

Making a Robot Dance to Music Using Chaotic Itinerancy in a Network of FitzHugh-Nagumo Neurons

Jean-Julien Aucouturier, Yuta Ogai, Takashi Ikegami

Department of General Systems Studies
Graduate School of Arts and Sciences, The University of Tokyo
3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
aucouturier@gmail.com, {yuta,ikeg}@sacral.c.u-tokyo.ac.jp

Abstract. We propose a technique to make a robot execute free and solitary dance movements on music, in a manner which simulates the dynamic alternations between synchronisation and autonomy typically observed in human behaviour. In contrast with previous approaches, we preprogram neither the dance patterns nor their alternation, but rather build in basic dynamics in the robot, and let the behaviour emerge in a seemingly autonomous manner. The robot motor commands are generated in real-time by converting the output of a neural network processing a sequence of pulses corresponding to the beats of the music being danced to. The spiking behaviour of individual neurons is controlled by a biologically-inspired model (FitzHugh-Nagumo). Under appropriate parameters, the network generates chaotic itinerant behaviour among low-dimensional local attractors. A robot controlled this way exhibits a variety of motion styles, some being periodic and strongly coupled to the musical rhythm and others being more independent, as well as spontaneous jumps from one style of motion to the next. The resulting behaviour is completely deterministic (as the solution of a non-linear dynamical system), adaptive to the music being played, and believed to be an interesting compromise between synchronisation and autonomy.

1 Introduction

Music makes people want to move - either in imagination or actually, as in dance. Quantitative psychological investigations reveal that humans associate gestural movements to music with remarkable consistency. When asked to translate music into free drawings, listeners systematically associate sound patterns composed of a percussive onset followed by a long decay with strokes composed of a steep slope followed by a long descent [Godoy et al., 2006]. Clarinetists are found to often perform semi-unconscious bell shape movements with the tip of their instrument, which boundaries correspond to those of musical phrases, and which amplitudes are dynamically modulated with the music's rhythmic and metric interpretation [Wanderley et al., 2005].

However, the same studies often reveal (or concede) that such directly interpretable mappings from sound to movement are not always predictable, and often found to vary within a given task. Movements without any clear correspondance with music are often performed in alternation with more interpretable gestures. Some types of clarinetists movements occur throughout an entire performance regardless of the change in character of the associated music. Even cyclic movements such as regular weight transfer between forward and back may not be synchronized with the piece’s rhythm, or depend on extrinsic physiological constraints (such as breathing rate). More generally, musicians and dancers exhibit an intriguingly fluid use of gestural patterns, “continuous but not repetitive” [Wanderley et al., 2005], with successive and seemingly unpredictable switches of attachment and detachment to the auditory stimulus.

Designing entertainment systems that exhibit such dynamic compromise between short-term synchronisation and long-term autonomous behaviour is key to maintain an interesting relationship between a human and an artificial agent in the long-term [Pachet, 2004]. However, this remains largely unaddressed in the recent academic and industrial efforts to design dancing robots. One common strategy is to design manually a number of dance presets (i.e. a fixed sequence of motor commands), which are then rendered to a given piece of music by adapting the execution speed of the sequence to the musical tempo (automatically extracted from the audio signal) [Goto, 2001]. The approach has merits, notably a convincing effect of synchronisation, but typically fails at sustaining long-term interest, since the dance repertoire of the robotic is rapidly exhausted and frequent patterns begin to reoccur without any variation. A more evolved approach relies on building imitative behaviour in the robot, which uses typically vision sensors to reproduce movements taught by a human [Nakazawa et al., 2002]. Behavioral studies [Michalowski et al., 2007] show that even passive rhythmic imitation by a robot can generate interesting patterns of interactions with human users (teaching, turn-taking, etc.). However, programming robots to initiate such interactions modes autonomously is still in the domain of speculation. Taking inspiration from the physiology of mirror neurons, [Tanaka and Suzuki, 2004] propose e.g. to use a learning model (Recurrent Neural Network with Parametric Biase) to switch between movement patterns dynamically stored in memory. Finally, richer interaction is often believed to come from physical contact between dance partners, and some recent research addresses the difficult motor control of a robot dancer’s haptic interaction with a human [Kosuge et al., 2003].

