Movement generation using dynamical systems : a humanoid robot performing a drumming task

Sarah Dégallier
School of Computer and Communication Sciences
EPFL, Ecole Polytechnique Federale de Lausanne
EPFL-IC-ISIM-GRIJ, Station 14
CH 1015, Lausanne, Switzerland
Email : sarah.degallier@epfl.ch

Cristina P. Santos
Industrial Electronics Department
University of Minho
Campus de Azurem
4800-058, Guimaraes, Portugal
Email : cristina@dei.uminho.pt

Ludovic Righetti, Auke Ijspeert
School of Computer and Communication Sciences
EPFL, Ecole Polytechnique Federale de Lausanne
EPFL-IC-ISIM-GRIJ, Station 14
CH 1015, Lausanne, Switzerland
Email : ludovic.righetti@epfl.ch, auke.ijspeert@epfl.ch

***Abstract*— The online generation of trajectories in humanoid robots remains a difficult problem. In this contribution, we present a system that allows the superposition, and the switch between, discrete and rhythmic movements. Our approach uses nonlinear dynamical systems for generating trajectories online and in real time. Our goal is to make use of attractor properties of dynamical systems in order to provide robustness against small perturbations and to enable online modulation of the trajectories. The system is demonstrated on a humanoid robot performing a drumming task.**

I. Introduction

While excellent progress has been made on designing efficient controllers for trajectory following in humanoid robotics, the problem of generating the trajectories themselves is still a complex, unsatisfactory solved problem. One of the main difficulties is that the trajectory generation problem is highly task-dependent and often requires extensive knowledge of the task to be solved. Some approaches use hand-coded trajectories or pre-recorded human trajectories to generate trajectory templates. Other approaches, e.g. in locomotion control, use stability criteria to do online trajectory modulation. But most approaches have significant difficulties when the environment is dynamic and partially unknown.

In this article, we explore an approach that uses dynamical systems, i.e. systems of differential equations, for doing online trajectory generation. See [1], [2], [3], [4], [5], [6], [7] for related work. Dynamical systems can be designed to have interesting attractor properties which makes them well-suited for trajectory generation. These properties include : intrinsic robustness against small perturbations, low computation cost which is well-suited for real time, possibility to change parameters on the fly (i.e. to do online modulation), possibility to synchronize with external signals, and possibility to integrate sensory feedback terms. A particularity of dynamical systems is that, instead of encoding a trajectory explicitly, they encode a whole state space and its time evolution. This means that the system must be integrated over time to generate the trajectory (i.e. the trajectory can not be instantaneously extracted out of the system), and that it encodes more than just the trajectory, since it also encodes how the trajectory evolves after a transient perturbation. This property makes dynamical systems difficult to design (i.e. the problem of how to encapsulate a trajectory into a dynamical system is difficult), but at the same time makes them interesting to deal with dynamic and partially unknown environments (i.e. once a dynamical system is properly designed, it can be robust enough to deal with a dynamic environment and to rapidly dampen out perturbations).

In this article, we address the problem of designing a dynamical system that can generate trajectories that have both discrete and rhythmic components, an issue that has so far received little attention. Typical examples requiring such trajectories include visually-guided locomotion, e.g. being able to rhythmically move limbs while making discrete adjustments for placing the feet at specific locations, and drumming.

While our goal is to apply our dynamical systems approach to the control of locomotion and movements in general, in this article we will focus on a simple drumming task. We chose this task as a starting point because it requires several important features of movement control, notably timing, synchronization and accuracy, without being too complex (e.g. no need to keep balance, avoid obstacles, etc., as in locomotion for instance). Furthermore it is a good example of movements that require both rhythmic and discrete components.

Drumming with dynamical systems has been studied before, for instance to explore synchronization with an external signal [8] and to learn complex rhythmic patterns by demonstration [9], but, to the best of our knowledge, not in the framework of the superposition of discrete and rhythmic movements. The novelty of the system presented here is inherent to the fact that the superposition of, and the switching between, rhythmic and discrete mode is achieved by a single simple system. The simplicity of the current system allows an easy control of the different patterns, which will be of crucial importance when considering autonomous switching between the modes. However, it is restrictive in the sense that the different possible patterns of motion are limited, compared to, for instance, the non linear filtering technique used by Ijspeert, Nakanishi and Schaal [6].

