . .	52
5.3	A) I_μ of the reservoir with analog sigmoidal neurons, obtained from $I(u; g(R))$ results, as shown in figure 5.1. The reservoir size is $N = 100$, and input has 16 sections. I_μ for MI is increased by increasing the w up to critical $w = 1$. With further increase in w , the I_μ starts to drop. Meaning, the consistency of the system in representing the input and output is decreased.	58

5.4 I_μ of the reservoir with analog sigmoidal neurons, obtained from $I(PAR_3(u); g(R))$ results, as shown in figure 5.2. The reservoir size is $N = 100$, and input has 16 sections. $I_\mu(PAR_3(u); g(R))$ also indicates the performance of the reservoir in solving PAR_3 over time. The rise and fall of $I_\mu(PAR_3(u); g(R))$ follows that of $I_\mu(u; g(R))$. That is to say, the performance of the reservoir in solving PAR_3 is highly dependent on the amount of stable memory in the reservoir. 59

5.5 Mutual information of the reservoir state and the input with 16 sections. Reservoir with analog LI neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.3$, E) $w = 2.0$. By increasing w , up to critical $w = 1.0$, MI is increased. By further increase in w , the MI starts dropping from $t = 1200ms$ on. This means that, the system no longer represents the input after the critical $w = 1$ 61

5.6 Mutual information of the reservoir state and PAR_3 of the input with 16 sections. Reservoir with analog LI neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and PAR_3 of section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.3$, E) $w = 2.0$. By increasing w up to critical $w = 1.0$, MI is increased. With further increase in w , the MI starts to drop from $t = 1200ms$ on. The system does not represent the input anymore; therefore PAR_3 cannot be extracted from the state of the reservoir after section 7 when w is greater than the critical $w = 1$ 67

- 5.7 A) I_μ of the reservoir with LI neurons, obtained from $I(u; g(R))$ results, shown in figure 5.1. The reservoir size is $N = 100$ and the input has 16 sections. I_μ for MI is increased by increasing w up to the critical $w = 1$. With further increase in w , the I_μ starts to drop. This means that, the consistency of the system in representing the input and output is decreased. The leak rate of the LI neuron is fixed at $a = 0.25$ 73
- 5.8 I_μ of the reservoir with LI neurons, obtained from $I(PAR_3(u); g(R))$ results, shown in figure 5.6. The reservoir size is $N = 100$ and the input has 16 sections. $I(PAR_3(u); g(R))$ also indicates the performance of the reservoir in solving PAR_3 over time. The rise and fall of $I(PAR_3(u); g(R))$ follows that of $I(u; g(R))$. That is to say, the performance of the reservoir in solving PAR_3 is highly dependent on the amount of stable memory in the reservoir. 74
- 5.9 Mutual information of the spiking reservoir state and the input with 16 sections. Reservoir with LIF neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) $w = 0$, B) $w = 1.5$, C) $w = 3$, D) $w = 5$, E) $w = 7$. By increasing w , up to critical $w = 3$, MI is increased and by further increase in w , the MI starts dropping from $t = 1200ms$ on. Meaning, the system does not represent the input anymore. 76

5.10 Mutual information of the spiking reservoir state and PAR_3 of the input with 16 sections. Reservoir with analog LIF neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and PAR_3 of section s_i over time. A) $w = 0$, B) $w = 1.5$, C) $w = 3$, D) $w = 5$, E) $w = 7$. By increasing w up to critical $w = 3$, MI is increased. With further increase in w , the MI starts to drop from $t = 1200ms$ on. The system does not represent the input anymore; therefore PAR_3 cannot be extracted from the state of the reservoir after section 7. 82

5.11 $I_\mu(u; g(R))$ of the reservoir with LIF neurons with two different time constants $\tau_m = 7$, and $\tau_m = 10$. Only the I_μ for $\tau_m = 10$ shown here by the green line with square marks corresponds to the $I(PAR_3(u); g(R))$ results shown in figure 5.9. Reservoir size is $N = 100$ and input has 16 sections. I_μ for MI is increased by increasing w up to critical $w = 3$ for $\tau = 10$ and critical $w = 9$ for $\tau = 7$. With further increase in w , the I_μ starts dropping. This means that, the consistency of the system in representing the input and output is decreased. 88

5.12	<p>$I_\mu(PAR_3(u); g(R))$ of the reservoir with LIF neurons and two different time constants $\tau_m = 7$, and $\tau_m = 10$. Only the I_μ for $\tau_m = 10$ shown here by the green line with square marks corresponds to the $I(PAR_3(u); g(R))$ results shown in figure 5.9. Reservoir size is $N = 100$ and input has 16 sections. $I(PAR_3(u); g(R))$ also indicates the performance of the reservoir in solving PAR_3 over time. $I(PAR_3(u); g(R))$ rise and fall follows those of $I(u; g(R))$. That is to say, the performance of the reservoir in solving PAR_3 is highly dependent on the amount of stable memory in the reservoir.</p>	89
5.13	<p>Performance of the reservoir with sigmoidal neurons ($N = 100, 200, \dots, 500$) in recognizing the isolated spoken-digit. The y axis represents the mean value of the performance over 30 trials for each experiment and the error bars show the standard deviation of the performance. We have run the experiment with the same w range shown on the x axis, covering the same range of w in the MI results.</p>	92
5.14	<p>Performance of the reservoir with LI neurons ($N = 100, 200, \dots, 500$) in recognizing one spoken-digit. The LI neurons' leak parameter is fixed to $a = 0.25$. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. We have run the experiment with same w range shown on the x axis, covering the same range of w in the MI results.</p>	93

5.15 The influence of the LI neuron's leak (a) parameter in the performance of the reservoir with LI neurons ($N = 500$) in recognizing one spoken-digit. We consider τ_m , $a = \frac{1}{\tau_m}$ as decay parameter and measure the performance against w when τ_m varies between 4 and 40. By increasing τ_m , the memory of the LI neuron is increased. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. We have run the experiment with the higher w range shown on the x axis (as compared to the range of w in the MI results) to ensure that the performance is not increased with further increase of w . Notice that the performance does not reach that of LIF reservoir shown in figure 5.16. It is worth recalling that in our MI result shown in figures 5.7 and 5.8, $\tau_m = 4$ ($a = 0.05$). 94

5.16	Performance of the reservoir with LIF neurons ($N = 100, 200, \dots, 500$) in recognizing one spoken-digit. The LIF membrane time-constant parameter is fixed to $\tau_m = 20ms$. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. We have run the experiment with $w = [0.0, 7.0]$ (shown on the x axis) so that the complete spectrum of MI for $\tau_{syn} = 10ms$ (shown in figures 5.11 and 5.12) is covered. The relevance of the MCMI results (shown in figures 5.9 and 5.10) in the performance of the reservoir at another spatio-temporal task is evident: the performance is maximized when the amount of stable memory is maximized. The LIF reservoir outperforms the analog non-spiking reservoirs in this spatio-temporal tasks.	95
5.17	Performance of the reservoir with sigmoidal neurons ($N = 100, 200, \dots, 500$) in recognizing the sequence of $n = 1, 2, \dots, 7$ spoken-digits. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance.	98

5.18 Performance of the reservoir with LI neurons ($N = 100, 200, \dots, 500$) in recognizing the sequence of $n = 1, 2, \dots, 7$ spoken-digits. To achieve the highest possible performance, the leak parameter and weight coefficient w of the LI neurons are fixed at $a = 0.05$ ($\tau_m = 20$) and $w = 2$. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. The mean-performance when $N = 100$ and $n = 1$ did not reach the maximum mean-performance shown in figure 5.15 of $N = 100$ because $w = 2$ is an optimum value for $N = 500$ but not $N = 100$ for $n = 1$ 99

5.19 Performance of the reservoir with LIF neurons ($N = 100, 200, \dots, 500$) in recognizing the sequence of $n = 1, 2, \dots, 7$ spoken-digits. The time constant parameter of the LIF neurons is fixed at $\tau_m = 20ms$. The y axis represents the mean value of the performance over 30 trials for each experiment and the error bars show the standard deviation of the performance. The maximum performance of the reservoir is achieved at the critical $w \approx 3$. The LIF reservoir outperforms the analog non-spiking reservoirs in a rather memory demanding spatio-temporal tasks. Recall that the LIF reservoir has the highest amount of stable memory of all the reservoirs in our experiments. . 100

Introduction

The objective of this thesis is to analyze and compare the impact of information representation and memory of the reservoir on critical dynamics – referred to as the edge-of-chaos [4,5] – in reservoirs with spiking neurons and none-spiking analog neurons. We also investigate the contribution of neuron types in this phenomenon. In all the reservoir types introduced, the optimum computational performance has been observed near to the edge-of-chaos, i.e., when the reservoir dynamics are under a critical regime [1–6]. Our results show that the amount of stable memory is maximized at critical dynamics and that it brings about the best performance in the vicinity of the edge of chaos.

We consider sigmoidal and LI neurons for the analog non-spiking reservoir connected by the ESN connectivity model. As for the spiking reservoir, we construct a simple reservoir with LIF neurons based on the ESN connectivity model. LIF neurons have conventionally been used in LSM [2]. One advantage of taking this approach is that the network connectivity differences between the ESN and LSM are eliminated so that it is possible to focus on the differences between the neurons. The other advantage of this LIF reservoir model over LSM is its simplification of the LIF reservoir construction. The approach may also be leveraged to simplify the construction of the nano-wired hardware RC systems with LIF neurons. Our results are useful for applications that focus on computing real-world spatio-temporal inputs like speech, vision.

1.1 Goals and Motivations

We pursued the following goals in this thesis.

- Representation and processing of spatio-temporal stimuli are crucial to living systems for perceiving and responding to the stimuli. Vision, auditory, and somatosensory systems exemplify the systems that receive and process such stimuli. The stimuli received by the neuro-receptors are transduced to trains of action potentials (a.k.a., spikes), which are carried along other neurons toward specific areas of the brain for perception and further processing [7,8].
- The amount of memory required to process such spatio-temporal stimuli – namely the input to the processing system – depends on the task and the expected output from the input. The memory of an RC system appears not to be enough for on-line processing of real-world problems, such as longtime vision or auditory tasks. Here we consider whether using spiking neurons in the reservoir can increase the overall memory of the RC system thereby solving real-world, long-time, spatio-temporal tasks consistently over time.
- Bertschinger *et al.* [4] and Natchaläger *et al.* [5] have shown that the computation power of the LSM is maximum in a critical regime just before it becomes chaotic (at the edge of chaos) and the reservoir then performs poorly in solving tasks when its dynamics becomes chaotic. We investigate this phenomenon from an information-representation perspective. We would like to know what changes the spiking and non-spiking RC systems undergo at the so-called edge of chaos in terms of information representation in the reservoir.
- It is not known how RC systems with spiking neurons represent and process spatio-temporal spike trains compared to analog non-spiking RC systems.

The major difference between analog non-spiking neurons and spiking neurons is in their neuron-model dynamics and their inter-neuron communication mechanism. Spiking neurons mainly transmit their states to their efferent neurons when they spike. Whereas, analog non-spiking neurons consistently communicate their states to their efferent neurons over time.

1.2 Contributions

The following list itemizes the contributions made by this thesis.

1. We show that the stable memory of the reservoir is maximized at the critical network dynamics (the edge of chaos) and driven by network connectivity weight coefficient w in our experiments. We also show that, by increasing the w , the reservoir cannot represent the input information consistently over time such that the amount of stable memory in the system drops dramatically. At this point the linear readout layer lacks the input information required to compute the temporal task.
2. With respect to memory and representing the information of the inputs, we compare analog non-spiking reservoirs built of sigmoidal and LI neurons, with spiking reservoirs built of LIF neurons by measuring the MI between the reservoir dynamics and the input. We accomplished the latter by measuring the MI between the reservoir dynamics and the input.
3. We expand the MI measurements, made in [9], from 800 *ms* to 3200 *ms* to unveil the implication of the edge-of-chaos phenomenon in information representation that have emerged from the reservoir dynamics.

4. We measure and compare the MI between the reservoirs' dynamics and the 3 digits temporal parity of the input to investigate the correlation with the MI obtained between the reservoir's dynamics and the input and to compare the reservoirs' ability to compute a linearly unsolvable task.
5. We propose and compute the *Mean Cumulative Mutual Information* (MCMI) of the above reservoirs to quantitatively show the memory and consistency of the reservoir in representing and computing the input over time.
6. We show how the input is consistently represented in the reservoir when the reservoir connectivity-weight-coefficient, w , is smaller than its critical value. When w is close to the critical value, the amount of stable memory, measured by MCMI, is maximized. At the same time, the performance of the reservoir peaks with respect to solving both temporal parity and isolated spoken-digit recognition tasks. Upon increasing w farther, the MCMI drops dramatically and so does task performance. In this situation, the reservoir dynamics become too random and lose their relevance to the given input, i.e., become chaotic [4, 5].
7. We compare the sigmoidal and LI reservoirs with the LIF reservoir with respect to the isolated spoken-digit recognition task discussed in [10]. It turns out that the LIF reservoir outperforms the other two reservoir types in this task. Our results showed that the LIF reservoir can perform the task with 97% mean-performance (with 100% performance in some trials). The reservoir with the optimized LI neurons could perform the same task with up to 79% mean-performance (with a maximum of 88% performance in some trials). The reservoir with sigmoidal neuron could not solve this task at all

(it yielded a 1% mean-performance).

8. We show that the reservoir with LIF neurons outperforms the non-spiking reservoir in recognizing a *sequence* of spoken-digits that demands up to 7 times more memory compared to the isolated spoken-digits recognition task. For sequence length $n = 7$, the reservoir with LIF neurons could perform this task with 68% mean-performance (with a maximum of 80% performance in some trials). The reservoir with LI neurons could perform this task with 33% mean-performance (with a maximum of 43% performance in some trials). The reservoir with sigmoidal neuron could not solve this task at all (it yielded a 1% mean-performance).
9. On the one hand, constructing a stable spiking reservoir has been a challenge. LSM has many complexities, such as dynamic synapses, that are hard to construct in both software and hardware. On the other hand, one is very likely to end up with an unstable and poorly functional reservoir who builds an LIF reservoir without considering LSM complexities or without making careful parameter adjustments to the reservoir connectivity, neurons, and synapses. We here argue that unstable reservoirs are severely lacking in stable memory: i.e. they lose their memory consistency over time and are not able to consistently compute the temporal tasks. We show that it is possible to achieve an LIF reservoir with stable dynamics and high performance by using an ESN connectivity matrix and adjusting the neurons and synaptic parameters.

The article is organized as follows. In chapter 2 we review the background and history of the RC systems and explain why we have considered them for

processing of spatio-temporal inputs. We also provide a brief overview of biological aspects that underlie spiking-neuron models and synapses. In chapter 3, we provide more detail on the formalisms for the neuron models and RC systems used in this study. Next, we walk through our methodology and measurements in chapter 4. Chapter 5 presents our results, discussions, and comparisons of MI measurements, and a performance analysis of isolated spoken-digits, and spoken-digits sequence-recognition tasks. We finally conclude our work in chapter 6.

Background and Related Works

Our brain is able to deal with the temporal tasks in real-time and on-line. In other words, the throughput of the brain in responding to the stimuli is so high that we cannot realize any delay between receiving and responding to temporal stimuli such as speech, vision, etc. To come up with a real-time and on-line system to solve temporal, real-word problems is one of the biggest challenges in the realm of Artificial Intelligence and Computational Neuroscience [1, 2, 11–17]. Here we present a brief history of attempts that have been made to achieve such a systems and provide an overview of the RC systems. We follow by considering the specifics of the ESN and LSM. Then, explain the biological aspects of the reservoir elements such as neurons and synapses.

2.1 Reservoir Computing

2.1.1 Prior Attempts

Some attempts have been made by using the feed-forward structures for solving temporal tasks. For example, *Time-Delayed Neural Network* (TD-FFNN) and *Nonlinear Autoregressive with eXternal input* (NARX) [18]. The main problem of these approaches is that the memory depth is fixed and they may not be suitable for solving many temporal problems, whereas in *Recurrent Neural Networks* (RNN) the memory depth is variable. Hopfield [13] took advantage of RNN by constructing network topologies with symmetric weights. In RNNs, connections between neurons form directed cycles, which allow the network internal state to

exhibit temporal dynamics. The disadvantage of the Hopfield network is the presence of point attractors. This is a disadvantage because it takes time for the system to reach point attractors and a system with this property cannot solve a task in real time. Werbos [19] used a back-propagation-through-time learning rule for a sparsely connected RNN. In his approach, all the weights were trained. Bengio *et al.* argued that training an RNN [20] for long-term dependencies is difficult.

RNNs with sigmoid activation function $\sigma(x) = [2/(1 + e^{-x})] - 1$ (and similar activation functions) have been shown to be Turing equivalent [21]. They are universal approximators [22] and can approximate an arbitrary finite-state automata [23]. Nevertheless, slow convergence and high computational training costs in the mentioned RNNs make them impractical for solving the real-world temporal tasks [11] that requires real-time and on-line computations.

2.1.2 Reservoir Computing Systems

Buonomano [17] proposed a recurrent network of spiking neurons (Integrate-and-Fire neurons) inspired by neocortical connectivity with short-term plasticity (paired-pulse facilitation and slow-inhibitory, post-synaptic potentials). He showed that this model can transform temporal information into spatial information. He trained a separate output layer by using a supervised correlation-based learning rule for solving temporal tasks. Steil [24] put forward a model of RNN called Back-Propagation Decorrelation (BPDC). In this model, the internal weights in the network are globally scaled up or down slightly and only the output weights are trained. The output nodes are nonlinear and the training is done in an on-line manner. Jaeger [1] introduced the Echo State Network (ESN), which consists of a random recurrent network of analog nodes with sigmoid activation function. The

RNN remains untrained, and its state is projected to a separate readout layer. The readout layer uses a simple linear regression and is trained to approximate the target output. Independent of this work, Maass *et al.* [2] introduced another model that was inspired by network models in neocortical columns, and coined the term *Liquid State machine* (LSM). In this model, a spatially distributed and locally connected RNN of LIF neurons is created in a 3-D space. In their model, the LIF neurons are connected by dynamic synapses, through a connectivity probability function. Like ESN, LSM also uses a simple linear readout layer to interpret the state of the RNN. Steil [24] showed that, in BPDC, global weight scaling can be ignored if the weights are initially well scaled and BPDC has the same weight dynamics as proposed by Jaeger and Maass, the major difference being that the output nodes are nonlinear and the training is done in an on-line manner.

Schrauwen *et al.* [11] in their review paper refer to Verstraeten *et al.* [25] as the proposers of the term *Reservoir Computing* (RC). This term was proposed to unify the last three ideas mentioned into a common research stream [11] and it has been used extensively in the literature. Figure 2.1 depicts a schematic view of a particular RC system.

RC systems are also comparable with kernel methods. In these methods, a function that transform an input into a higher-dimensional vector is referred to as kernel. Examples of methods that use kernel functions include Support Vector Machines, FFNN, Radial-Basis Function approximators, Slow Feature Analysis, and various Probability Mixture models [26]. RC systems transform the input into a higher-dimensional feature space as kernel methods do. Nonetheless, the kernel methods are unable to deal with temporal tasks on their own. In RNNs, such as reservoirs in RC systems, the history of the input is also transformed into a

higher-dimensional feature space automatically.

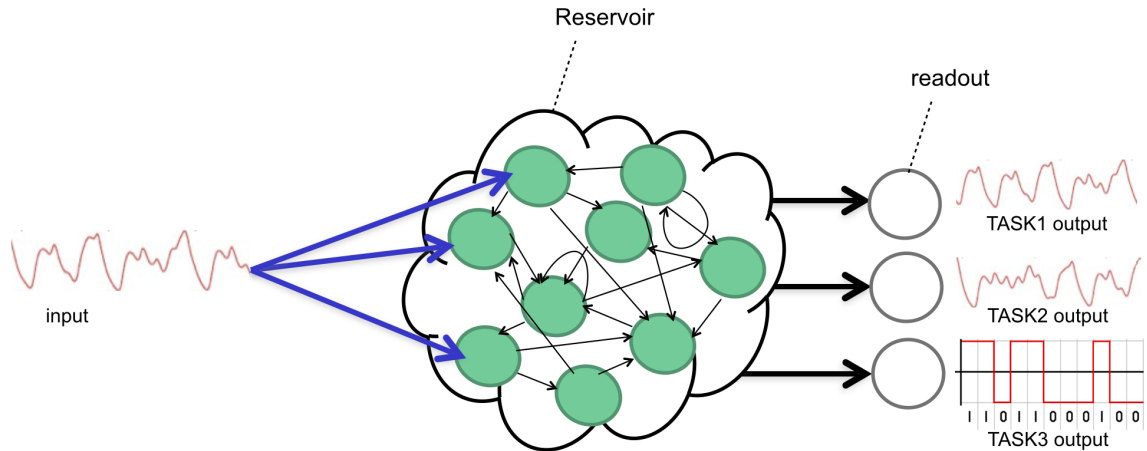


Figure 2.1: A schematic view of a generic RC system. Reservoir is the main component of the RC system and is comprised of recurrently connected computing units such as spiking neurons or non-spiking analog nodes. In this specific example, an input signal is injected into three reservoir nodes to perturb the reservoir. The temporal input causes a trajectory in the reservoir’s network dynamics. The three white nodes compose a readout layer. Each trained readout node receives inputs from all or a subset of the reservoir nodes and interprets the perturbations in the reservoir in a distinct way. To solve a particular task, one or more readout nodes may be required, depending on the task. For instance, the readouts for tasks 1 and 2 have analog outputs. These readout nodes are trained by a linear-regression algorithm over the reservoir node states. The readout for task 3 is a binary classifier. This readout is trained by a linear classification algorithm, yielding two classes of 0 and 1.

RC systems have been targeted by scientists from different disciplines and backgrounds; therefore a spectrum of RC systems of different types and flavors has been proposed in the past decade. Other computational units have been used in the reservoir in different studies: e.g., threshold gates [27], memristors [28], and random automata [29].

2.2 Echo State Network and the Critical Connectivity

Echo State Network (ESN) is a well-known RC system that was introduced by Jaeger [1]. The reservoir in ESN is composed of sigmoidal neurons that are connected through a random weight matrix. The reservoir connectivity is defined by a random weight matrix that is normalized by the spectral-radius of the matrix, by which the network’s connectivity weights are controlled. The readout layer is a standard linear regression. ESN was originally introduced with standard sigmoidal neurons in the reservoir; later LI neurons were considered for the reservoir in ESN. Compared to the sigmoidal neurons (described in section 2.5), LI neurons showed better performance at continuous-time tasks such as Mackey-Glass system prediction [1]. LI neurons have continuous-time dynamics and more memory than the sigmoidal neurons due to the leak parameter that is incorporated into them. Echo-state property is an essential property in the reservoir of the ESN for solving temporal tasks and it has a direct relation to the performance of the ESN in solving the tasks [1]. Interestingly, the echo-state property is guaranteed to exist when spectral-radius, $|\lambda_{max}|$, of the reservoir’s network connectivity matrix is smaller than 1 and it is maximized when the spectral-radius is close to 1. The echo-state is not guaranteed to hold when the reservoir Spectral Radius is higher than 1 [1]. In all the tasks tested in [1], the optimum performance of the reservoir is achieved when $|\lambda_{max}| \approx 1$.

2.3 Liquid State Machine and the Critical Connectivity

Liquid State Machine (LSM) is another well-known RC system that was introduced independently by Maass *et al.* [2]. The reservoir in LSM is composed of a recurrent neural network of LIF neurons that are spatially distributed over a

three dimensional x, y, z space. The spatial distribution of the neurons is used to provide a control over the connectivity in the network and the properties of synaptic connections. The distance between any node along each spatial axis is set to unity. The probability of the connectivity between two neurons is defined according to the Euclidean distance between them. The probability that neuron i forms a synaptic connection to neuron b is defined as $P_{conn}(a, b) = Ce^{-D(a,b)/\lambda}$, where D is the Euclidean distance between neuron a and b , and where λ controls the average number of connections as well as the average distance between connected neurons. C is the probability of a synaptic connection. The probabilities that a synaptic connection will be excitatory-excitatory (EE), excitatory-inhibitory (EI), inhibitory-excitatory (IE), or inhibitory-inhibitory (II) are 0.3, 0.2, 0.4, 0.1 respectively [2]. Synaptic connection from a to b , is modeled according to a dynamic synapse model proposed by Markam *et al.* [30]. Reservoir's separation and fading memory properties have a direct relationship with the computational power of the LSM. It has been shown that the reservoir holds the optimum separation and fading-Memory properties, when $\lambda = 2$. In addition, the computation power of the LSM in classification of a spoken-words tested in [2] is maximized when $\lambda = 2$. By increasing λ , the performance of the reservoir is decreased and the network dynamics become chaotic [2]. Later, Bertschinger *et al.* [4], Natschläger *et al.* [5] looked at this phenomena from the perspective of complexity theory and referred to the critical value $\lambda = 2$ as the edge of chaos.

2.4 Relation of the Work with the Edge of Chaos

An input driven neural network is a dynamic system [3]. Derrida *et al.* [31] defined the chaotic and ordered phases of an input driven network as follows. For

autonomous systems, consider two (initial) network states with a certain (normalized) Hamming distance. These states are mapped to their corresponding successor states (using the same weight matrix) with the same input in each case and the change in the Hamming distance is observed. If small distances tend to grow this is a sign of chaos; if the distance tends to decrease this is a signature of order. Bertschinger *et al.* [4] and Natschläger *et al.* [5] followed the same approach to defining order and chaos dynamics. A neural network can be seen as a dynamical system, and the dynamics of the system can be changed from ordered to chaotic by changing some global parameters of the system: e.g., connectivity structure or strength of the connectivities.

Related dynamical systems have been studied extensively in various contexts. For instance, Derrida [32] studied Ising-spin models (networks of threshold elements), Langton [33], Packard [34] and Michel *et al.* [35] considered Cellular Automata (CA), Kauffman [36] studied random Boolean networks. Langton [33] claimed that CA are capable of performing non trivial computation most likely when their dynamics are in the vicinity of "phase transitions" between order and chaos. He termed this phase transition the "edge of chaos". Bertschinger *et al.* [4] showed that networks of threshold elements with online inputs expose the optimum computational performance at the edge of chaos: i.e., at the transition from ordered to chaotic dynamics. Mitchell *et al.* [35] argued that the region of best computational performance should depend on the task at hand. Therefore, the best computational power does not necessarily correspond to the edge of chaos.

In this study we do not provide rigorous measures for order and chaotic dynamics. Hence, we have neither enough evidence to address the debate about whether or not the optimum computation performance happens exactly at the edge of chaos

nor enough to consider the consequences of the input and task variations on this proposition. In most of the works cited above, the optimum computational performance of the dynamic systems under consideration is reported to be near the edge of chaos. We leave exhaustive study of the exact correlation between the information representation and the dynamics phases of the system for future works. In this thesis, one may think of chaotic dynamics as reservoir dynamics in which the desired input information is not represented in the reservoir due to the very high entropy of the reservoir state.

2.5 Neurons

Neurons are electrically excitable cells which can be found in many simple and complex cellular organisms in nature. Neurons are known to be responsible for information processing and transmission. Many models have been inspired by pyramidal neurons, which are mostly found in areas of the mammalian brain. Anatomically, pyramidal neurons are comprised of a triangular soma (cell body), a single axon, and a large apical dendrite. Dendrites receive inputs from other neurons and propagate them to the soma. The axon then carries the neuronal output to other cells by means of an action potential, (a.k.a., spike). An *action potential* is a short-lasting rise and fall in the electrical membrane potential of a cell which propagates along the membrane. Membrane potential is defined as the difference in electrical potential between the interior of a neuron and the surrounding extracellular fluid. Under resting conditions, the potential inside the cell membrane of a neuron is about -70 mV relative to that of the surrounding fluid. Typically, the membrane potential rapidly rises to a peak potential of +40 mV when a spike happens. Figure 2.2 shows a cartoon model of two connected neurons. Neurons have

enormous morphological and functional variations, and their properties have been studied from different perspectives. Although we now have a fair amount of understanding about the dynamics and behaviors of the neurons, researchers generally are managed to focus on specific aspects of them judicious simplification of the neurons. A broad range of models of neurons including detailed bio-physiological, compartmental, kinetics, and simplified mathematical models have been proposed in literature e.g. [37–39].

Simple artificial neuron models are mathematical models that are designed only to simulate the integration of inputs and firing spikes in neuron’s axons. They are still extensively used in studies that are more focused on the general properties of neural networks. The first artificial neuron was introduced by McCulloch *et al.* [40]. It basically produces an output of 1 if the integration of weighted inputs passes a threshold and of -1 otherwise as shown in equation 2.1.

$$y = \varphi\left(\sum_{i=1}^n w_i x_i\right), \quad (2.1)$$

where y is the output of the neuron and w_i is the weight of input x_i . The variable n denotes the number of inputs to the neuron. φ here is a signum function called the transfer (activation) function of the neuron. In this article, we consider the same model but with a continuous sigmoid activation function as defined in equation 2.2. We refer to this model as sigmoidal neuron.

$$y = \tanh\left(\sum_{i=1}^n w_i x_i\right) \quad (2.2)$$

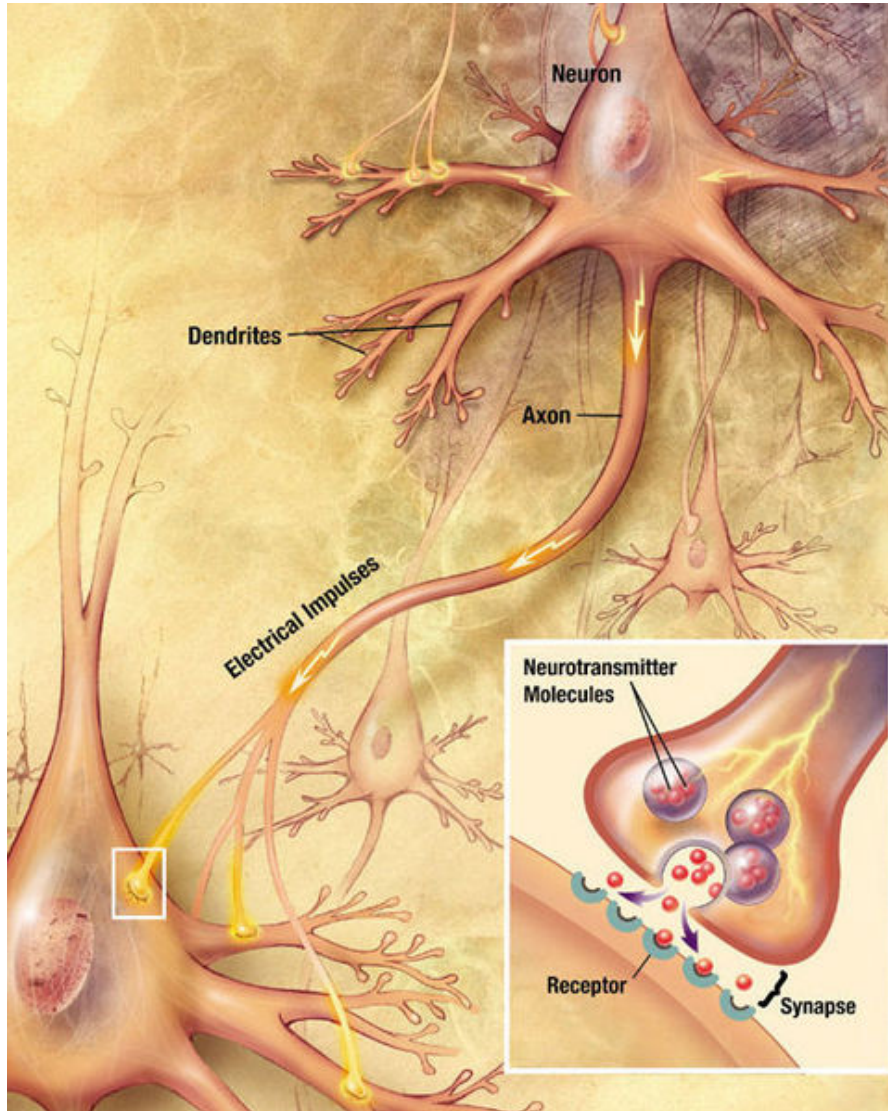


Figure 2.2: Neurons and synaptic transmission: an action potential is generated and travels down the axon to the axon terminal, where it provokes a neurotransmitter release that acts on the post-synaptic end. Reproduced with permission from the National Institute on Aging - US National Institutes of Health.

