Studying a Self-Sustainable System by making a Mind Time Machine

Takashi Ikegami

Graduate school of Interfaculty Initiative
in Information Studies , University of Tokyo,
3-8-1 Komaba, Tokyo, 153-8902, Japan
ikeg@sacral.c.u-tokyo.ac.jp

Abstract

We present our pilot study on a special machine that self-sustains its rich dynamics in an open environment. We made a machine called "MTM" (Mind Time Machine) that runs all day long, receiving massive visual data from the environment, processing by an internal neural dynamics with a learning capability, and showing sustainable complex adaptive dynamics. The System's internal time structure is also self-organized as a result of coupling with the environment. By observing MTM over 2 and half months, we argue for the possibility of machine consciousness in an artificial system.

Keywords mind time, massive data, plasticity, sustainability

1. Robustness and system design

It is time for bringing artificial life in silicon into the real world. In contrast to the artificially simulated environment, the real world presents many unexpected complex encounters, and living systems are essentially adaptive to these real world complexities. In this pilot study, we designed an artificial system that can be a first test system for overcoming various problems for artificial systems to "survive" in an open ended environment. We required that any artificial life should simultaneously cope with various kinds of sensory flows while simultaneously maintaining its own identity and autonomy over a relatively long period of time.

In creating such a machine, our main concern is how to design a system's time structure. A human has subjective time structures which is different from objective time. Our hypothesis is that this should be true for all intentional/functional systems, whether natural or artificial. Objective time structures, i.e. the physical Newtonian time scale, can be measured by a mechanical clock, but our mind's time scale, the so-called Bergsonian time scale, may not be treated the same way. That is, a minimal-length time segment can be regarded as infinitesimally small in the case of Newtonian time, but in Bergsonian time it can be bounded. I submit that there is no continuous time flow which can be assumed, as it is always perturbed by the inflow from an open-ended environment.

Wiener's definition of Bergsonian time, as opposed to Newtonian time (chapter 1 in [20]), is the emergence of an irreversible time flow in an ensemble of particles. Deleuze argues that Bergson's time is a duration which is also equivalent to memory, consciousness and free will [2]. Bergson's duration is equivalent to self-sustainability in the sense that humans can utilize their long term/short term memory to be sustainable over their life span. I have been elaborating upon the idea of self-sustainability at some length and have argued for biological robustness in various ways in the field of artificial life. Our concern is how to make a robust system that runs over a long period of time in an open-ended environment and, in order to design such a system, we have to care about the internal Bergsonian time structure. We explain robustness below, before returning to the internal time issue.

Some authors [14] and [15] have already developed insightful definitions of robustness in complex systems. Our discussion of robustness has served to demonstrate that, even in the highly diverse field of artificial life, such an approach may still flourish (e.g. [12, 19]).

In our case, a novel biochemical experiment together with simulation and robotics approaches are being used to develop an in-depth understanding of robustness and how we may quantify and examine its effects [4, 6]. Oil droplets demonstrate a simple chemical experiment of high pH: water reacting with oleic anhydride generates self-moving droplets which maintain the reaction on its surface, sustaining its self-mobility [5]. Here, the environmental conditions, pH and oleate concentration, are controlled by the droplet motion.

Workshop on Self-sustaining Systems (S3) 2010
September 27–28, 2010, The University of Tokyo, Japan

We define robustness of the droplets with respect to their ability to sustain self-moving behavior. In contrast, if we pick an example from the game of Life, gliders (the simplest moving pattern in the game) appear to display self-moving behavior but do not actually function in this way. This evolution of self-movement, autonomy and individuality appears to be a key prerequisite for developing robust behaviors.

Using a robotic platform, we used pure Hebbian learning dynamics to show how auditory and visual modules cooperatively work together to self-organize robust goal-oriented behavior [6]. When giving a robot a capacity for self-organization, however, the robot cannot sustain its autonomous movement quite so easily. Developing robustness in this case appears to depend on the development of an appropriate use of time-scales for its behavior; in particular, parameter settings for the robot's learning and forgetting during the process of Hebbian learning can affect the time-scales of the robot's behaviors. Finding a range of these parameters which allow proper functioning by utilizing the background noise of the environment will allow the development of more robust behaviors.

By increasing our understanding of how we can connect artificial systems with natural environments, we can further our development of a theoretical framework that provides a background of assumptions to inform our robotic and simulated models. One of my proposals is the Maximal Design Principle [11, 13], which underlines the importance of "halfway design" (of the initial states and architecture of a system) and letting a system self-organize in interacting with an environment, which can later lead to robust behaviors.

