A modular, bio-inspired architecture for movement generation for the infant-like robot *iCub*

Sarah Degallier, Ludovic Righetti, Lorenzo Natale, Francesco Nori, Giorgio Metta and Auke Ijspeert

***Abstract*— Movement generation in humans appears to be processed through a three-layered architecture, where each layer correspond to a different level of abstraction in the representation of the movement. In this article, we will present an architecture reflecting this three-layered organization and try to demonstrate the advantages of applying such an architecture to the control of movements.**

Basing ourselves on a modular approach to human movement generation, we will show that our architecture is well suited for the online generation and modulation of a particular behavior, but also for switching between behaviors. This will be illustrated respectively through a interactive drumming task and through the switching between reaching and crawling. This latter task has been tested with the ODE-based simulator Webots$^{\text{TM}}$ while drumming has been implemented on the real robot *iCub*.

I. Introduction

In the framework of the European project RobotCub [REF], which aims at developing a infant-like robot, *iCub*, with the motor and cognitive abilities of a 2 years-old child, we are currently developing a functional model of the human motor system. Our motor architecture will be integrated in a larger cognitive architecture that has been developed by Vernon, Metta and Sandini [REF].

To build our model of the human motor system, we define a three-layered architecture whose layers can be functionally defined as the planner, the manager and the generator. Functionally, the planner (i.e the motor cortex in humans) builds the mental representation of the task. Indeed, in order to choose to perform a given action, we must be able to predict its consequence on the environment and on ourselves and an internal model of the environment and the self is thus needed [REF JEANNEROD]. The manager (the brain stem, the basal ganglia and the cerebellum in humans) is involved in the selection, timing and coordination of the appropriate behaviors (motor programs). Finally, the generator (the spinal cord) generates trajectories through central pattern generators, i.e. networks of neurons involved in the production of rhythmic and discrete movement primitives. ADD REFERENCES

Sensory feedback seems to be also distributed along the motor structure in the same three-layered fashion [REFERENCE], accordingly to its degree of processing, namely (i) local information (cutaneous information, state of the muscles, load, ...), that is required for fast, protective movements (reflexes), (ii) contextual information (balance, position of objects for reaching, timing, ...), that triggers learned responses to the specific context (automatisms) and finally at the higher level, (iii) semantic information (signification of the environment, state of mind, ...), that contributes to the mental representation of the context involved in voluntary movements.

Note that as our particular interest is movement generation here, we do not focus on the high cognitive abilities needed to define and choose the action, i.e. in terms of the architecture we do not focus on the development of the planner. Such questions are treated by other labs in the framework of the RobotCub project [REFS].

In order to develop an efficient model reflecting those principles, we make the assumption that movements generation is highly modular, both in terms of motor primitives (i.e. units of movement) [REF] and in terms of motor programs (i.e. behaviors) [REF], as it will be discussed more in details in section II.

We assume the existence of two basic types of motor primitives, i.e. discrete (aperiodic and finite) and rhythmic (periodic) movements. Such a distinction is convenient for modeling purposes. Indeed, our trajectories are generated using a network of coupled dynamical systems, accordingly to the biological concept of central pattern generators (see [1] for a review); we use a system similar to the ones that we had previously developed ([2],[3]) which allows the generation of discrete (i.e. short-term) and rhythmic movements and the combination of both (i.e. oscillations around time-varying offsets). Such an approach allow us to use the stability properties of dynamical systems to ensure a robust control of the movements.

STATE OF THE ART

In this article, we present our current implementation of this functional architecture (section II) as well as the future improvements that we are planning (section VI). The current implementation allows for an easy and fast online modulation of trajectories as well as the possibility of easily switching between behaviors according to sensory information; this will be illustrated through two applications, namely *interactive drumming* (sectionIV) and the switching between *crawling*, *reaching* on the fours and *reaching while crawling* (sectionV). Interactive drumming has been tested on the real robot while switching between behaviors has been teste using the physics based simulator Webots$^{\text{TM}}$. In section III, the infant-like robot *iCub* is presented.

II. Presentation of the architecture

We present here more in details the current implementation of the architecture, which is depicted on 1. Note that since it is an ongoing work, only some parts of the architecture have been yet tested.

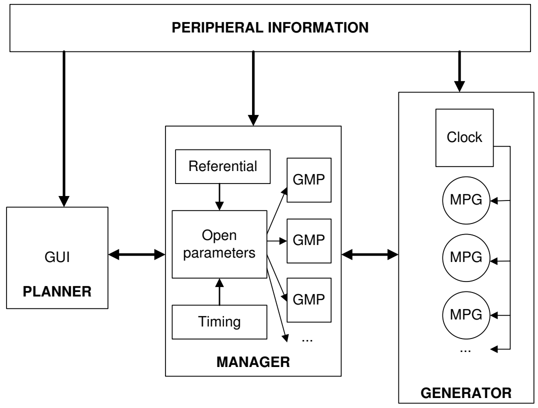


Fig. 1. Schematic of the functional organization of the architecture.