2.6 Synapses

Synapses are the main point of communication between neurons. They allow neurons to pass an electrical or chemical signal to another cell. Chemical synapses are known to be the main mechanisms of neural communication [41]. Figure 2.2 shows a schematic view of the synaptic transmission process. The voltage transient of an action potential in the pre-synaptic neuron propagates along the membrane of the cell and reaches the axonal terminals. Then neurotransmitter molecules are released into a small space between the axonal terminal of the pre-synaptic neuron and the cell body (mostly dendrites) of the post-synaptic neuron, which is called synaptic cleft. These molecules then bind to receptors on the post-synaptic cell's side of the synaptic cleft. Finally, the neurotransmitters are cleared from the synapse through several potential mechanisms to terminate the transmitter action. This process creates an electrical response that is generated in the post-synaptic cell, which may excite (increase the membrane potential) or inhibit (avoid membrane potential to increase toward the threshold) the post-synaptic neuron. We have used a simplified mathematical model of this mechanisms in our model, which is presented in chapter 3.

3

Model

In this chapter we describe our RC system model including the reservoir and the neuron types that we have considered in this study. We base our reservoir connectivity for sigmoidal, LI, and LIF neurons on ESN. LIF neurons however are conventionally used in LSM reservoir. Nevertheless, we used ESN weight matrix for connecting the LIF neurons in our spiking reservoir. This involves readjustment of the neuron and synaptic parameters to obtain stability in the overall reservoir dynamics.

3.1 The Reservoir with Non-Spiking Analog Neurons

We use an ESN model to build our non-spiking reservoirs. The reservoir is composed of a neural network of $N \in \mathbb{N}$ neurons. They receive input from $K \in \mathbb{N}$ input channels, the state of which is defined as $U(t) = u_0(t), u_1(t), \dots, u_N(t), u_i \in \mathbb{R}$. The state of the reservoir at time t is defined as $X(t) = x_0(t), x_1(t), \dots, x_N(t), x_i \in \mathbb{R}$. x_i is the state of each neuron at time t . The state update formalism for each type of neuron is provided later in this chapter. The state of the reservoir ($X(t)$) is projected into L output units. L is assumed to be 1, since we have only one readout in this study. Real-valued connection weights are collected in an $N \times K$ weight matrix W_{in} for the input weights. An $N \times N$ weight matrix W is used for the reservoir internal connections, and an $N \times L$ matrix W_{out} is used for the connections to the output units in the readout layer. We consider a connection weight coefficient w and multiply the W matrix by a variable w . No feed-back

from the readout layer is considered in our RC system.

We consider sigmoidal neurons and Leaky-Integrator (LI) neurons for the reservoirs with analog non-spiking neurons.

3.1.1 sigmoidal Neuron

Sigmoidal neuron is derived from the artificial neuron model introduced by McCulloch *et al.* [40]. It has a *tanh* activation function. The update state of sigmoidal neuron i receiving connection from neurons j in discrete time is as follows.

$$x_i(n+1) = \tanh(u(n+1) + \sum_{j \in \mathbb{N}} w_j x_j(n)), \quad (3.1)$$

where x_i is the state of the neuron i , w_j is the connection weight between neuron j and i , and u is the input to the neuron i .

ESN with sigmoidal neuron is a discrete-time system. The vectorized state update for the reservoir with sigmoidal neurons at time-step n is defined as

$$X(n+1) = \tanh(W_{in}U(n+1) + WX(n)), \quad (3.2)$$

where X is the state of the reservoir, W_{in} is the input weight matrix, U is the input vector, and W is the reservoir connectivity weight matrix.

3.1.2 Leaky Integrator Neuron

The standard sigmoidal neurons are better suited for modeling discrete-time systems and they are not adequate for the tasks with continuous-time dynamics as they cannot capture slow dynamics in continuous tasks [1]. LI neurons are better fits for continuous-time tasks and RC systems with LI neurons have more memory

than those with sigmoidal neurons [1]. The state update of the LI neurons is as follows.

$$\dot{X} = C(-aX + \tanh(W_{in}U + WX)), \quad (3.3)$$

where C is a time constant and a is the leak rate. Other variables are the same as those of the sigmoidal neuron, mentioned above.

3.2 The Reservoir with Spiking Neurons

LSM [2] is a well-known spiking reservoir. The reservoir in LSM is comprised of LIF neurons. Not only does LSM differ from ESN in terms of neuron type, it also differs in terms of reservoir-network connectivity and in having dynamic synapses. The reservoir in LSM is composed of LIF neurons that are distributed in a 3-D space and are connected with dynamic synapses through a distribution probability function. Hence, direct comparison of ESN and LSM appears not to be a fair neuron-wise comparison. Moreover, constructing a stable LIF reservoir per-se is a challenge. One is very likely to end up with an unstable and useless reservoir who builds an LIF reservoir without considering LSM complexities or without making careful parameter adjustments in the reservoir connectivity, neurons, and synapses. We here show that unstable reservoirs severely lack stable memory and perform poorly over time in chapter in 5. ESN has proven successful to yield a stable reservoir with analog non-spiking neurons by adjusting the spectral radius of its connectivity matrix [1]. This simplicity extremely favors the construction of an LIF reservoir that is comparable neuron-wise with analog non-spiking reservoirs. Hence, we leverage the ESN connectivity matrix W , as explained in section 3.1, and performed an MI analysis, as explained in section 4.1, to adjust the neuron and synaptic parameters, thereby achieving a reservoir that

exposed all the stable, critical, and chaotic dynamics. To construct the reservoir, we used connectivity matrix W in ESN and adjusted the spectral-radius of the connectivity matrix $|\lambda_{max}|$. The spectral-radius of the reservoir weight matrix, W , is an eigenvalue of W with the largest absolute value (more on this later in section 4.2). We constructed the connectivity matrix with $|\lambda_{max}| = 1$ and adjusted the synaptic efficacy q , as described in subsection 3.2.2, which eventually becomes a coefficient for $|\lambda_{max}|$ when the weight matrix is applied to the network. For the sake of terminological consistency between the spiking and non-spiking reservoirs we refer to w as the reservoir’s network weight coefficient throughout the text. One can consider $w = |\lambda_{max}|$ in the article. Description of the LIF neuron and synaptic models in the reservoir follows in the next two subsections. We tested our neuron and synaptic parameters so that the overall reservoir dynamics expose stable, critical, and chaotic dynamics (see our MI and MCMC results for the LIF reservoir in chapter 5).

3.2.1 The Leaky Integrate-and-Fire Neuron Model

We regard the LIF neuron as our spiking neuron model for this study. The dynamics of the membrane potential (v_t) of the LIF neuron are defined by

$$\tau_m \dot{v} = -v_t + I_t R, \tag{3.4}$$

where R is the resistance, $\tau_m = RC$ is the membrane time constant (C is the capacitance), and I_t is the input current induced to the neuron over time. When the voltage of the neuron exceeds the threshold v_{th} , a spike is said to have occurred and the voltage is reset to the resting potential v_r after a specific absolute refractory period τ_{ref} ($5ms$ in our study). We set R and τ_m to 1 and $20ms$, respectively. We

assume that our model is dimensionless. We set the threshold $v_{th} = 1$ and the resting potential $v_r = 0$ to achieve the same range of state values as in sigmoidal and LI neurons for a more robust comparison.

3.2.2 The Synaptic Connection

We use a static synaptic connection model with an exponential decay [2], $g(t) = qe^{(-s/\tau_{syn})}$, with decay constant τ_{syn} , and maximum conductance $q = 1$ (the latter is multiplied by network connectivity weight coefficient w) for the reservoir with LIF neurons, where g is the synaptic conductance, $s = t - t^{(f)}$ is the time deference from the post-synaptic spike time $t^{(f)}$ and the current time t . We do not consider depression and facilitation dynamics in the dynamic synapses used in [2] to make the reservoir model as close as possible to the analog non-spiking reservoirs. In more biologically realistic synapse models, the input current for each neuron is obtained by $I_{syn}(t) = g(t)(v(t) - E_{syn})$, in which I_{syn} is the pre-synaptic current contributing to the input of the post-synaptic neuron and E_{syn} is the reversal potential of the synapse used for excitatory and inhibitory synapses. However, in our model the positive and negative values of the connectivity weight matrix account for inhibition and excitation of the neurons. For simplicity, we omit the voltage factor in input current and assume that the input current is applied directly to the membrane voltage for simplicity of the model by assuming $v(t) - E_{syn} = 1$. The total input current to neuron i is then achieved by the sum over all current pulses that result from the spikes of all pre-synaptic neurons j as $I_{syn_i}(t) = \sum_j g_j(t)$.

Methodology

The measuring of the computational power of the reservoir has been addressed from different perspectives. Maass *et al.* [2] looked first at computational power as a function of separation, fading memory in the reservoir, and approximation of the readout layer [2]. Separation is defined as the Hamming distance between two different internal states of the reservoir over time caused by two different extraneous inputs. The amount of separation between the dynamics of the reservoir which results from two different input signals should be well above the separation caused by any internal noise or any imposed extraneous noise. Also, the reservoir must possess a fading memory property to avoid the history of input data from lasting so long that it impairs the separation property of the recent inputs. The approximation property is the capability of the readout(s) to distinguish and transform different internal states of the liquid into the target outputs. In this study, we do not examine the approximation property, as our main focus is on the effect of different types of nodes on the reservoir's dynamic and performance. We use a linear learning algorithm for our readout layer; this remains the same in all of our experiments.

Natschläger *et al.* [5] introduced a predictor for the computational capabilities of a typical recurrent network of binary nodes that is based on the separation capability, which was numerically evaluated by a mean field approximation of the Hamming distance between different network states evolving from distinct input sequences. Mean field theory studies the behavior of large and complex stochastic

models by studying a simpler model: i.e., by reducing a many-body problem to a one-body problem. Büsing *et al.* [6] extended this method to investigate the dynamics of the reservoir composed of non-spiking analog nodes with sigmoid functions by quantizing the states of the nodes and mapping that to a smaller state space with a few bins. In both approaches the mean field approximation is possible when we are aware of the distribution and connectivity of the network. This measurement becomes impractical when the complexity of the network (in terms of both connectivity distribution and nodes' dynamics) is considerably high.

Computation is, in essence, a form of information reduction. In fact, the maximum amount of information we can extract from an input is less than or equal to the amount of the input information [42]. The capability of representing the input information is an essential condition for computing an output from a given input. Moreover, many real-world temporal stimuli are in the form of a sequence of spatio-temporal patterns. For instance, in an auditory system, words are composed of letters, or sentences are composed of words. The visual cortex is able over time to process the temporal movements of objects. Therefore, it is important for a computing system, such as RC system, to be able to represent and process this type of information.

In the following sections, we present our methodology for measuring the capability of the RC systems in representing such spatio-temporal inputs and processing them to solve a temporal parity task. We also describe two speech-recognition tasks for benchmarking the performance of the reservoirs in solving more realistic spatio-temporal tasks.

4.1 Measuring the Mutual Information between the Reservoir and Spatio-Temporal Inputs

According to data-processing inequality, the information content of a signal cannot be increased via a local physical operation. Hence, the computational capability of a system has a close relationship with the amount of input information that can be represented in the system [42–44]. We measure the mutual information to quantify this characteristic of the reservoir. Mutual information (MI) quantifies a relationship between two random variables. In particular, it measures how much information is communicated, on average, in one random variable about another. Informally stated, it measures how much one random variable tells us about another. For example, suppose X represents the roll of a fair 6-sided die, and R represents whether the roll is even (0 if even, 1 if odd). The value of R tells us something about the value of X and vice versa. In this example, X and R variables share mutual information. In contrast, if R does not tell us anything about the roll of the die, X and R do not share any mutual information. In our case, in the analogy mentioned, X represents the input value at time t , and R represents the reservoir dynamics at time t' . Later, we consider X to represent the output from the reservoir so as to measure the amount of information shared between the reservoir state and the output.

The mutual information between two random variables X and R has been defined by Shannon [45] as

$$I(X; R) = H(X) - H(X|R) = H(R) - H(R|X), \quad (4.1)$$

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x), \quad (4.2)$$

$$H(X|R) = \sum_{x \in \mathcal{X}} p(x) H(R|X = x) = - \sum_{x \in \mathcal{X}} \sum_{r \in \Psi} p(x, r) \log_2 p(r|x), \quad (4.3)$$

where $H(X)$ is the entropy of the random variable X and $H(X|R)$ is the conditional entropy of the random variable X given R (see Cover and Thomas [42]). In our case, X corresponds to the input and R corresponds to the state of all or a subset of reservoir components at time t_0 . The classical empirical measurement of the mutual information is subject to under-sampling effect, which results in over estimation of the overall mutual information [9]. Natschläger *et al.* [9] exploited the data processing inequality and derived a useful method to calculate a close lower bound for mutual information. According to data-processing inequality we have the following:

$$I(X; R) \geq I(X; g(R)), \exists g(.) \text{ independent from } X. \quad (4.4)$$

$g(R)$ can be a linear function, which maps the reservoir state to an output.

4.1.1 Measuring MI between the Reservoir and Input

We will now explain in more detail the empirical measurement of the mutual information proposed by Natschläger *et al.* [9]. Let S be a sample space for all examples $X = x$. There are two steps to generate the input for each example in the sample space of the random variable X : 1) generate two input templates, and then, 2) generate the actual inputs from the templates. Recall that the input corresponds to the input random variable X when we measure the mutual information between the input and the reservoir state. Therefore, each instance of the input is considered a possible value x for random variable X among all inputs in the sample set

S. First, we need two templates of spatio-temporal spike patterns consisting of 5 spike trains (i.e. 5 channels of binary series) with length $n \times L$ time-steps, from which we will generate the actual input examples. n is the number of input sections and L , here, is the constant length of each sections. $L = 200$ time steps in the entire study. Each template is obtained by combining 5 binary time series generated from a Poisson distribution with frequency of $20Hz$. The patterns are to simulate the spatio-temporal firing patterns from arrays of sensory neurons or from other brain areas caused by, e.g., visual or audio stimulation [9]. Figure 4.1 (A) shows two instances of the templates mentioned. Second, we generate the input examples randomly from the two input templates. For instance, the following binary sequences are 3 arrays of input sections generated randomly from the input templates with n sections.

$$\begin{array}{r}
s_1, s_2, s_3, s_4, s_5, s_6, \dots s_n \\
1, 0, 1, 1, 0, 0, \dots 1 \\
1, 0, 1, 1, 1, 0, \dots 0 \\
0, 0, 1, 0, 1, 1, \dots 0
\end{array} \tag{4.5}$$

A generated input consists of n randomly selected sections s_i , $i = 1, 2, 3, \dots, n$, $200ms$ long each, and we define the value of s_i to be equal to the label of the template, from which the section s_i has been chosen. In other words, for any i in template zero, we say $s_i = 0$; and for any i in template one, we say $s_i = 1$. In order to generate an input instance, we pick n sections from the corresponding template section, each of which is randomly and independently selected from section s_i of template zero or one.

MI measurement with this approach was shown to be more practical, since it

yields a close lower-bound for MI with a reasonably small sample size $|S|$. Empirical MI obtained with this approach can fit well to the expected theoretical lower-bound with $|S|$ as small as 100 [9]. In our experiments, we set $|S| = 250$ to be on the safe side in terms of accuracy. We set the same sample size (different sample set) for training the readout to obtain g .

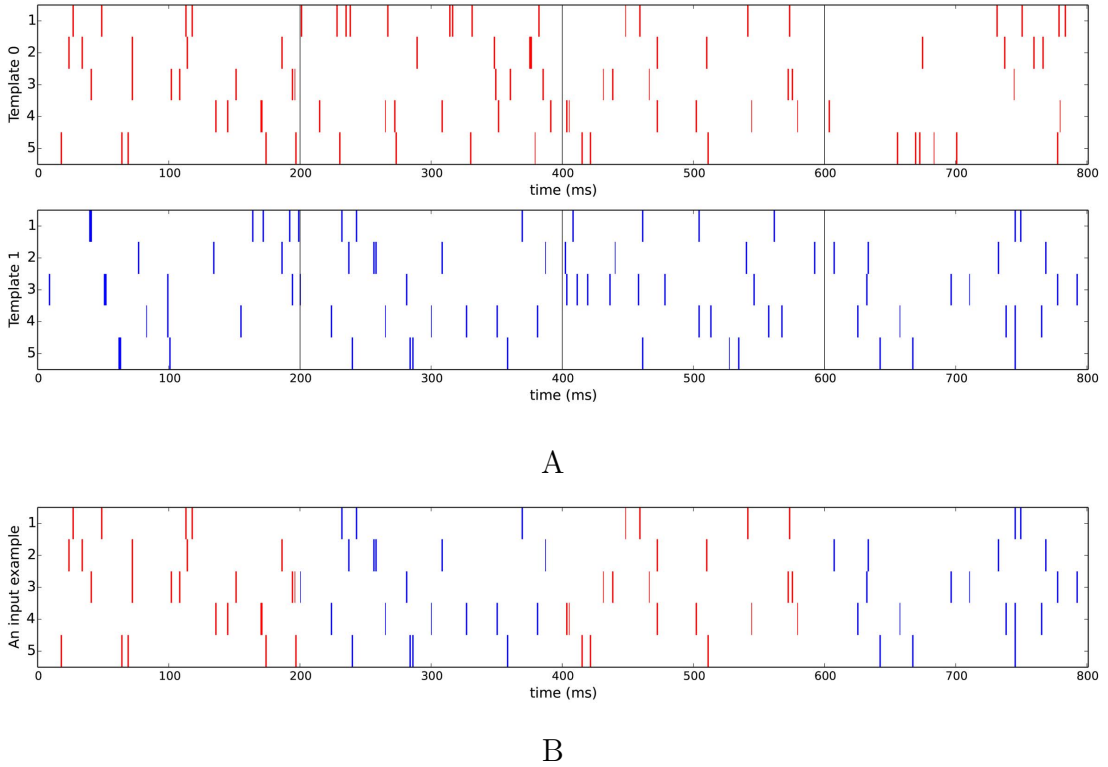


Figure 4.1: A) Template 0 (in red) and template 1 (in blue) consist of 4 sections s_i of spike patterns. Each section s_i is comprised of 5 randomly generated $200ms$ spike trains: i.e., 5 channels over $200ms$, from a Poisson distribution with the frequency of $20Hz$. B) An input instance, comprised of a random selection of the sections s_i , chosen from template 0 ($s_i = 0$) or from template 1 ($s_i = 1$). In this particular example, $s_1 = 0, s_2 = 1, s_3 = 0, s_4 = 1$.

Next, we need to obtain the function $g(R)$, described in equation 4.4. We divide our sample space S into two training and testing samples. The training sample

is used to obtain $g(R)$ and the testing sample for measuring the MI. We apply the inputs in the training sample to the reservoir and train our linear readout by a linear regression and then classification, as we will explain shortly. Note that the inputs in the training set are not used for mutual information measurement so that g remains independent from the inputs used in the measurement phase. Otherwise, the data processing inequality would not hold [9, 42].

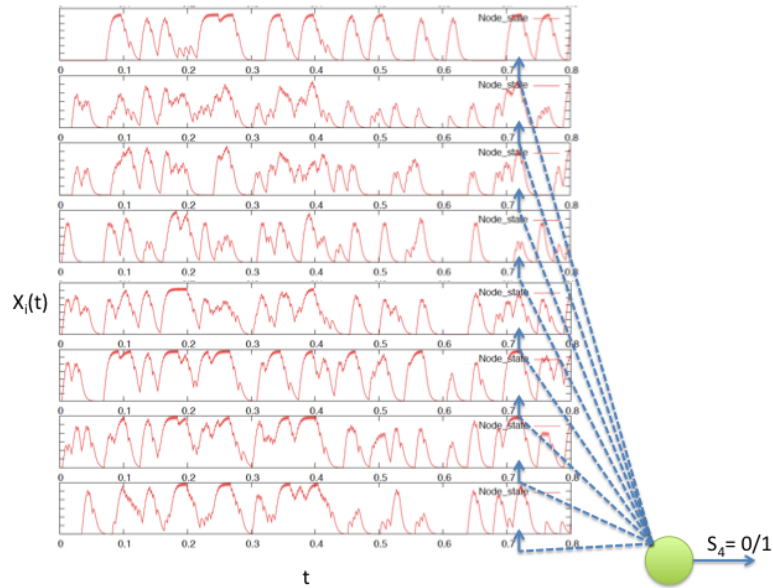
In order to measure the MI, we perturb the reservoir with a generated input of length $n \times L$ ($L = 200ms$). For every $t_0 = 20, 40, \dots, n \times L$ we measure the MI between each section of the input and the state of a subset of nodes in the reservoir, according to the equations 4.1 and 4.4. We consider the MI between the state of the reservoir at time t and all s_i .

To obtain the optimized classification result for g , we classify the reservoir state $X(t)$ to two classes of 0 and 1 at time t_0 after regression in the training phase. We define the regression output of our linear readout layer as $h(t) = \sum_{j=1}^N w_j x_j(t) + b$, where w_j is the adjusted weight by the linear regression and b is the bias. $x_j(t)$ refers to the state of neuron j at time t . In order to find the best possible classification, we use the *Receiver Operating Characteristic* (ROC) curve analysis [46] for classification hypothesis testing and to find the best threshold for the classifier $g(u(t))$. The accuracy of our binary classification is achieved from,

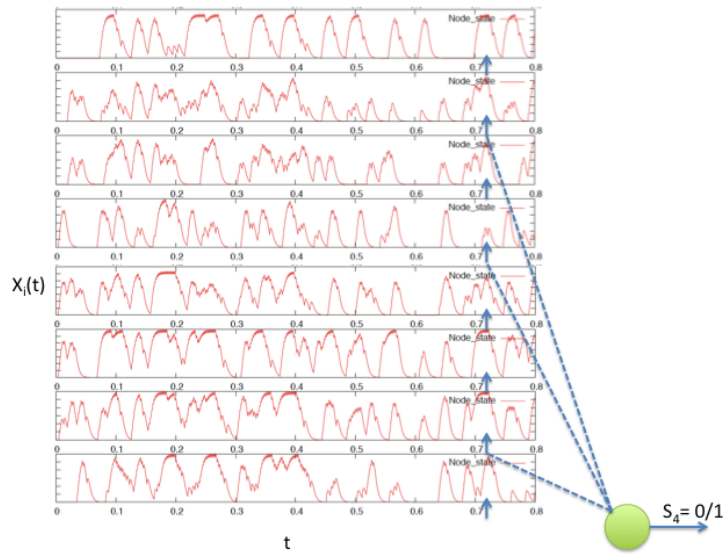
$$ACC = \frac{TP + TN}{P + N}, \quad (4.6)$$

where TP denotes the number of truly classified 1s, and TN denotes the number of truly classified 0s. $P + N$ is the sum of the number of all miss-classified and truly-classified 1/0s respectively. ACC gives us a good indicator for quality of the classifier g , though we do not use ACC , as the MI is informative enough for our

analysis and comparisons. Figure 4.2 shows a schematic view of reading out the state of the nodes or a subset of the nodes in a reservoir at time $t_0 = 720ms$ and their mapping into two classes of 0 and 1 when measuring the mutual information. Figure 4.3 shows a schematic view of training the readout layer over trials in a reservoir at time $t_0 = 720ms$ to obtain the g function.



A



B

Figure 4.2: A schematic view of reading out the state of a reservoir of size $N = 8$ at time $720ms$ when measuring the mutual information. A) The readout node is receiving inputs from all nodes in the reservoir. B) A randomly selected subset of reservoir components is selected.

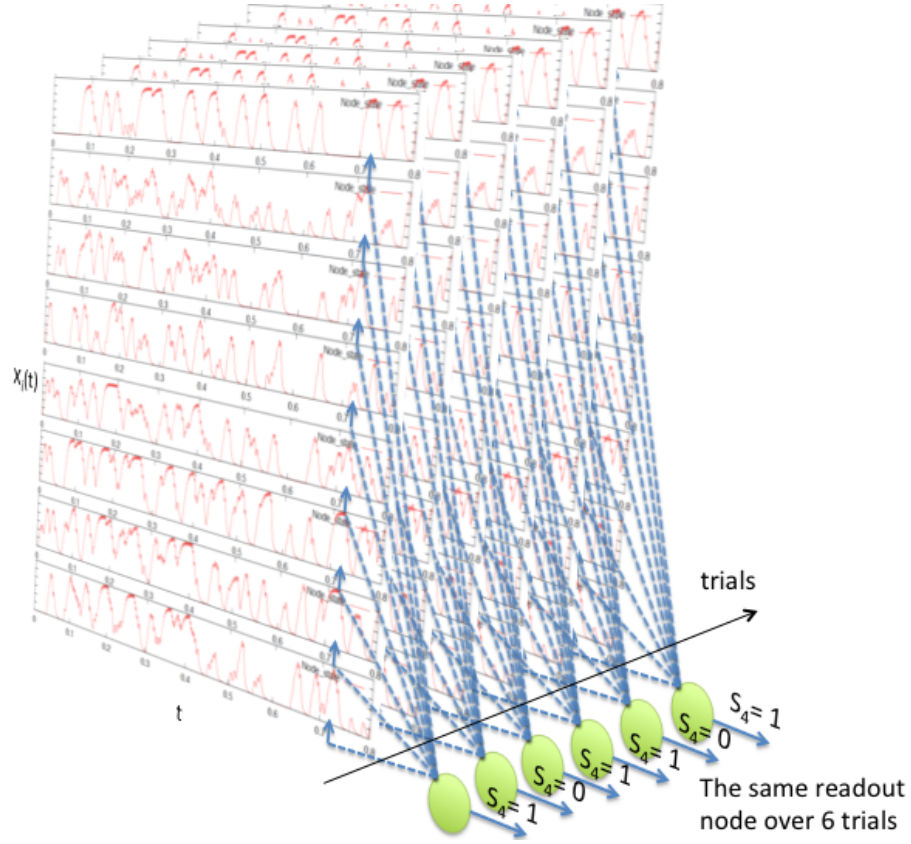


Figure 4.3: A schematic view of training the readout layer over trials with a reservoir of size $N = 8$ at time $720ms$ and input with $n = 4$ sections to obtain the g function. Later at the MI measurement phase, the mutual information $I(u, g(R))$ is measured between the input u in sections $s_i = 1, 2, 3, 4$ and the state of the reservoir at time $t = 720ms$.

4.1.2 Measuring the MI between the Reservoir and PAR_n of the Input

In order to measure the performance of the RC system in solving a linearly unsolvable temporal task, we consider a linearly inseparable temporal task denoted by PAR_n : the temporal parity of n sequential binary digits. The MI measurement

steps here are identical to those of the MI between the reservoir and the input; the only difference is that $PAR_3(u)$ supersedes the input u in representing the random variable X . The PAR_n function is defined as follows:

$$PAR_n(u, t) = \begin{cases} 1 & , \sum_{k=0}^n u(t - (k\tau)) \text{ is even,} \\ 0 & , \sum_{k=0}^n u(t - (k\tau)) \text{ is odd ,} \end{cases} \quad (4.7)$$

where $u(t) \in \{0, 1\}$ is a binary digit at time t , $\tau \in \mathbb{R}$ here is the time distance between the digits, and $k = 0, 1, 2, \dots, n$. Below is an example of the temporal parity 3. u is the input and the resulting $PAR_3(u)$ is the output of the temporal parity 3 function over time. The s_i variables here are used to show the correspondence of the input sections in MI measurement to the binary labels and their PAR_3 .

$$\begin{aligned} & s_1, s_2, s_3, s_4, s_5, s_6, s_7, \dots \\ u : & 1, 0, 1, 0, 0, 1, 1, \dots \\ PAR_3(u) : & -, -, 0, 1, 1, 1, 0, \dots \end{aligned}$$

After measuring $I(u; g(R))$, we measure $I(PAR_3(u); g(R))$ to observe the capability of the reservoir in solving PAR_3 and compare it to $I(u; g(R))$ side-by-side, thereby visualizing the contribution of the amount of input information represented in the system in solving a linear unsolvable temporal task: i.e. PAR_3 .

4.2 Measuring Memory and the Stability of the Reservoir

The stability of the network dynamics is an important property for the reservoir. The readout layer can interpret the reservoir dynamics consistently over time only if the reservoir dynamics are stable. Note that the notion of stability here should not be confused with convergence to point attractors as in the Hopfield network [13].

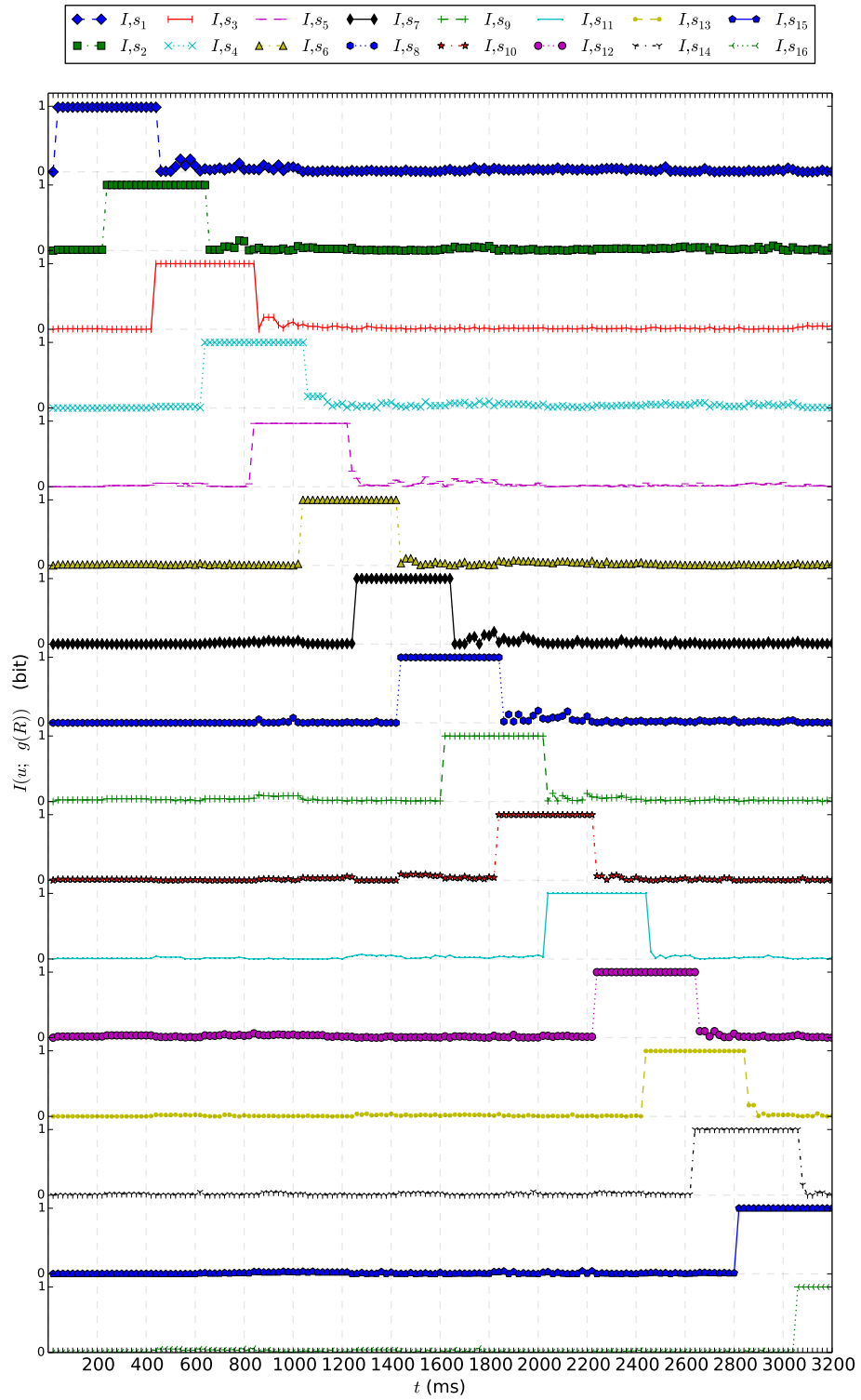
Maass *et al.* [2] refer to a similar property as fading memory. In a reservoir with fading memory property, when a finite time input $u(\cdot)$ is injected to the reservoir over the time course of $t \in [t_0, t_n]$, the memory of the input should not remain in the reservoir after a finite time $\tau = t - t_n$, $t > t_n$.

Jaeger [1] put forward the network state forgetting property as an essential condition for the RC system. The reservoir is said to have a forgetting property if the distance between the states (X and X') of two identical networks with identical inputs goes to zero at some time $t \rightarrow \infty$.