In our experiment, a Hoap-2 humanoid robot is controlled to play a given score on a drum set composed of two bongos and a cymbal. A dynamical system is designed with two components per degree of freedom of the robot arms : (1) a discrete system with a single point attractor which generates

discrete trajectories towards a goal and (2) a rhythmic system (a nonlinear oscillator) which generates periodic movements of controlled amplitude and frequency around the discrete trajectories. By transforming the drumming scores into time-varying parameters controlling the dynamical systems, drumming trajectories can be generated in real time. Couplings between the different degrees of freedom ensure that the arms stay synchronized. Interesting properties of the system include : (1) the fact that drumming trajectories can be generated for arbitrary drumming scores (i.e. no need to re-design the system for a given score), (2) the trajectories smoothly superpose, and/or switch between, discrete and rhythmic behavior, and (3) the possibility of online modulation of the trajectories. Since our system is partly inspired from the biological concept of central pattern generators (CPGs), i.e. neural networks capable of producing coordinated rhythmic output signals without rhythmic inputs [10], we will call it a CPG.

In the rest of the article, we will first present the dynamical system we designed and its properties (Section 2). We then present the technical setup for the application to drumming (Section 3). In Section 4, we present our results, both theoretical and empirical. We conclude by presenting possible improvements to the system (Section 5).

II. Dynamical System

In this section we present our model of the generic CPG we use to control motion for one degree of freedom (DOF). First, we present in detail the architecture of the CPG which is made of discrete and rhythmic primitives and then we discuss the intrinsic properties of the system that makes it suitable to generate more complex movements and online trajectories modulation.

A. Architecture of the CPG

The mechanism of control of a generic CPG is illustrated on fig. 1. The system is built on the hypothesis that complex movements can be generated through the superposition and sequencing of simpler motor primitives implemented as dynamical systems. In particular, our system is made of sets of motor primitives which implement dynamical goal directed and also rhythmic movements.

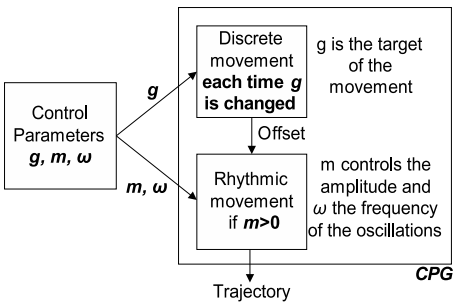


Fig. 1. A generic CPG. Modes of movements are switched on and off by parameters m and g. Discrete movement is incorporated to the final trajectory as an offset to the rhythmic movement. Trajectory is modulated by particular choices of m and g.

The CPG for a given DOF i is divided in two subsystems, one generating the discrete part of the movement and another generating the rhythmic part. The discrete part is given by the following system of equations

$$\dot{y}_i = v_i \quad (1)$$

$$\dot{v}_i = \frac{-b^2}{4}\,(y_i - g_i) - b\,v_i \quad (2)$$

and the rhythmic part by a Hopf oscillator, i.e. by the following system

$$\dot{x}_i = a\left(m_i - r_i^2\right)(x_i - y_i) - \omega_i z_i \quad (3)$$

$$\dot{z}_i = a\left(m_i - r_i^2\right) z_i + \omega_i\,(x_i - y_i) \quad (4)$$

where $r_i = \sqrt{(x_i - y_i)^2 + z_i^2}$.

Eqs 1 and 2 describe a discrete motion whose solutions converge asymptotically and monotically to a goal g_i with speed of convergence controlled by b. As g_i is a stable, globally attractive point, no stability problem will occur for any value of g_i.

Thus, each time g_i is changed, the system will be attracted by the new goal g_i and modify the resulting position x_i, generating a discrete movement towards g_i (fig. 1).

Eqs. 3 and 4 describe an Hopf oscillator where m_i controls the amplitude of the oscillations, ω_i is the oscillator intrinsic frequency and a controls the speed of convergence to the limit cycle. This oscillator contains a bifurcation from a fixed point (when $m_i < 0$) to a structurally stable, harmonic limit cycle with radius $R = \sqrt{m_i}$ for $m_i > 0$. The output x_i of the system has an offset given by y_i which is the state variable of the discrete system. For $m_i < 0$ the system exhibits a stable fixed point at $x_i = y_i$.