Concerning the above robustness issue, we designed a machine called MTM (Mind Time Machine). In order to take into account the system's internal time structure, Benjamin Libet's neuro-physiological early experiments [16] were very instructive (although we couldn't take many of his points into MTM). He said that there are time differences in processing information between time and mind time which is temporally editted backward and forward. For example, Libet showed that we actually have a 0.5 second delay in sensing a stimulus to our foot or hand, but our mind corrects the delay by backward referring the event. This backward referral never occurs when we directly stimulate the somatosensory region of the brain. Namely, the way a stimulus is delivered to the brain through our embodiment determines the subjective sense of the momentary "now."

In our pilot work, the system receives and edits the video inputs, while it self-organizes the momentary "now," in agreement with Libet's arguments. Its core program is a neural network that includes chaos (a mechanism that expands the small difference) inside the system, and a meta-network that consists of neural networks. Using this system as a hardware, and chaotic itinerancy [9, 10] as a conceptual framework, we like to describe system's sustainable behavior in terms of internal time scales. This is a proposal of designing an artificial life that self-organizes in a real world (i.e. coupling with the massive data input flow) to ultimately understand consciousness as a temporal order.

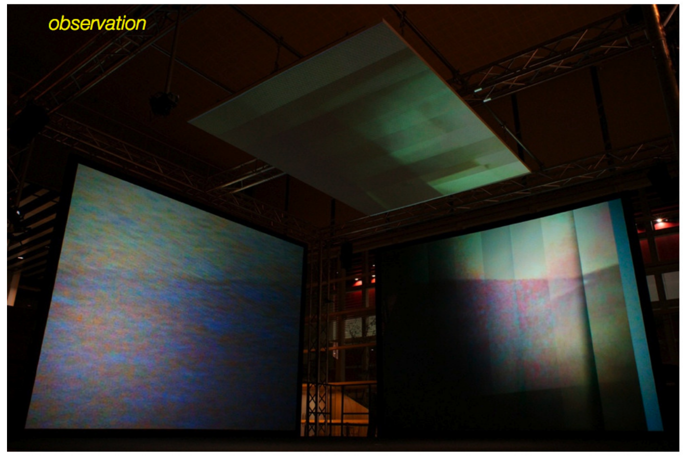

Figure 1. View of MTM displayed at the Yamaguchi Center for Arts and Media, 2010. (Photo taken by Kenshu Shintsubo)

In section 2, we illustrate the architecture of MTM explaining the underlying neural dynamics. In section 3, we report how MTM behaves over 2.5 months and show some characterization of the behaviors—its temporal complexity and dynamics of the internal clock. In section 4, we briefly describe how a sound version of MTM might function, and report on the pilot study of it. In section 5, we discuss how a system's sustainability is restored by the asynchronous memory updating and sensory networks. We then return to the Bergsonian vs. Newtonian time scale issue.

2. Architecture of MTM

2.1 A rough sketch

We presented our MTM for the first time at the Yamaguchi Center for Arts and Media in March 2010 as an art project . While demonstrating this MTM at the museum, we took data for analysis every day from the machine. The system's initial states were reset every morning, but its long term memory accumulated over 67 days.

This machine consists of three screens, right, left and above at the ceiling, displayed as faces of a cubic skeleton of the 5400 mm width on each side (see figure 1). Fifteen video cameras attached to each pole of the skeleton view things happening in the venue. The movie images are decomposed into frames, and a neural dynamics combine, reverse and superpose them to produce new frame sequences, which will be described in detail in the next section.

The operating principle is to run a "plastic" neural dynamics and optical feedback to make autonomous timeorganizing phenomena. Received images from the video cameras are progressively embedded into the neural connections of the first layer via the Hopfield method. Connections from the first layer to the second layer are progressively self-modified in the form of modified Hebbian dynamics (i.e. the connection strength is modified in proportion to the correlation between the pre synaptic neuronal state and the post synaptic neural state change). The connections between neurons in the second layer are also modified by the same dynamics.

By fixing the memory-embedded connections, every neuronal state is updated by the particular neural dynamics which is known as chaotic neural dynamics [1, 17]. By changing the connection strengths, a network demonstrates a fixed point, with periodic and chaotic behaviors (see Appendix for model equations). Each neuron in the second layer is associated with a set of visual images, and a simple "winner-takes-all" rule is applied here. Namely, a combination of visual images corresponding to the largest neural state is selected to project onto the screen. A detailed schema will be given in the next section.