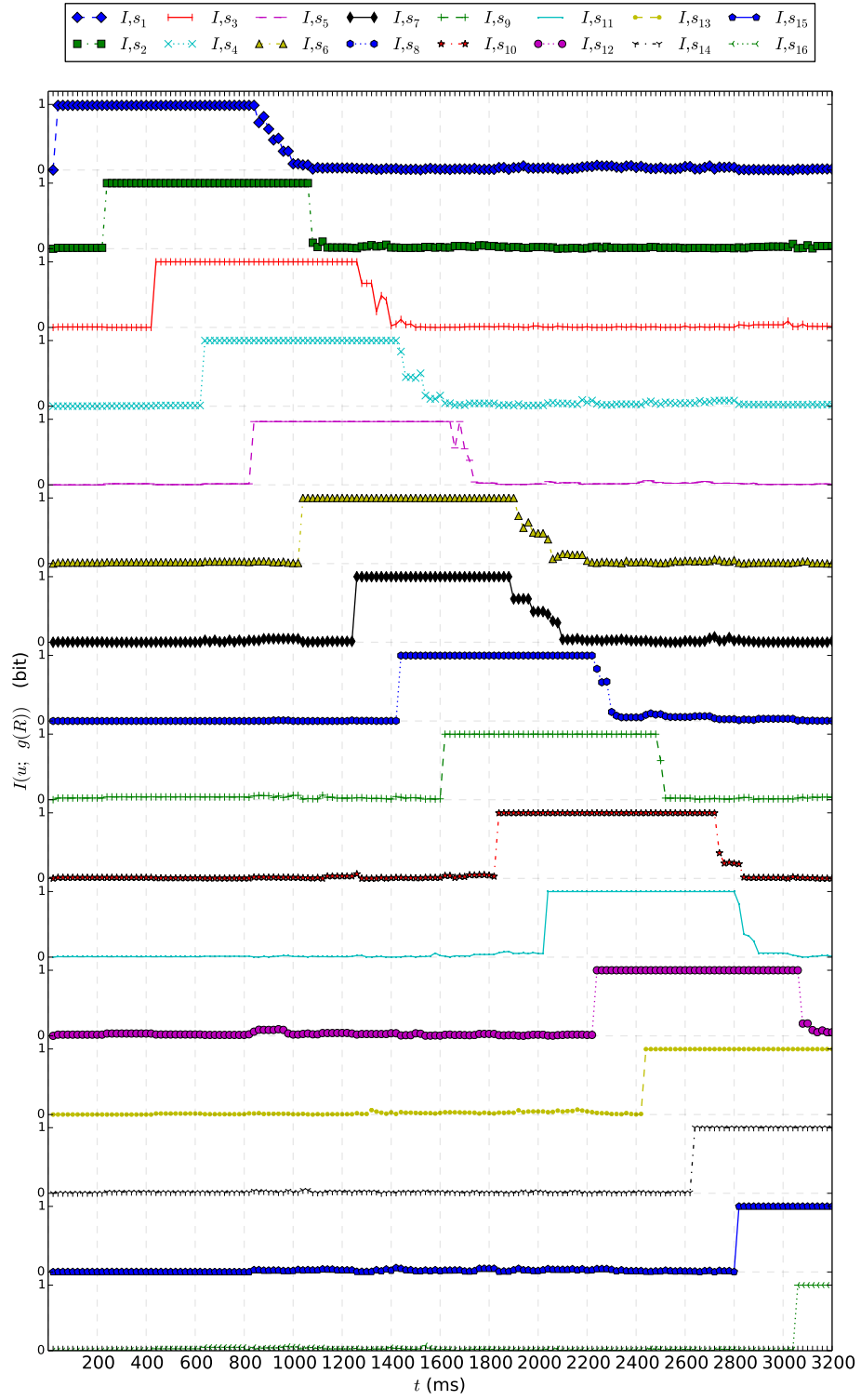
Here, we refer to the similar quality as stability in the reservoir dynamics. Our MI results taken from different RC systems show this property from an information representation perspective. Figure 4.4 shows the effect of the reservoir stability on the information representation of the reservoir by using MI between the state of the reservoir and the input taken from ESN when $\lambda_{max} = 1$ (A) and when $\lambda_{max} = 2$ (B), where λ_{max} is the spectral-radius of the weight matrix. Spectral-radius of the reservoir weight matrix W is an eigenvalue of W with the largest absolute value. Jaeger [1] showed that the ESN with $\lambda_{max} < \approx 1$ has state forgetting property. As λ_{max} increases, this property is no longer guaranteed. We present the full extent of our MI results in chapter 5.

Figure 4.4: Three examples of $I(u; g(R))$ taken from ESN with size $N = 100$ and input with 16 sections using the method described in section 4.1. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) When $\lambda_{max} = 0.9$. Notice that at any time t_i , the information about the past 2 sections of the input is present in the state of the reservoir, denoted by random variable R . B) When $\lambda_{max} = 1$. Notice that at any time t_i , the information about the past 4 sections of the input is present in the reservoir state, denoted by random variable R , which is much more than when $\lambda_{max} = 0.9$ (shown in A). C) When $\lambda_{max} = 1.5$, the reservoir is only able to represent the past sections at time t in the beginning of the input; then the reservoir no longer has any memory of the past input sections. (A) and (B) show the stable reservoir dynamics, as the MI consistently remains at a certain level. (C), however, shows the information dynamics of an unstable (chaotic) reservoir.

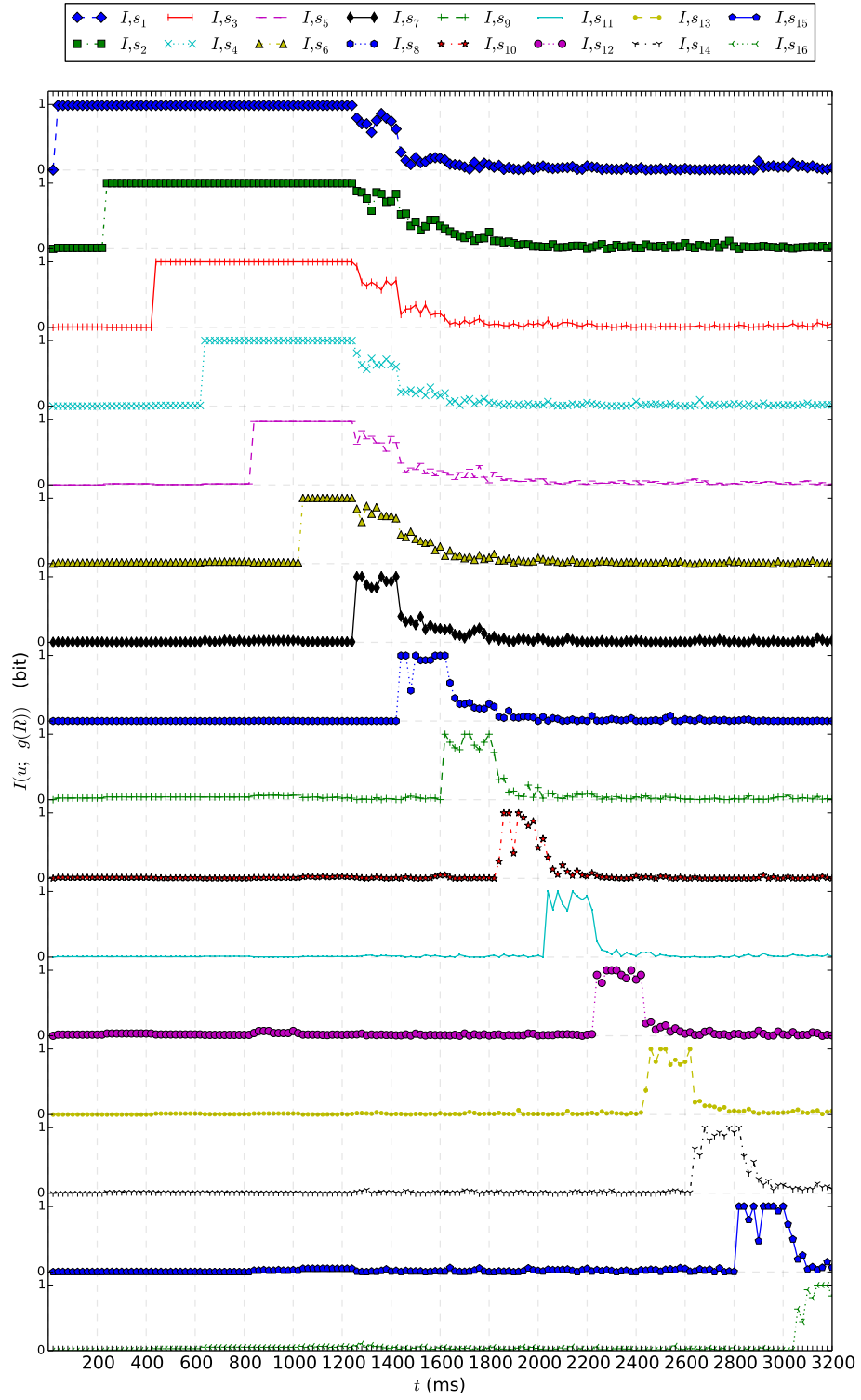
A



B



C



In order to define a comparative indicator of the capability of the reservoir in consistently representing the input and solving PAR_3 , we define *Mean Cumulative Mutual-Information* (MCMI) I_μ as in equation 4.8.

$$I_\mu(X_s; g(R)) = \frac{1}{N - k_0} \sum_{k=k_0}^N \sum_{i=1}^n I(X_{s_i}; g(R(k\tau))), \quad (4.8)$$

$$N = \frac{T}{\tau},$$

where X_s is either the section label (s_i) of the input u and is denoted by $I_\mu(u; g(R))$ or is the result of $PAR_3(u)$ for (s_i) of the input u , which is denoted by $I_\mu(PAR_3(u); g(R))$. I is the MI described in equation 4.1. n is the total number of input sections. T is the total simulation time. τ , here, is the constant time-bin of measuring MI over time ($\tau = 20$ time-steps in our study), N is the number of MI measures taken over time for each section s_i of either u or $PAR_3(u)$, and k_0 is the number of starting time-bin to ignore the MI results before $t = k_0\tau$. We fix k_0 so that all s_i ($i \leq 7$) sections are ignored from I_μ value – that is, $k_0 = 36$ throughout the study. Informally stated, I_μ is the average of accumulated I 's for all sections of the input. Since the MI for the future input sections is 0, I_μ is actually the average of accumulated I 's about the past sections of the input. Figure 4.5 shows the I_μ 's for reservoir with analog sigmoidal neurons (ESN) obtained from $I(u; g(R))$'s, as shown in figure 4.4, which are computed for 3 different λ_{max} values 0.9, 1, and 1.5.

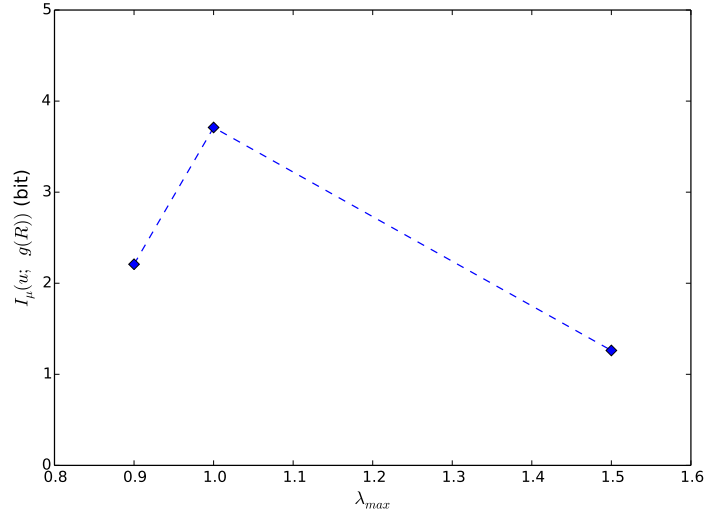


Figure 4.5: I_μ of the reservoir with analog sigmoidal neurons (ESN) obtained from $I(u; g(R))$ shown in figure 4.4. Notice that I_μ at $|\lambda_{max}| = 1.5$ is less than I_μ at $|\lambda_{max}| = 0.9$ because we ignore the memory of the reservoir before section 7. The memory about past input sections is decreased after section 7 as $|\lambda_{max}|$ is increased (and $|\lambda_{max}| > 1$) in ESN.

4.3 Benchmarking the Reservoirs' Performance in Recognizing the Isolated Spoken-Digits

In order to test the performance of the in-question reservoir types in solving a real-world spatio-temporal task, we consider an isolated spoken-digits recognition task. The task is to recognize ten isolated digits spoken by five different female speakers available in the TI46 corpus. We use a subset of this corpus, consisting of 500 samples for ten different utterances of the isolated digits ‘zero’ to ‘nine’, spoken by five different speakers. The preprocessed speech data has a spatio-temporal form of 40 spike trains, with an average $\approx 300ms$ long each. We inject the input after

$t = 7 \times 200ms$ to ignore the initial unstable memory of the reservoirs in case of a chaotic reservoirs. We are particularly interested in understanding whether higher MCMI brings about a higher performance among the reservoir types in-question.

The corpus was first used in [47] to perform speech recognition by a simple Spiking Neural Network. It was later exploited by Verstraeten *et al.* [10] to analyze the LSM performance in this task with different preprocessing methods. The speech was preprocessed using a biological model of the human cochlea by Lyon [48] resulting in a cochleagram. This cochleagram was converted into 40-channel spike trains using a filter-coding method called BSA, as proposed by Schrauwen *et al.* [49]. The preprocessing method was shown to yield a high performance (95%) with LSM in [10].

We follow the same readout and classification method as in [10]. The read out is a simple linear projection that linearly maps the reservoir state $X(t)$ onto the output $Y(t)$ using a weight matrix W obtained by pseudo matrix inversion [10]. I.e., $Y(t) = WX(t) + W_0$. Every $20ms$, and when the entire input is injected to the reservoir, the $Y(t)$ is obtained and the maximum value of the output vector $Y(t)$ is taken as the actual class selection (winner-take-all selection). Then the best performance among all the max values is determined as the final classification. This method has been shown to perform better in [10] than some other readout functions, such as a Fisher discriminant or a pool of parallel perceptrons [50]. The performance of the task is expressed as the proportion of incorrectly classified words (False Positives + False Negatives) to the total number of word samples and it is referred to as the Word Error Rate (WER) in [10]. Here, we use $(1 - \text{WER})$ to indicate the performance. The performance thus ranges from 0 to 1, and we express that as performance% in the text. From 500 samples, we use 450 samples

for training and 50 samples for testing. We validate our classification results with 10-fold cross validation.

4.4 Testing the Impact of Memory and Stability of the Reservoirs in Recognizing Spoken-Digit Sequences

In section 4.3, we discussed our method for measuring the performance of the reservoir in recognizing an isolated spoken-digit, which requires at most about 350 time-steps (*ms* for continuous-time neurons) memory. Nevertheless, shorter memory could be enough for this task. On the other hand, we observed a maximum of 4.5 sections of memory in our MI and MCMI results from the reservoir with LIF neurons and a maximum 3.7 sections of memory in the reservoir with LI neurons. That is to say, a 350 time-steps-long input does not stress a reservoir capable of even 3.7×200 time-steps memory. Thus, we construct a much more memory-demanding task based on the same task to compare the ability of in-question reservoirs to solve a memory-demanding and linearly non-separable task with a spatio-temporal spike train input. Each spoken-digit is comprised of 40 channels of 250 to 350 time-step spike trains. We concatenate the digits together randomly to obtain 10 different sequences of length $n = 1, 2, \dots, 7$ inputs. A 7-digit sequence would be as long as approximately 7×300 time-steps. For instance,

$$\begin{aligned}
 & - 1033456 \longrightarrow 0, \\
 & - 6348134 \longrightarrow 1, \\
 & - 9795020 \longrightarrow 2, \\
 & \dots, \\
 & - 3382975 \longrightarrow 9
 \end{aligned} \tag{4.9}$$

are 10 sequences of length $n = 7$ that are mapped to 0 to 9 digits. The RC system is supposed to distinguish between these digit-sequences. The generated inputs here have a complexity similar to our MI measurements inputs in terms of memory requirement and exhibit different spatio-temporal patterns over time. In other words, each digit is analogous to a spatio-temporal pattern and the system is to compute on a sequence of them, as we did in the PAR_3 task. This task challenges the reservoirs for long-term and stable memory. We perform this task with the sigmoidal, LI, and LIF reservoirs. The objective of this task is to reveal the impact of the memory and stability of the reservoir (i.e. MCMI) in the performance of the reservoirs and to find the best reservoir for solving a memory-demanding spatio-temporal and real-world task.

Results

We begin this chapter by presenting our MI and MCMI results measured for the reservoirs with sigmoidal, LI, and LIF neurons by the approach discussed in section 4.1. We then compare the ability of these reservoirs to recognize the isolated spoken-digits. We further challenge the mentioned reservoirs and test their ability to solve problems that require more memory by having them recognize the sequences of up to 7 spoken-digits.

We start by presenting the MI results for reservoirs with non-spiking analog neurons i.e., sigmoidal neurons and LI neurons and follow by presenting MI results for the reservoir with LIF neuron. For each reservoir type, we measure $I(u(t_i); R(t_r))$, which is the MI between the state of the reservoir R at time t_r and the input $u(t_i) = s_1, s_2, \dots, s_{16}$ at time t_i , where $t_r, t_i \in \{0, 20ms, 2 \times 20ms, \dots\}$, to measure the amount of input information represented in the reservoir over time. $I(u(t_i); R(t_r))$ also yields the memory of the reservoir about the input for all $t_r > t_i$. Then, we measure $I(PAR_3(u(t_i)); R(t_r))$, I is the MI between the state of the reservoir, R , at time t_r and $PAR_3(u)$ at time t_i . This implies that the RC system not only represents the input but also computes a linearly non-separable function of the input. The input is comprised of 16 spike patterns. Each pattern consists of 5 channels of $200ms$ spike-trains in all of the MI measurements presented in what follows.

5.1 Mutual Information Results for Analog Non-Spiking and Spiking Reservoirs

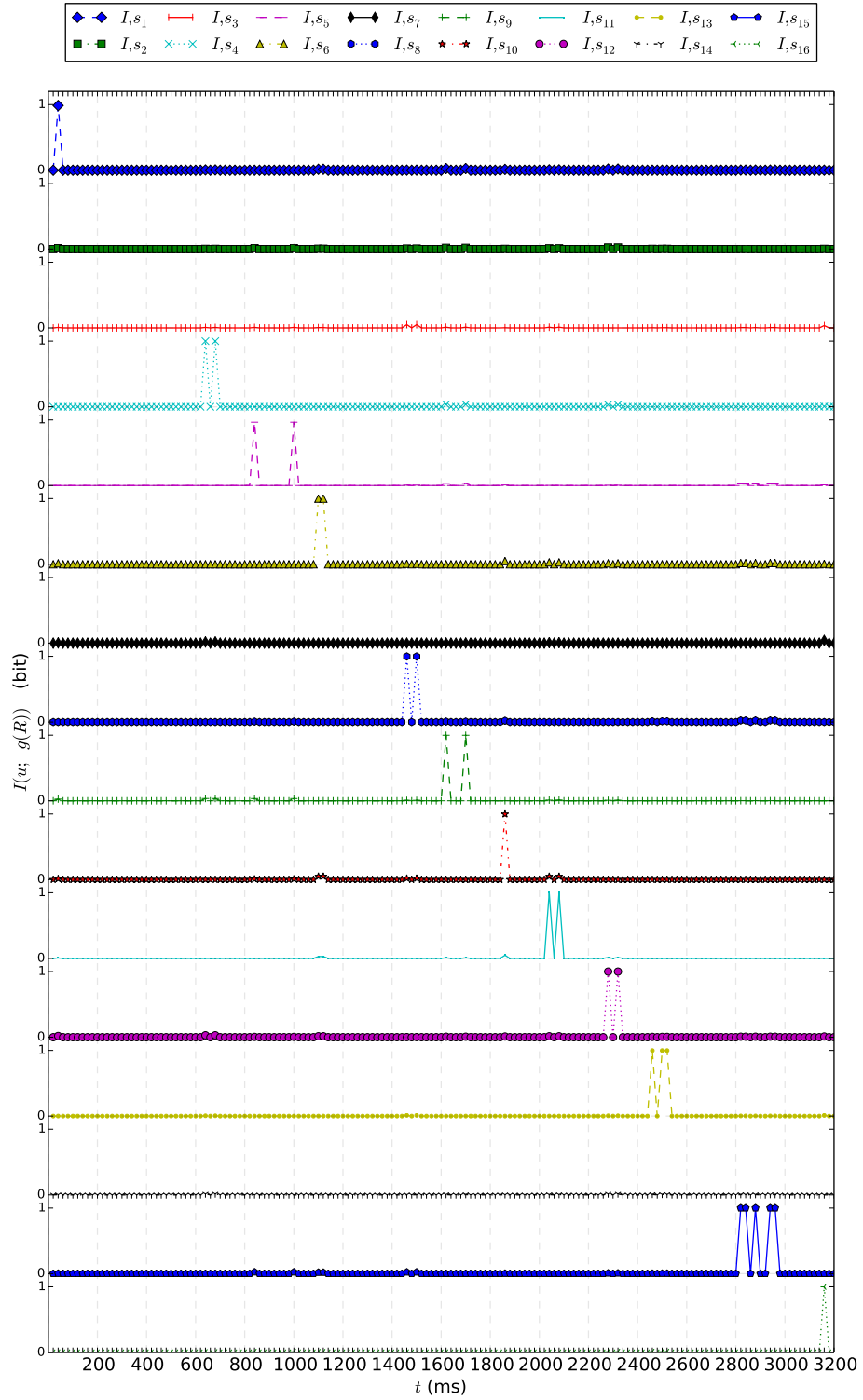
5.1.1 MI Results for the Reservoir with Sigmoidal Neurons

The MI between the reservoir state and the input is shown in figure 5.1. When $w = 0$ very little of the input is represented in the reservoir (A). As w increases the amount of information about the input is increased and it reaches its maximum level when $w = 1$ (B and C). With further increase in w , the reservoir becomes incapable of consistently representing the input. Even though the memory is highly increased for the beginning sections of the input, after section s_i , $i \geq 7$ the memory about the input is considerably decreased (D) until it finally reaches 0 (E). Figure 5.2 shows the MI between the reservoir state and the PAR_3 of the input for reservoir with sigmoidal neurons. With $w = 0$, though multiple sections enough for computing $PAR_3(u)$ are injected to the reservoir, the information represented about the input by the reservoir is not enough to compute $PAR_3(u)$ (A). Recall that information for at least 3 sections of the input has to be available in order to compute PAR_3 . Information about the PAR_3 of the input is only represented as soon as enough information about at least 3 sections becomes available in the reservoir (B) and it maximizes with $w = 1$ (C). By further increasing w , the consistency of the reservoir in representing the input is decreased. Information about the past sections again drops to less than 3 sections after section 7, and PAR_3 cannot be computed anymore from section 7 onward (D, E).

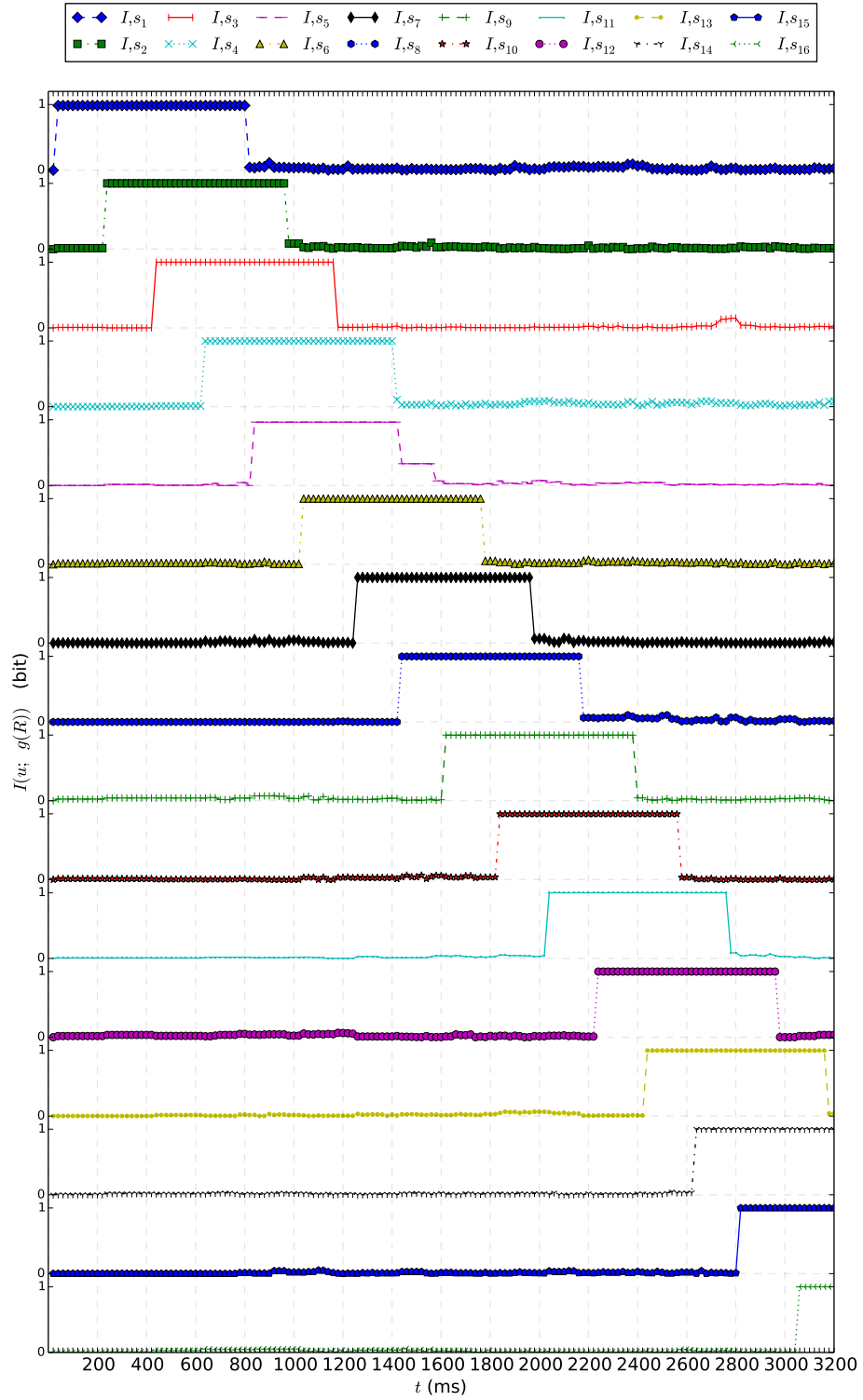
Figure 5.3 shows I_μ calculated by equation 4.8 over the MI results shown in figures 5.1. Figure 5.4 shows I_μ calculated by equation 4.8 over the MI results shown in figures 5.1 and 5.2 respectively.

Figure 5.1: Mutual information of the reservoir state and the input with 16 sections. Reservoir with analog sigmoidal neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.5$, E) $w = 2.0$. By increasing w up to $w = 1.0$, MI is increased. By further increase in w , the MI starts dropping from $t = 1200ms$ on. This means that the system does not represent the input anymore after the critical $w = 1$.