Rhythmic motion can be switched on or off by simply setting m_i to a positive or a negative value respectively. Moreover, the amplitude of the movement is specified by m_i and its frequency by ω_i (fig. 1).

Thus, by modifying on the fly the g_i and m_i parameters, the system can switch between purely discrete movements ($m_i < 0$), purely rhythmic movements (g_i fixed), and combinations of both (i.e. rhythmic movements around time-varying offsets).

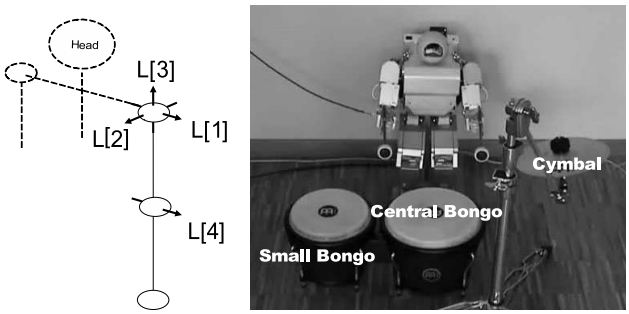
As will be described in the next section, the complete system is made of a network of CPGs coupled together.

III. Drumming Task

A. The Overall Architecture

We use a Hoap-2 robot, which is a 25 DOFs humanoid robot made by Fujitsu. We control 8 of the 25 DOFs of the robot, that is 4 DOFs in each arm : 3 in the shoulder and 1 in the elbow. Figure 2(a) shows a schematic view of the controlled DOFs of the Hoap-2 robot. From now on, we will refer to the different joints as L[1], L[2], L[3] and L[4] for the left arm and as R[1], R[2], R[3] and R[4] for the right arm. The others DOFs remain fixed to an appropriately chosen value during the task.

For each controlled DOF, we use the generic CPG presented in section 2 with the addition of couplings between the CPGs



(a) (b)

Fig. 2. (a) Schematic view of the controlled DOFs of the Hoap-2 left arm. The corresponding axes of rotation are also represented, (b) Picture of the real humanoid Hoap-2 robot sitting in front of 3 intruments : a central bongo, a small bongo and a cymbal (top view).

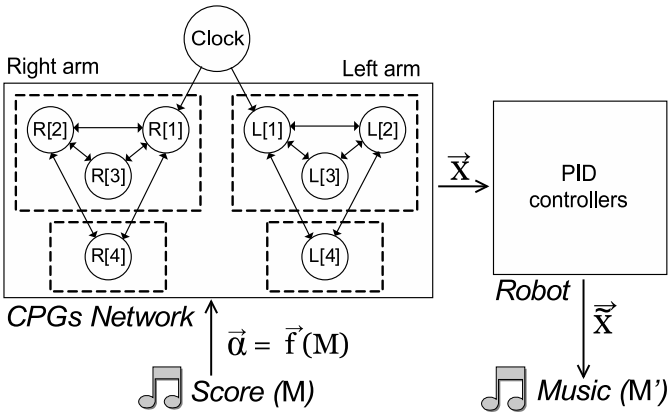


Fig. 3. Overall architecture of the system. A score matrix M is translated into a time-varying control vector α, this vector specifying the parameters of the different CPGs of the network. Each circle represents a CPG and the arrows represent couplings between them. By integrating the dynamical system of the different CPGs, we obtained desired trajectories $\vec{x}$ which are transformed in the actual trajectories $\vec{\tilde{x}}$ by the PID controllers of the robot.

for different DOFs, as described in subsection III C. The CPGs network constitutes the controller generating the trajectories which are used as input for the PID controllers of each joint.

The overall system architecture is depicted in figure 3. A arbitrary score matrix M is transformed onto a time-varying parameter vector, $\vec{\alpha}$, that controls the parameters of the CPGs network that generate drumming trajectories in real time. More precisely, for each DOF i and for each beat, this vector specifies the goal g_i and the amplitude of the movement, controlled by m_i. Desired trajectories $\vec{x}$ of each DOF are obtained by integrating the CPGs dynamical systems. These trajectories are used as input for the PID controllers of each joint and result in the actual trajectories $\vec{\tilde{x}}$.

B. Robotic Setup

The robot sits in front of a drum set composed of three instruments : two bongos and a cymbal, as illustrated on figure 2(b). The central bongo can be hitten by both arms, the cymbal only by the left one and the small bongo only by the right one.