In this work, we propose a technique to make a robot execute free and solitary dance movements on music, in a manner which simulates the dynamic alternations between attachment/detachment typically observed in human behaviour. In contrast with previous approaches, we preprogram neither the dance patterns nor their alternation, but rather build in basic dynamics in the robot, and let the behaviour emerge in a seemingly autonomous manner. To this aim, we make use a special type of chaotic dynamics, namely chaotic itinerancy (CI). CI is a relatively common feature in high-dimensional chaotic systems, which shows itinerant behaviour among low-dimensional local attrac-

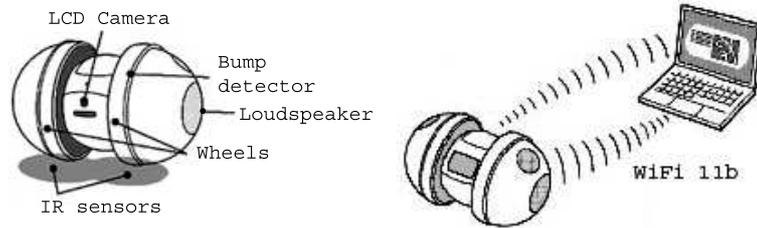


Fig. 1. Our robotic platform, the MIURO manufactured by ZMP Inc, is a two-wheeled musical player equipped with an iPod mp3 player interface and a set of loudspeakers. Wheel velocities can be controlled in real-time through wireless communication with a computer. (*Illustration courtesy of ZMP Inc.*)

tors through higher-dimensional chaos. Recently, CI was proposed to model many exploratory behaviours in living systems, such as insect flight trajectory [Takahashi et al., 2007], neurodynamics in the rat olfactory system [Kay, 2003], or attachment/detachment mechanisms in conscious states [Ikegami, 2007]. In each of these domains, CI appears as an elegant model to describe seemingly spontaneous switches between exploratory/motion styles, with alternations between local periodic patterns and global exploratory wanderings.

Here, we generate CI with a dynamical system composed of a network of artificial spiking neurons, each controlled by a biologically-inspired model (FitzHugh-Nagumo (FHN)). FHN neurons are connected to one another randomly with time delays¹. We showed in a recent experiment [Ikegami, 2007] that such an architecture could generate CI when sensorimotor coupling exists (i.e. when the network outputs influence its input). Here, we demonstrate that this still holds without any coupling, i.e. when the network is fed with a sequence of pulses corresponding to the beats of the music being danced to, and its output is converted to motor command in real-time. We find that chaotic itinerancy in the network output can be converted to control a robot to a variety of motion styles, some being periodic and strongly coupled to the musical rhythm and others being more independent, as well as spontaneous jumps from one style of motion to the next. The resulting behaviour is completely deterministic (as the solution of a non-linear dynamical system), adaptive to the music being played, and believed to be an interesting compromise between synchronisation and autonomy.

We demonstrate the system using a relatively simple vehicle-like robot, the MIURO manufactured by ZMP Inc.² (Figure 1). Note that the dance movements performed by the robot are two-dimensional trajectories controlled with

¹ note that this involves no learning: we only use the network to specify a non-linear dynamical system, using fine-tuned parameters

² ZMP Inc., 10F Aobadai Hills 4-7-7 Aobadai, Meguro Ward Tokyo, Japan. Miuro Homepage: <http://miuro.com>. ZMP Official Homepage <http://www.zmp.co.jp>