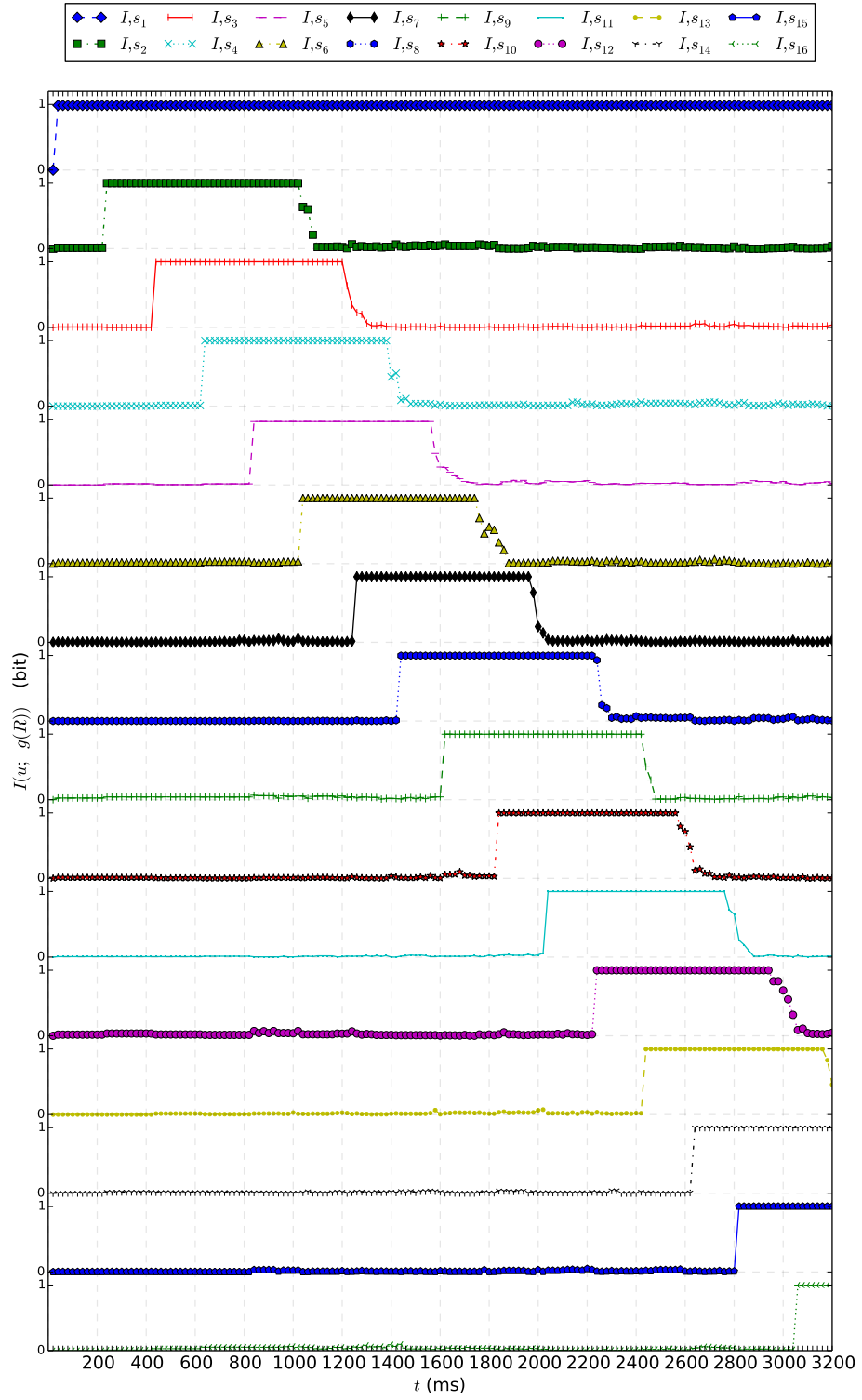
A



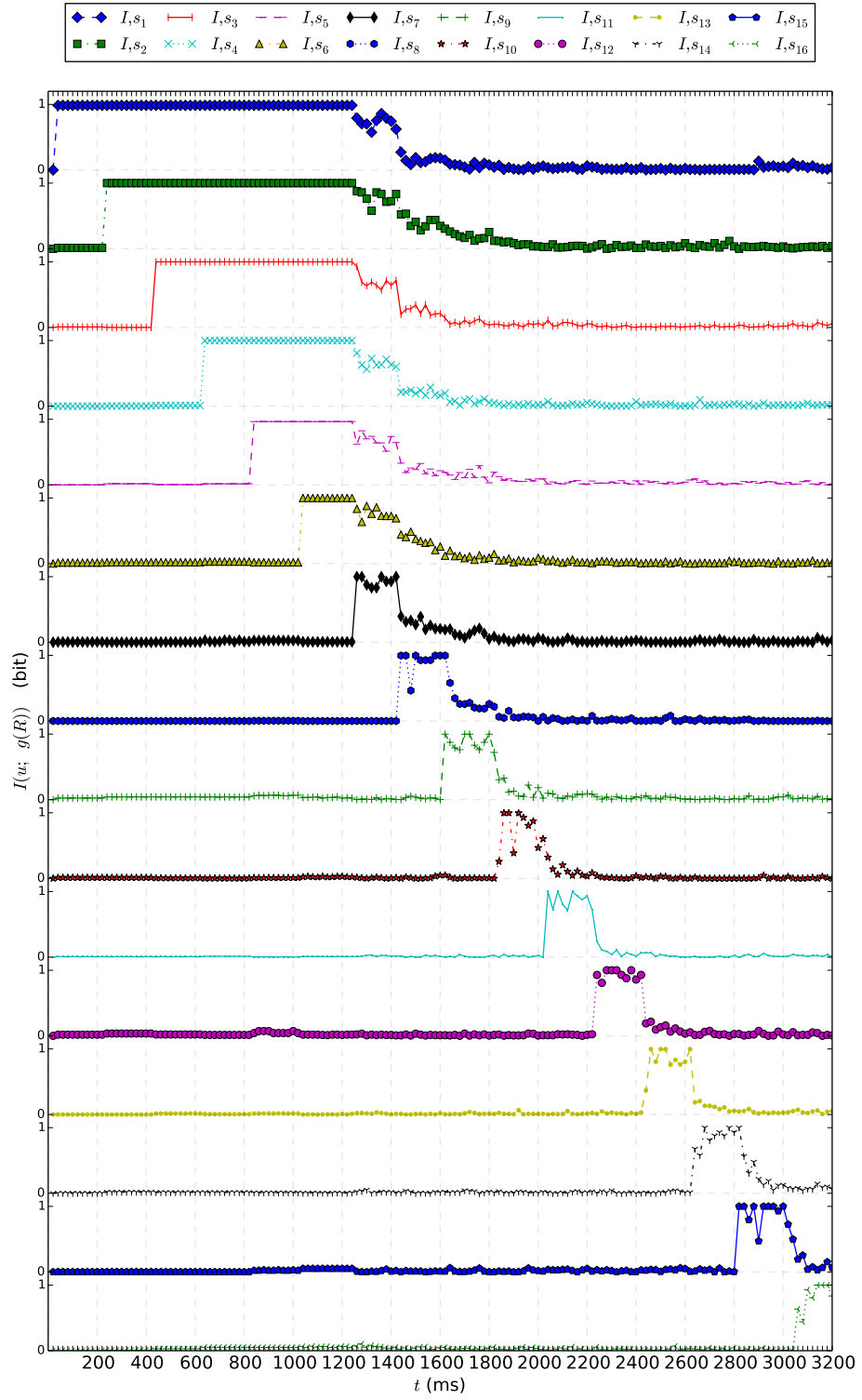
B



C



D



E

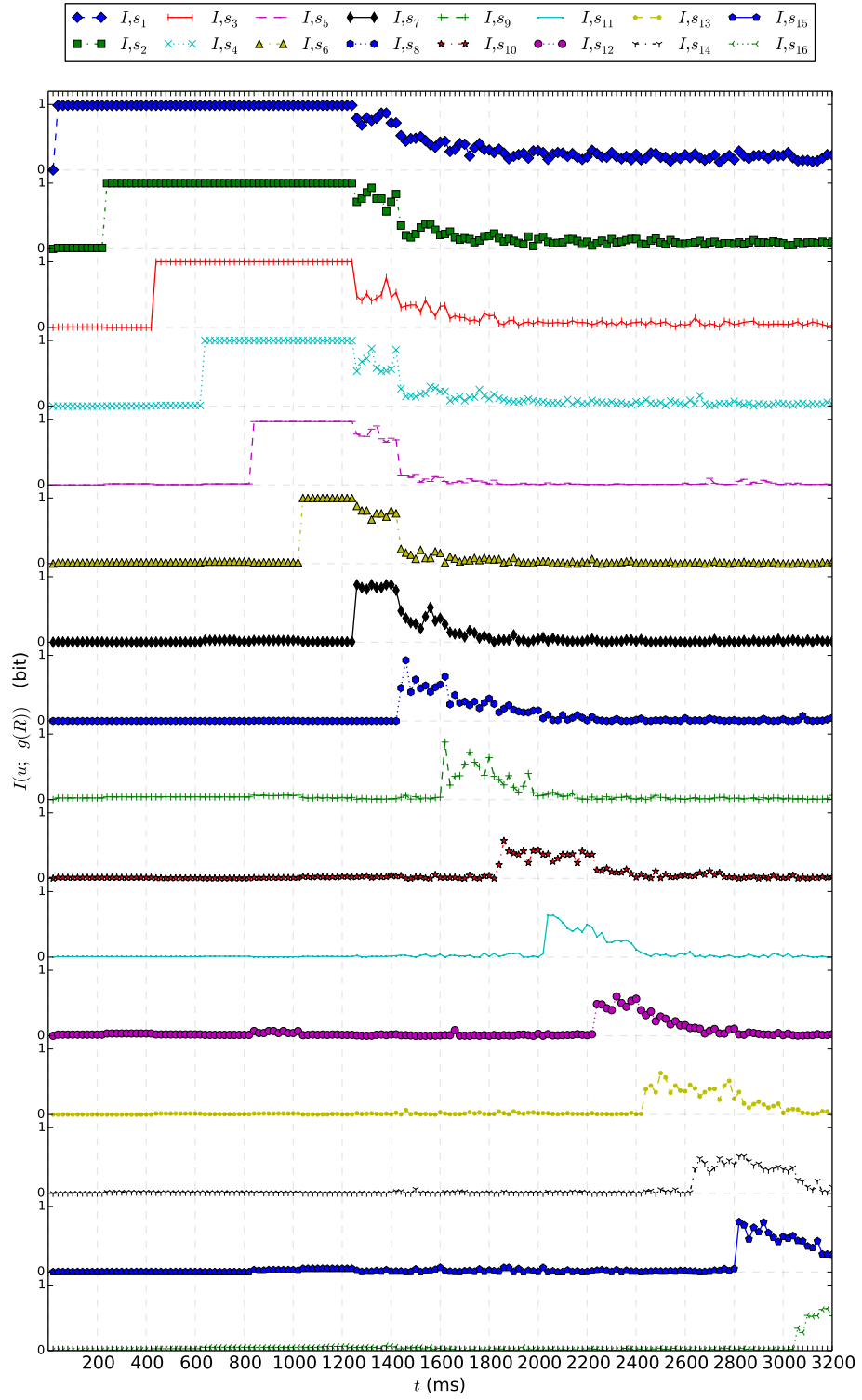
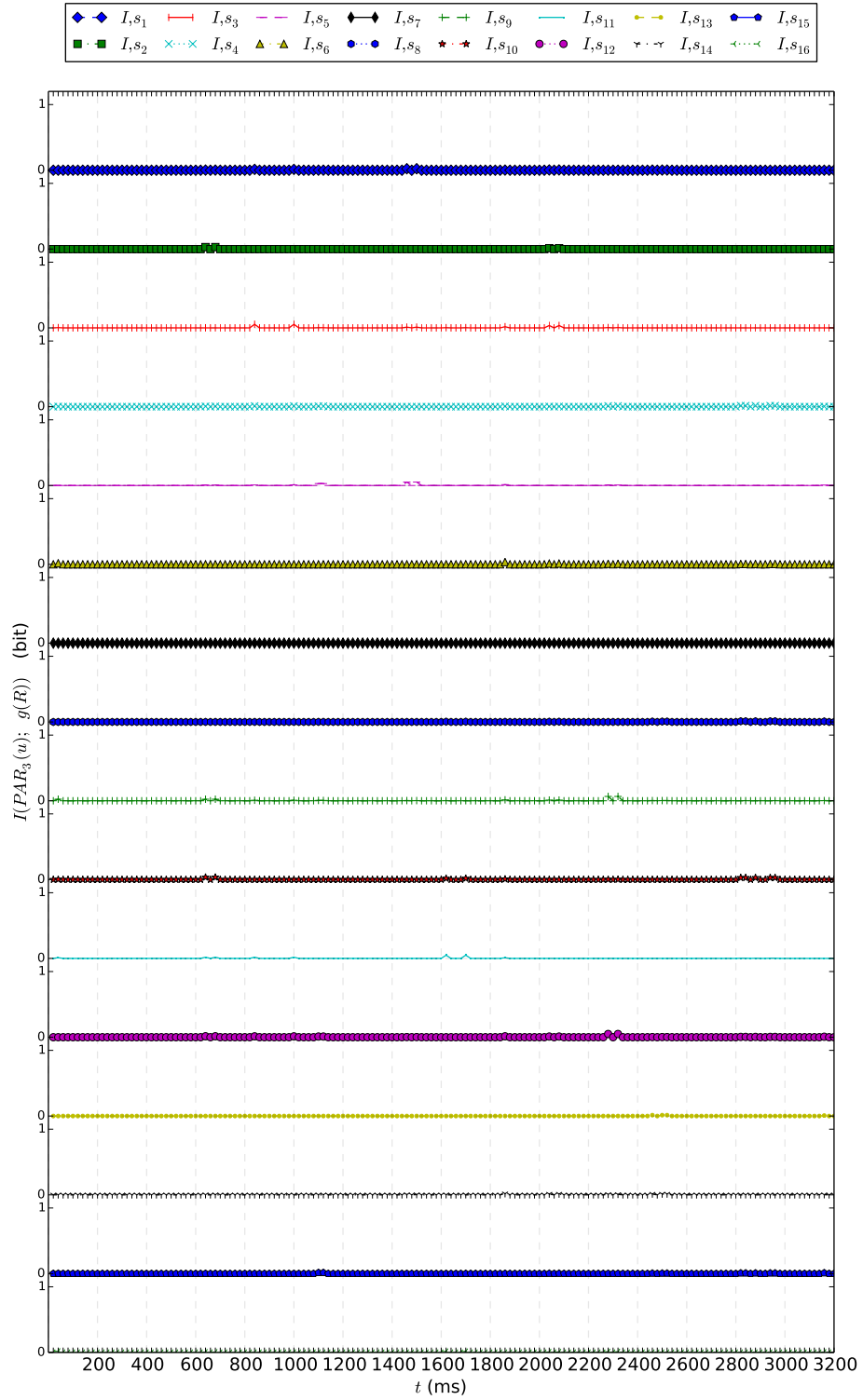
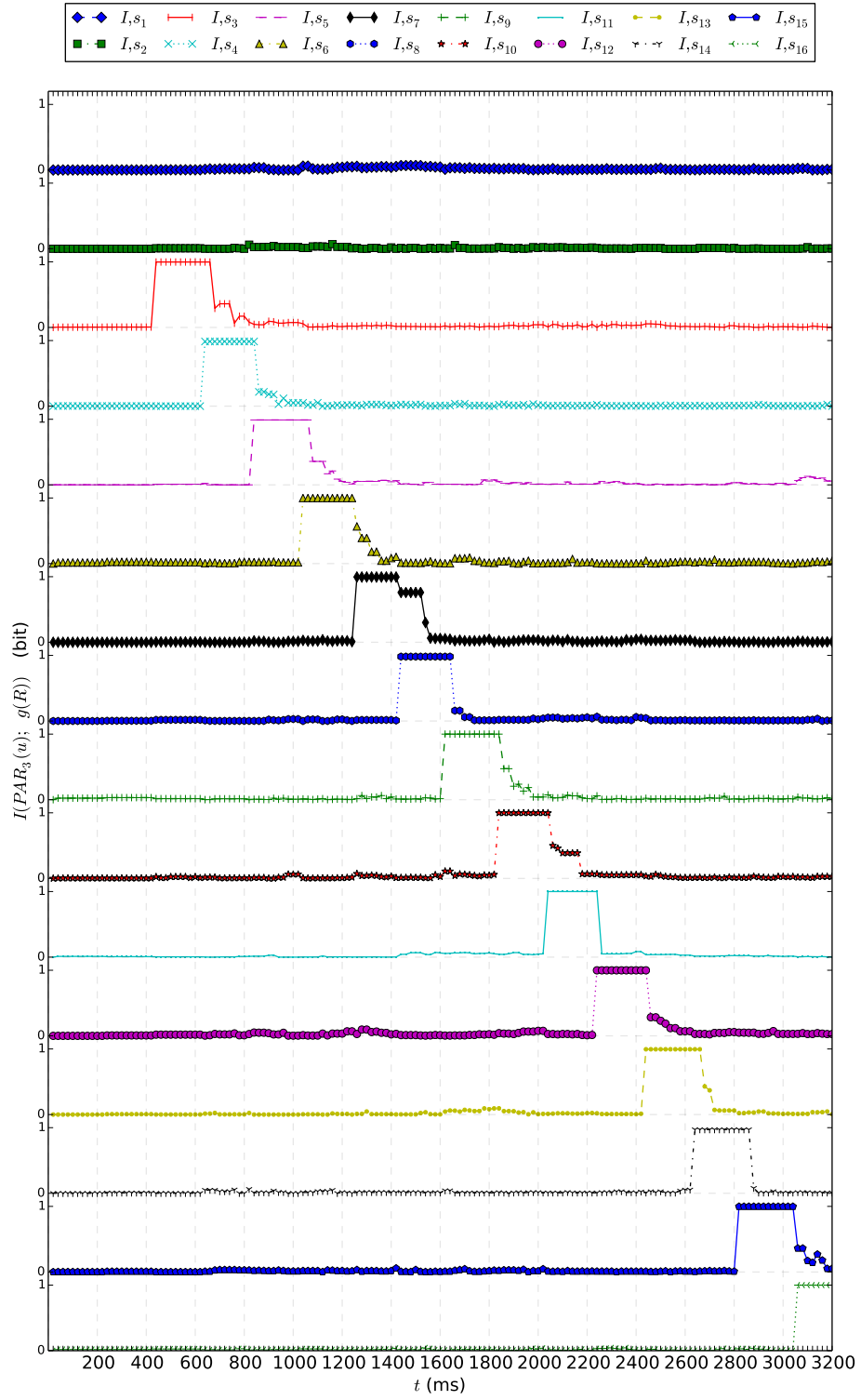


Figure 5.2: Mutual information of the reservoir state and PAR_3 of the input with 16 sections. Reservoir with analog sigmoidal neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and PAR_3 of section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.5$, E) $w = 2.0$. By increasing w up to $w = 1.0$, MI is increased. By further increase in w , the MI starts dropping from $t = 1200ms$ on. The system does not represent the input anymore; therefore PAR_3 cannot be extracted from the state of the reservoir after the critical $w = 1$.

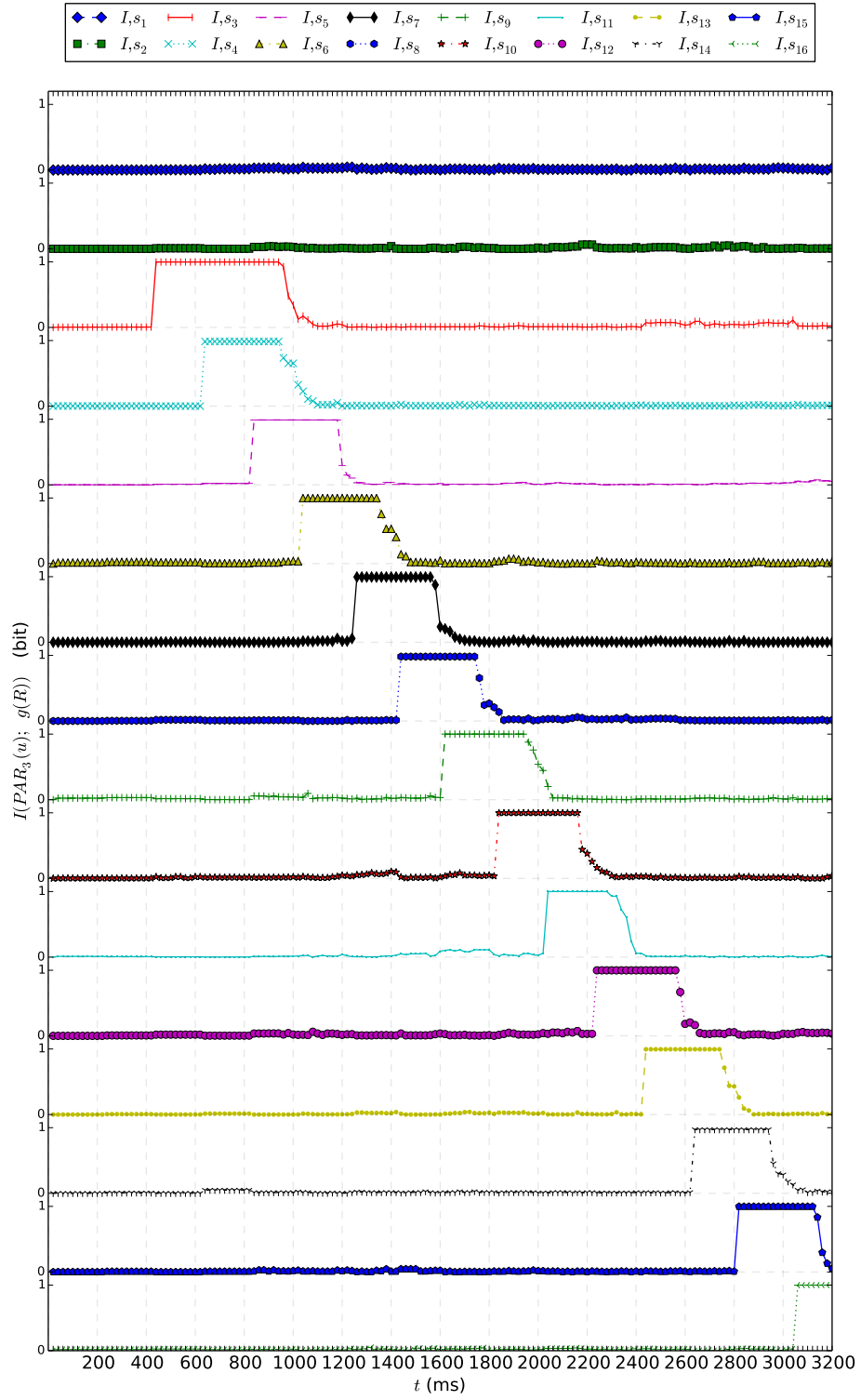
A



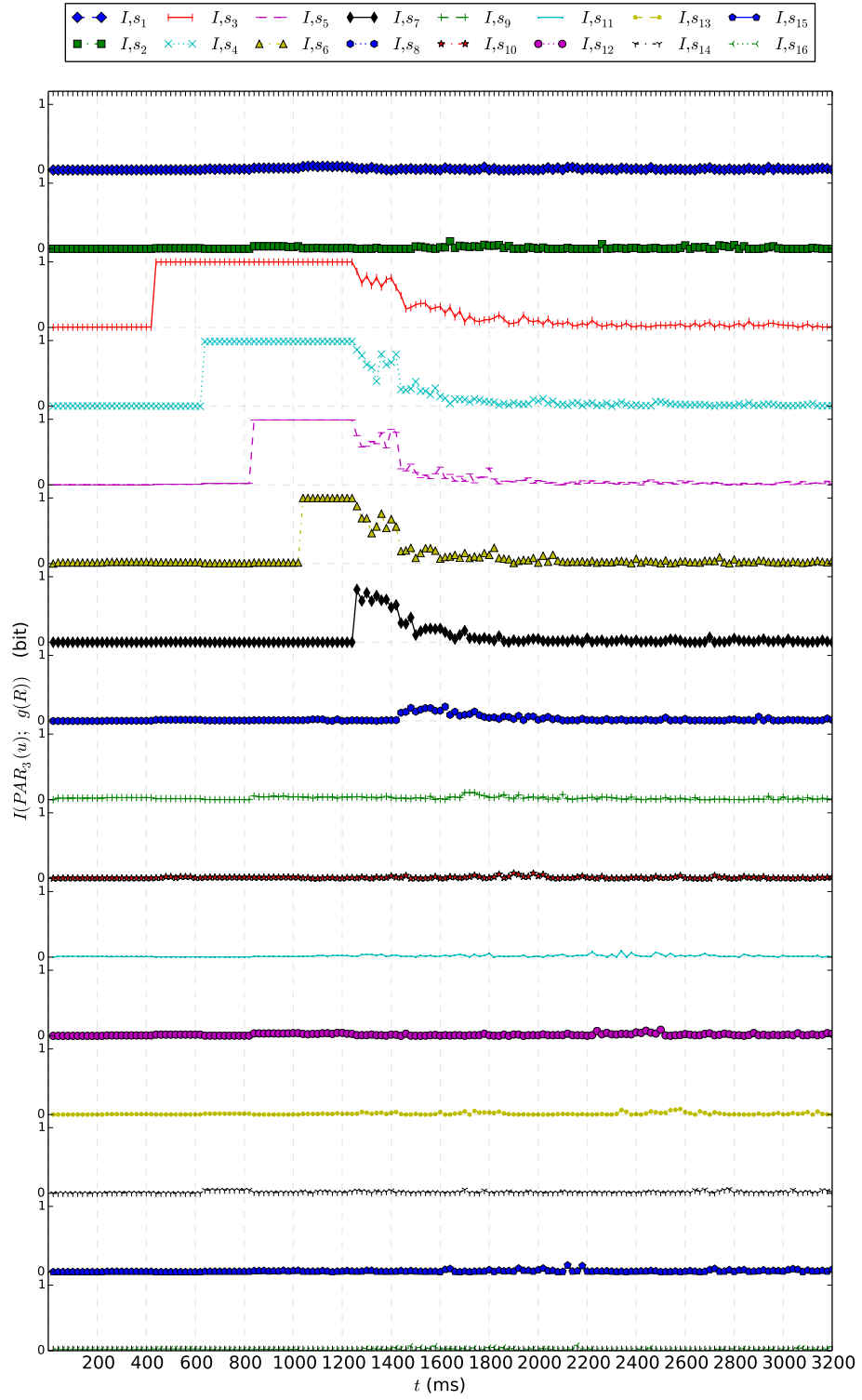
B



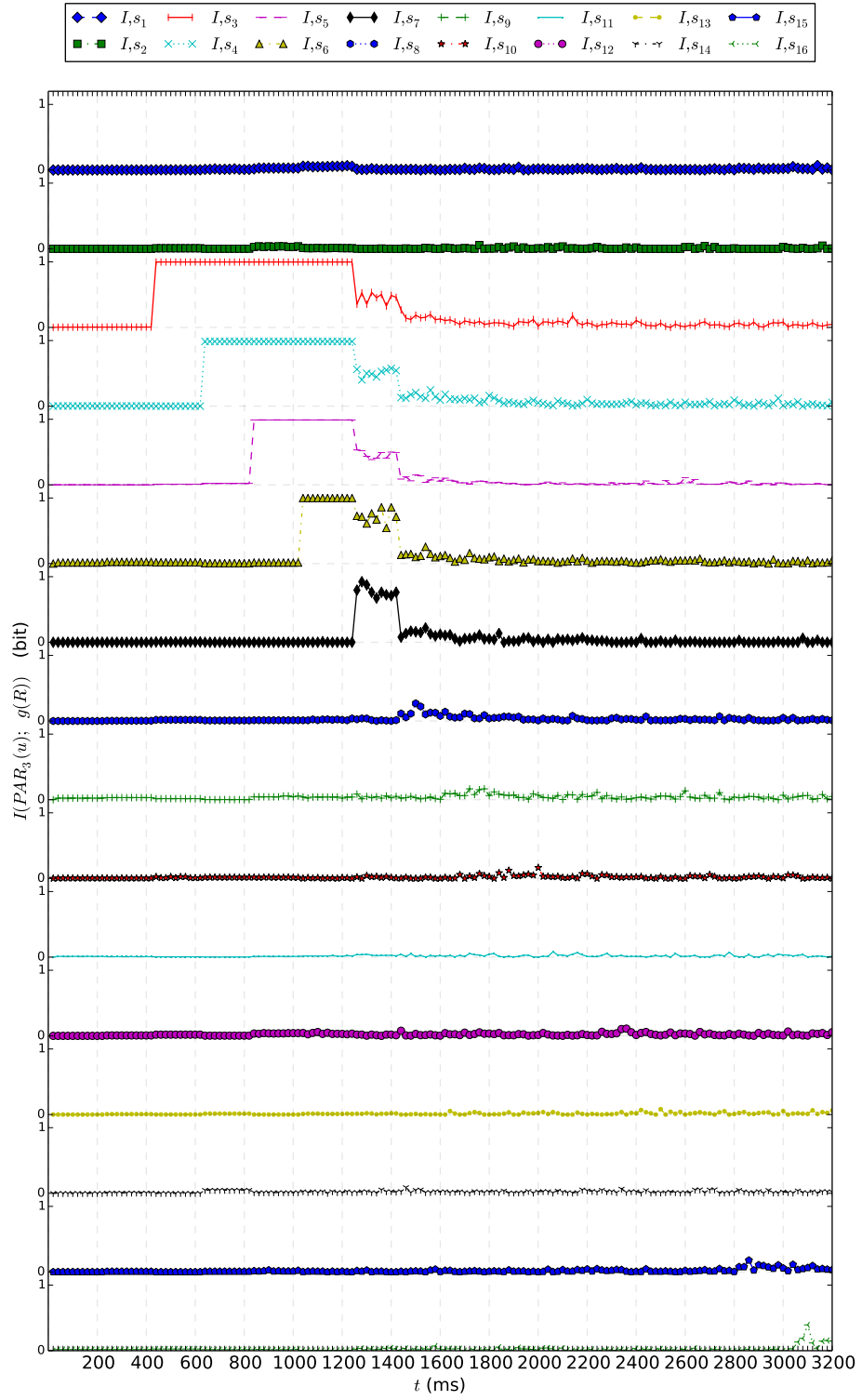
C



D



E



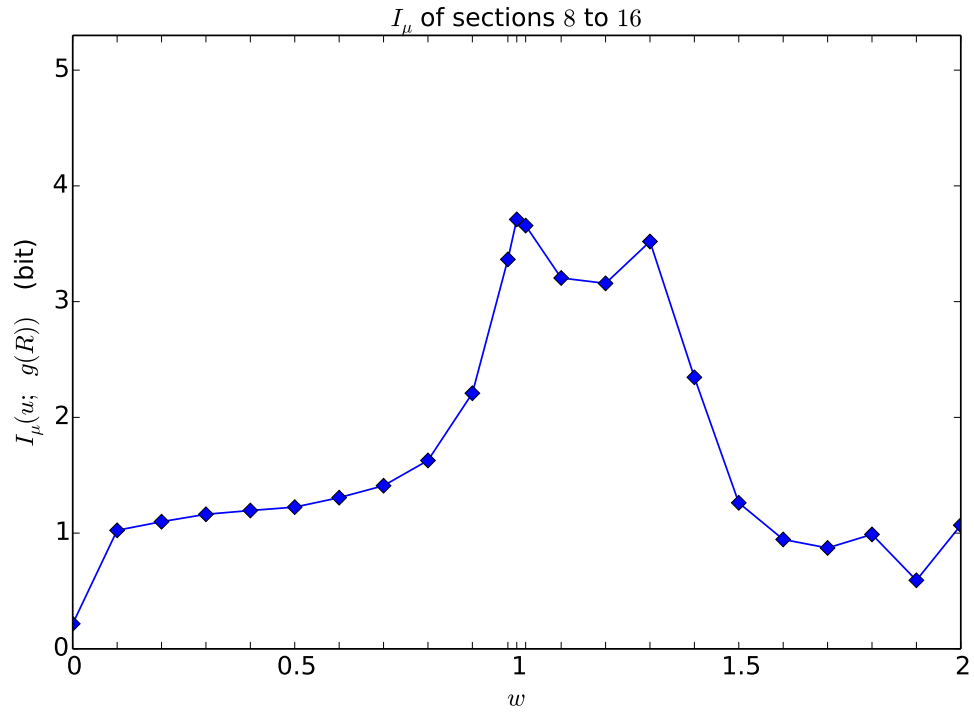


Figure 5.3: A) I_μ of the reservoir with analog sigmoidal neurons, obtained from $I(u; g(R))$ results, as shown in figure 5.1. The reservoir size is $N = 100$, and input has 16 sections. I_μ for MI is increased by increasing the w up to critical $w = 1$. With further increase in w , the I_μ starts to drop. Meaning, the consistency of the system in representing the input and output is decreased.

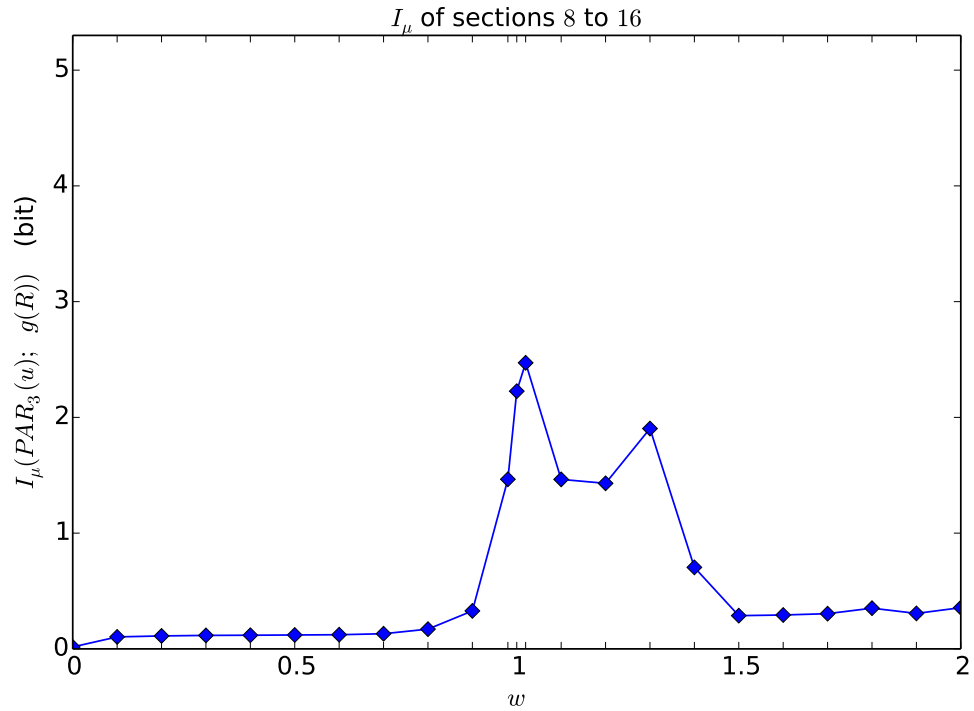


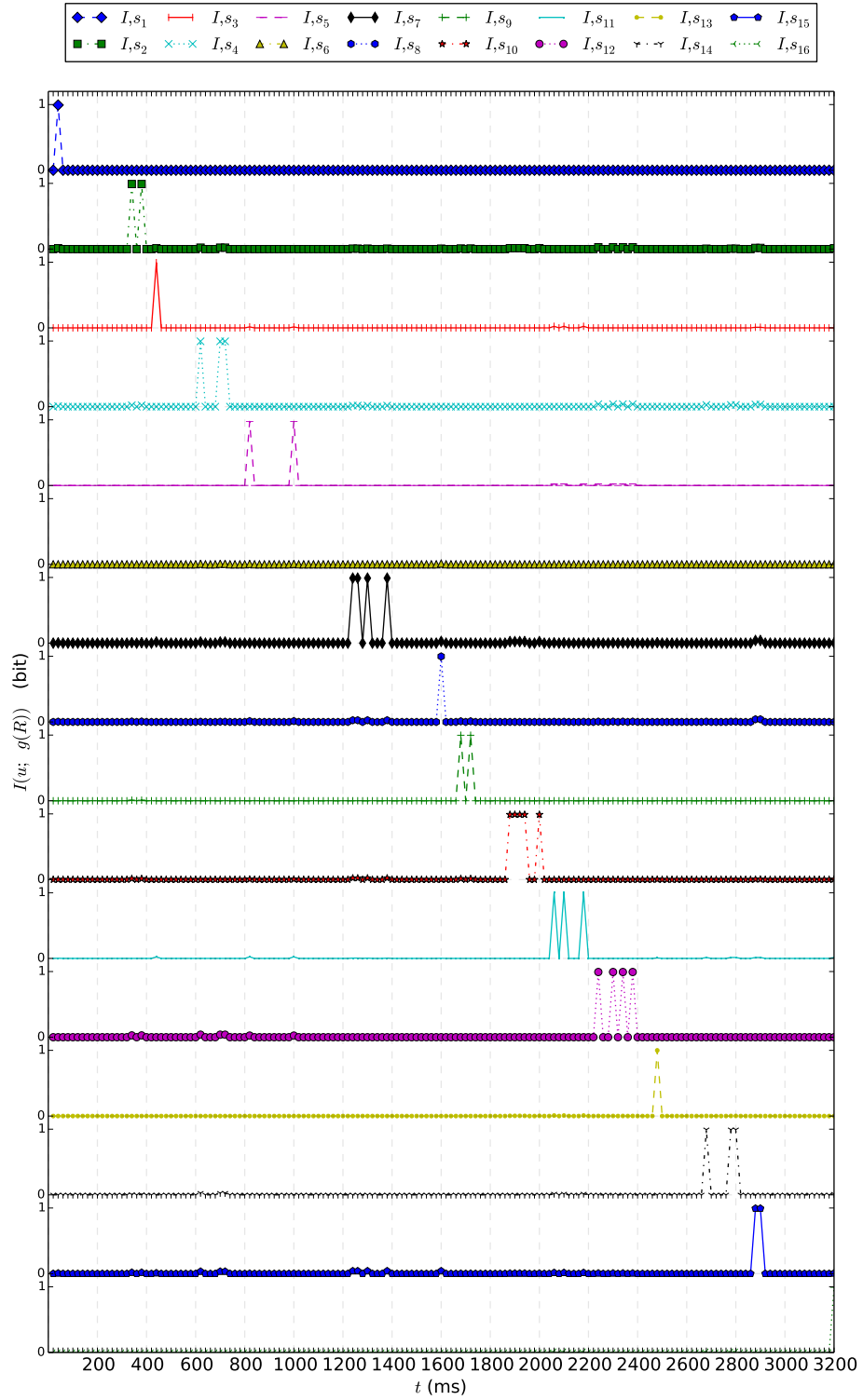
Figure 5.4: I_μ of the reservoir with analog sigmoidal neurons, obtained from $I(PAR_3(u); g(R))$ results, as shown in figure 5.2. The reservoir size is $N = 100$, and input has 16 sections. $I_\mu(PAR_3(u); g(R))$ also indicates the performance of the reservoir in solving PAR_3 over time. The rise and fall of $I_\mu(PAR_3(u); g(R))$ follows that of $I_\mu(u; g(R))$. That is to say, the performance of the reservoir in solving PAR_3 is highly dependent on the amount of stable memory in the reservoir.