For each CPG of the network, we set $b = 20$ and $a = 100$. Those parameters, which control the speed of convergence of each system (b for the discrete one and a for the rhythmic one), were fixed to appropriate values in regard to stability during the integration process and to feasability of the desired trajectory. The frequency ω is fixed to π by default with regard to to motor speed limitations. It can be modified to play the score faster or slower. The system is conceived to play any score and we tried 8 different scores. In this article we will only present the results obtained for the two scores shown in figure 4, as results were the basically the same for all the scores.

We will now explain more in details fig. 3, and especially the transformation of a standard score into a time-varying parameter $\vec{\alpha}$ controlling the CPGs network. This transformation is made in two steps : first, we translate the score into a binary matrix M, and then, we use a look-up table to assign to α the values needed to make the robot play this score.

We translate the scores in four rows matrices M, where the rows stand for, respectively, the cymbal, the central drum for the left arm, the central drum for the right arm and the small drum. Thus, the two first rows of M correspond to instructions for the left arm and the two last ones to instructions for the right arm. Column elements correspond to a sixteenth note, which will be our unit of time. We will denote by t_j the time corresponding to the jth column. We set $M_{\text{i,j}}$ to 1 if instrument i has to be hit at time t_j and to 0 otherwise. So, for instance, the score A will be translated in the following matrix

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & ... & 0 & 1 \\ 1 & 0 & 1 & 0 & ... & 1 & 0 \\ 1 & 0 & 1 & 0 & ... & 0 & 1 \\ 0 & 1 & 0 & 1 & ... & 1 & 0 \end{pmatrix}$$

Let's examine in details the rows corresponding to the left arm. For a given column j (i.e. at a given time t_j), we have three possibilities : the left arm has to hit the cymbal ($M_{1,j} = 1$, $M_{2,j} = 0$), the central drum ($M_{1,j} = 0$, $M_{2,j} = 1$) or doesn't have to hit anything ($M_{1,j} = 0$, $M_{2,j} = 0$). Note that both rows can not be simultaneously equal to one because that would mean that both instruments have to be hit at the same time by the same arm. If, for instance, the left arm has no instrument to hit at a given time t_j (i.e. if $M_{1,j} = 0$ and $M_{2,j} = 0$), its posture will not be specified directly by the score (or, more precisely, by the instrument to be hit). To overcome this, we anticipitate the movement in the sense that, if, for instance, $M_{1,j+1} = 1$ and $M_{2,j+1} = 0$ (i.e. cymbal has

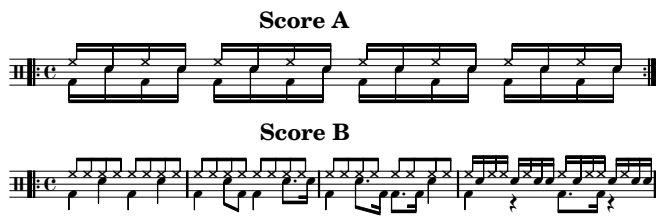


Fig. 4. The scores we used for the described drumming application.

	CB	Cymbal
$g_{L[1]}$	-1.11	-0.95
$g_{L[2]}$	0.20	0.41
$g_{L[3]}$	-0.26	0.22
$g_{L[4]}$	-1.04	1.31
$m_{L[1]}$	0.01	0.01
$m_{L[2]}$	-15	-15
$m_{L[3]}$	-15	-15
$m_{L[4]}$	-15	-15

(a) Left Arm

	CB	SB
$g_{R[1]}$	1.11	1.14
$g_{R[2]}$	-0.20	-0.17
$g_{R[3]}$	0.26	-0.34
$g_{R[4]}$	1.04	0.96
$m_{R[1]}$	0.01	0.01
$m_{R[2]}$	-15	-15
$m_{R[3]}$	-15	-15
$m_{R[4]}$	-15	-15

(b) Right Arm

Fig. 5. Tables of the values of m and g we used in the experiment.

to be hit at time t_{j+1}), then the arm is set in advance (i.e. at time t_j) upon the cymbal. If no instrument has to be hit a time t_{j+1}, we try at time t_{j+2} and so on.