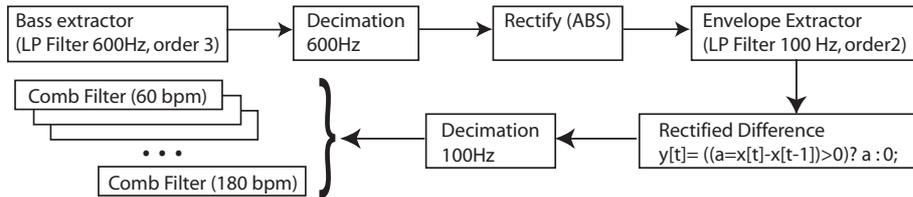


Fig. 2. Block diagram of the beat-tracking algorithm used to send pulses to the neural network in real-time correspondence with the music.

wheel speed, therefore well below the complexity and expressivity aimed at other proposals closer to human kinesiology [Tanaka and Suzuki, 2004].

Finally, note also that this work is not the first to propose to use chaotic dynamics to simulate dance movements. Most notably, [Bradley and Stuart, 1998] exploit the ever-changing trajectories of symbolic states around a common attractor (Rössler) to generate variations on ballet choreographic movements. Chaos is used to explore the compromise between novelty and consistency: chaotic dependency on initial conditions guarantees that each variation is different from the original, while the attractor structure maintains consistency between the two. In contrast, in the current paper, we use chaos to compromise between attachment and detachment to the auditory stimulus, in order to simulate autonomous dance movements to music.

2 System Description

2.1 Audio Analysis

The audio front-end for our system is responsible for sending pulses to the neural network, in real-time correspondence with the beats of the music being danced to. We use a stripped-down implementation of the beat-tracking algorithm introduced in [Scheirer, 1998]. Figure 2 shows a block diagram of the algorithm. Buffers of audio are sent to the algorithm at regular time intervals (typically 5 ms, see Section 3). Each buffer of audio is processed by successively filtering out all frequencies above 600Hz, then extracting the amplitude envelope by a succession of drastic decimations and low-pass filtering. The signal is then fed to a filterbank of comb-shaped filters (or resonators), each tuned to a specific tempo (from 60 to 180 beat-per-minutes). The output of a comb filter of period T is given by:

$$y(t) = \alpha y(t - T) + (1 - \alpha)x(t) \quad (1)$$

where $x(t)$ is the input signal of the filter, and $|\alpha| < 1$ is a gain factor regulating the respective importance of novelty vs memory (here we put $\alpha = 0.9$). Each resonator has an output buffer of the last T samples $y(t)$ in the past.

After each processed buffer, the algorithm selects the resonator which output has the highest energy (this gives the current tempo), finds the position of the latest maximum in the resonator’s output buffer (this gives the position of the latest beat), and if this position is different from that of the previous beat (i.e. this is a *new* beat), sends an impulse to the neural network with fixed width (typically 50ms) and a height proportional to the beat’s amplitude (computed as the root-mean-square of the original audio signal in a 50 ms window around the beat position, normalized in $[0, 1]$).

The system is not tuned for consistency: nothing is done to prevent switches between locally optimal solutions. Musical passages with ambiguous or complex rhythm typically generate rapid switches between correlated bpm estimates (e.g. with integer ratio), which are then processed by the network to result in greater complexity than with more stable rhythms. Note that an abundant literature exists to extract robust tempo and beat estimates which prevent such switches, should it be needed (see e.g. [Gouyon et al., 2006] for a recent review).

2.2 FitzHugh-Nagumo Neuron

The FitzHugh-Nagumo (FHN) model is a simplification of the Hodgkin-Huxley model describing the depolarization of a neural membrane in a squid axon [Fitzhugh, 1961]. Each neuron is a coupled system of a fast variable u responsible for the excitation of membrane potential and a slow variable ω controlling its refractory state:

$$\frac{du}{dt} = c\left(u - \frac{u^3}{3} - \omega + I(t)\right) \quad (2)$$

$$\frac{d\omega}{dt} = a + u - b\omega \quad (3)$$

where $I(t)$ is an input signal (in our case a pulse train), and we take $a = 0.7$, $b = 0.8$ and $c = 10$. The neuron is said to overshoot (or generate a spike) when its output u reaches above 0. In this work, we integrate the $\left\{\frac{du}{dt}, \frac{d\omega}{dt}\right\}$ system with 4th-order Runge-Kutta [Press et al., 1986] from some initial conditions $u_0 = -1.2$, $\omega_0 = -0.62$.