A. Generator

The generator, which is responsible for the generation of the trajectories, is built on the concept of central pattern generators (PGs), that we take in the sense of a network of generators responsible for the generation of units of movements called motor primitives (REF).

The trajectories corresponding to the different behaviors are generated through a unique set of differential equations, which is designed to produce complex movements modeled as periodic movements around time-varying offset. More precisely, it is built on the hypothesis that complex movements can be generated through the superimposition and sequencing of simpler motor primitives generated by rhythmic and discrete pattern generators[1]. The discrete primitive is injected in the rhythmic primitive as an offset, although other combinations of them could be considered such as a sequencing or a simple addition of their output.

The discrete PG is modeled by the following system of equations

$$\dot{h}_i = d(u - h_i) \quad (1)$$

$$\dot{y}_i = h_i^4 v_i \quad (2)$$

$$\dot{v}_i = u^4 \frac{-b^2}{4} \; (y_i - g_i) - b \; v_i. \quad (3)$$

The system is critically damped so that the output y_i of Eqs 2 and 3 converges asymptotically and monotically to a goal g_i with a speed of convergence controlled by b, whereas the speed v_i converges to zero. The first equation ensures a bell-shaped velocity profile and is reset to zero at the end of each movement. The output of such a system is depicted on Fig. **??**.

ADD FIGURES

Concerning the rhythmic PG, it is modeled by a system with a Hopf bifurcation:

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

$$\dot{z}_i = a\left(m_i - r_i^2\right) z_i + \omega_i (x_i - y_i) + \sum k_{ij} z_j + u_i \quad (5)$$

$$\omega_i = \frac{\omega_{down}}{e^{-bz_i} + 1} + \frac{\omega_{up}}{e^{bz_i} + 1} \quad (6)$$

where $r_i = \sqrt{(x_i - y_i)^2 + z_i^2}$. When $m_i > 0$, Eqs. 4 and 5 describe an Hopf oscillator whose solution x_i is a periodic signal of amplitude $\sqrt{m_i}$ and frequency ω_i with an offset given by g_i (see Fig. **??**) . A Hopf bifurcation occurs when $m_i < 0$ leading to a system with a globally attractive fixed point at (g_i,0). The term $\sum k_{ij} z_j$ controls the couplings with the other joints j; the k_{ij}'s denote the gain of the coupling between joints i and j. The expression used for ω_i allows for an independent control of the speed of the ascending and descending phases of the periodic signal (see [5] for more details). Finally the term u_i is a control term generated by feedback information.

Qualitatively, by simply modifying on the fly the parameters g_i and m_i, the system can thus switch between purely discrete movements ($m_i < 0, g_i = g(t)$), purely rhythmic movements ($m_i > 0, g_i =$cst), and combinations of both ($m_i > 0, g_i = g(t)$) (see [2] fore more details). More elaborate movements can be achieved by sending time-varying parameter to the system and by integrating feedback signals to the generator.

TODO: add figure

Feedback.

B. Manager

The manager is built upon the concept of motor program, which is defined as "*a set of muscle commands which are structured before a movement begins and which can be sent to the muscle with the correct timing so that the entire sequence is carried out in the absence of peripheral feedback*" by Marden et al. ([6]). This concept is a nice way of explaining the rapidity with which we react to stimuli and the stereotypy present in human movements. Moreover, the notion of generalized motor program (GMP), that is motor programs with open parameters, allows the generation of movements adapted to the environment (see [7] for instance).

Functionally speaking, the manager is mainly responsible of sending the right parameters (in joint space) to the generator, at the right timing. We define a (generalized) motor program (MP) as a sequence of parameters sent to the generator to produce the desired trajectories, that is in our case $\vec{g}(t)$, $\vec{m}(t)$, $\vec{\omega}(t)$, the couplings between the oscillators, the initiation of an appropriate feedback and a time interval that defines the rapidity with which the sequence of movements is performed. Some of the parameters are fixed, as the coupling between the limbs for crawling for instance, others are open and need to be defined relatively to the environment (the desired angles in reaching for instance). An inverse kinematics is also needed to transform task space goals into target joint angles.

Every time a MP is launched by the manager, the first command sent correspond to a predefined initial position. The

[1]Note that Schaal et al. ([4]) have shown that rhythmic movements are not a particular case of discrete movements by using fmri techniques; some brain areas involved in discrete task are not active during rhythmic movements.