Now, we have to transform the score into the corresponding trajectories. As ω is independant of the score and will be externally set, the trajectories corresponding to the score are specified by only two parameters for each DOF i, i.e. g_i and m_i (fig. 1). Matrix M has thus to be transformed onto a time-varying parameter vector

$$\overrightarrow{\alpha}(t) = [g_{\mathrm{L[1]}}(t), m_{\mathrm{L[1]}}(t), ..., g_{\mathrm{R[4]}}(t), m_{\mathrm{R[4]}}(t)]^T$$

, where $g_{\mathrm{L[1]}}$ stands for the parameter g for the L[1] DOF at time t, and so on. This transformation will be achieved by a function, called f on fig. 3, matching an instrument to be hit at a certain time t_i to the $\vec{g}$ and $\vec{m}$ values corresponding to that instrument, those being stored in a look-up table such as the one shown on fig. 5.

To determine this look-up table, we first parse the motion in two : the placement of the arm upon the correct instrument (the discrete part, controlled by $\vec{g}$) and then the beat of this instrument (the rhythmic part, controlled by $\vec{m}$). To achieve the placement of the arm, we have to specify for each of the four joints a target angle (the parameter vector $\vec{g}$ in the joint's CPG). To define those angles, the robot is manually set in an appropriate posture for each instrument and then its encoders are read (see fig. 5 for the values we get with Hoap-2,in radians). For the rhythmic part, only one appropriate DOF, L[1] here, is required to achieve the beat of the instrument. This is done by setting the parameter $m_{\mathrm{L[1]}}$ to a positive value. Larger values for m will lead to larger amplitudes of the oscillation and thus m can be used to control the strength of the beat. Once this look-up table has been set for a given drumming setup, it can be used for any score.

C. The Controller Architecture

In the overall, discrete movements are achieved by the four DOFs of the arm (this implying redundancy, which could be used to achieve more natural motion) whereas the rhythmic movement can be achieved by the elbow (L[4]) or by the shoulder's DOF generating an up-down movement (L[1]). For the sake of generality, all DOFs are controlled by the same type of CPG, that is the discrete and rhythmic system introduced above (Eqs.1-4 and fig. 1).

In order to ensure synchronization, we couple the different CPGS toghether. Moreover we add an external clock (see fig. 3) used as a metronome to provide for a unit of time, i.e. as an external reference to be synchronized with. This clock is only coupled to one DOF (L[1]) to avoid oscillations to be forced in a direct way. We discuss details of the left arm controller since both arms have similar controller architectures.

We bilaterally couple the Hopf oscillators of the CPGs of the left arm, those couplings being illustrated by right-left arrows on fig. 3. We modify Eqs. 3 and 4 of all the left DOFs as follows :

$$\begin{pmatrix} \dot{x}_{\mathrm{L[i]}} \\ \dot{z}_{\mathrm{L[i]}} \end{pmatrix} = \ldots + w \sum_{j \neq i} \mathbf{R}(\theta^{\mathrm{L[j]}}_{\mathrm{L[i]}}) \begin{pmatrix} \frac{x_{\mathrm{L[j]}} - y_{\mathrm{L[j]}}}{r_{\mathrm{L[j]}}} \\ \frac{z_{\mathrm{L[j]}}}{r_{\mathrm{L[j]}}} \end{pmatrix}, \quad (5)$$

where $r_{\mathrm{L[j]}}$ is the norm of vector $(x_{\mathrm{L[j]}}, z_{\mathrm{L[j]}})^{\mathrm{T}}$. We normalize the coupled vectors in order to reduce the impact of the couplings on the amplitude and thus avoid distortion. The linear terms are rotated onto each other by a rotation matrix $R(\theta^{\mathrm{L[j]}}_{\mathrm{L[i]}})$, where $\theta^{L[j]}_{\mathrm{L[i]}}$ is the desired relative phase among L[i]'s and L[j]'s CPGs (i, j = 1,2,3,4). In our case, we set it to 0 degrees as we want all oscillators to be in phase. Coupling strength w was set to 1 in the experiments. Due to the properties of this type of coupling among oscillators, the generated trajectories will stay smooth.