The dynamic properties (attractor, bifurcations) of the FHN equations have been studied intensively (see e.g. the review by [Kostova et al., 2004]). It is known for instance that the membrane spiking behaviour is well controlled by the periodicity of the input spike train $I(t)$. Figure 3 shows the joint histogram of the inter-spike intervals (counted in numbers of Runge-Kutta update steps) in the input and output spike trains of a FHN neuron. It appears that for large ranges of input periods, the output spike train is periodic with a proportional period. However, chaotic, non periodic responses also occur for certain input periods. Finally, fast input spikes (with periods smaller than 100 steps) do not generate any spike in output, as the neuron is caught in permanent refractory state.

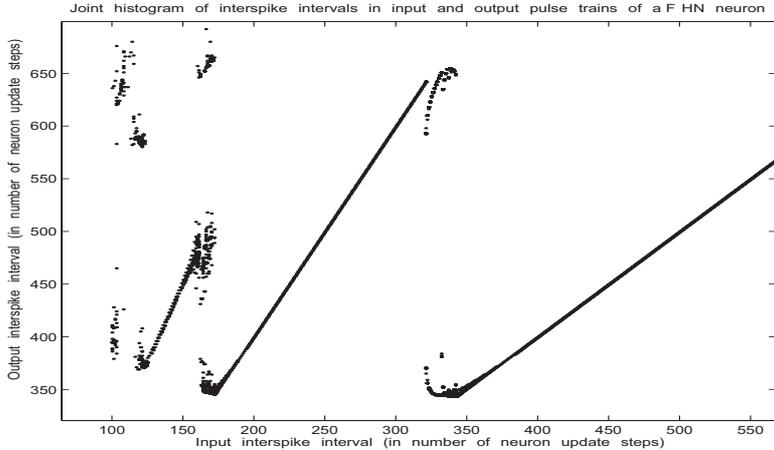


Fig. 3. Joint histogram of the inter-spike intervals (counted in numbers of Runge-Kutta update steps) in the input and output spike trains of a FHN neuron. Output periods are measured over 100 periods of the input train, for each input period

2.3 Network architecture

The robot is equipped with a sparse network of FHN neurons, randomly connected with a probability $p_c = 0.2$. Neurons are connected to one another with time-delayed connections of two types “fast” $\Delta_f = 100$ and “slow” $\Delta_s = 300$, decided randomly upon initialisation with probability $p_{fast} = 0.6$. When a neuron overshoots ($u > 0$), a pulse (with a given width W_p and height H_p) is transmitted to the neurons to which it is connected, with the appropriate time delay. Coincident pulses at a recipient neuron are not integrated, and equivalent to a single pulse. In the experiments reported here, we use 12 neurons divided into 3 groups:

- 4 *sensory* neurons which all receive the same input $I(t)$ from the audio analyser, namely a pulse train with local periodicity corresponding to the local musical tempo, width $W_p^{sens} = 10$ and height H_p^{sens} corresponding to the audio signal’s energy around the beat.
- 4 *internal* neurons, which generate pulses with $W_p^{int} = 10$ and $H_p^{int} = 0.7$
- 4 *motor* neurons, which generate pulses with $W_p^{mot} = 300$ and $H_p^{mot} = 0.7$.

All neurons are equipped with FHN dynamics, integrated with Runge-kutta as described above.