Visual images are acquired and re-played in a recursive way. The system itself is a completely deterministic system, using no random numbers, but it projects different images depending on the inherent instabilities of the neural dynamics that reflect environmental light conditions, movement of people coming to the venue, and the system's stored memory.

A momentary "now" of the system progresses when the system's memory is updated, where the memory is a mixture of nested images. How does the inherent timescale progress with respect to the physical timescale? The MTM experiment is designed to answer such questions,

It should be remembered that MTM is not a large chaotic dynamic system that updates visual inputs randomly. First, the state of the neural system is not directly perturbed by the inputs. Second, it is a plastic system and changes its approximately 30,000 parameters (connection strengths) all day long. As opposed to the mere chaotic system, MTM is designed as a life-like system since its dynamics are controlled by an environment, and the system has a short and long term memory to sustain its dynamics. We claim that MTM is "artificial life," since we designed it to

1. receive information from its environment,
2. memorize data (in the form of the Hopfield type learning which tunes the parameters of the overall dynamics),
3. have "episodic memory,"
4. change the network structure (continuously, by way of Hebbian dynamics)

As a result. it organize its overall dynamics as an adaptation to environmental changes.

We will come back to these points in the discussion parts. Those who are not interested in the details of the MTM setup may skip the next section and go on to the observation and discussion sections.

2.2 Content of the program processing visual information

MTM consists of two parts:

A: Programs for sensory inputs. Programs combine and edit images from video camera inputs.

B: Programs for internal processes. Sensory inputs are processed by the internal neural networks with learning capabilities.

In the A programs, we have 4 different kinds of editing modes.

1) Slit-scanning movies: Picking up a vertical pixel line from different time frames to generate one image sequentially.

2) Superposing images from different time frames. We used the simplest alpha-blending method to do this.

3) Reversing the time order of frames to make a reverse-time movie.

4) Video Feedback: Dividing an input into 2^K regions, each of which contains the same image. The video then shuts that divided screen so that it inevitably generates self-similar images except when there is no division. This is also a time-related operation since, during this period of time, Newtonian time elapses but the internal time is recursively processed by the converging feedback

The B programs are composed mainly of neural network dynamics that produce the parameter value to control which mode of program B to produce and when to rewrite memory weights.

In the B programs,

5) We use two layered artificial neural networks. Each neuron in the first layer receives a weighted alpha-blending of inputs of 4 different modes.

6) Each neuron obeys an artificial neural dynamics called chaos neural networks [1, 17] (Appendix A.1).

7) Synaptic connections within the first layer are organized by the Hopfield type [7, 8] construction with a constant decay term (Appendix A.2).

8) Synaptic connections from the first to the second layer and those within the second layer are controlled by the modified Hebbian dynamics [3, 18] (Appendix A.3).

9) Each neuron on the second layer is associated with the set of visual inputs and the set associated with the most activated neural state will be selected for projecting onto the output screen. This is a "winner-takes-all" rule. However, non-selected sets of images are also superimposed in the neural weight. This was intended to reflect the idea that perception is essentially working in parallel. Non selected modes should also commit to make up the system's memory pattern.

10) Each neural state as well as synaptic strength is updated Q times (which is assigned 10 times in this experiment) for each T real time duration. T is ranging from 0.1

second to 1 sec and is determined by a designated neural state of the second layer.

There are several other important features of the MTM architecture.

1. A set of A and B consists of one module. MTM has three such independent modules, where each module has its own projection screen and the memory of those modules will be updated asynchronously.

   The 15 video cameras used in MTM are classified into two kinds: those which always shoot at the same angle (spatially fixed ones) and those which change their shooting directions (freely moving ones) controlled by artificial neural networks. How to combine and edit the video images are directed by program B, while program A controls the parameters. Those moving cameras are also controlled by the output from the neural networks.

   i) Module No. 0 is tentatively called the "unconscious" module as its inputs are from 9 passive video cameras (the camera positions are fixed in the space.)
   ii) Module No. 1 is called the "conscious" module as its inputs are from 2 active and 2 passive video cameras.
   iii) Module No. 2 has no camera inputs. Instead, its inputs are from screen 0 or 1 or from the buffer that accumulates long term images of its own.