5.1.2 MI Results for the Reservoir with Leaky Integrator Neurons

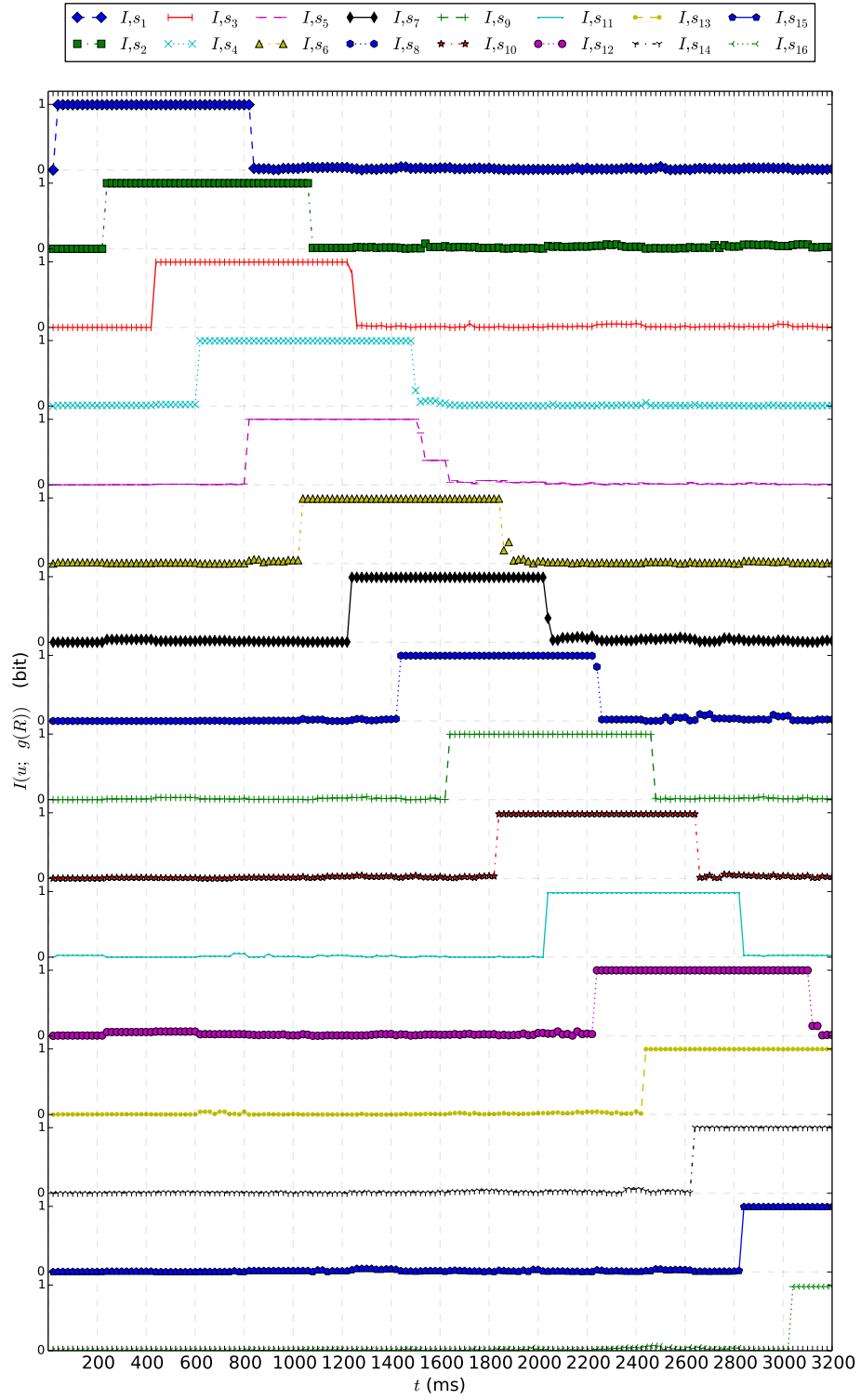
Figure 5.5 shows the MI between the reservoir state and the input. Figure 5.6 shows the MI between the reservoir state and the PAR_3 of the input for a reservoir with LI neurons. Our MI results for the reservoir with LI neurons are very similar to that of the reservoir with sigmoidal neurons (shown in figures 5.1 and 5.2) in the sense that the memory measured by MI is maximized when w is close to a critical value of 1 and decreased as w is increased and $w > 1$. The MI between the reservoir state and PAR_3 also follows this rule. Figure 5.7 and 5.8 help us visualize of the behavior by showing the I_μ calculated by equation 4.8 over the same MI results, as presented in figures 5.5 and 5.6.

Figure 5.5: Mutual information of the reservoir state and the input with 16 sections. Reservoir with analog LI neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.3$, E) $w = 2.0$. By increasing w , up to critical $w = 1.0$, MI is increased. By further increase in w , the MI starts dropping from $t = 1200ms$ on. This means that, the system no longer represents the input after the critical $w = 1$.

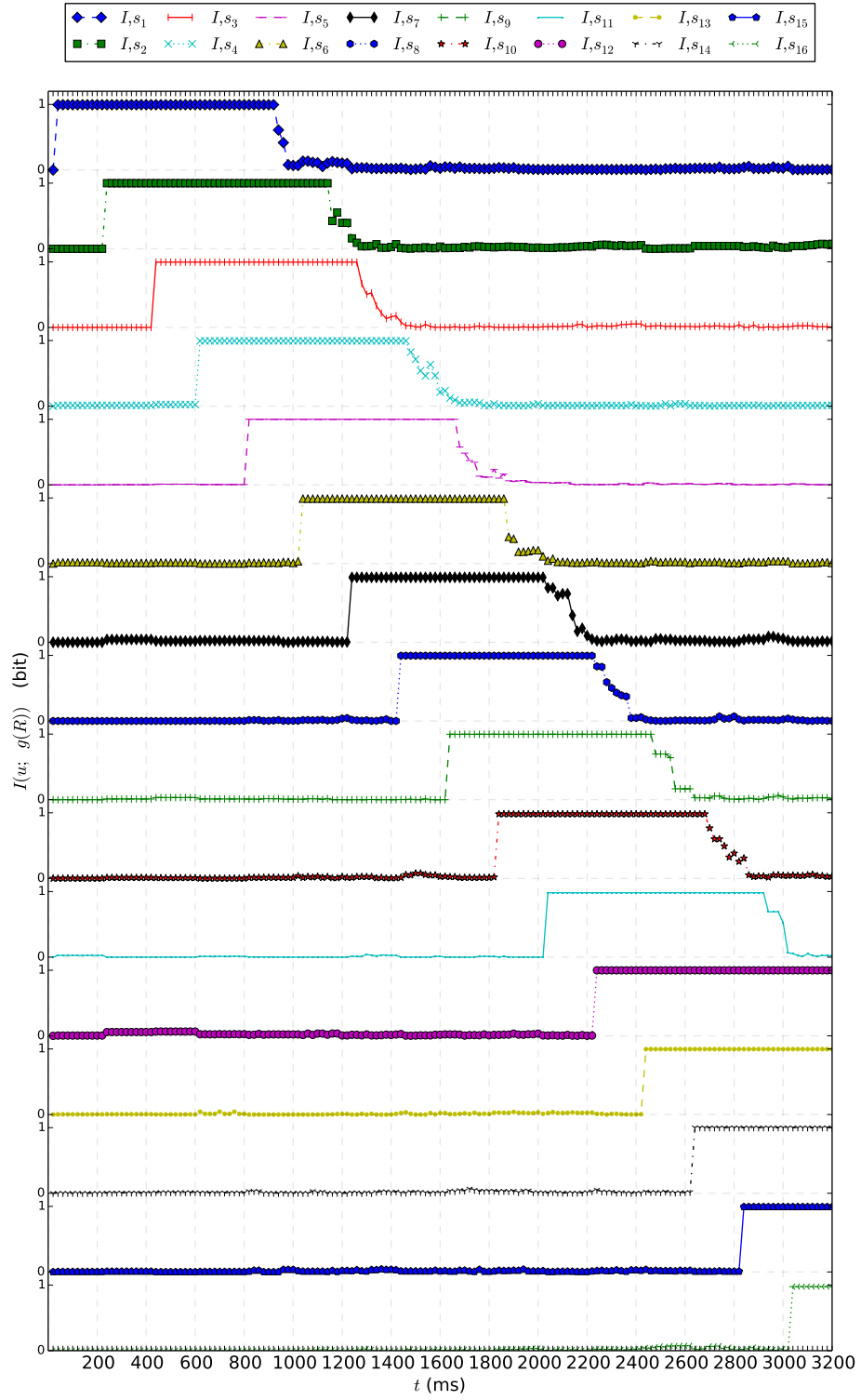
A



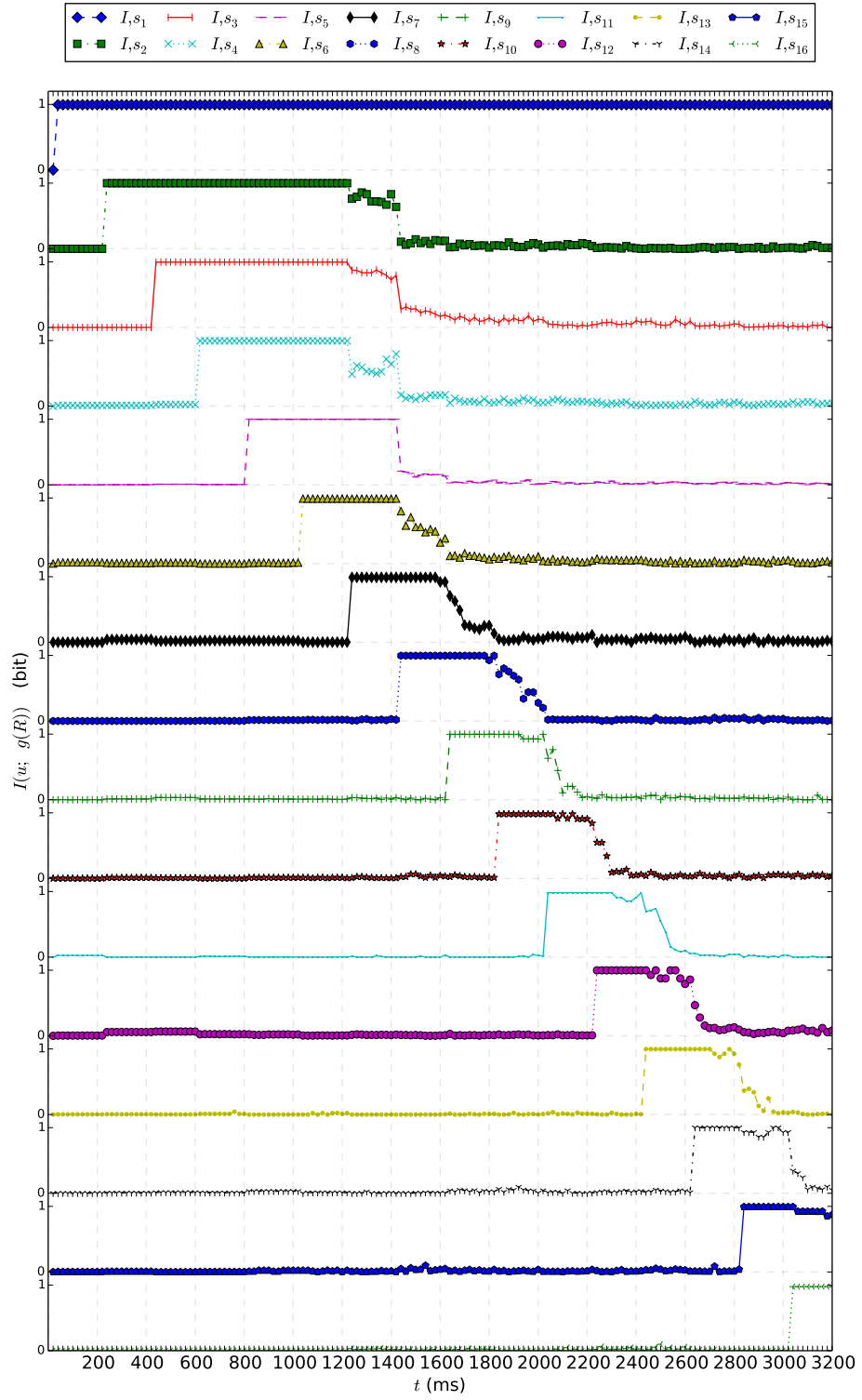
B



C



D



E

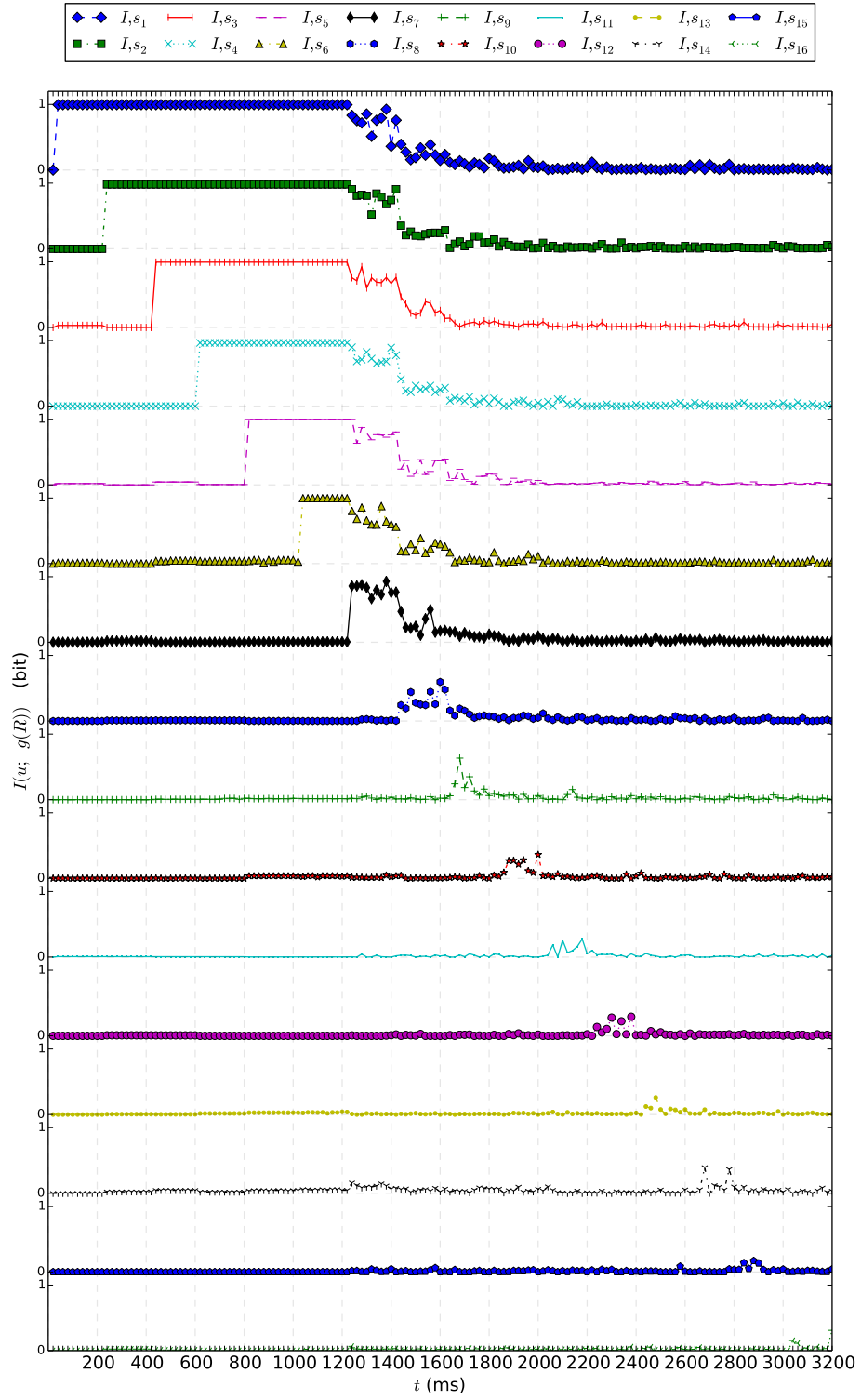
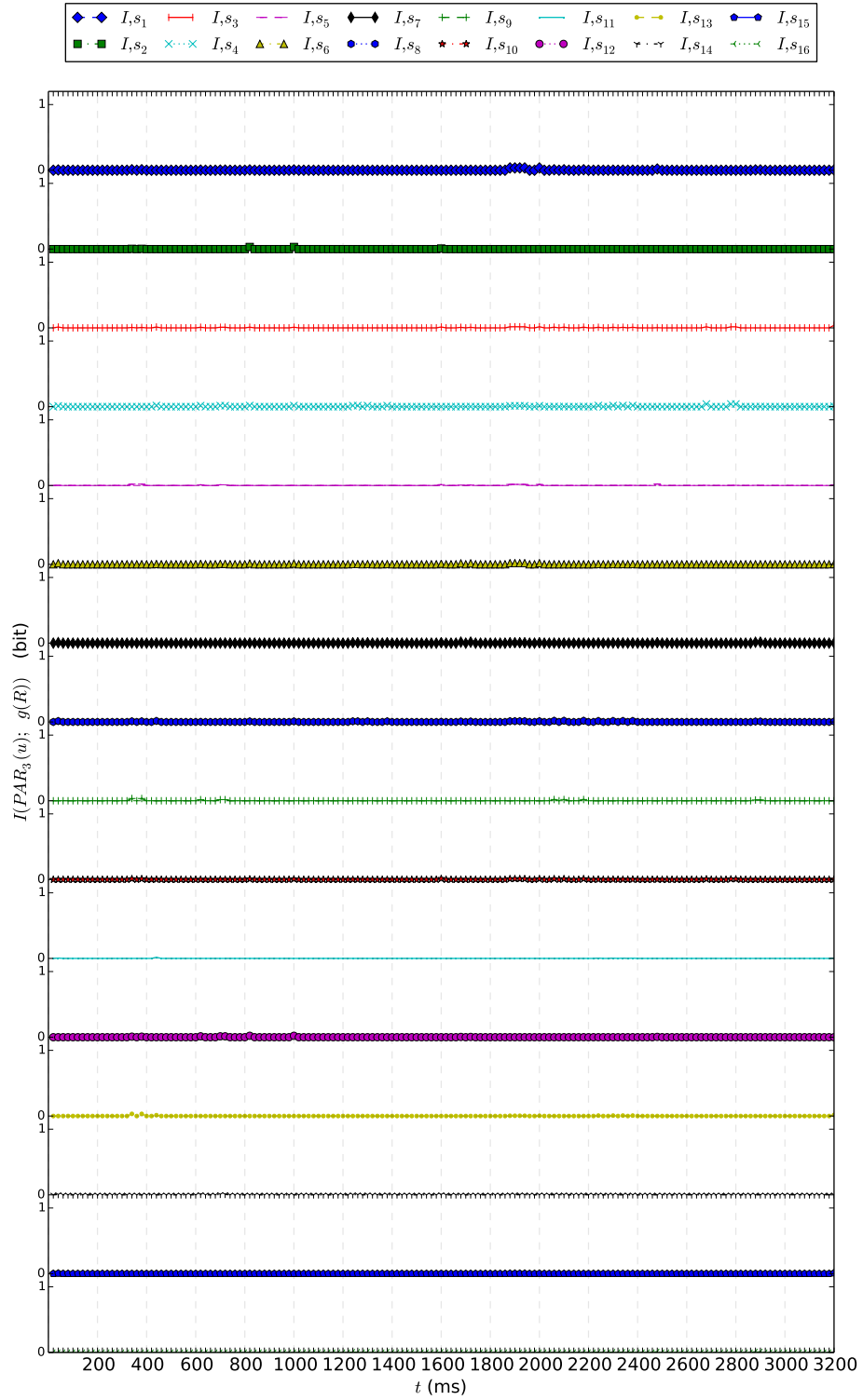
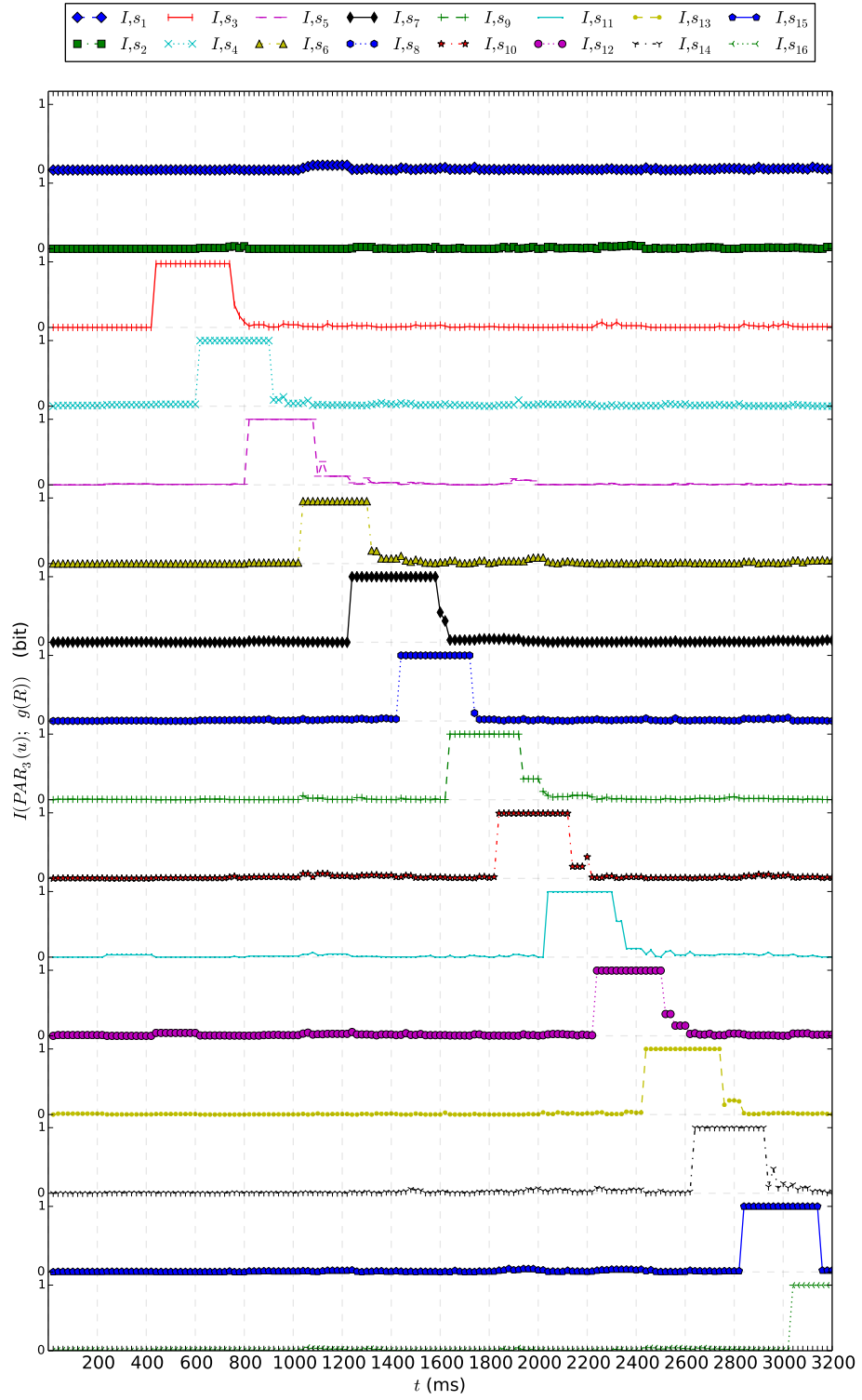


Figure 5.6: Mutual information of the reservoir state and PAR_3 of the input with 16 sections. Reservoir with analog LI neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and PAR_3 of section s_i over time. A) $w = 0$, B) $w = 0.95$, C) $w = 1.0$, D) $w = 1.3$, E) $w = 2.0$. By increasing w up to critical $w = 1.0$, MI is increased. With further increase in w , the MI starts to drop from $t = 1200ms$ on. The system does not represent the input anymore; therefore PAR_3 cannot be extracted from the state of the reservoir after section 7 when w is greater than the critical $w = 1$.