2.4 Motor output

Finally, the motor neurons collaborate to constitute the motor commands (left and right wheel velocities) sent to the robot

$$V_L(t) = \tanh(h_1(t) + h_2(t)) \quad (4)$$

$$V_R(t) = \tanh(h_3(t) + h_4(t)) \quad (5)$$

where $h_i(t)$ is a test function holding on the output *spike train* (and not output *variable* u) of the i^{th} motor neuron, returning 1 if a spike is active at time t (i.e. was generated within W_p time steps in the past), else 0. Note that the time scale corresponding to iterator t needn't be the same time scale as the network time scale: one may downsample the motor output spike train before computing the motor commands (see Section 3).

Finally, the trajectory of the robot can be simulated on computer using the following approximations:

$$\frac{dx}{dt} = g_1(V_L(t) + V_R(t)) \cos \theta(t) \quad (6)$$

$$\frac{dy}{dt} = g_1(V_L(t) + V_R(t)) \sin \theta(t) \quad (7)$$

$$\frac{d\theta}{dt} = g_2(V_L(t) - V_R(t)) \quad (8)$$

where x, y is the space displacement vector and θ the heading direction. We use $g_1 = 50$ and $g_2 = 10$.

3 The 3-time-scale problem

Information in the network, from the input beat pulse to the output of the motor neurons can be processed at different time scales, upon which hold the following constraints:

1. As seen in Figure 3, input pulse trains with periods smaller than 100 neuron time steps do not generate pulses in output (due to the slow refractory dynamics of the FHN neurons). Typical beats in music occur every 500-1000ms (i.e. 60 to 120 beats-per-minute). Neurons should therefore update faster than every 5-10ms (we call this time scale *network time scale* (NTS), and take it equal to 5 ms)
2. Because of hardware limitations, it is generally impossible to update the speed of a robot at a rate faster than a few tens of milliseconds. The practical limitation we observed for our specific platform (Miuro) was 100ms. We call this time scale the *robot time scale* (RTS).
3. Also from Figure 3, output pulses, generated when neurons overshoot, are sent with period roughly equivalent to the neurons' input, i.e. at longest 500-1000ms. The width of such output pulses W_p^{mot} should therefore be smaller than a few 1000ms, else they would concatenate to continuous output, and command the robot to a straight line. We take here $W_p^{mot} = 100$ NTS steps (i.e. 500 ms).

It follows that, if equ. 4-5 are processed at the RTS, output pulses are downsampled from W_p^{mot} NTS steps down to $\frac{W_p^{mot} NTS}{RTS}$ steps, which is in the order of 1-5. This turns out to be too crude, especially since we are interested in chaotic dynamics resulting in fine overlaps between pulses. Therefore, the activation of

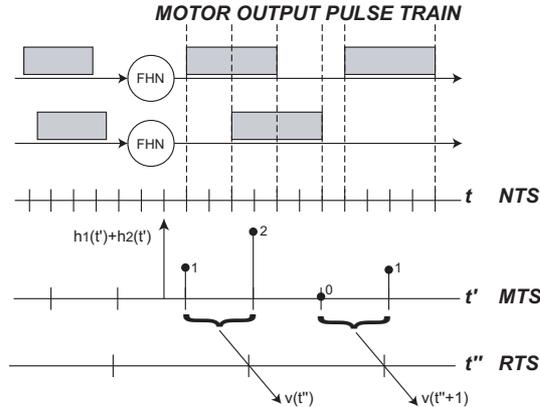


Fig. 4. Converting the network’s chaotic dynamics into suitable motor commands of the robot requires 3 simultaneous time scales. Output pulse train of the motor neurons are generated at NTS (e.g. 5ms), sampled at MTS (e.g. 30ms), and the corresponding speeds interpolated and sent to the robot at RTS (100ms).

the robot (under the RTS constraints) using chaotic dynamics at a time scale NTS constrained both by the dynamics of the FHN neurons and the typical time scale of the environment (music) requires the introduction of a third, intermediary time scale (the *motor time scale* MTS), at which to sample the output pulse train of the motor neurons. Figure 4 illustrates this 3-time-scale process: Motor output pulses are generated by updating the network at NTS. Pulses are sampled to generate a new set of wheel velocities at MTS. Then velocities are interpolated (averaged) over the RTS period, and sent to the robot’s. MTS should be intermediary between NTS and RTS:

- If MTS is too close to RTS, we downsample the output pulses too much, and lose most of the chaotic dynamics happening at NTS.
- If MTS is too close to NTS, we interpolate among too many consecutive velocity vectors, and smooth the resulting trajectory too much.