   Images from module 0 are sent to screen1 (visualization of unconscious states: passive images) and those from module 1 go to screen 2 (visualization of conscious states: active images). Images from module 2 are sent to screen 3 (which we call episodic memory: bundles of previously produced images from either the conscious or unconscious screen).

2. Most cameras are shooting screen images, so video feedback is ready to happen. Most cameras are zooming the screen images, so they sometimes induce oscillation excitation. It also happens that the same image pattern is circulated for a while among the three screens due to the mutual feedback loop (i.e. one camera shoots the screen from the other camera and its own screen is shot by the other camera, and so on).

3. This network "remembers" the previously received inputs but it doesn't mean that the system can stably retrieve the memory. Actually we know that neural dynamics with a fixed connection strength also produce complex and temporally unstable behavior such as chaotic itinerancy [17]. A synaptic connection between neurons is strengthened when the connected neuronal activities are temporally correlated. This is called the Hebb's learning rule. The point is that this network copies the spatial temporal correlation that exists in its environment into the correlation of synaptic weights in the network.

4. We assume that perceiving something requires changes in the functional state of the underlying neural dynamics. In particular, that which a system has perceived before determines what to perceive next. The above architecture reflects this hypothesis.

2.3 Relationship between MTM and Human Interaction

When people come to the venue, their images are taken by the video cameras, which are then recognized and processed by the neural dynamics. Intake images are explicitly affected by the human movements but also by the projection mode. The 4 projection modes detailed in the previous section have the following effects:

1. Movements are strangely stretched or contracted in the images. For example, when you toss balls into air, the number of balls increases. (1)
2. You see yourself being replicated in the movie. (2)
3. You are moving backwards, in reverse time. (3)
4. Your bodily movement is embedded in the screen at different scales. (4)

Those effects are certainly affecting MTM's behavior as people come to see and interact with it. Its quantitative analysis is a future problem.

In addition, we generate sound images by scanning each screen. We raster scan the frames successively and translate them into sound amplitudes and play the sound by filtering through a prepared frequency spectrum. In this way, the dynamics of visual images are translated into sound patterns, which also affect the human observers. The generated sounds thus co-vary as the visual images change with our programs that stretch, fold, superpose and remove the speed of sound and the frequency spectrum.

3. Observation/Analysis

In order to study the behavior of such a system, our first trial was to observe and make a diary of its everyday behavior from the morning when the museum opened to the evening when the museum closed, for over 67 days. For tracking the behavior, we recorded i) a return map of the neural states, ii) changes of the weight strengths; its average value and the standard deviation, iii) changes of the video images of three screens, iv) long term memory images and v) the internal memory updating schedule.

3.1 Return map analysis

We define a return map of the neural states as the superposition of the two successive neural states in one figure. Due to the characteristic of the neural dynamics, the weight strength of the neural network plays the role of a parameter that controls the dynamic neural state. Too big or too small a weight strength means the network has a stable fixed point (i.e. showing a temporally constant state). Only in the middle values (around 0.25 and 0.75 due to the symmetric structure)

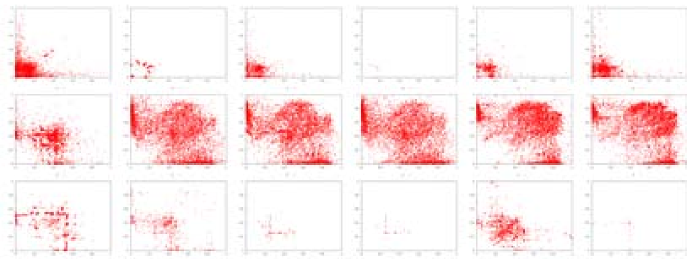

Figure 2. Temporal changes of return maps by randomly picking up a neuron from three modules, respectively for one day.

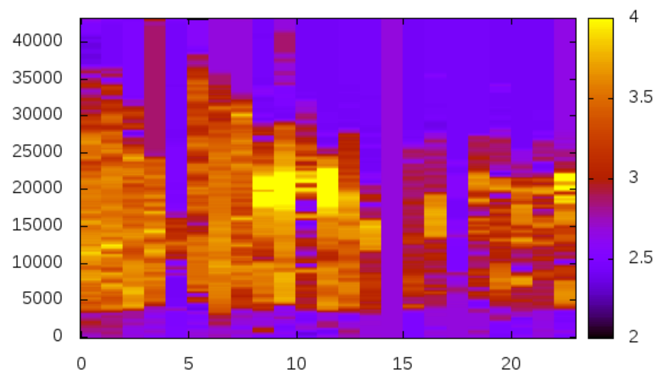

Figure 3. Temporal variations of entropy quantity as a function of dates (numbered from April 25th.) The vertical line is hours from 8am to 10pm. The brighter color indicates the larger entropy.

does the network show complex dynamic behavior such as long term periodicity and chaos.