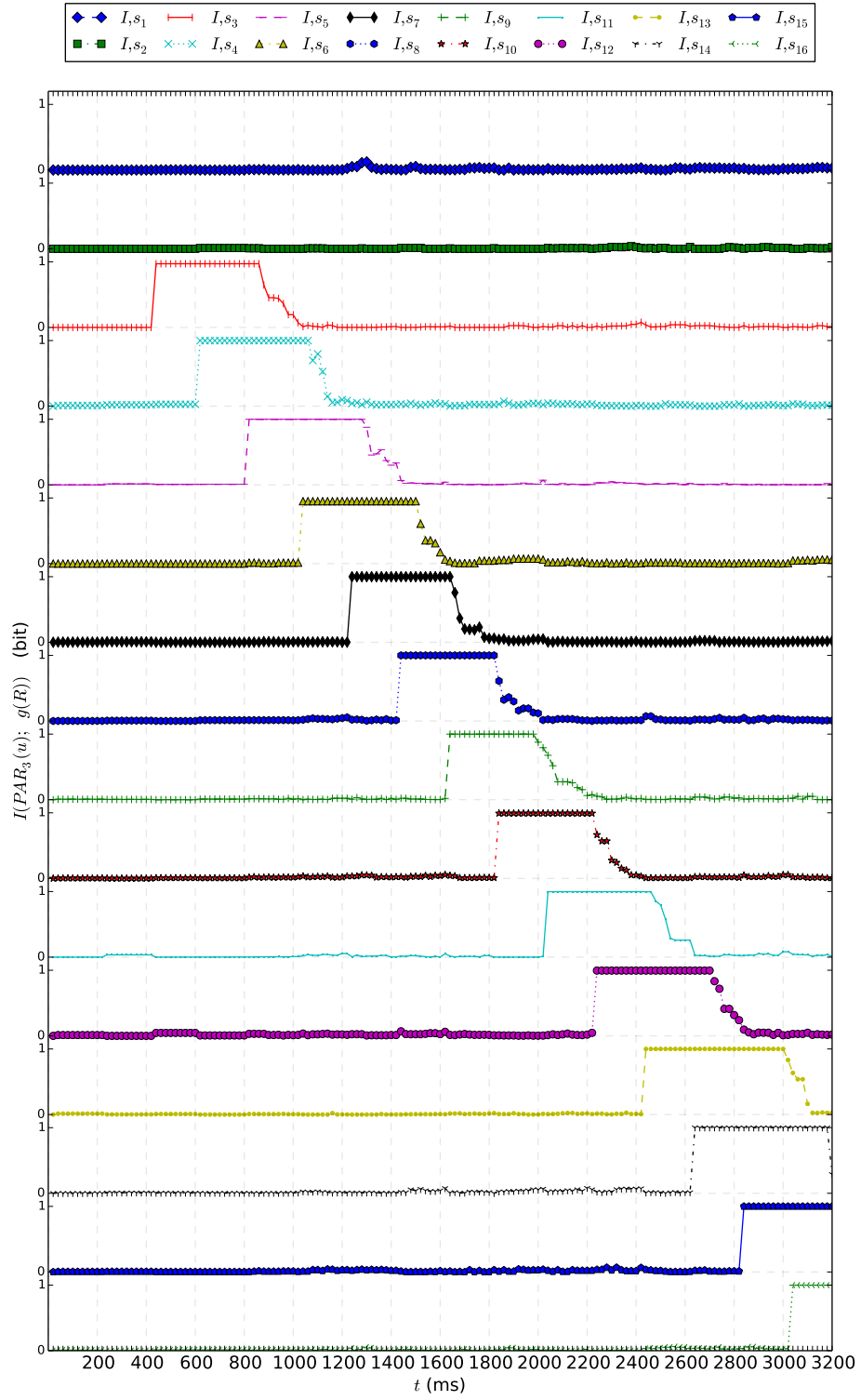
A



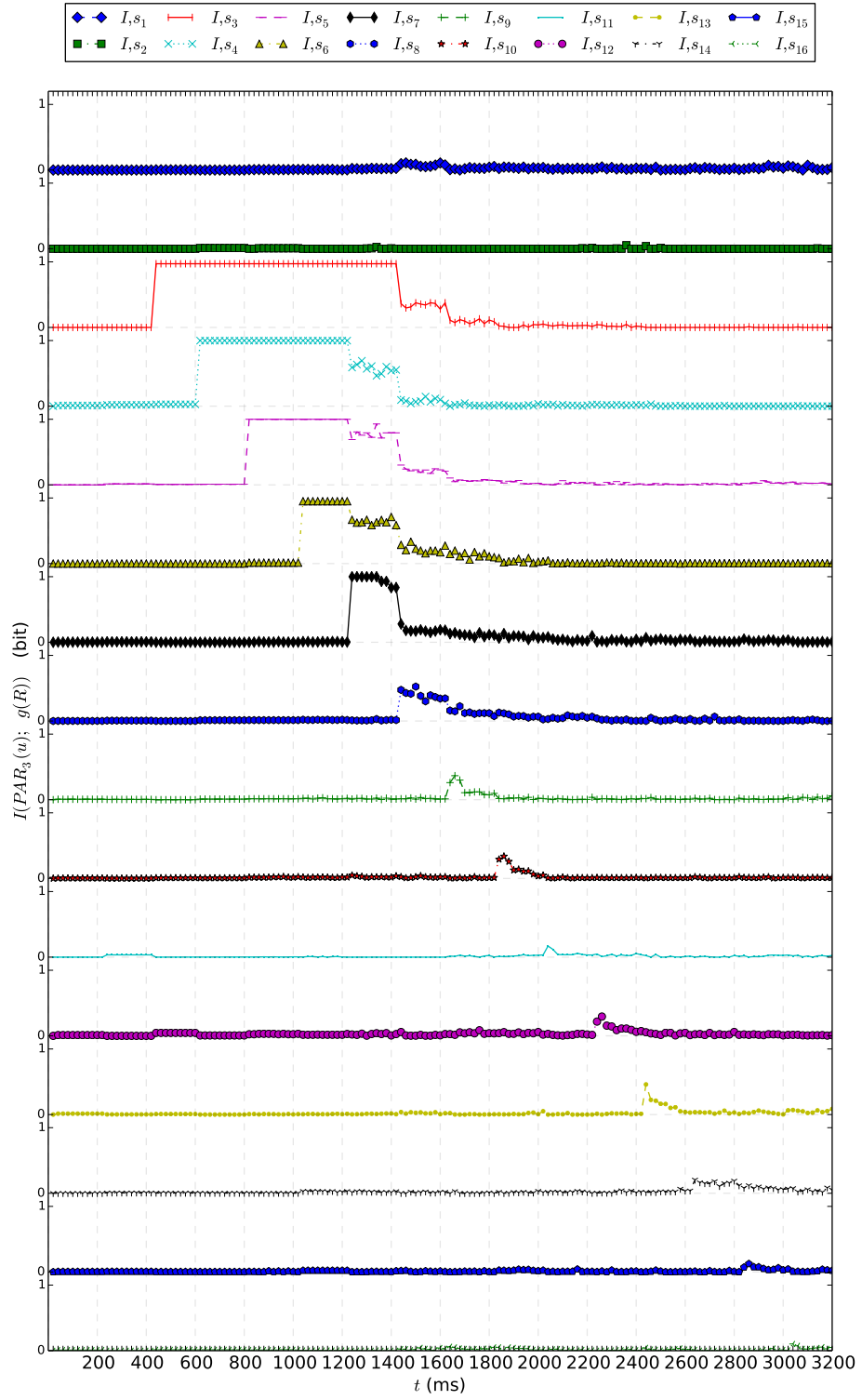
B



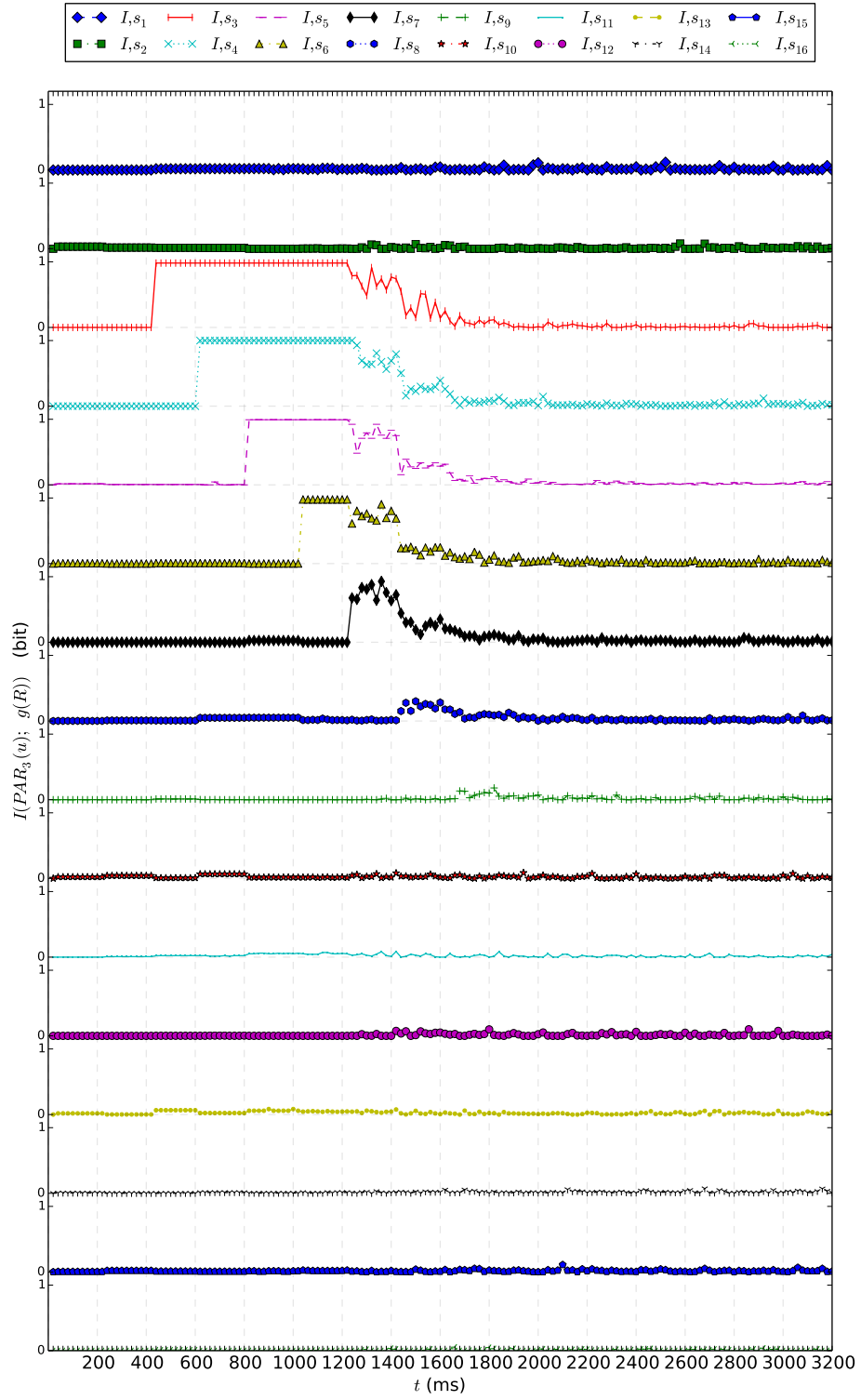
C



D



E



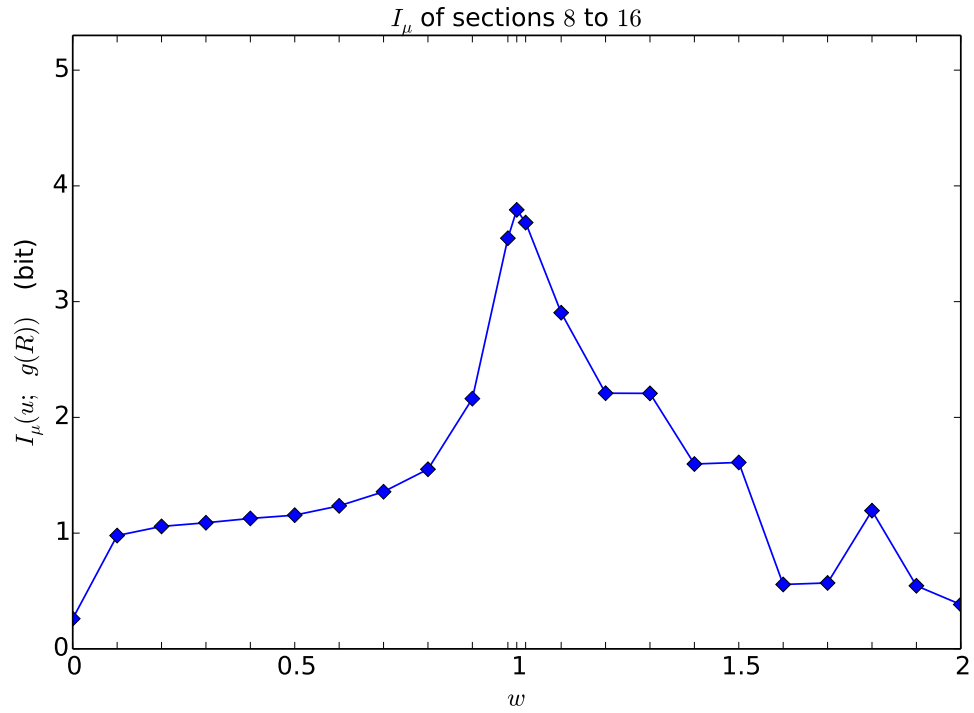


Figure 5.7: A) I_μ of the reservoir with LI neurons, obtained from $I(u; g(R))$ results, shown in figure 5.1. The reservoir size is $N = 100$ and the input has 16 sections. I_μ for MI is increased by increasing w up to the critical $w = 1$. With further increase in w , the I_μ starts to drop. This means that, the consistency of the system in representing the input and output is decreased. The leak rate of the LI neuron is fixed at $a = 0.25$

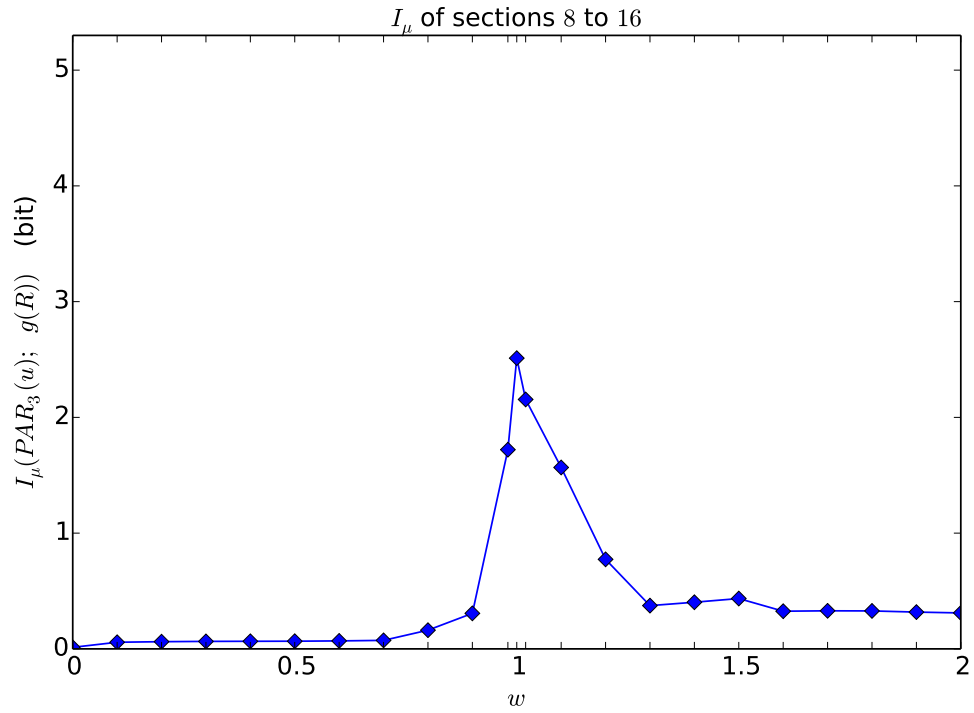


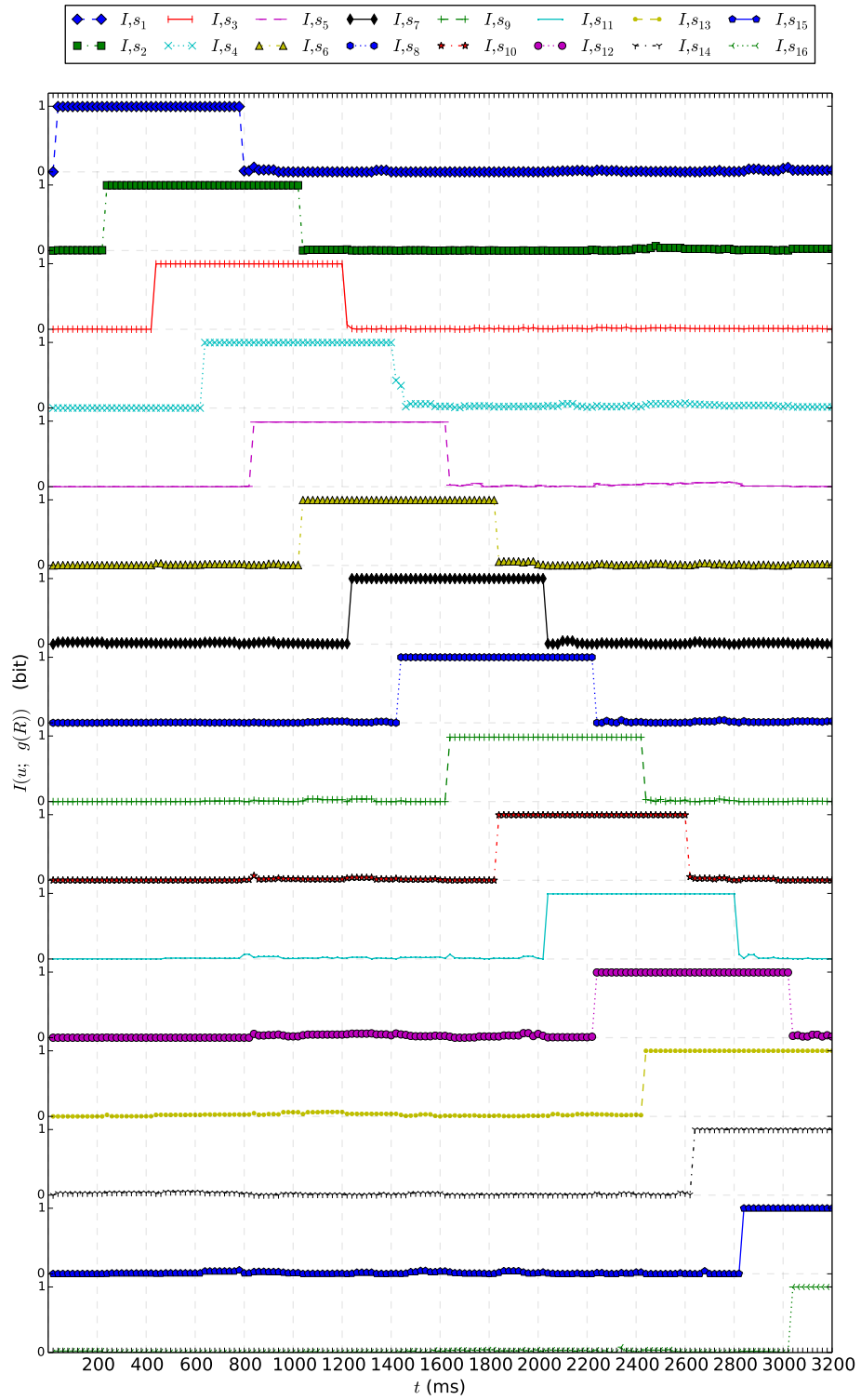
Figure 5.8: I_μ of the reservoir with LI neurons, obtained from $I(PAR_3(u); g(R))$ results, shown in figure 5.6. The reservoir size is $N = 100$ and the input has 16 sections. $I(PAR_3(u); g(R))$ also indicates the performance of the reservoir in solving PAR_3 over time. The rise and fall of $I(PAR_3(u); g(R))$ follows that of $I(u; g(R))$. That is to say, the performance of the reservoir in solving PAR_3 is highly dependent on the amount of stable memory in the reservoir.

5.1.3 MI Results for the Reservoir with Leaky Integrate-and-Fire Neurons

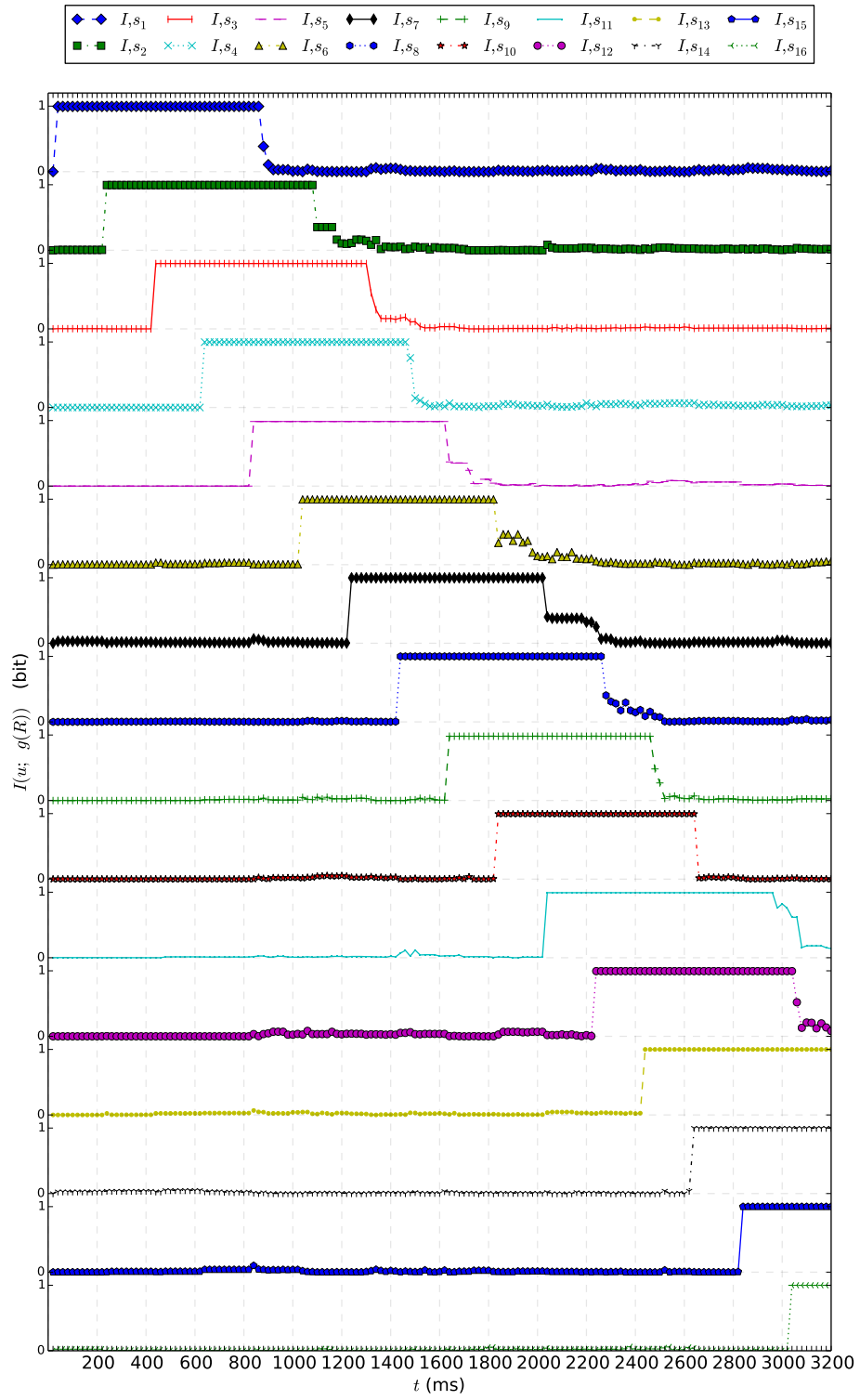
MI results between the reservoir state and the input for reservoir with LIF neurons when $\tau_{syn} = 10$ are presented in figure 5.9. When $w = 0$ (A), the memory of the reservoir about the past input sections is considerably higher than it is for analog reservoirs i.e., sigmoidal and LI reservoirs and as w increases, the amount of information about the input is increased (B) until it reaches its maximum level when $w = 3$ (C). The maximum I is no longer at $w = 1$, since LIF dynamics differ from the preceding analog neurons. With further increase in w , the reservoir becomes unable to consistently represent the input, and, like analog non-spiking reservoirs, after section s_i , $i \geq 7$, the memory about the past input sections is decreased (D and E). Figure 5.10 shows the MI between the reservoir state and the PAR_3 of the input for a reservoir with LIF neurons. The system able to compute the $PAR_3(u)$ when $w = 0$ (A). Increasing w results in an increase in the memory in the reservoir (B). It maximizes when $w = 3$ (C). By further increasing the w , the consistency of the reservoir in representing the input is decreased and information about the past sections is again decreased after section 7 compared to the earlier sections of the input (D). Finally the PAR_3 of the input cannot be computed anymore from section 7 onward with larger w values (E). Figure 5.11 shows I_μ , calculated by equation 4.8, over the MI results shown in figures 5.9. Figure 5.12 shows I_μ calculated by equation 4.8 over the MI results shown in figures 5.10. The maximum achieved $I_\mu(u; g(R))$ and $I_\mu(PAR_3(u); g(R))$ for LIF reservoir turns out to be higher than the measured I_μ for LI and sigmoidal reservoirs. That is, the critical w for LIF reservoir with $\tau_{syn} = 10$ is 3.

Figure 5.9: Mutual information of the spiking reservoir state and the input with 16 sections. Reservoir with LIF neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and section s_i over time. A) $w = 0$, B) $w = 1.5$, C) $w = 3$, D) $w = 5$, E) $w = 7$. By increasing w , up to critical $w = 3$, MI is increased and by further increase in w , the MI starts dropping from $t = 1200ms$ on. Meaning, the system does not represent the input anymore.

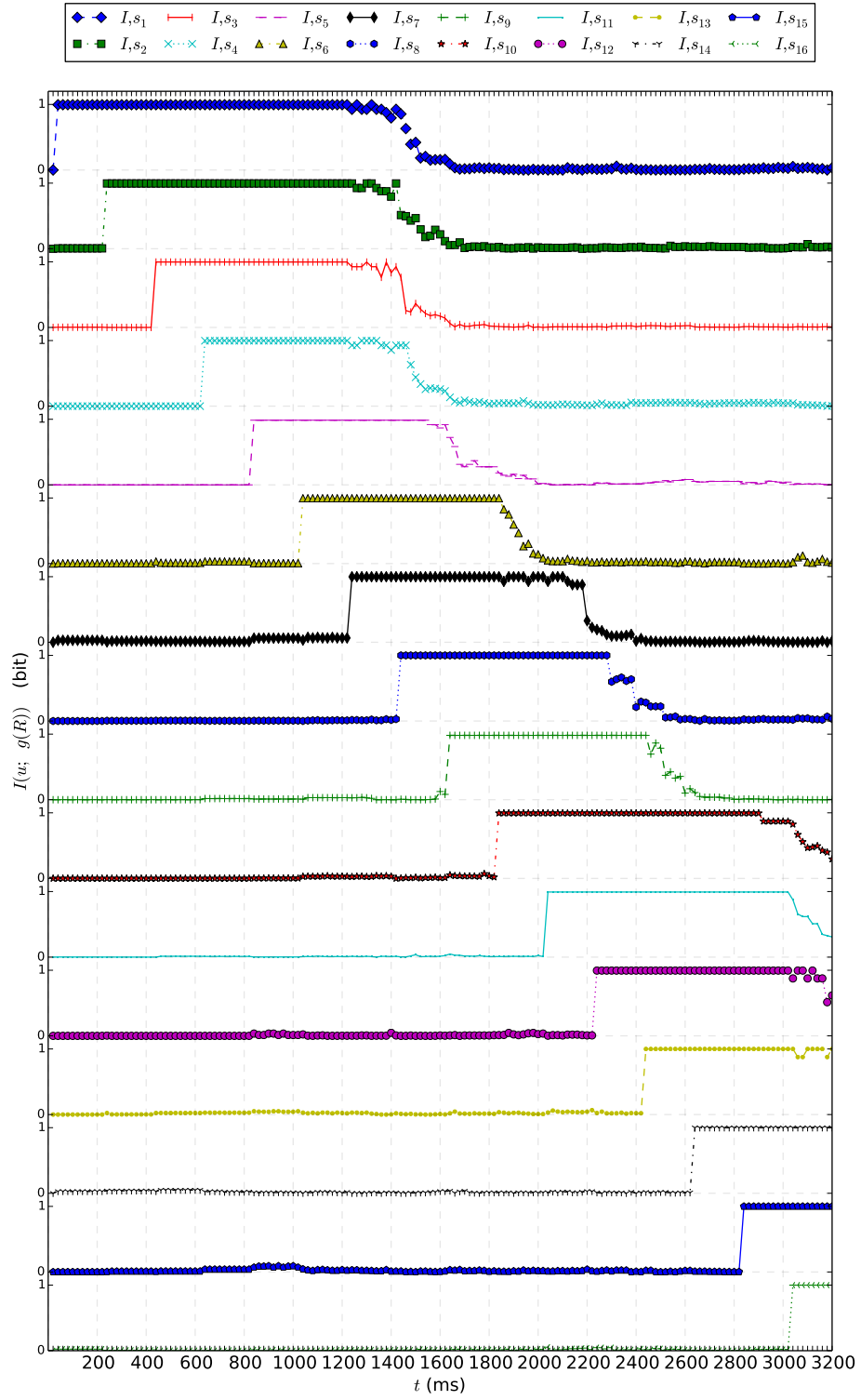
A



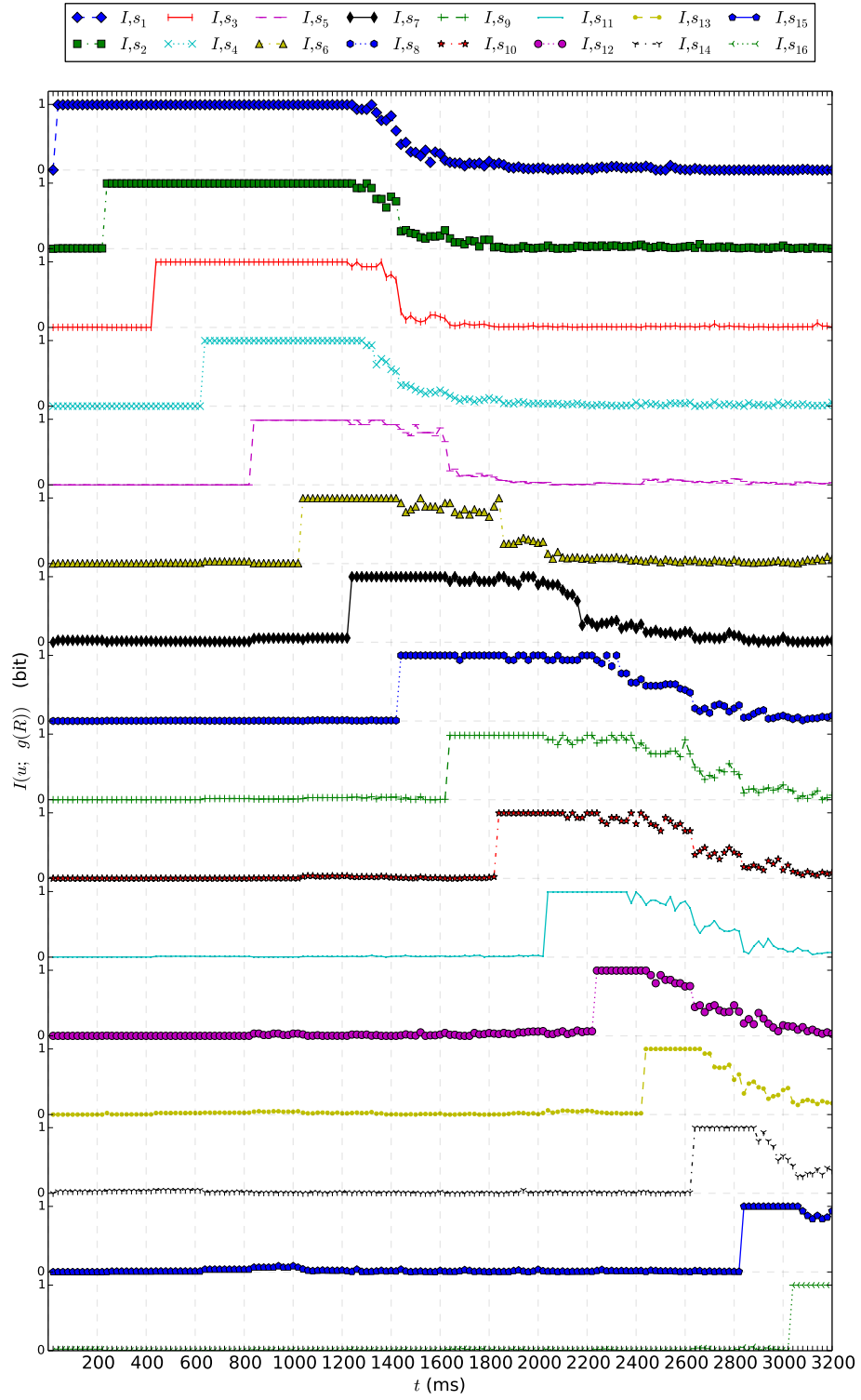
B



C



D



E

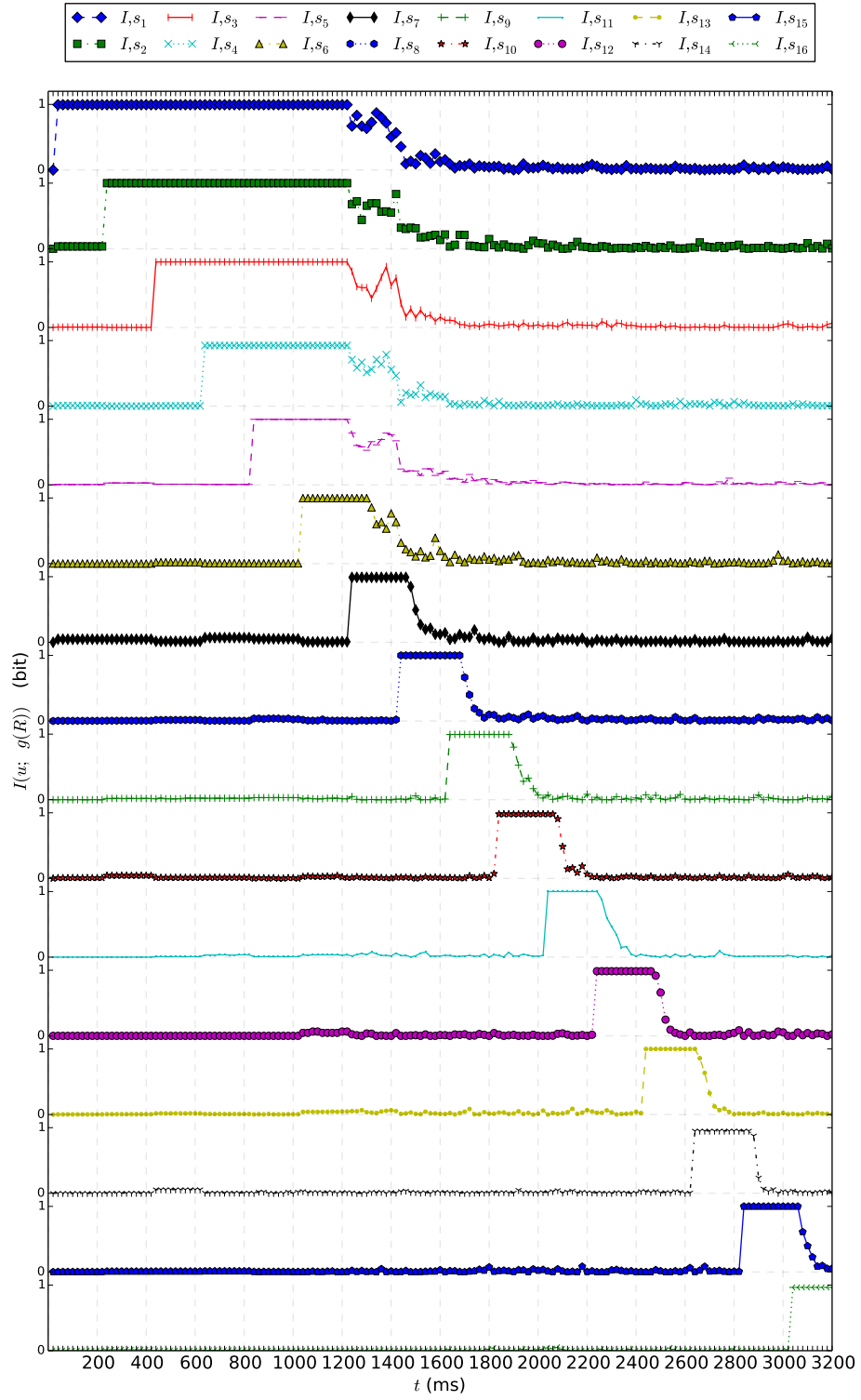
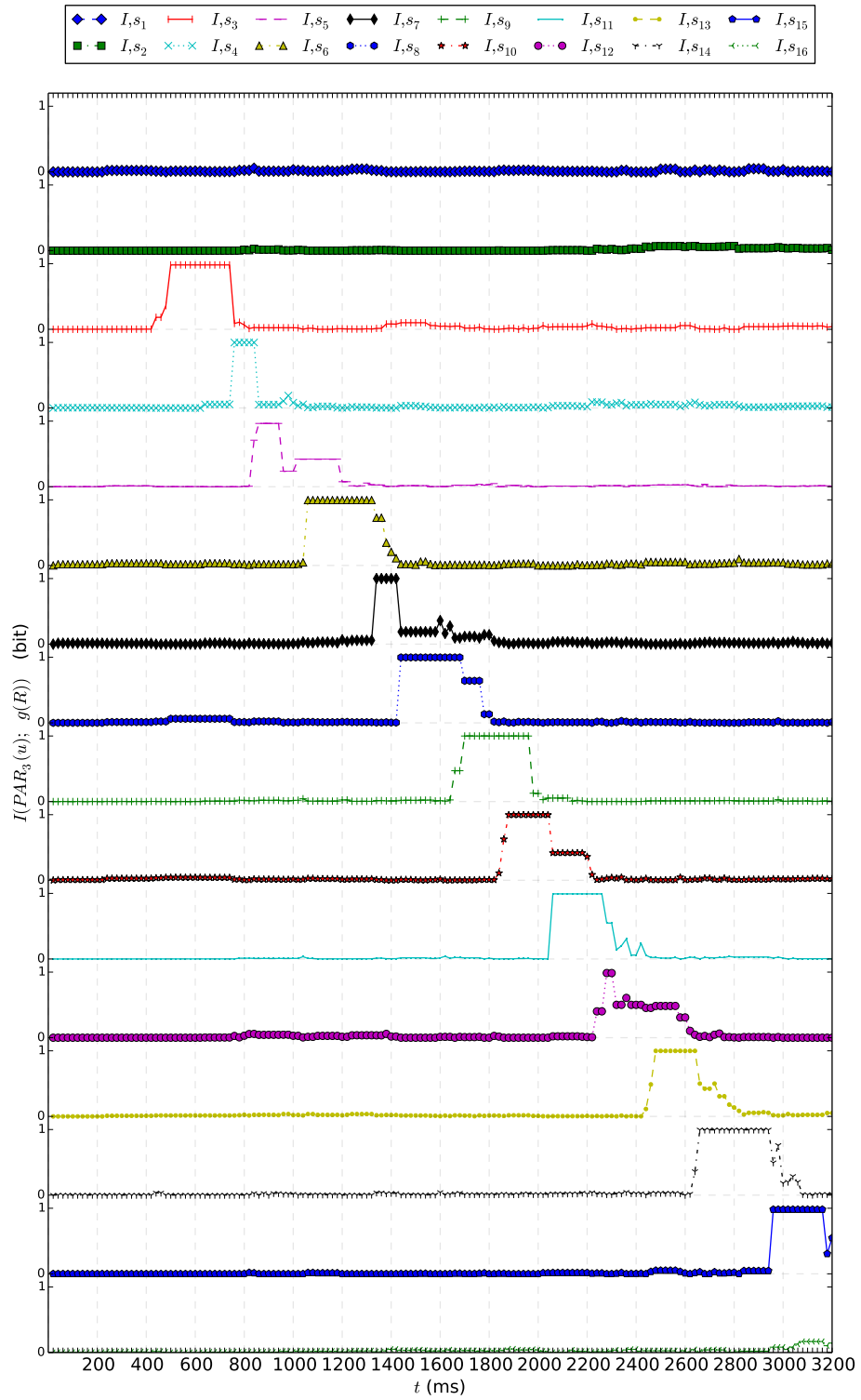
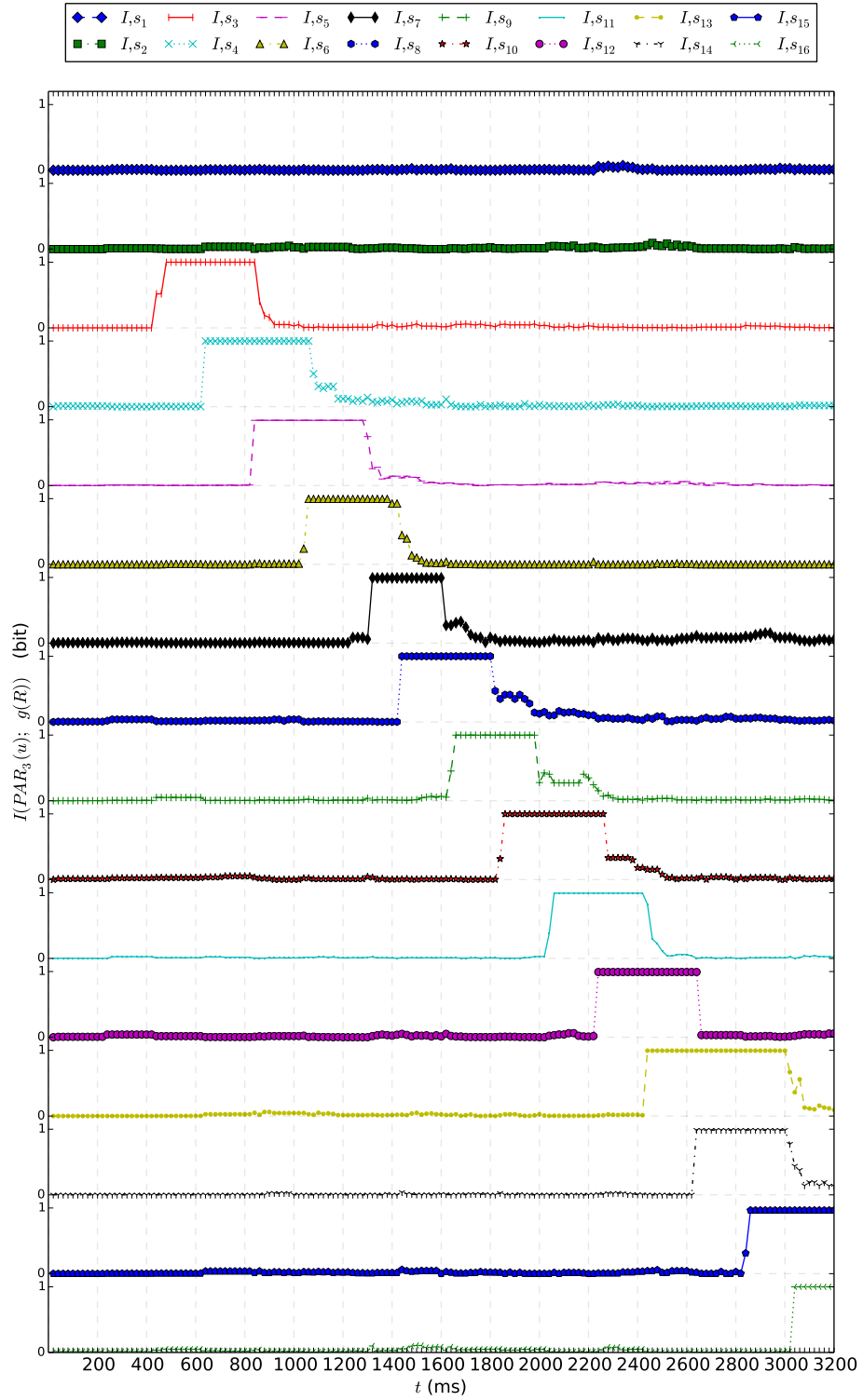


Figure 5.10: Mutual information of the spiking reservoir state and PAR_3 of the input with 16 sections. Reservoir with analog LIF neurons, $N = 100$. The figures follow in the next pages. Each line represents the MI between the reservoir state and PAR_3 of section s_i over time. A) $w = 0$, B) $w = 1.5$, C) $w = 3$, D) $w = 5$, E) $w = 7$. By increasing w up to critical $w = 3$, MI is increased. With further increase in w , the MI starts to drop from $t = 1200ms$ on. The system does not represent the input anymore; therefore PAR_3 cannot be extracted from the state of the reservoir after section 7.