4 Results and Experiments

We report here on some qualitative aspects of the robot trajectory generated with our system. This report is supplemented by online material³ including some parameter effects, animations of simulated orbits synchronized to music, as well as videos of real robot experiments.

4.1 Chaotic itinerancy

Figure 5 shows successive steps of a simulation of the robot trajectory (using eq. 6-8), for a given music piece. Animations of the motion of the miuro robot

³ <http://www.jj-aucouturier.info/docs/miuro>

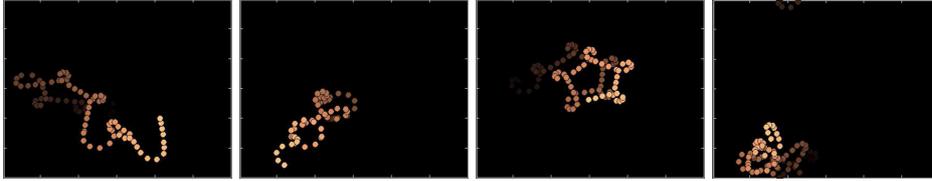


Fig. 5. Simulation of the robot trajectory in the (x,y) plane for a given music piece. Each figure is an overlay of 100 successive robot time steps (at $RTS = 100ms$), and successive figures correspond to different stages of the simulation (every 25 sec.). The trail shows typical chaotic itinerancy behaviour.

can be found in the online supplementary material. Each figure is an overlay of 100 successive RTS steps. Time scales were chosen as $NTS = 5ms$, $MTS = 30ms$, $RTS = 100ms$. The orbit shows typical chaotic itinerancy behaviour, with locally quasi-periodic trails in attractors of various shapes, and abrupt transitions from one attractor to the next through higher-dimensional chaos. We observe that different songs generate different types of orbits and styles of motion. Fine variations of time scales create considerable variation of the robot motor behaviour, with disappearance of CI for limit values (please refer to online supplementary material).

4.2 Attachment and detachment

Although not designed explicitly to this aim, the robot trajectories exhibit some degree of correlation with the musical input. First, we observe that transitions from one attractor to the next are generally triggered by sudden changes of periodicity in the network input (which in turn often correspond to changes of rhythm in the music). Second, thanks to fast signalling from sensory to motor neurons, quasi-periodic motion generated by most of the attractors tends to happen with a period correlated to the input bpm. However, such correlations are difficult to quantify, and part of their experience may be more subjective than objective. The reader is suggested to refer to online supplementary material.

4.3 Robotic Implementation

Our robotic platform, the MIURO manufactured by ZMP Inc, is a two-wheeled musical player equipped with an iPod mp3 player interface and a set of loudspeakers (Figure 1). It has built-in autonomous behaviour, which can be overridden by sending duplets of instantaneous speed values (left and right wheel), through wireless communication with a computer. In our implementation, musical playback is initiated on the robot using the manufacturer's remote control. Music is played by the embedded mp3 player on the robot, and rendered on the robot's loudspeakers. Simultaneously, the robot sends a notification, including an identifier of the song being played, to the client PC via wireless connection.

The iPod music database is duplicated on the PC. Upon reception of the robot playback notification, real-time beat-tracking of the corresponding audio item is started (on the PC), and beats are sent to the network (also running on PC) as they are found (quantized to NTS). After propagation through the network, output pulse trains are converted to wheel velocities, and sent back via wireless to the robot at rate RTS. When playback is finished, no more beats are found by the audio analysis module, and the network quickly converges to zero output, putting the robot to halt. The system was demonstrated in public in Apple Store Ginza, Tokyo, Japan on May 31st, 2007 (see online supplementary material for a video footage).