In producing return maps for every two hours, we observe that the pattern of the return map varies during the day time (figure 2 and figure 3). As for the general tendency, modules 0 and 2 easily lose their complexity in return maps, but module 1 sustains it. Modules 0 and 2 sometimes return a rich pattern after a few hours, and may be indirectly affected by module 1.

3.2 Taking an everyday diary

The first thing to do with our long term 67-day experiment is to calculate and compare the everyday return maps (figure 3). We noticed that everyday maps look similar but not exactly the same (some return maps were more complex than others) and some return maps have much less denser patterns.

We then studied how weather conditions looked and found that the return maps became quiet when it was raining all day long. On the other hand, there were quiet return maps even when the weather was fine.

Since MTM receives its visual inputs from the venue, and since those inputs accumulate in the weights that connect neurons of the first layer, the neural states will gradually go through the bifurcation cascade. In some cases, when a certain attractor is hit by the system, it won't recover the original state, so that it eventually becomes quiet. Whether it will recover by posing some effective visual inputs is not fully understood. As well as the weather condition, the neural states also became quiet after the sun set. The light intensity is the main factor that suppresses the neural state.

Figure 4 shows the change of neural pattern complexity between April 25$^{\text{th}}$ and May 10$^{\text{th}}$. Entropy is computed in the form of Shannon entropy from a time sequence of length 1000 bits. We hypothesized that on the 27$^{\text{th}}$, the museum put a black light shield around MTM which made it less robust against the insufficient day light. Also, as you can see, the entropy is relatively low during rainy days.

This rather simple light condition also limits the MTM dynamics, but whether the venue is busy or not should be a potential factor. On Sunday, more kids come to the venue which may activate MTM's dynamics. Complexity in the visual inputs must be copied as memory complexity of MTM. We computed the weighted average and the standard deviation, tracking from 8 am to 8 pm to see what is responsible for complex behavior. We found a correlation between the dispersion of weight values and the complex time series (figure 5).

3.3 System's subjective time

As we stated in the introduction to this paper, living systems have organized their own time scales driven by the memory structure. This is the memory updating and projection loop used in this system: [visual inputs $\rightarrow$ memory organization $\rightarrow$ dynamics change $\rightarrow$ determining what to project $\rightarrow$]

When to update the memory is determined simply by a neural state multiplied by the parameter TIME (which is assigned 1 sec). If that neural state is suppressed at the lower value, memory is frequently updated. On the other hand, if its value takes the large value, memory is rewritten every 1 sec maximally. We take this memory updating time scale as a candidate of a system's subjective time scale, which is a direct outcome of neural dynamics but indirectly as a function of memory organization.

Figure 6 shows the correlation between a system's time scale as a function of physical time. When there is a plateau, the subjective time goes slower than the physical time scale, and when it becomes a large increasing line, it means that subjective time goes faster. Our 67-day experiment revealed that the total amount of subjective time (i.e. number of memory updates) varies depending on the day. In particular, early acceleration of the subjective time scale is not observed. We speculate that the daylight condition also matters in this accelerating pattern. On the other hand, since this subjective time depends on our system settings, we will test a future possible algorithm for changing time scales, even though our

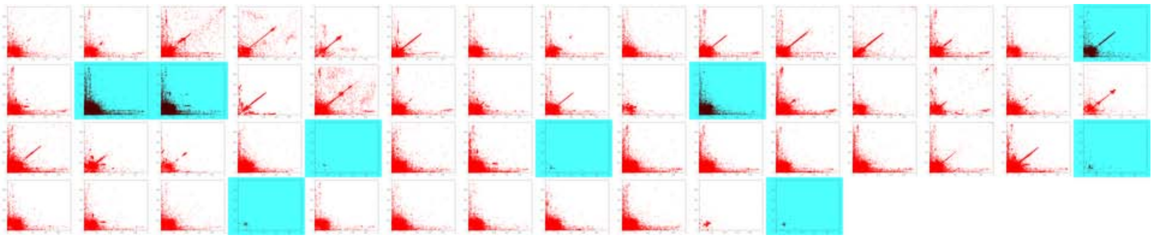

Figure 4. A daily return map from module 0 from March 25th to June 2nd 2010. Blue background indicates a rainy day.