We couple one of the CPG's rhythmic component with an external clock (see fig. 3). This clock is an Hopf oscillator of parameters m_{clock} and ω_{clock}. We use the same type of coupling than above. The Hopf oscillator of the L[1]'s CPG was thus further modified as follows :

$$\begin{pmatrix} \dot{x}_{\mathrm{L[1]}} \\ \dot{z}_{\mathrm{L[1]}} \end{pmatrix} = \ldots + w_{\mathrm{clock}} \mathbf{R}(\theta^{\mathrm{clock}}_{\mathrm{L[1]}}) \begin{pmatrix} x_{\mathrm{clock}} \\ z_{\mathrm{clock}} \end{pmatrix} \quad (6)$$

where w_{clock} is the coupling strength and the clock vector $(x_{\mathrm{clock}}, y_{\mathrm{clock}})$ corresponds to the normalized variables of Hopf oscillator (eqs. 3 and 4). We set the relative phase $\theta^{\mathrm{clock}}_{\mathrm{L[1]}}$ to 0 degrees as we want both oscillators to be in phase. In the experiments w_{clock} was set to 4, m_{clock} to 0.001 and ω_{clock} to the same frequency that the other oscillators, i.e. $\omega_{\mathrm{clock}} = \pi$.

As shown on fig. 3, the architecture of the right arm is similar to the one described for the left arm. Analogously, R[1] DOF is unilateraly coupled to the clock. There is currently no connection among arms, so that, if one arm is disturbed by a perturbation, then the other arm won't be affected.

IV. Results

Our aim was to build a system able to superpose, and switch between, discrete and rhythmic modes of movement and easy to control. In the introduced system, switching between the modes is controlled by only two parameters, g and m. Indeed, each time g is changed, a discrete movement is generated and oscillations occurs only when m is positive. Moreover, those two parameters allow modulations of the trajectories. In the results presented here, we set $a = 100$, $b = 20$ and $\omega = 2\pi$. We solve the equations using Euler integration, with a timestep of one millisecond.

We first present the behavior of a single CPG without any coupling. On fig. 6, we plot the trajectories obtained when varying parameters g and m. In part A., both types of

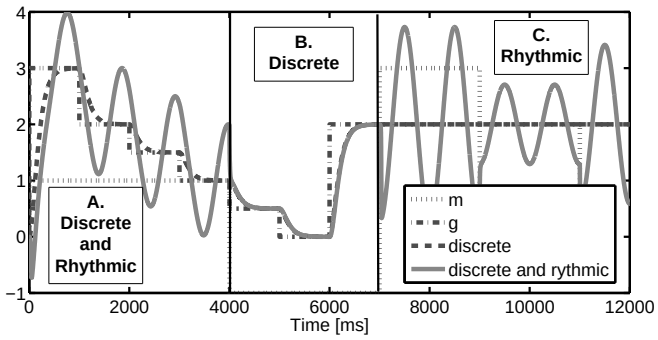


Fig. 6. Illustration of switch between the different modes and modulation of trajectories using parameters g and m.

movement are superposed, i.e. discrete movement is a time-varying offset of the rhythmic movement. In part B., we set m to a negative value, so that the attractor of the Hopf oscillator is no longer a limit cycle but a fixed point specified by the offset, i.e. by g. The resulting discrete trajectory converges asymptotically to the current value of g. In part C., g is constant, so that no discrete movement is produced, while m is fixed to a time-varying positive value, producing oscillations of amplitude $\sqrt{m}$. Whatever the change is, the system converges almost immediately to the new solution of the sysem.

For the drumming application, we test the entire architecture presented on fig. 3 on several scores, to ensure the generality of the approach. Snapshots of the robot while drumming are shown on fig. 7. We present here the results for scores A and B (fig. 4). We set the frequency to $\omega = \pi$ in regards with the motor limitations.

We recorded the trajectories from the joints incremental encoders ($\vec{\tilde{x}}$ on fig. 3) and the planned trajectory ($\vec{x}$ on fig. 3). The actual trajectories corresponding to the score A and B are shown on the upper graph of fig. 8 (left arm) and of fig. 9 (right arm) respectively. On the middle graph, we have plotted the difference between the planned and the actual trajectory, i.e before and after the PID controllers of the robot. The error being quite small, i.e. about twenty times smaller than the trajectories, we can conclude that the speed of the system is well adapted to the motor performance of the robot. Comparing the errors with the actual trajectory, we can see that they occurs mainly because of an instrument has been hit or just after the initiation of a discrete movement. Indeed, as said before, the discrete system converges asymptotically towards g, i.e. the movement is really fast in the beginning and slows down afterwards.