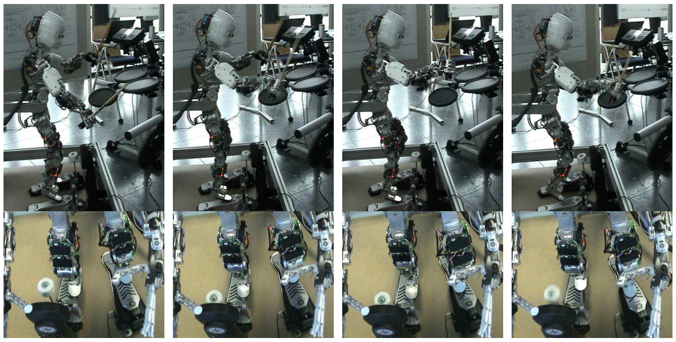


Fig. 2. Snapshots of the iCub drumming at CogSys 2008 in Karlsruhe.

parameters are then sent at regular time interval to the generator. At the end of the sequence, a command corresponding to a final target position is sent. This makes the switching between task easier, as it will briefly be illustrated with crawling and reaching.

A MP can be elicited either by the planner (voluntary movements) or by the contextual sensory information (automatisms). We will illustrate this by two examples, drumming and reaching while crawling.

Feedback.

C. Planner

The planner is for now a simple GUI that allows the user to specify the task the robot has to perform. However, higher cortical abilities can be implemented to defined the task to be performed. The input that should be given to the manager is the target goal (or the target trajectory) in cartesian space.

Feedback.

III. Presentation of *iCub*

IV. Application to drumming

A. Results

V. Application to crawling and reaching

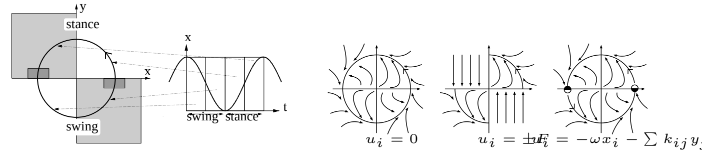
We define three tasks: *reaching*, *crawling* and *reaching while crawling*. Each of this task is triggered by color marks on the ground, i.e. a red mark on the ground launches *reaching*, a blue mark *reaching while crawling* and no mark *crawling*. No visual processing is considered here; the position and color of the mark are directly provided to the robot. The robot crawls in an environment where it has to switch between those three behaviors according to marks arbitrarily placed on the ground.

We define here the motor programs corresponding to the different tasks; each behavior is simply triggered through the specification of the amplitudes $\vec{m}$ and the offsets $\vec{g}$ by the manager.

Crawling ($\vec{m} > \vec{0}, \vec{g} = \vec{0}$). Analysis of crawling infants have shown that most infants crawl on hands and knees, using a walking trot gait. The couplings are thus set so to obtain a trot gait and g is set to a fixed value (0 here) so to obtain a purely rhythmic movement.

Reaching ($\vec{m} < \vec{0}, \vec{g} \neq \vec{0}$). Once a mark is close enough to be reachable, *crawling* is turned off by setting $\vec{m}$ to a negative value at an appropriate time for each limb. Then the IK algorithm is used to get the target position $\vec{g}$ which is sent to the generator.

Reaching while crawling ($\vec{m} > \vec{0}, \vec{g} = \vec{g}(t)$). When a mark is reachable and when the arm is in an appropriate state (i.e. is in the swing phase), the desired position g is sent to the generator by the manager; the actual position of the system is then compared to the desired position so to reach the correct position through a modification of the offset. $\vec{m}$ is kept to a positive value so that the resulting movement is rhythmic with a time varying offset.



(a) Feedback activation zone (b) Oscillator phase spaces

Fig. 3. Phase space of an oscillator with its activation zone for the feedback (light gray for switch and dark gray for stop controls) and the correspondence with the x variable of the oscillator is shown on the left figure. Right figure shows the schematic phase plot of the oscillator for the different types of feedback.

Feedback integration. A phase dependent sensory feedback is also included in the rhythmic PG to make the crawling locomotion more robust and adaptive to the environment, as we did previously in [8]. Information from the touch sensors located on the hands and knees of the robot is used to modulate the onset of the swing and stance phases, as mammals do ([9]). The transition from stance to swing phases is delayed as long as the other limbs cannot support the body weight and is triggered sooner when the limb leaves unexpectedly the ground. Analogous policies are used for the swing to stance transition. More precisely, the term u_i of Eq 5 is defined as

$$u_i = \begin{cases} -\text{sign}(y_i)F & \text{fast transitions} \\ -\omega x_i - \sum k_{ij} y_j & \text{stop transition} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where F (200 in our case) controls the speed of the transition. Fig. 3 shows the activation of the feedback depending on the phase of the limb and the resulting modification of the phase space of the oscillator.