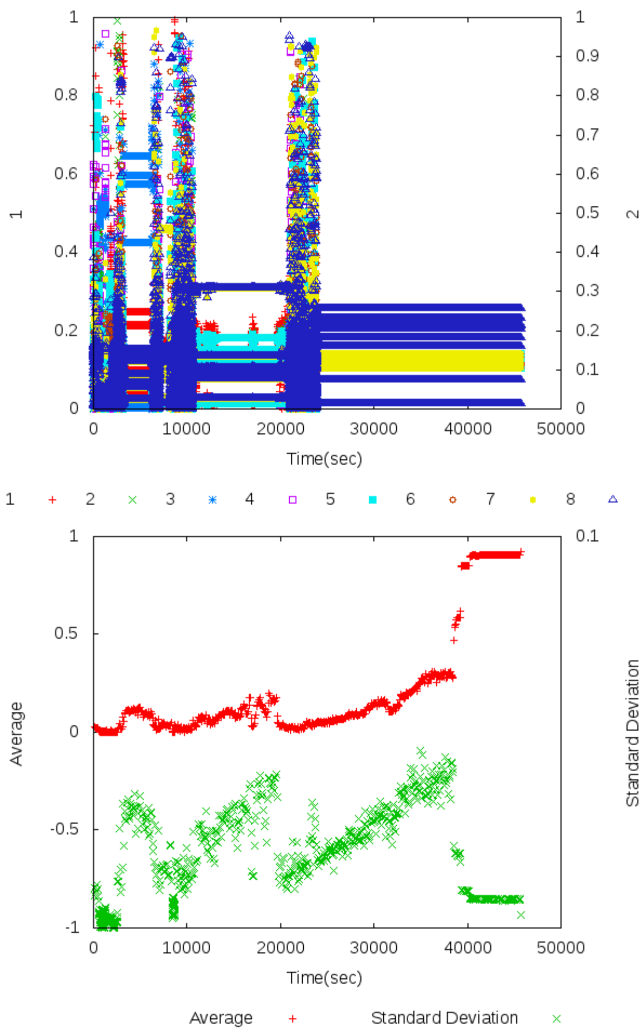

Figure 5. Temporal Changes of neural states (overlaid) (up figure) and average value (in red colors) and deviations (in green colors) of connection strengths (bottom figure) for one day.

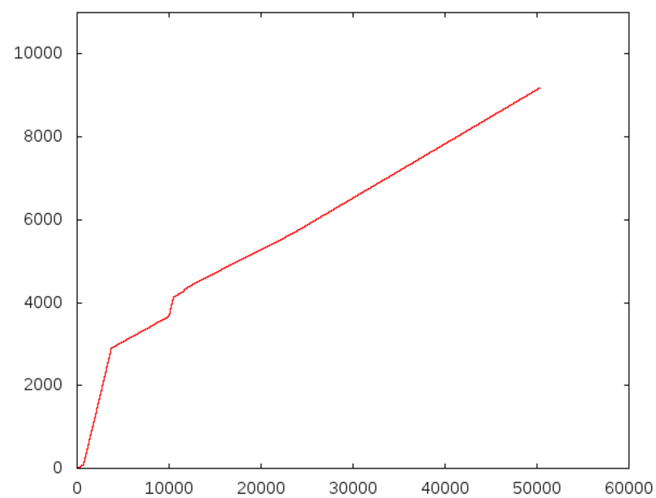

Figure 6. An example showing how the internal clock changes from 8am to 8pm. You can see several step-wise behaviors as a function of Newtonian time.

principle here is that subjective time is a function of memory update timing.

The side effect of this algorithm is that a system becomes more robust against noise. But if one module updates its memory much more frequently than the other, its robustness becomes different between the two. Here the robustness is what we discussed in the introduction.

4. Plans for a sound MTM

Using sound inputs in addition to visual inputs demonstrates how the soundscape is self-organized within MTM. In our pilot study of a sound MTM, I had 3 dancers perform in a designated space (10 m square) where 3 proximity sensors were set. When the dancers came close to the sensors, sound files associated with the sensors were played. At the same time, sound in the space was continuously recorded through three microphones into the MTM memory buffer which is about 10 minutes long. Depending on the neural state of the MTM, the sound from the position of the buffer, designated by the neural state, is replayed several times.