Throughout the score, both arms stay synchronized, as illustrated by the bottom graph of figs. 8 and 9. We only plot the trajectory of up and down shoulder DOF for left and right arm (L[1] and R[1]] resp.), as those are the only ones producing rhythmic movements. As the orientations of the axes are opposite for those DOFs, we had to plot the inverse values for R[1] (this is denoted by the -R[1] or by -L[1] in the legend of the graphs).

Movies of the robot while drumming and animations done with Webots are available on our website (birg.epfl.ch).

V. Conclusion

In this article, we introduce simple, unique system able to produce both discrete and rhythmic movements and also to easily switch from one mode to another or produce a combination of both modes. In the proposed system, the resulting movement can be modulated by a simple change of the parameters of the dynamical systems ($\vec{g}$ for the discrete system and $\vec{m}$ for the rhythmic one). Moreover, due to the properties of dynamical systems, the trajectory will always

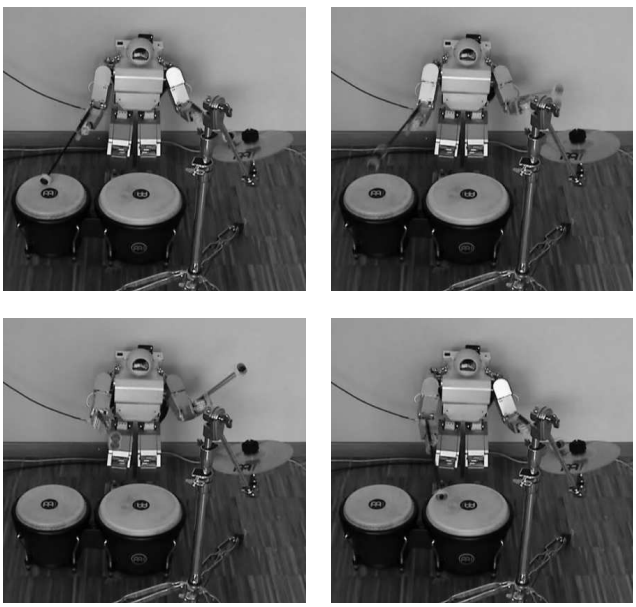


Fig. 7. Snapshots of the drumming robot.

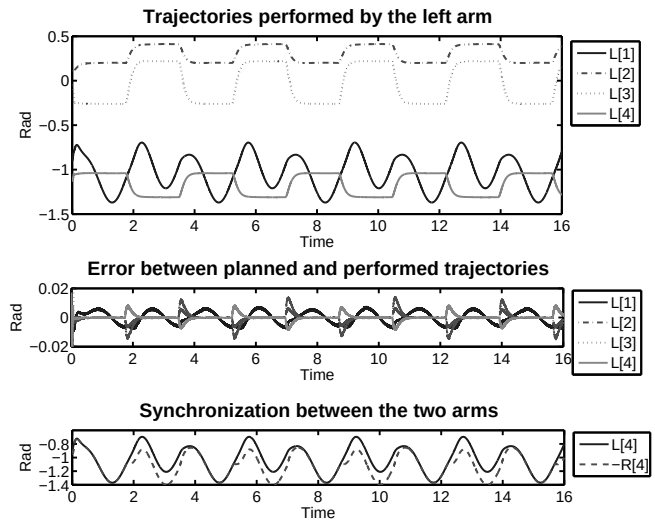


Fig. 8. Results for score A. Up : Actual trajectories of the left arm, middle : Difference between the actual trajectories and the desired one, bottom : Actual trajectories of the up and down shoulder DOF of each arm.