5 Conclusion

When excited with a pulse train corresponding to beats in a musical signal, a network of FitzHugh-Nagumo neurons is able to generate chaotic itinerancy dynamics. Using a 3-time-scale architecture, the output of the network can be converted to motor commands able to drive the trajectory of a robot in real-time. The resulting dance alternates in a seemingly autonomous manner between a variety of motion styles, some being periodic and strongly coupled to the musical rhythm and others being more independent. This illustrates that interesting compromises between synchronisation and autonomy can emerge from appropriate non-linear dynamics, without requiring patterns and their alternations to be programmed a priori.

6 Acknowledgements

This work was partially supported by the European Community project ECAgent (IST-1940) and a Postdoctoral Fellowship of the Japanese Society for the Promotion of Science. The authors thank the ZMP team for their willingness for collaboration and the technical help provided with robot miuro.

References

- [Bradley and Stuart, 1998] Bradley, E. and Stuart, J. (1998). Using chaos to generate variations on movement sequences. *Chaos*, 8:800–807.
- [Fitzhugh, 1961] Fitzhugh, R. (1961). Impulses and psychological states in theoretical models of nerve membrane. *BioPhys. Journal*, 1(1):445–466.
- [Godoy et al., 2006] Godoy, R., Haga, E., and Jensenius, A. R. (2006). Exploring music-related gestures by sound-tracing. - a preliminary study. In *2nd ConGAS International Symposium on Gesture Interfaces for Multimedia Systems, Leeds (UK)*.
- [Goto, 2001] Goto, M. (2001). An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171.
- [Gouyon et al., 2006] Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. (2006). An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1832–1844.

- [Ikegami, 2007] Ikegami, T. (2007). Simulating active perception and mental imagery with embodied chaotic itinerancy. *Journal of Consciousness Studies*, 14(7):111–125.
- [Kay, 2003] Kay, L. (2003). A challenge to chaotic itinerancy from brain dynamics. *Chaos*, 13(3):1057–1066.
- [Kostova et al., 2004] Kostova, T., Ravindran, R., and Schonbek, M. (2004). Fitzhugh nagumo revisited: Types of bifurcations, periodical forcing and stability regions by a lyapunov functional. *International Journal of Bifurcation and Chaos*, 14(3):913–925.
- [Kosuge et al., 2003] Kosuge, K., Hayashi, T., Hirata, Y., and Tobiyama, R. (2003). Dance partner robot -ms dancerr-. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [Michalowski et al., 2007] Michalowski, M. P., Sabanovic, S., and Kozima, H. (2007). A dancing robot for rhythmic social interaction. In *Proceedings of HRI*.
- [Nakazawa et al., 2002] Nakazawa, A., Nakaoka, S., and Ikeuchi, K. (2002). Imitating human dance motions through motion structure analysis. In *Proceedings of International Conference on Intelligent Robots and Systems*.
- [Pachet, 2004] Pachet, F. (2004). *On the Design of Flow Machines*. The Future of Learning. IOS Press.
- [Press et al., 1986] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. (1986). *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, Cambridge.
- [Scheirer, 1998] Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustic Society of America*, 103(1):588–601.
- [Takahashi et al., 2007] Takahashi, H., Horibe, N., Ikegami, T., and Shimada, M. (2007). Analyzing house flies exploration behavior with ar methods. *Journal of the Japanese Phycis Society (submitted)*.
- [Tanaka and Suzuki, 2004] Tanaka, F. and Suzuki, H. (2004). Dance interaction with qrio: A case study for non-boring interaction by using an entrainment ensemble model. In *Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication*.
- [Wanderley et al., 2005] Wanderley, M. M., Vines, B., Middleton, N., McKay, C., and Hatch, W. (2005). The musical significance of clarinetists’ ancillary gestures: An exploration of the field. *Journal of New Music Research*, 34(1):97–113.