After 10 minutes of free dancing in the space, I let the dancers leave the area and observed how the soundscape auto-progressed in time. MTM replayed the sound fragment from its memory and recorded it at the same time. As a result, the soundscape self-modifies as time elapses.

5. Discussion

MTM is composed of many different dynamics and algorithms. We used the classical Hopfield memory construction, chaos neural net, Hebbian dynamics, video feedback, slit scanning, and so on. This is what I call the “all-in-one” approach. In contrast to the physicist’s minimalism, this all-in-one approach is based on Maximalism. We believe that maximal complexity is necessary to develop conscious states in artificial systems. In particular, MTM is designed for the full complexity of the real world. Differing from the usual meaning of “machine,” MTM becomes sensitive to its environmental changes. It cannot hold the same functionality which is independent from the variety of environmental conditions.

With respect to the notion of self-sustainability, we are concerned with what would be a relevant dynamics that introduces self-sustainability. As we briefly reviewed in the first section, the self-moving oil droplets and the pure Hebbian learning robots provide examples of self-sustainability mechanisms, and MTM adds another possibility. Sometimes MTM fails after time to come back to the “alive state,” but at other times MTM becomes self-sustainable, i.e. it always returns to the “alive state” maintaining its sensitivity to environmental change.

Where does this sustainability come from? From 67 days of observation, we learned the following points.

We should carefully determine where to put video cameras in a space. Module 1 used mobile videos and the network state sustained its complex dynamics almost throughout the day. On the other hand, modules 1 and 2 failed to maintain. Interestingly, that lost complexity was regenerated after 6 hours, i.e. a fixed point attractor such as behavior was destroyed while complex chaos, such as attractor, emerged again. We don’t know of this kind of transition from a fixed point to chaotic one in normal dynamic systems (6 hours is equal to 3600 $\times$ 6 iterations). We deduce that module 1 is helping the other modules to recover, and in this sense sustainability requires several different modules helping each other to achieve total adaptive dynamics (see figure 4). Also, the fact that those modules are asynchronous in real time scales may be an important factor to take into account.

We also find it interesting that MTM tends to self-organize in intermediate time scales. There exist three fundamental time scales in this system. One is the neural state updating, which is the fastest time scale. The second one is the memory accumulation in the first layer. The third one is the Hebbian time scale. However, as we see in figure 5, there are two or three temporal changes about every 10,000 seconds. These intermediate time scales are generated from the total MTM architecture. Likewise, in the circadian rhythm, MTM tends to have a particular time scale under daylight change. However, this rhythm is not observable every day. This is due to the fact that we reset weight values every morning. Defining self-sustainability in terms of sustaining its own time scale is an interesting next step in creating MTM.

All these observations indicate that self-sustainability is about the self-organized time scale underlying MTM’s dynamics. In this experiment, timing of memory updating was attributed to a single neural state, and figure 6 depicts that the system’s internal time scale, i.e. Bergsonian time scale, shows a stepwise behavior as a function of the Newtonian time scale. We interpret this as meaning that the Bergsonian time scale cannot have a fixed minimal length (or we cannot make it infinitesimally small) but should be variable as a matter of memory and perception. This notion of Bergsonian time should be explored and examined in different MTM organizations.

Acknowledgments

I thank Yuta Ogai and Motoi Ishibashi for stimulating arguments and their technical supports. I also thank Kenshu Shintsubo for his camera setup for MTM and Miyuki Kawamura for supporting this project and the sound MTM plans. I also thank Evala and Keiichiro Shibuya for helping the sound effects on MTM.

This project is partially supported by the challenging pilot study by the Ministry of Education, Culture, Sports, Science, and Technology, Japan, I also like to thank Kazunao Abe and the Yamaguchi Center for Arts and Media for giving an opportunity to present MTM for 3 months.

A. Appendix

A.1

A chaotic neural network (of the k-th neuron at time step n) obeys the following equation [1, 17];

$$p^k_{n+1} = r_1 p^k_n + \left(1 - \frac{1}{1 + e^{(q^k_n - p^k_n)/\beta}}\right) \tag{1}$$

where q^k_n is defined as

$$q^k_n = r_2 \sum_{j \neq k} w_{kj} p_j \tag{2}$$

and w_{kj} is the synaptic onnection coupling from the j-th neuron to the k-th neuron. This value q^k_n dynamically controls the stability of this neural dynamics additional to the static parameters; r_1, r_2 and β.