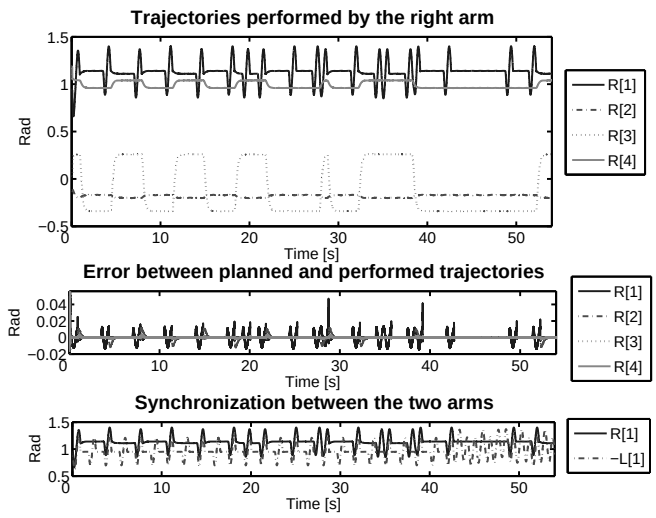


Fig. 9. Results for score B. Up : Actual trajectories of the right arm, middle : Difference between the actual trajectories and the desired one, bottom : Actual trajectories of the up and down shoulder DOF of each arm.

remain smooth, whatever the change is. The system can take an arbitrary drumming score and generate the necessary trajectories for playing it with the robot.

The trajectories are generated online by integrating differential equations in real time. This offers several interesting options such as (1) the possibility to change parameters on the fly (e.g. the score or the drum set could be abruptly modified and the system would smoothly change the trajectories accordingly) and (2) the possibility to include feedback loops in order to do online trajectory modulation and take external perturbations into account.

We are currently using those two properties in simulation (using a robot simulation program called Webots, see [11] for details). Indeed, we have implemented an inverse kinematics algorithm to calculate the corresponding angles for a given cartesian position. This algorithm is useful to generate movements towards new cartesian goals. For instance, if an instrument is moved during the experiment, the system will update the g_i values in the look-up table and smoothly adapt the trajectory accordingly to the new position of the instrument. Moreover, to emphasize the possibility online modulation of the trajectories, we will add in the future a simple interface to allow users to create a score, so that it could be played by the robot on the fly.

In addition, we have introduced a feedback loop to enable us to take perturbation into account. This feedback is based on the error between planned and actual trajectories and slows down the system whenever this error exceeds a threshold.

In the drumming application presented here, the type of movement (discrete or rhythmic) is specified by an external score. However, in further works, we would like to extend our system to allow autonomous switches from one type to another, for instance for obstacle avoidance or foot placement during locomotion. For this, we will need to introduce a second level of control in our system, this level specifying the parameters of the dynamical system and thus the type of movement produced.

ACKNOWLEDGMENT

This work was made possible thanks to the support of the Swiss National Science Foundation (A.I.) and of the European Commission's Cognition Unit, project no. IST-2004-004370 : RobotCub (S.D.).

REFERENCES

[1] G. Schöner and M. Dose. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems*, 10(4) :253–267, 1992.

[2] J. Tani, Y. Ito, and Y. Sugita. Self-organization of distributedly represented multiple behavior schemata in a mirror system : reviews of robot experiments using rnnpb. *Neural Networks*, 17 :1273–1289, 2004.

[3] S. Schaal, S. Kotosaka, and D. Sternad. Nonlinear dynamical systems as movement primitives. In *International Conference on Humanoid Robotics. Cambridge, MA, Sept 6-7, 2001*, pages 117–124. Springer, 2001.

[4] Y. Fukuoka, H. Kimura, and A.H. Cohen. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. *The International Journal of Robotics Research*, 3–4 :187–202, 2003.

[5] G. Schöner and C.M.P. Santos. Control of movement time and sequential action through attractor dynamics : A simulation study demonstrating object interception and coordination. In *SIRS 2001*, 2001.

[6] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In S. Thrun S. Becker and K. Obermayer, editors, *Neural Information Processing Systems 15 (NIPS2002)*, pages 1547–1554, 2003.

[7] T. Pongas, A. Billard, and S. Schaal. Rapid synchronization and accurate phase-locking of rhythmic motor primitives. In *Intelligent Robots and Systems (IROS 2005)*, 2005.

[8] S. Kotosaka and S. Schaal. Synchronized robot drumming by neural oscillator. In *The International Symposium on Adaptive Motion of Animals and Machines. Montreal, Canada.* 2000.

[9] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Learning rhythmic movements by demonstration using nonlinear oscillators. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2002)*, pages 958–963. 2002.

[10] F. Delcomyn. Neural basis of rhythmic behavior in animals. *Science*, 210 :492–498, 1980.

[11] 0. Michel. Webtos tm : Professional mobile robot simulation. *International Journal of Advanced Robotic System*, 1 :39–42, 2004.