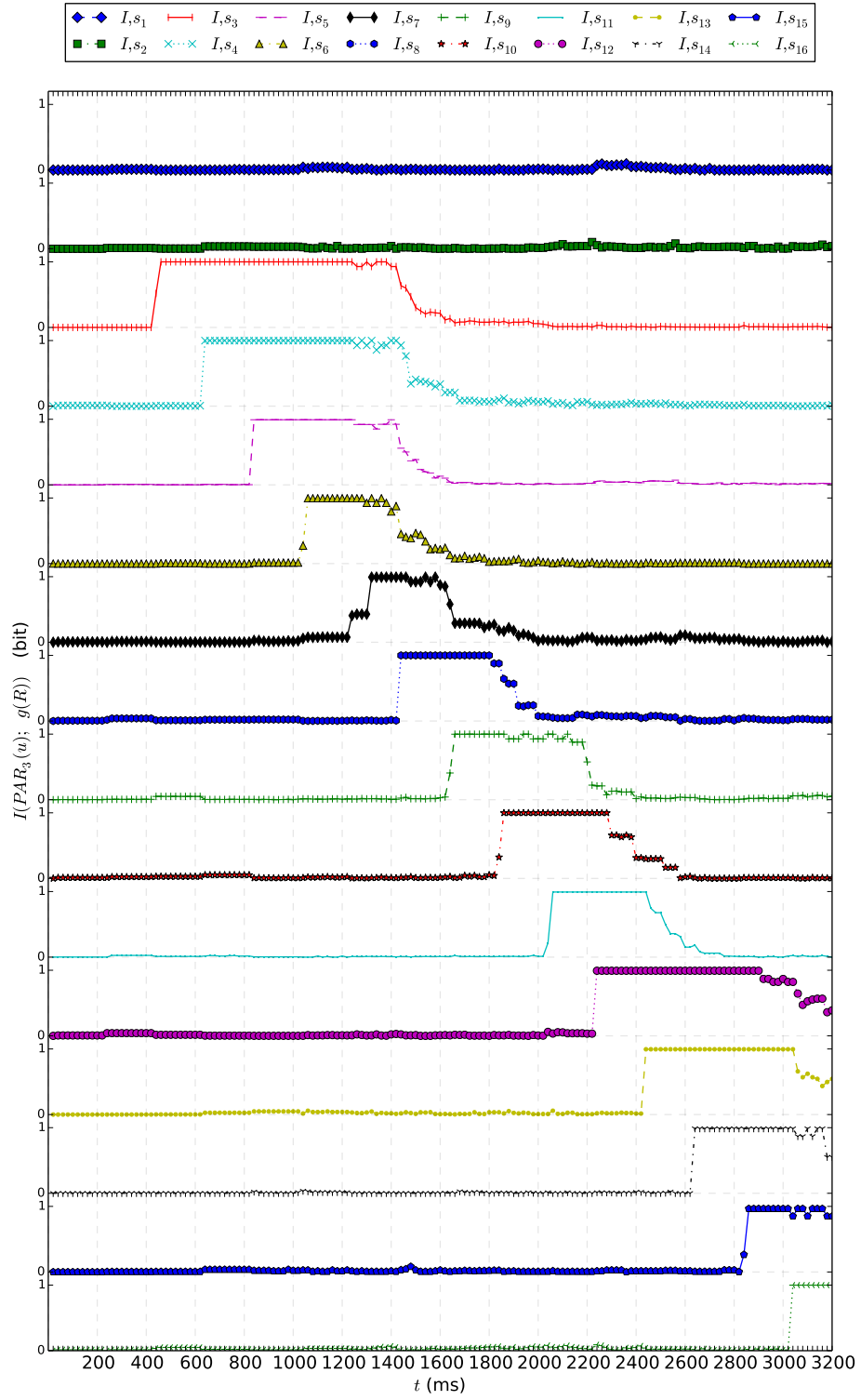
A



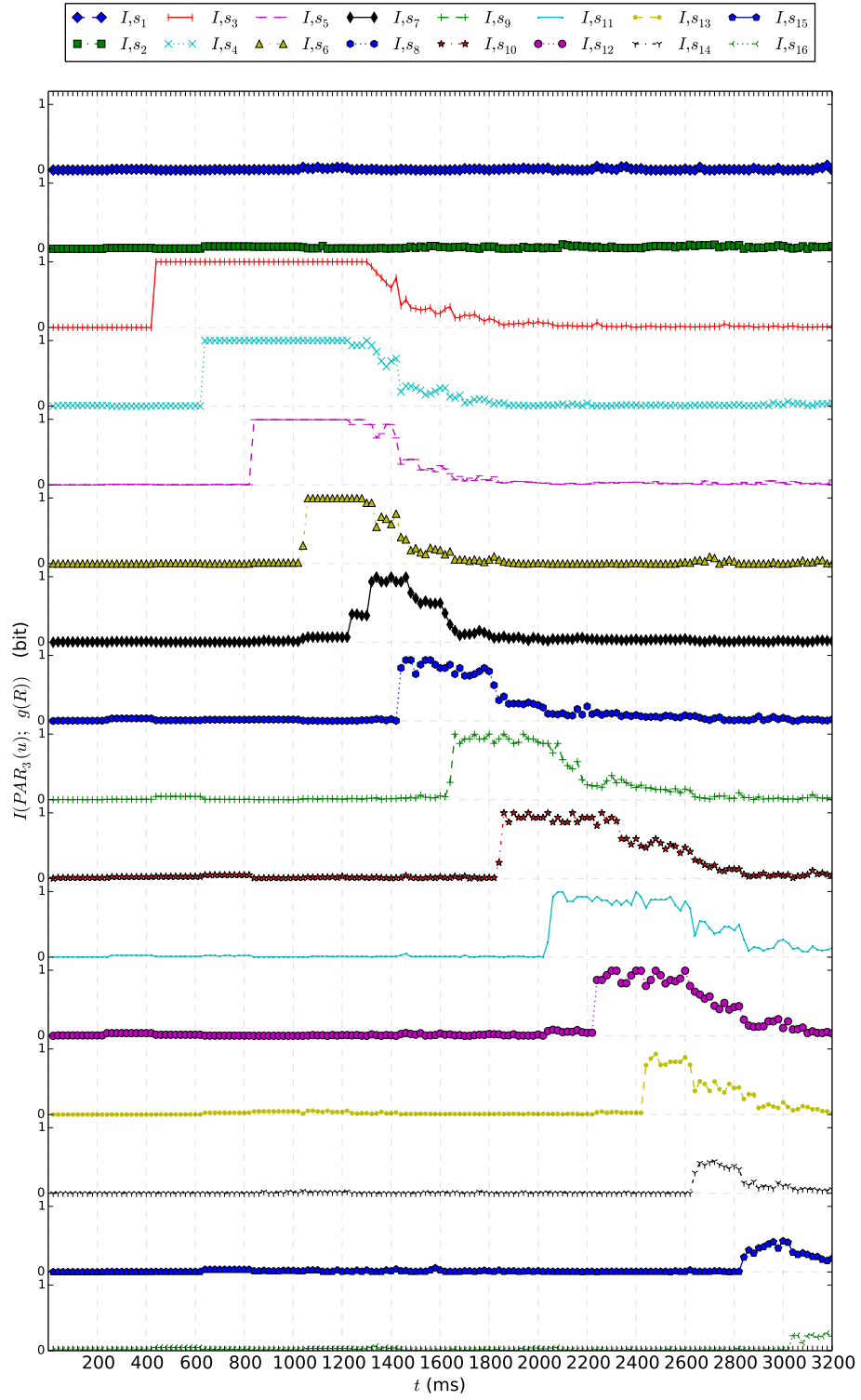
B



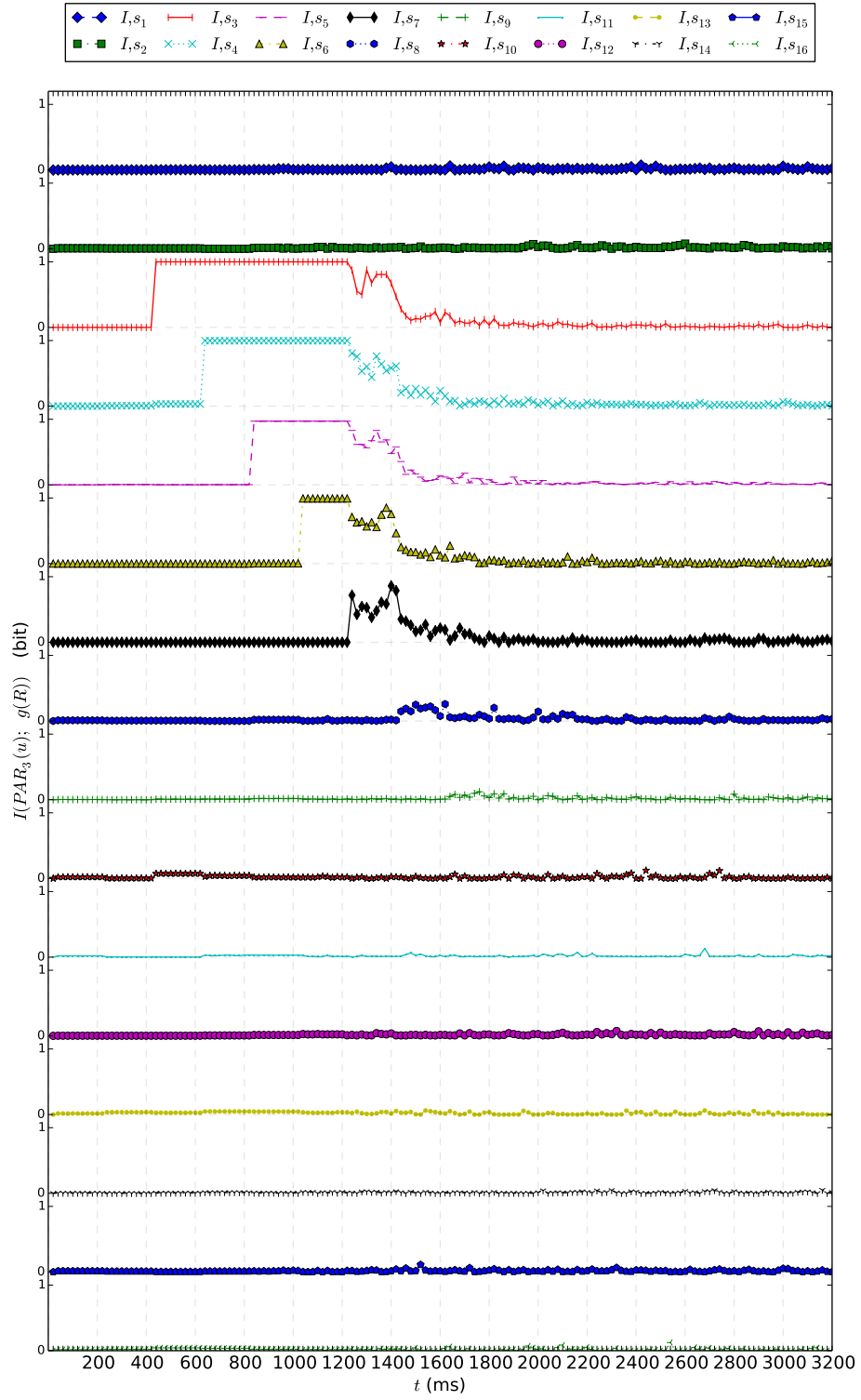
C



D



E



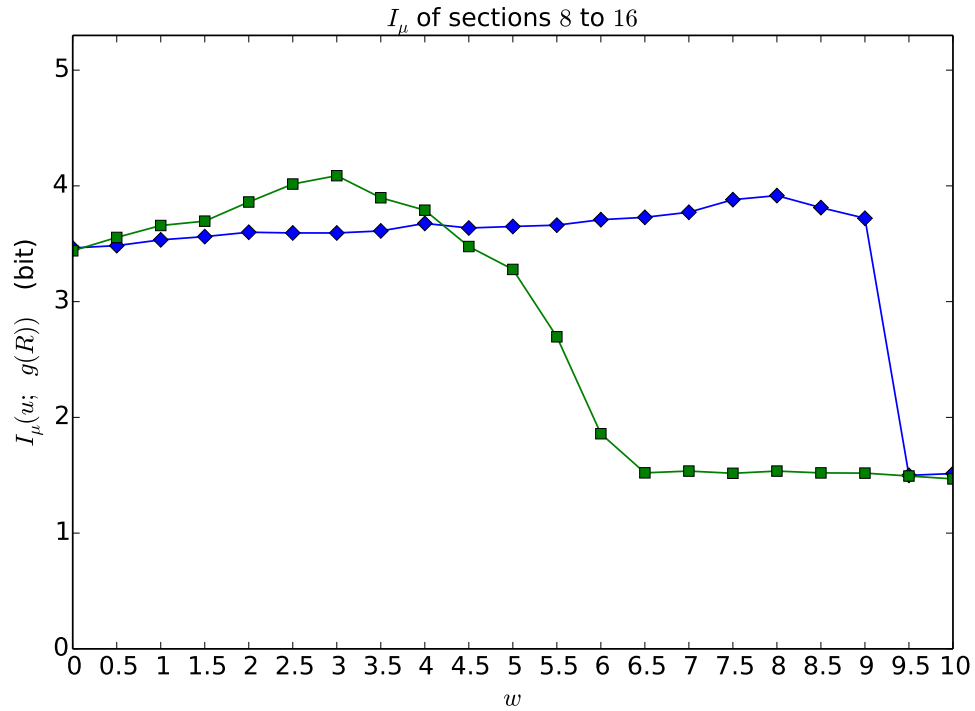


Figure 5.11: $I_\mu(u; g(R))$ of the reservoir with LIF neurons with two different time constants $\tau_m = 7$, and $\tau_m = 10$. Only the I_μ for $\tau_m = 10$ shown here by the green line with square marks corresponds to the $I(PAR_3(u); g(R))$ results shown in figure 5.9. Reservoir size is $N = 100$ and input has 16 sections. I_μ for MI is increased by increasing w up to critical $w = 3$ for $\tau = 10$ and critical $w = 9$ for $\tau = 7$. With further increase in w , the I_μ starts dropping. This means that, the consistency of the system in representing the input and output is decreased.

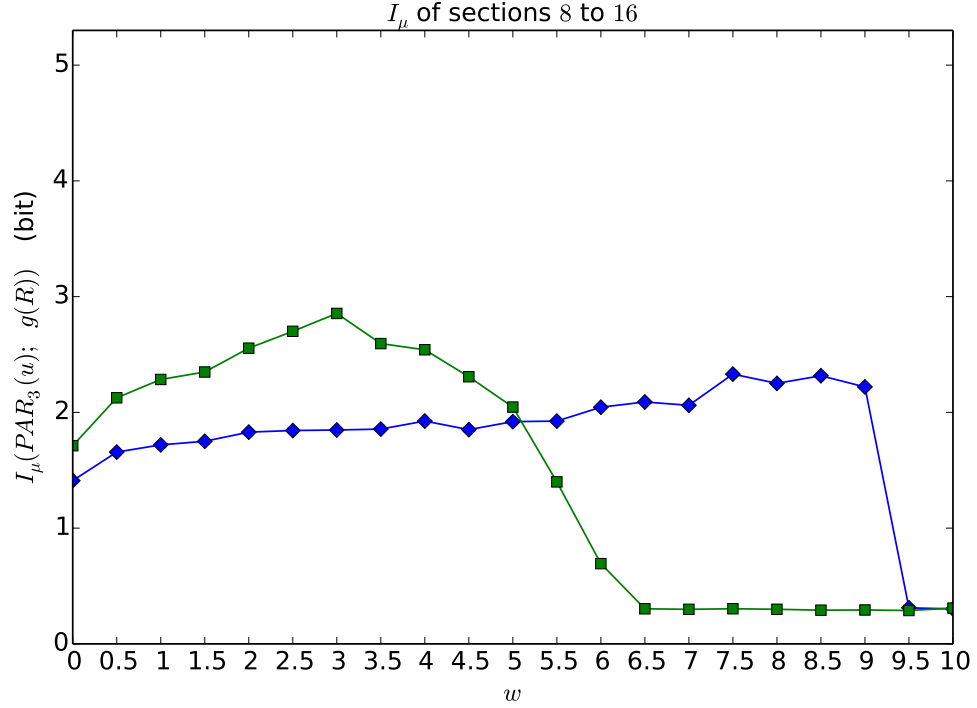


Figure 5.12: $I_\mu(PAR_3(u); g(R))$ of the reservoir with LIF neurons and two different time constants $\tau_m = 7$, and $\tau_m = 10$. Only the I_μ for $\tau_m = 10$ shown here by the green line with square marks corresponds to the $I(PAR_3(u); g(R))$ results shown in figure 5.9. Reservoir size is $N = 100$ and input has 16 sections. $I(PAR_3(u); g(R))$ also indicates the performance of the reservoir in solving PAR_3 over time. $I(PAR_3(u); g(R))$ rise and fall follows those of $I(u; g(R))$. That is to say, the performance of the reservoir in solving PAR_3 is highly dependent on the amount of stable memory in the reservoir.

5.1.4 Discussion

We compared the reservoirs with sigmoidal, LI, and LIF neurons in terms of information representation and memory by measuring $I(u; g(R))$, $I(PAR_3(u); g(R))$,

$I_\mu(u; g(R))$, and $I_\mu(PAR_3(u); g(R))$. I_μ yields a more convenient handle on finding the critical w and comparing the reservoirs with respect to memory, and stability. We first considered whether our MI results can elucidate the distinction between the information representation of the reservoir before and after the critical w in ESN. In both reservoirs with sigmoidal and LI neurons, the distinction was apparent and the measured I_μ 's were maximized at the critical $w \approx 1$. This relationship conforms with the ESN's maximum performance when $w \approx 1$ in [1]. We pursued the same MI and MCMC measurement methods for the LIF reservoir and found the critical w to be ≈ 3 for $\tau_{syn} = 10$ and ≈ 9 for $\tau_{syn} = 7$. We outlined the comparison of the MI and MCMC results for all three reservoir types in table 5.1. The reservoir with LIF neurons (shown in bold face in the table) achieved the highest maximum $I_\mu(u; g(R))$ and $I_\mu(PAR_3(u); g(R))$, 4.21 and 2.92 respectively. The reservoir with LI has a slightly higher $I_\mu(u; g(R))$ and $I_\mu(PAR_3(u); g(R))$ than the reservoir with sigmoidal neurons.

	Neuron Type	Connectivity. Model	N neurons	Max. I_μ input	Max. $I_\mu PAR_3$
1	sigmoidal	ESN	100	3.71	2.47
2	LI	ESN	100	3.79	2.51
3	LIF	ESN	100	4.21	2.92

Table 5.1: Comparison of the reservoir types in terms of the maximum obtained MCMC for the input and PAR_3 of the input. Max. I_μ input and Max. $I_\mu PAR_3$ refer to the maximum achieved $I_\mu(u; g(R))$ and $I_\mu(PAR_3(u); g(R))$ respectively. The reservoir with LIF neurons (shown in bold face) has the highest input and PAR_3 MCMC's (4.21 and 2.92).

5.2 Recognizing the Isolated Spoken-Digits with Analog Non-Spiking and Spiking Reservoirs

We first feed in 7 irrelevant random digits to the reservoir, each of which requires $250ms$ on average, to make sure that the reservoir is not in the early stages of perturbations with higher, but inconsistent, memory. Then, we feed in the actual digit and measure the performance of the task to determine whether the reservoir is reliably able to recognize the digit at any time. This insures that an unstable reservoir would not solve the task with high performance. In our experiments, we consider each discrete time step to be equivalent to one millisecond in continuous time. We tested the task for 5 different reservoir sizes: $N = 100, 200, \dots, 500$. We ran each experiment 30 times and obtained the mean performance to reduce variance in the performance due to stochasticity in the reservoir connectivity weight matrix. Our performance plots are the mean value of 30 trials of the spoken-digit recognition task with the error-bars showing the standard deviation of the performance over 30 trials. For each trial, a new reservoir is generated. Figure 5.13 shows the performance of the reservoir with sigmoidal neurons in recognizing a spoken-digit. This reservoir is completely unable to perform the task. Figure 5.14 and 5.16 show the performance of the reservoirs with LI and LIF neurons, respectively. The reservoir with LIF neurons recognizes the digit with 0.81 mean performance with only 100 neurons and 0.97 mean performance with 500 neurons. The LI reservoir can recognize the digits with up to 0.70 mean performance, yet with 500 neurons. The LIF reservoir could recognize all the digits i.e., with *performance* = 1 in some trials.

We also probed the reservoir with LI neurons in recognizing the spoken-digits with different leak rates (a) to find the optimum performance of the LI reservoir by

varying a and w . We considered τ_m as a decay parameter and define it as $a = \frac{1}{\tau_m}$. Then we measured the performance against w when τ_m varies between 4 and 40 (see figure 5.15). As shown in figure 5.15, the LI reservoir with size $N = 500$ can recognize the digits at the best with 0.80 mean-performance when $\tau_m = 20$ and $w = 1.9$. Yet, it falls short of the LIF reservoir mean-performance (0.81) with size $N = 100$.

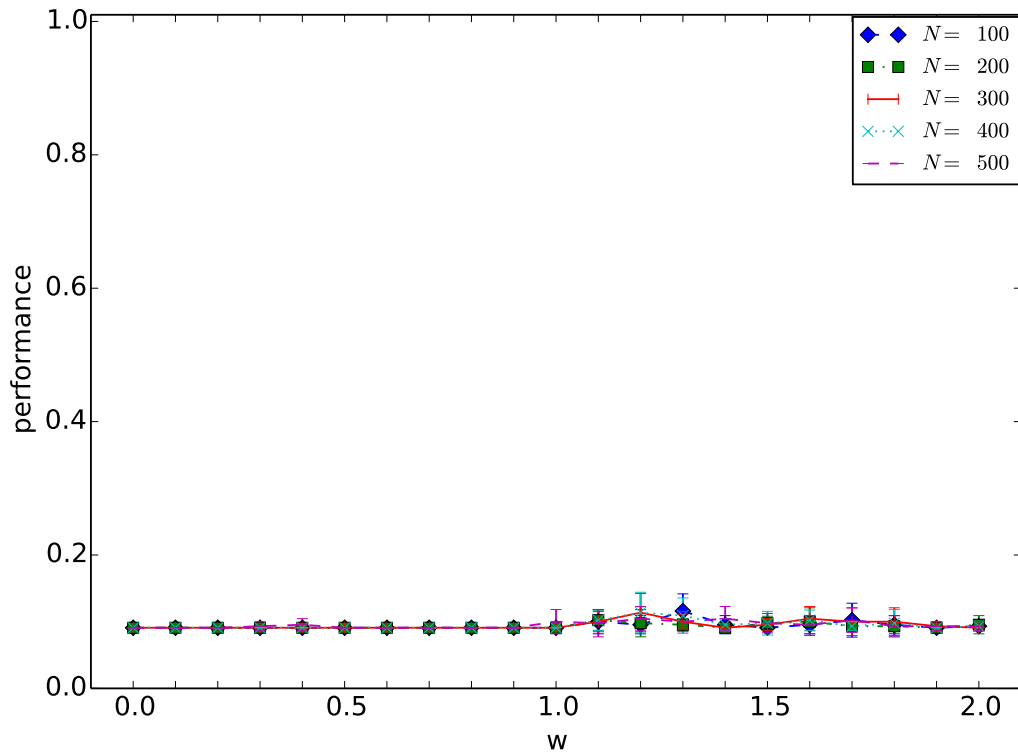


Figure 5.13: Performance of the reservoir with sigmoidal neurons ($N = 100, 200, \dots, 500$) in recognizing the isolated spoken-digit. The y axis represents the mean value of the performance over 30 trials for each experiment and the error bars show the standard deviation of the performance. We have run the experiment with the same w range shown on the x axis, covering the same range of w in the MI results.

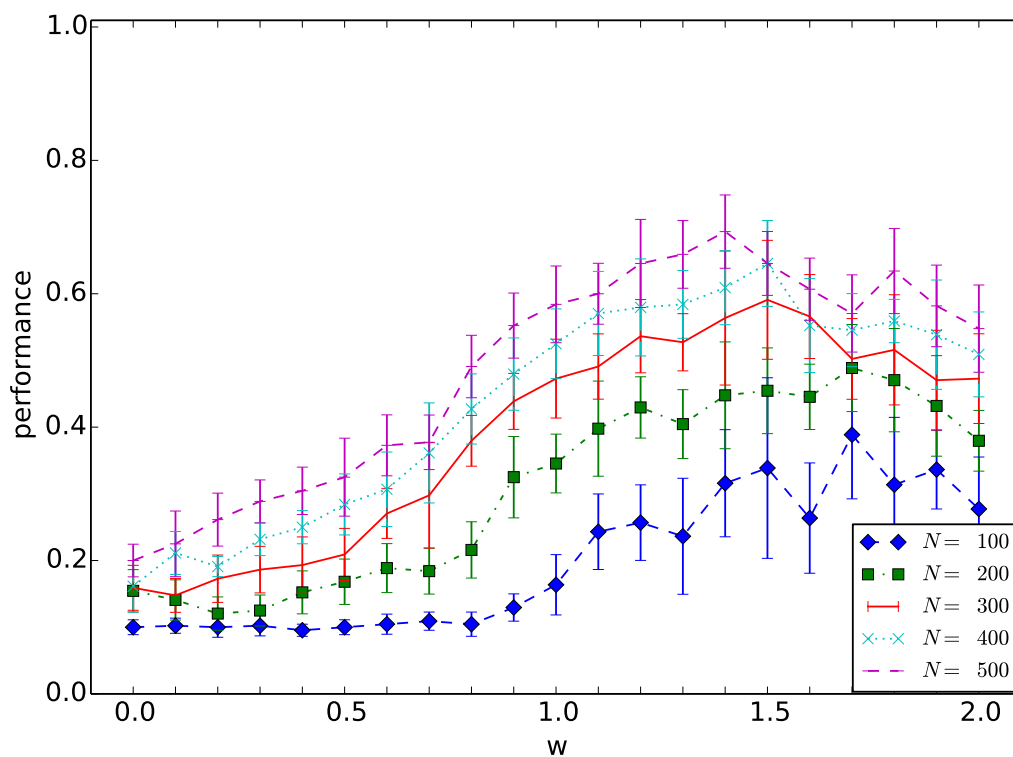


Figure 5.14: Performance of the reservoir with LI neurons ($N = 100, 200, \dots, 500$) in recognizing one spoken-digit. The LI neurons' leak parameter is fixed to $a = 0.25$. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. We have run the experiment with same w range shown on the x axis, covering the same range of w in the MI results.

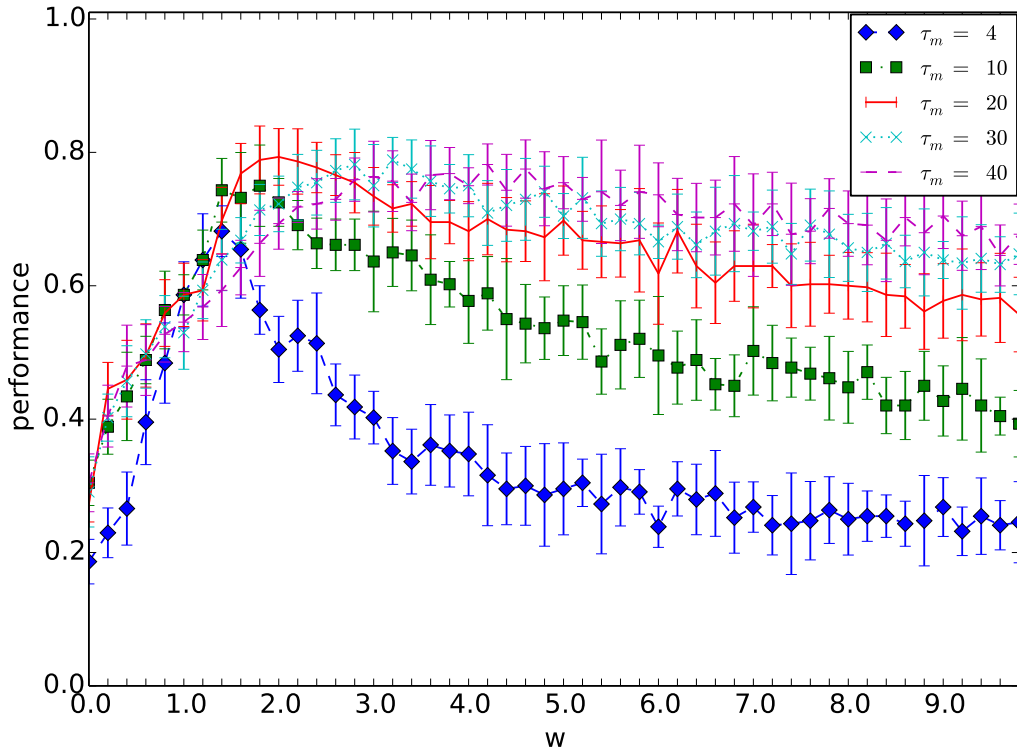


Figure 5.15: The influence of the LI neuron’s leak (a) parameter in the performance of the reservoir with LI neurons ($N = 500$) in recognizing one spoken-digit. We consider τ_m , $a = \frac{1}{\tau_m}$ as decay parameter and measure the performance against w when τ_m varies between 4 and 40. By increasing τ_m , the memory of the LI neuron is increased. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. We have run the experiment with the higher w range shown on the x axis (as compared to the range of w in the MI results) to ensure that the performance is not increased with further increase of w . Notice that the performance does not reach that of LIF reservoir shown in figure 5.16. It is worth recalling that in our MI result shown in figures 5.7 and 5.8, $\tau_m = 4$ ($a = 0.05$).