A.2

Hopfield proposed the following memory embedding in the connection strength [7, 8]:

$$\Delta w_{ij} = \sum_{s=1}^{M} (2V^s_i - 1)(2V^s_j - 1) \tag{3}$$

where V^s_i denotes the pixel value of the spatially re-normalized site i which is normalized in between 0 and 1, and the total number of memory is M. Here, we coarse-grain 1024×768 pixels into 10×10 re-nomarlized pixels, so that the value V_i^s is also an averaged value of the original pixels contained in that site.

In addition to this, we introduced the forgetting parameter r (< 1) so that w_{ij} changes over time.

$$w_{ij}^{n+1} = r w_{ij}^n + \Delta w_{ij}^n \quad (4)$$

A.3

Hebbian dynamics [3, 18] uses the idea that the connection strength ω changes in proportional to the cross correlation of the pre and post synaptic neural states, p_k and p_j. We used the modified version of it.

$$\frac{d\omega_{kj}}{dt} = \gamma\left(\frac{dp_k}{dt} p_j - \alpha \omega_{kj} p_j^2\right) \quad (5)$$

Namely, the strength is not simply proportional to the correlation between pre- post neural states, but is proportional to the correlation between the pre synaptic neural state and the rate of the post synaptic neural state change. By using this formula, the synaptic strength becomes sensitive to the sensitivity of post-synaptic neural state responding to the pre-synaptic state. Here the second term is introduced in order to avoid bound the weight strength.

References

[1] K. Aihara, T. Takabe, and M. Toyoda. Chaotic neural networks. Phys. Lett. A 144(6/7), 333–340.

[2] G. Deleuze. *Bergsonism* (Zone Books 1990)

[3] D. O. Hebb. The Organization of Behavior. Wiley, New York, 1949.

[4] M. M. Hanczyc, and T. Ikegami. Chemical Basis for Minimal Cognition. In *Artificial Life* Vol. 16, No. 3, (2010) 233–243

[5] M. Hanczyc, T. Toyota, T. Ikegami, N. Packard, and T. Sugawara. Chemistry at the oil-water interface: Self-propelled oil droplets In J. Am. Chem. Soc.; (2007); 129(30) 9386–9391

[6] J. Hubert, E. Matsuda, E. Silvermann, and T. Ikegami. A Robotic Approach to Understanding Robustness. In *Proceedings of the 3rd Mobiligence Conference* (2009) 361–366.

[7] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. J. Proc Natl. Acad. Sci. USA 79 (1982) 2554–2558;

[8] J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. 81 (1984) 3088-3092.

[9] T. Ikegami. Simulating Active Perception and Mental Imagery with Embodied Chaotic Itinerancy. Journal of Consciousness Studies Vol.14 (2007) 111–125.

[10] T. Ikegami, and G. Morimoto. Chaotic Itinerancy in Coupled Dynamical Recognizers. Chaos 13 (2003) 1133–1147.

[11] T. Ikegami, and M. M. Hanczyc. The search for a First Cell under the Maximalism Design Principle. J. Technoetic Arts, 2009, 153–164.

[12] T. Ikegami, and K. Suzuki. From Homeostatic to Homeodynamic Self, BioSystems, 91 (2008) 388–400

[13] T. Ikegami. Rehabilitating Biology as a Natural History. Adaptive Behavior, Vol.17, (2009) 325–328.

[14] E. Jen. Stable or robust? what's the difference? In *Robust design: a repertoire of biological, ecological and engineering case studies*. Oxford University Press. 2005.

[15] H. Kitano. Towards a theory of biological robustness. Molecular Systems Biology, 3: (2007) 137.

[16] B. Libet. *Mind Time* (Harvard University Press, 2004).

[17] H. Nozawa. A neural network model as a globally coupled map and applications based on chaos. Chaos, 2 (1992) 377–386.

[18] E. Oja. Simplified neuron model as a principal component analyzer. Journal of Mathematical Biology,15(3), (1982) 267–273.

[19] E. Silverman, and T. Ikegami. Robustness in Artificial Life. Int. J. Bio-Inspired Computation, Vol. 2, No. 3, (2010) 197–203

[20] N. Wiener. *Cybernetics Cybernetics, Second Edition: or the Control and Communication in the Animal and the Machine* The MIT Press; 2 edition (1965).