A. Results

Crawling and reaching have been tested in simulation using the ODE-based software Webots^{TM}. We performed simulation of the three different behavior with and without feedback. First we note that the feedback makes the crawling more stable and adaptive to the environment (i.e. the robot can crawl on inclined slopes and uneven terrain), for details on the improvement of crawling by feedback refer to [8]. Second, thanks to our modular approach, the integration of the different behaviors and the feedback is successful (see Fig. 4 for snapshots of the three different behaviors).

On Fig. 5, the trajectories of the left shoulder corresponding to the snapshots of Fig. 4 are shown together with the parameters sent by the manager. Both trajectories with and without

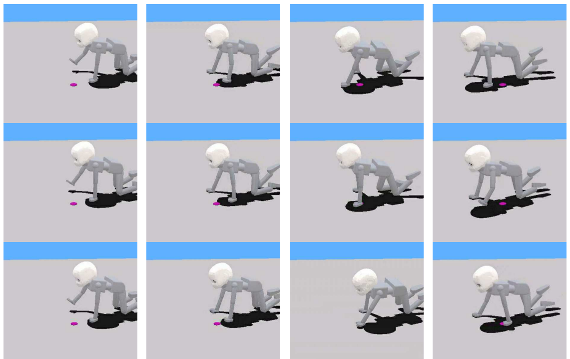
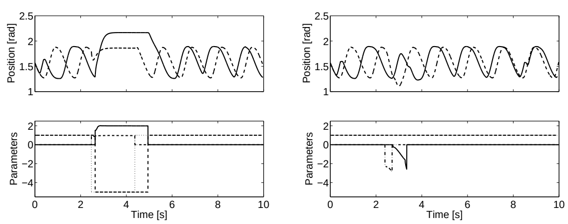


Fig. 4. Snapshots of the three behaviors with feedback. Upper line: Only crawling; middle line: Crawling while reaching; bottom line: iCub crawls, stops and then reaches the mark.

feedback are presented. It can be seen that with feedback the trajectories are constantly modulated accordingly to the local sensory information showing fast adaptive behaviors. Specially at the beginning of the crawling, the amplitude of the trajectory is modulated to account for the initial posture of the robot which does not correspond to the usual crawling posture.

Concerning behaviors, we notice that simple parameters change allows to generate trajectories complex enough to fulfill the task. Thanks to the robustness of dynamical systems, the trajectories resume to simple crawling after reaching the mark in both tasks. The generated trajectories are smooth despite the discontinuous nature of the parameters defining the GMP.



(a) Reaching without crawling (b) Reaching while crawling

Fig. 5. Results of the simulation for both reaching while crawling and reaching when crawling is stopped and resumed after. The experiments were done with and without feedback integration. The upper graphs show the left shoulder joint (plain line with feedback, dashed line without), the lower graphs show the corresponding parameters (m plain line with feedback, dashed line without. g dash-dot line with feedback, dotted line without). Refer to the text for discussion of the results.

VI. Discussion

FUTURE IMPROVEMENTS:

- dealing with constraints in the manager - task space control integration - adding feedback

VII. Conclusion

Acknowledgments

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

References

[1] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks (to appear)*, 2007.

[2] S. Degallier, C. P. Santos, L. Righetti, and A. Ijspeert. Movement generation using dynamical systems: a humanoid robot performing a drumming task. In *IEEE-RAS Inter. Conf. on Humanoid Robots*, pages 512–517, 2006.

[3] S. Degallier, L. Righetti, and A. Ijspeert. Hand placement during quadruped locomotion in a humanoid robot: A dynamical system approach. In *IEEE-RAS International Conference on Intelligent Robots and Systems (IROS07).*, 2007.

[4] S. Schaal, D. Sternad, R. Osu, and M. Kawato. Rhythmic arm movement is not discrete. *Nat. Neuroscience*, 7(10):1136–1143, 2004.

[5] L. Righetti and A.J. Ijspeert. Design methodologies for central pattern generators: an application to crawling humanoids. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[6] C.D. Marsden, P.A. Merton, and H. Morton. The use of peripheral feedback in the control of movements. *Trends Nerosci.*, 7:253–258, 1984.

[7] R.A. Schmidt and T.D. Lee. *Motor control and learning: A behavioral emphasis*. Human Kinetics, Champaign, IL, USA, 2005.

[8] L. Righetti and A.J. Ijspeert. Pattern generators with sensory feedback for the control of quadruped locomotion. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation ICRA*. Accepted.

[9] S. Frigon and S. Rossignol. Experiments and models of sensorimotor interactions during locomotion. *Biological Cybernetics*, 95(6):607–627, 2006.