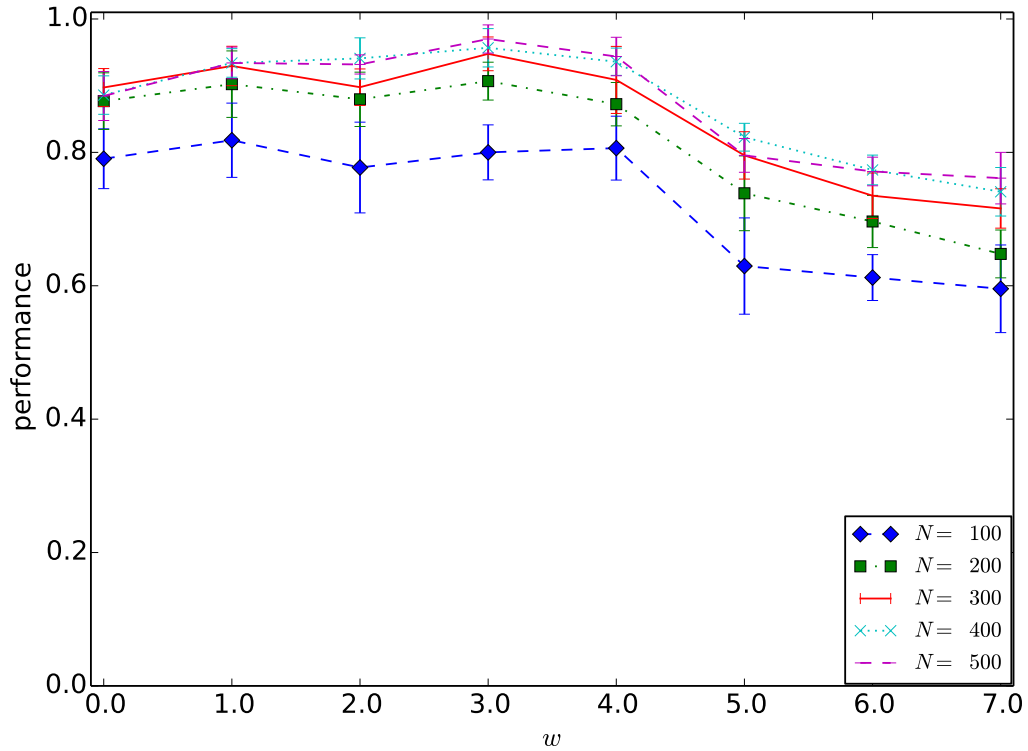


Figure 5.16: Performance of the reservoir with LIF neurons ($N = 100, 200, \dots, 500$) in recognizing one spoken-digit. The LIF membrane time-constant parameter is fixed to $\tau_m = 20ms$. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. We have run the experiment with $w = [0.0, 7.0]$ (shown on the x axis) so that the complete spectrum of MI for $\tau_{syn} = 10ms$ (shown in figures 5.11 and 5.12) is covered. The relevance of the MCMI results (shown in figures 5.9 and 5.10) in the performance of the reservoir at another spatio-temporal task is evident: the performance is maximized when the amount of stable memory is maximized. The LIF reservoir outperforms the analog non-spiking reservoirs in this spatio-temporal tasks.

5.2.1 Discussion

The objective of performing this task was to benchmark the in-question reservoirs in solving a real-world, spatio-temporal task. The proposed reservoir with LIF neurons outperformed its rivals in solving this task. We achieved 100% performance in some of the trials in our experiments. The mean performance of our proposed LIF reservoir showed a slight improvement compared to the LSM tested in [10] (97% as opposed to 95% mean performance). The rise and fall of the performance achieved by varying the w was compatible with our observations preceded earlier in the MI results sections. This compatibility implies that the memory and stability of the reservoir have an explicit impact on the performance of this task. The performances were maximized after, but very close to, the critical w in all cases. This relationship is in agreement with the postulation that the performance of the reservoir is maximized at the edge of chaos [5]. Upon looking at figures 5.9 and 5.10, one can see the impact of the amount of stable memory in the reservoir, as measured by MCIMI, in the performance of this task is evident.

Another observation is that in most of the experiments, as the reservoir size grows the performance improvement becomes less obvious. For instance, the performance of the reservoir with 400 neurons is very close to the reservoir with 500 neurons, but they both have at least a 10% higher mean-performance than the reservoir with 100 neurons. The maximum mean performances obtained from all the reservoir types with sizes 100 and 500 in recognizing the isolated spoken-digits are compared in table 5.2 in ‘I. S-D’ columns.

The rather poor performance of the sigmoidal reservoir in solving this task came as quite a surprise. It is not clear to us why the reservoir with sigmoidal neurons cannot solve this task despite the fact that its MCMI measures were

not severely lower than those of the LI reservoir. In other words, MCMI of the sigmoidal reservoir does not seem to be so small that the performance of the spoken-digit recognition task would become $\approx 1\%$. This observation also highlights the proposition that a high enough stable memory is essential but not sufficient to solve any spatio-temporal task with high performance, but it is not sufficient. It may be that might that the sigmoidal neuron lacks sufficient separation. ON the other hand, we showed that the sigmoidal reservoir could solve PAR_3 for an approximately $2.4 \times 200ms$ long input.

5.3 Recognizing the Spoken-Digits Sequences with Analog Non-Spiking and Spiking Reservoirs

We present our results for the reservoir types in recognizing a sequence of spoken-digits. Figures 5.17 to 5.19 show the performance of the reservoirs with sigmoidal, LI, and LIF neurons respectively. It is obvious that by increasing the length of the sequences the performance of all reservoir types decreases because longer sequences demand higher memory to be recognized by the system. Sigmoidal neurons are obviously unable to perform this task, as they could not perform the single digit task at all. Our result show that the LIF reservoir is the best reservoir of the three for solving this task. Compare the performance of the LI reservoir in figures 5.18 and 5.19. This task also reveals the memory limits of the reservoir with LIF neurons in solving a temporal task such as spoken-digits recognition when more memory is required in 7-digit long sequences.

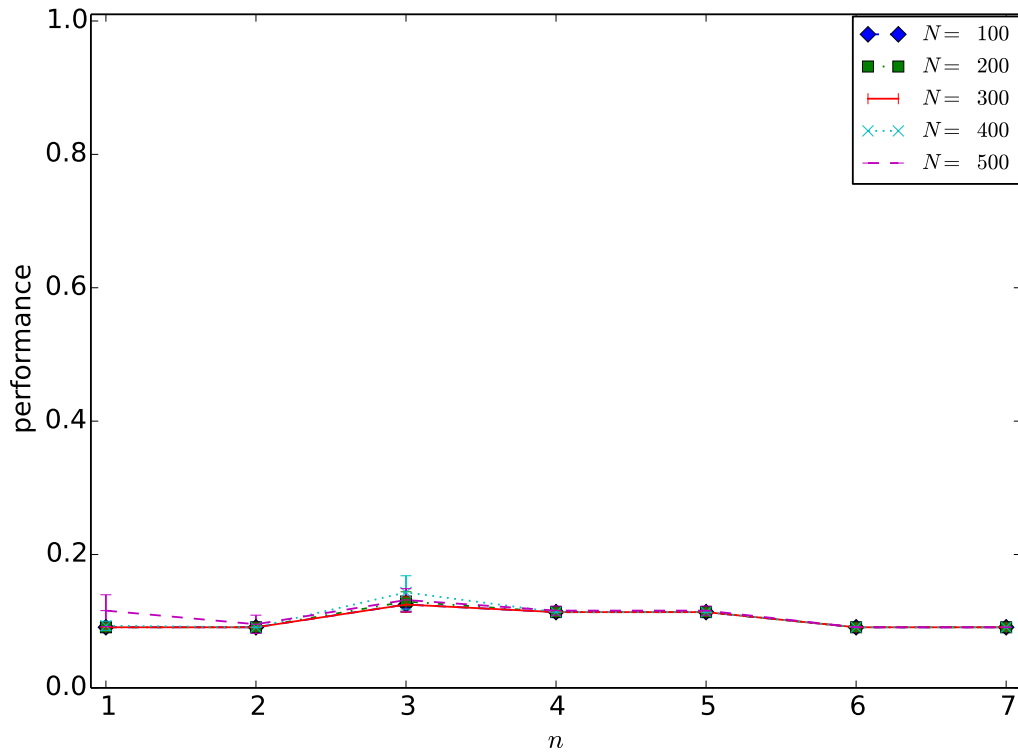


Figure 5.17: Performance of the reservoir with sigmoidal neurons ($N = 100, 200, \dots, 500$) in recognizing the sequence of $n = 1, 2, \dots, 7$ spoken-digits. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance.

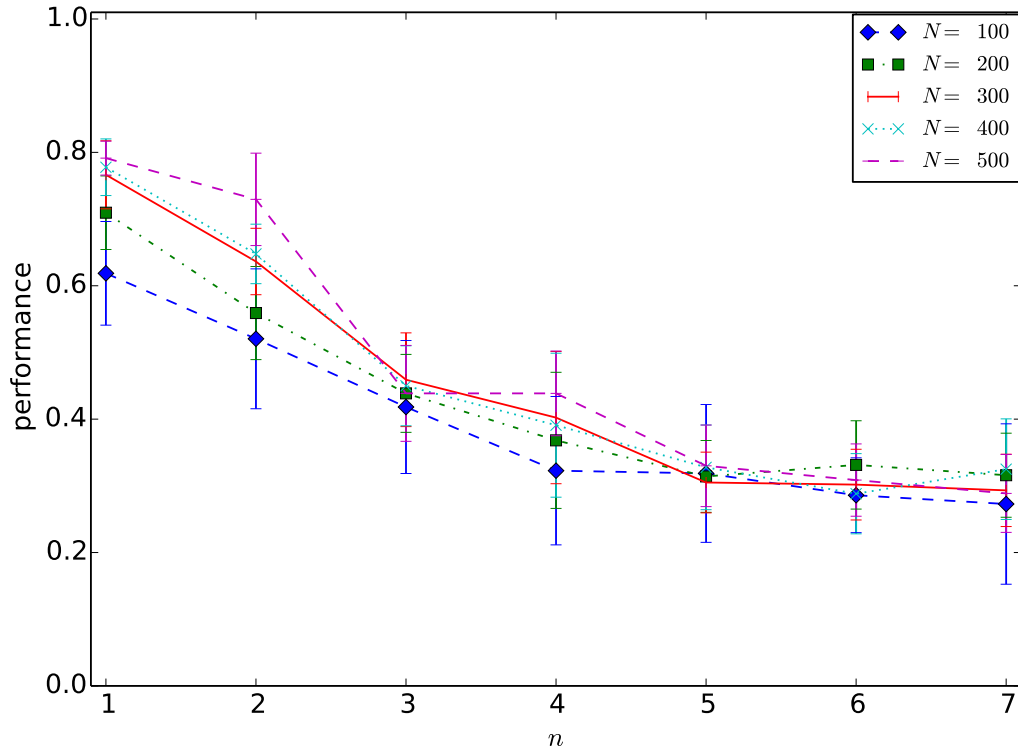


Figure 5.18: Performance of the reservoir with LI neurons ($N = 100, 200, \dots, 500$) in recognizing the sequence of $n = 1, 2, \dots, 7$ spoken-digits. To achieve the highest possible performance, the leak parameter and weight coefficient w of the LI neurons are fixed at $a = 0.05$ ($\tau_m = 20$) and $w = 2$. The y axis represents the mean value of the performance over 30 trials for each experiment, and the error bars show the standard deviation of the performance. The mean-performance when $N = 100$ and $n = 1$ did not reach the maximum mean-performance shown in figure 5.15 of $N = 100$ because $w = 2$ is an optimum value for $N = 500$ but not $N = 100$ for $n = 1$.

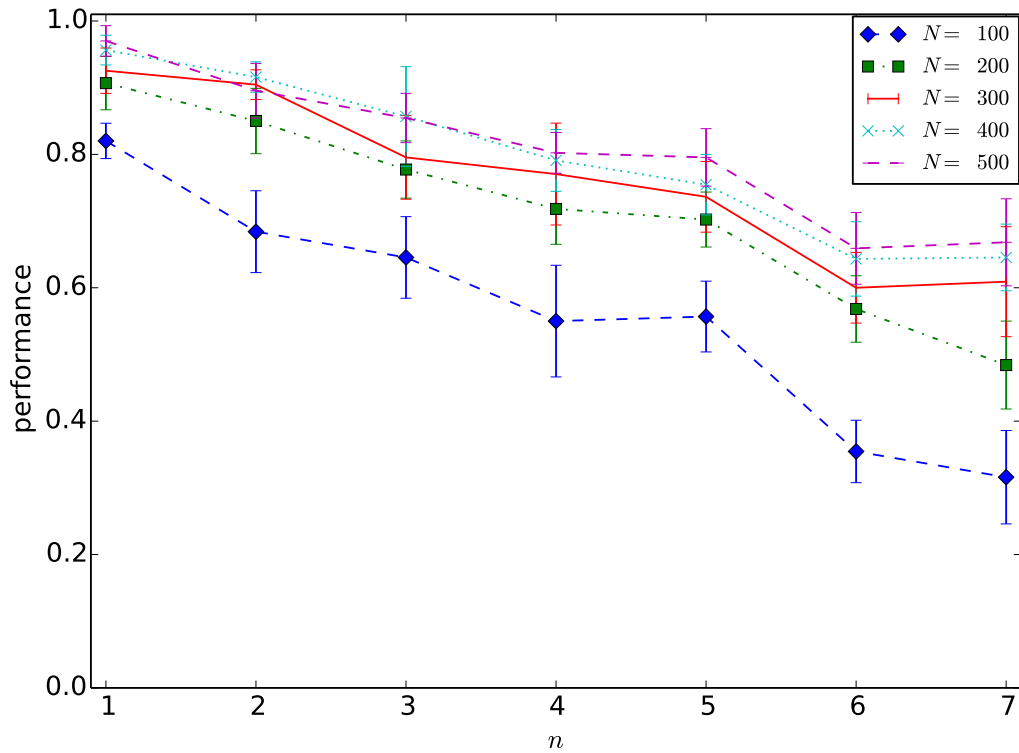


Figure 5.19: Performance of the reservoir with LIF neurons ($N = 100, 200, \dots, 500$) in recognizing the sequence of $n = 1, 2, \dots, 7$ spoken-digits. The time constant parameter of the LIF neurons is fixed at $\tau_m = 20ms$. The y axis represents the mean value of the performance over 30 trials for each experiment and the error bars show the standard deviation of the performance. The maximum performance of the reservoir is achieved at the critical $w \approx 3$. The LIF reservoir outperforms the analog non-spiking reservoirs in a rather memory demanding spatio-temporal tasks. Recall that the LIF reservoir has the highest amount of stable memory of all the reservoirs in our experiments.

5.3.1 Discussion

The main goal of proposing and performing this task was to conduct a stress test over all three types of reservoirs to determine the memory and stability of the reservoirs in representing the input information and eventually to compute the desired output. It turns out that the memory and stability of the LIF reservoir is highly suited for solving the memory demanding spoken-digit sequence recognition task. The LIF reservoir could solve the task much better than the other two reservoirs. The maximum mean performances obtained from all the reservoir types, with sizes 100 and 500, in recognizing the spoken-digit sequences, are compared in table 5.2 in ‘Seq. S-D 7’ columns. The LIF reservoir maintains the highest record in recognizing the spoken-digit sequences. It shows more promise in solving memory-demanding, spatio-temporal, real-world tasks. Another considerable result is that, the LIF reservoir with 100 neurons beats the reservoir with 500 optimum LI neurons (80% vs. 79%). This result shows that, the LIF reservoir works remarkably more efficient than the LI reservoir. It is very interesting with respect to the feasibility of hardware implementation and energy efficiency.

Neuron Type	Connectivity. Model	I. S-D maximum mean perf. (N = 100)	I. S-D maximum mean perf. (N = 500)	Seq. S-D 7 maximum mean perf. (N = 100)	Seq. S-D 7 maximum mean perf. (N = 500)
sigmoidal	ESN	0.9% ($\pm 1.3\%$)	1% ($\pm 1.8\%$)	0.9% ($\pm 0\%$)	1% ($\pm 0.6\%$)
LI	ESN	68% ($\pm 7.5\%$)	79% ($\pm 3\%$)	27% ($\pm 12\%$)	32% ($\pm 5.8\%$)
LIF	ESN	80% ($\pm 4.1\%$)	97% ($\pm 2.1\%$)	32% ($\pm 7\%$)	67% ($\pm 6.5\%$)
LIF	LSM	N/A	95%	N/A	N/A

Table 5.2: Comparison of the reservoir types with respect to the maximum obtained mean-performance in recognizing isolated-spoken-digit recognition (I. S-D) and a sequence of 7 spoken-digits (Seq. S-D 7) when $N = 100$ and $N = 500$ neurons are used in the reservoir. The standard deviations of the performances are shown in parentheses. The LIF reservoir is shown in boldface. Our reservoir with $N = 500$ LIF neurons could recognize the isolated spoken-digits with 97% mean-performance. Compared to a 95% mean-performance obtained from LSM in [10], it showed a slight improvement. Comparing that to the mean-performance of 79% obtained from the reservoir with LI neurons, it has a considerable performance increase. The proposed LIF reservoir also showed higher performance than the LI reservoir even with $N = 100$ neurons. Performance results for isolated spoken-digits with $N = 100$ is not available in [10], and the sequence-recognition tasks have not been performed in [10]. The sigmoidal reservoir failed to solve the problem. All the numbers are rounded values.

Conclusion

We investigated the impact of information representation and memory in processing spatio-temporal spike-train inputs in RC systems. From this perspective, we compared RC systems with LIF neurons and analog non-spiking RC systems with sigmoidal and LI neurons. Our study has been deeply under the influence of stable, transition phase (edge of chaos), and chaotic reservoir dynamics imposed by network weight coefficient w . In other words, we questioned what happens to the information representation of the reservoir when w is smaller than a critical value w_0 , when $w \approx w_0$, and when $w > w_0$. To answer this question, we expanded the MI measurement method outlined in [9] and measured for all three types of reservoirs. Then, we investigated the impact of that on the performance of the reservoir in solving memory-demanding spatio-temporal tasks. Below, we outline the findings we obtained in four major steps and then draw an overall conclusion by connecting those findings.

First, we measured the MI and MCMCMI between the reservoir dynamics and the input to show the capability of the reservoir for input information representation, memory capacity, and memory stability. Our observation on MI measures obtained from the reservoir with sigmoidal and LI neurons conforms to the fact that ESN's performance in solving tasks reaches its apex at the critical w close to 1 [1]. These reservoirs also maintain their stability when $w < \approx 1$. By further increasing w , both the stability and the performance are decreased. MI results obtained from the reservoir with LIF neurons shows a similar behavior. However, the critical w for

this type of reservoir varies with the neuron and synaptic parameters. Based on our experiments in this study, the critical values in the reservoir with LIF neurons for w turns out to be $w \approx 3$, when $\tau_{syn} = 10ms$, and $w \approx 9$, when $\tau_{syn} = 7ms$. According to our results, the reservoir with LIF neurons has higher memory capacity at the critical w than the reservoirs with LI and sigmoidal neurons. We achieved a maximum input MCMI of 4.5 for the reservoir with LIF neurons. The maximum input MCMI's for the reservoir with sigmoidal and LI neurons were 3.71 and 3.79 respectively.

Second, for all the three reservoir types, we measured the MI and MCMI between the reservoir dynamics and temporal parity of the input to show the correlation between the performance of the reservoir and the capability of the reservoir in input information representation, memory capacity, and memory stability. We achieved a maximum PAR_3 MCMI of 2.92 for the reservoir with LIF neurons. The maximum PAR_3 MCMI's for the reservoir with sigmoidal and LI neurons were 2.47 and 2.51, respectively. Comparison of PAR_3 MCMI's with the input MCMI results shows that the data-processing inequality holds in computing the PAR_3 task for reservoirs with all three neuron types. In other words, the reservoir can only compute over an input data if the amount of represented input information is enough to solve PAR_3 . In fact, the amount of memory and the memory stability in the reservoir have a direct impact on the performance of the reservoir.

Third, we compared the RC systems with all three types of neurons in recognizing spoken digits represented by 40 channels of spike trains. An RC system with sigmoidal neurons fell short of solving this task with the maximum mean-performance of 1%. The RC system with LIF neurons outperformed the RC system with LI neurons. We witnessed 97% and 76% maximum mean-performance

for reservoir with LIF and LI neurons in our experiments, respectively. We further tested the reservoir with LI neuron but with other leak ($a = \frac{1}{\tau_m}$) values. The performance of the reservoir with optimized LI neurons did not yet approach that of the reservoir with LIF neurons. Compare 78% in the LI reservoir to 97% maximum mean-performance in the LIF reservoir.

Forth, we further investigated the ability of the mentioned reservoirs to solve a similar but rather more memory-demanding task. We concatenated the spike trains of the digits to form sequences of digits, and we compared the reservoirs in recognizing the constructed spoken-digit sequences. The reservoir with LIF neurons outperformed the reservoir with non-spiking analog neurons in recognizing spoken-digit sequences up to 7 digits long. The LIF and LI reservoirs solved the task with 76% and 32% maximum mean-performance, respectively, for a sequence of 7 spoken-digits. Obviously, sigmoidal neurons failed to solve this task, yielding 1% maximum mean-performance. It is evident that the LIF reservoir possesses a higher memory compared to its rivals. With this task, we also explored and revealed the memory limits of the reservoir with LIF neurons in computing the spoken-digit sequence-recognition task. The performance dropped from 97% mean-performance for one digit to 76% mean-performance for the sequence of seven digits.

In all the three reservoir instances in our study, MI measures showed stable information representation in the reservoir, when connectivity weight coefficient w is less than or close to a critical value w_0 (at the edge of chaos [4,5]). The memory and consistency of the reservoir in representing the input (measured by MCMI) was maximized when $w \approx w_0$. After the w was increased to a value higher than w_0 , both the MCMI and the performance of the reservoir in solving PAR_3 were

decreased dramatically. This stage of the reservoir dynamics is referred to as a chaotic regime in [5].

The bottom-line of our observations is that the network dynamics have a direct impact on input representation, memory, and memory stability in the reservoir. As a consequence, an increase or decrease in these factors gives rise in turn to the increase or decrease of the task performance. The reservoir can perform the best when MCMI is high enough for a given task. Lack of enough linearly separable memory in the reservoir results in decreasing the performance of the RC system as is shown in the MCMI results measured for both input and PAR_3 of the input in the so-called chaotic regime. Our performance analysis in spoken-digit recognition tasks confirm that this MCMI difference (maximum input MCMI of 4.5 and 3.7) causes a large difference in the performance of the reservoirs in solving this complex spatio-temporal task. We witnessed a 97% and 76% maximum mean-performance for reservoir with LIF and LI neurons, respectively, in our experiments. With respect to the performance peak and the critical w in spoken-digit recognition task, we observed that the performance peak varies with w , neuron parameters, reservoir size. The exact critical w in PAR_3 and isolated spoken-digit recognition tasks obtained from the LI reservoir were not exactly the same; and that implies that the exact critical w also depends on the input characteristics.

Our reservoir with LIF neurons yielded slightly higher performance (97% mean-performance) in isolated spoken digit-recognition tasks than the LSM used in [10] with 95% mean-performance. Yet the detailed in-depth comparison of this reservoir with LSM will be left to the future work. The reservoir with LIF neurons appears to be more promising for building nano-electronic RC systems. It outperformed the sigmoidal and LI reservoirs in all the tasks and solved the isolated spoken-digit

recognition more efficiently. From an engineering stand-point, our findings have applications in building nano-electronic RC systems with LIF neurons for solving real-world, memory-demanding, spatio-temporal problems. The study paves the road toward solving real-world, spatio-temporal problems by shedding light on the impact of information representation on the performance of RC systems.

References

- [1] H. Jaeger, “The ”echo state” approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, p. 287289, 2001.
- [2] W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [3] R. Legenstein and W. Maass, “Edge of chaos and prediction of computational performance for neural circuit models,” *Neural Networks*, vol. 20, no. 3, pp. 323–334, 2007.
- [4] N. Bertschinger and T. Natschläger, “Real-time computation at the edge of chaos in recurrent neural networks,” *Neural Computation*, vol. 16, no. 7, pp. 1413–1436, 2004.
- [5] T. Natschläger, N. Bertschinger, and R. Legenstein, “At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks,” *Advances in Neural Information Processing Systems*, vol. 17, pp. 145–152, 2005.
- [6] L. Büsing, B. Schrauwen, and R. Legenstein, “Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons,” *Neural Computation*, vol. 22, no. 5, p. 12721311, 2010.

- [7] A. I. Basbaum, *The senses: a comprehensive reference*. Amsterdam; Boston: Elsevier, 2008.
- [8] V. E. Abraira and D. D. Ginty, “The sensory neurons of touch,” *Neuron*, vol. 79, no. 4, pp. 618–639, 2013.
- [9] T. Natschläger and W. Maass, “Dynamics of information and emergent computation in generic neural microcircuit models,” *Neural Networks*, vol. 18, no. 10, pp. 1301–1308, 2005.
- [10] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, “Isolated word recognition with the liquid state machine: a case study,” *Information Processing Letters*, vol. 95, no. 6, pp. 521–528, 2005.
- [11] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, “An overview of reservoir computing: theory, applications and implementations,” in *Proc. of the 15th European Symposium on Artificial Neural Networks*, 2007.
- [12] F. Takens, “Detecting strange attractors in turbulence,” in Rand, D. A. and Young, L. S., editors, *Lecture Notes in Mathematics*, vol. 898, pp. 366–381, 1981.
- [13] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proc. of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [14] P. J. Werbos., “Beyond regression: New tools for prediction and analysis in the behavioral sciences,” Ph.D. dissertation, Harvard University, Boston, MA, 1974.

- [15] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proc. of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [16] D. Rumelhart, G. Hinton, and R. Williams, *Parallel Distributed Processing*. MIT Press, 1986, ch. Learning internal representations by error propagation.
- [17] D. V. Buonomano and M. M. Merzenich, “Temporal information transformed into a spatial code by a neural network with realistic properties,” *Science*, vol. 267, no. 5200, pp. 1028–1030, 1995.
- [18] S. Haykin, *Neural Networks and Learning Machines (3rd Edition)*. New York, NY: Pearson Education, Inc., 2009.
- [19] P. J. Werbos, “Generalization of backpropagation with application to a recurrent gas market model,” *Neural Networks*, vol. 1, no. 4, pp. 339–356, 1988.
- [20] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.
- [21] J. Kilian and H. T. Siegelmann, “The dynamic universality of sigmoidal neural networks,” *Information and Computation*, vol. 128, no. 1, pp. 48–56, 1996.
- [22] K. Funahashi and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” *Neural Networks*, vol. 6, no. 6, pp. 801–806, 1993.
- [23] C. W. Omlin and C. L. Giles, “Constructing deterministic finite-state automata in sparse recurrent neural networks,” in *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, vol. 3. IEEE, 1994, pp. 1732–1737.

- [24] J. J. Steil, “Backpropagation-decorrelation: online recurrent learning with $O(N)$ complexity,” in *Proc. of the Neural Networks, 2004. IEEE International Joint Conference on*, vol. 2. IEEE, 2004, pp. 843–848.
- [25] D. Verstraeten, B. Schrauwen, M. D’Haene, and D. Stroobandt., “A unifying comparison of reservoir computing methods,” *Neural Networks*, vol. 20, pp. 391–403, 2007.
- [26] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.
- [27] R. Legenstein and W. Maass, “What makes a dynamical system computationally powerful,” in S. Haykin, J. Príncipe, T. Sejnowski, and J. McWhirter, editors, *New Directions in Statistical Signal Processing: From Systems to Brain*, pp. 127–154, 2007.
- [28] J. Burger and C. Teuscher, “Variation-tolerant computing with memristive reservoirs,” in *Nanoscale Architectures (NANOARCH), 2013 IEEE/ACM International Symposium on*. IEEE, 2013, pp. 1–6.
- [29] D. Snyder, A. Goudarzi, and C. Teuscher, “Computational capabilities of random automata networks for reservoir computing,” *Physical Review E*, vol. 87, no. 4, p. 042808, 2013.
- [30] H. Markram, Y. Wang, and M. Tsodyks, “Differential signaling via the same axon of neocortical pyramidal neurons,” *Proc. of the National Academy of Sciences*, vol. 95, no. 9, pp. 5323–5328, 1998.

- [31] B. Derrida and Y. Pomeau, “Random networks of automata: a simple annealed approximation,” *Europhysics Letters*, vol. 1, no. 2, pp. 45–52, 1986.
- [32] B. Derrida, “Dynamical phase transition in nonsymmetric spin glasses,” *Journal of Physics A: Mathematical and General*, vol. 20, no. 11, pp. 721–725, 1987.
- [33] C. G. Langton, “Computation at the edge of chaos: phase transitions and emergent computation,” *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 12–37, 1990.
- [34] N. H. Packard, “Adaptation toward the edge of chaos,” in J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, *Dynamic patterns in complex systems*, pp. 293–301, 1988.
- [35] M. Mitchell, P. Hraber, and J. P. Crutchfield, “Revisiting the edge of chaos: Evolving cellular automata to perform computations,” *Complex Systems*, no. 7, pp. 89–130, 1993.
- [36] S. A. Kauffman, *The origins of order: Self organization and selection in evolution*. New York: Oxford University Press, USA, 1993.
- [37] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *The Journal of physiology*, vol. 117, no. 4, p. 500, 1952.
- [38] R. Rose and J. Hindmarsh, “A model of a thalamic neuron,” *Proc. of the Royal Society of London. Series B. Biological Sciences*, vol. 225, no. 1239, pp. 161–193, 1985.

- [39] E. M. Izhikevich, *Dynamical systems in neuroscience*. Cambridge, MA: MIT press, 2007.
- [40] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [41] D. L. Schacter, D. T. Gilbert, and D. M. Wegner, *Psychology (2nd ed.)*. New York: Worth Publishers, 2011.
- [42] T. M. Cover and J. A. Thomas, *Elements of information theory*. Hoboken, NJ: John Wiley & Sons, 2012.
- [43] N. J. Beaudry and R. Renner, “An intuitive proof of the data processing inequality,” *Quantum Information & Computation*, vol. 12, no. 5-6, pp. 432–441, 2012.
- [44] J. B. Kinney and G. S. Atwal, “Equitability, mutual information, and the maximal information coefficient,” *Proc. of the National Academy of Sciences*, vol. 111, no. 9, pp. 3354–3359, 2014.
- [45] C. E. Shannon, “A mathematical theory of communication.” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [46] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [47] J. J. Hopfield and C. D. Brody, “What is a moment? “cortical” sensory integration over a brief interval,” *Proc. of the National Academy of Sciences*, vol. 97, no. 25, pp. 13 919–13 924, 2000.

- [48] R. F. Lyon, “A computational model of filtering, detection, and compression in the cochlea,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’82.*, vol. 7. IEEE, 1982, pp. 1282–1285.
- [49] B. Schrauwen and J. Van Campenhout, “BSA, a fast and accurate spike train encoding scheme,” in *Proc. of the International Joint Conference on Neural Networks*, vol. 4, 2003, pp. 2825–2830.
- [50] P. Auer, H. Burgsteiner, and W. Maass, “Reducing communication for distributed learning in neural networks,” in *Artificial Neural Networks—ICANN*. Springer, 2002, pp. 123–128.