

Hand placement during quadruped locomotion in a humanoid robot: A dynamical system approach

Sarah Degallier, Ludovic Righetti and Auke Ijspeert

School of Computer and Communication Sciences

EPFL, Ecole Polytechnique Fédérale de Lausanne

EPFL-IC-ISIM-GRIJ, Station 14

CH 1015, Lausanne, Switzerland

email: sarah.degallier@epfl.ch, ludovic.righetti@epfl.ch and auke.ijspeert@epfl.ch

Abstract—Locomotion on an irregular surface is a challenging task in robotics. Among different problems to solve to obtain robust locomotion, visually guided locomotion and accurate foot placement are of crucial importance. Robust controllers able to adapt to sensory-motor feedbacks, in particular to properly place feet on specific locations, are thus needed. Dynamical systems are well suited for this task as any online modification of the parameters leads to a smooth adaptation of the trajectories, allowing a safe integration of sensory-motor feedback.

In this contribution, as a first step in the direction of locomotion on irregular surfaces, we present a controller that allows hand placement during crawling in a simulated humanoid robot. The goal of the controller is to superimpose rhythmic movements for crawling with discrete (i.e. short-term) modulations of the hand placements to reach specific marks on the ground.

I. INTRODUCTION

This work is part of the RobotCub project which aims at building a 54-degrees of freedom baby robot with the cognitive abilities of a 1.5-years old child (see [1] for more details on the project and on the robot). Visually-guided locomotion, more precisely crawling, and hand placement during locomotion are required for the robot to be able to explore its, possibly unknown, environment.

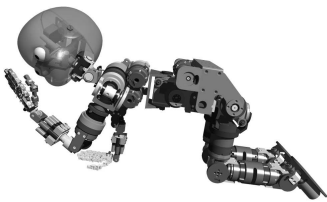


Fig. 1. Drawings of the baby robot iCub.

Generating trajectories in a time-evolving environment is still a hard and unsolved problem for robotics. Indeed, if efficient controllers for following predefined trajectories have been designed, adaptation to unpredicted variations of the outside world is still an open problem.

Successful results have been achieved by the dynamical systems approach ([2],[3],[4],[5],[6],[7] and many more), where the desired trajectories are no longer pre-computed, but rather are generated on line and in real-time relatively to the aim of the movement.

The use of dynamical systems has several interesting properties: (i) they generate trajectories in real-time with little computation, (ii) they can smoothly modulate the trajectories when control parameters (e.g. defining the goal posture) are changed, and (iii) they can be designed to have useful stability properties such as in our case limit cycle behavior of the oscillators of the rhythmic module, phase-locking between the different oscillators for different degrees of freedom, and asymptotic stability of single point attractors for the discrete module.

However, designing dynamical systems is a difficult task, and little has been done in this direction ([8]). Indeed, the system does not encode only one trajectory, but a whole space of trajectories corresponding to the different possible states of the system. Moreover, those systems may in some cases be extremely sensitive to modifications of initial conditions or of parameters, and the qualitative behavior of the system may drastically change subject to small variations. Thus, when using dynamical systems, a previous deep theoretical study of the behavior of the system is needed.

In order to make the design of a dynamical system easier, we choose a modular approach, meaning that we see any complex trajectory as the combination of simpler movements, that we call motor primitives. Our aim is to model those primitives by simple, known stable, dynamical systems. The modular approach is supported by neurological results demonstrating that the spinal cord seems to produce many behaviors through the combination of a small number of muscles synergies (see, for instance, the results obtained, notably on spinalized frogs, by Bizzi's group ([9],[10])).

We distinguish two abstract types of primitives: discrete and rhythmic primitives. This distinction is supported by research on human motor control by Schaal *et al* ([5]) who has provided evidence that discrete movements are not a special case of rhythmic movements. Indeed, some planning cortical areas activated during point-to-point movements of the wrist are not activated when the same joint executes a rhythmic movement.

In this article, we present a controller for hand placement on marks on the ground during crawling. This problem is very challenging since crucial issues in robotics such as perception, planning, timing, and stability are involved in this task. However, as a first approach to this problem, we uniquely

address the problem of modulating trajectories (i.e. we assume that the presence and position of marks are known) and our aim is to design a controller as simple as possible.

The controller is composed of two motor pattern generating modules, a discrete and a rhythmic one. Both modules are constructed using dynamical systems. The rhythmic module is constructed as a central pattern generator (CPG), i.e. a circuit which, like its biological counter part, can produce stable multidimensional rhythmic patterns while receiving only simple control inputs. This CPG has been designed by Righetti et al. ([11]) to produce joint angle trajectories that resemble crawling trajectories recorded from infants.

We have previously developed a controller for combining discrete and rhythmic movements in the context of a drumming task ([7]). However, when drumming, the discrete movement is the predominant task. Here, on the contrary, it is locomotion that is regarded as the main task. Placing ourselves in the context of quadrupedal locomotion rather than bipedal locomotion decreases the need for the control of balance. Indeed, during crawling, there are always at least two supporting limbs, namely an arm and the opposite side leg, which provides a higher stability than during bipedal walking.

To validate the model, and as the robot iCub is still under construction, we use the mobile robot simulator Webots ([12]), which uses ODE, a rigid body dynamics simulator, to simulate the dynamics of the robot. In the experiment, the baby robot is given the coordinates of marks it has to place its hands on. An inverse kinematic algorithm is used to obtain the joint angles needed to reach the mark.

Note that we do not address in this paper the (important) problem of visually recognizing and localizing a mark: the presence and position of the mark is directly provided to our controller in cartesian coordinates and then transformed into a desired posture using inverse kinematics.

We present results showing how the controller successfully generates crawling motion while episodically placing the hands on specific marks on the ground. We also analyze current limitations of the method (e.g. situations in which the robot falls over) and suggest improvements as future work.

Foot placement during locomotion has been studied before, see Hodgins and Raibert for a quick review ([13]). Usually, the presented methods use an algorithm to find suitable footholds relatively to the dynamics of the robot. By suitable footholds it is meant a position of the foot that maintains balance and allows the legged system to keep on walking. With those methods, the foot placement is very accurate but the locomotion is very slow.

In this article, accuracy of the foot placement is not our main concern. Indeed, our aim is to design an architecture such that the robot can modify its trajectory to reach the mark, while keeping its velocity and the step length induced by the rhythmic movement. Thus instead of modifying the rhythmic movement, we will add a external, independent discrete movement to modify the final trajectory in such a way that the foot will reach the mark.

We start by presenting the mathematical systems we used

to model the rhythmic and the discrete motor primitives (Section II). Then we briefly present how we combine both movements (Section III). We then present the results obtained in simulation using iCub and discuss the possible improvements of the system, in particular visual and sensory-motor feedbacks (Section V). We conclude by discussing the main results we obtained and the future issues we are planning to work on.

II. COORDINATION OF THE MOTOR PRIMITIVES

An interesting question arise when considering the arm joints producing the rhythmic movement: how do we modulate the rhythmic trajectory to reach the target? Is it by modulating the rhythmic movements, for instance by modifying the length of the steps or do we superimpose a discrete movement to the rhythmic one? As symmetry is a key component in locomotion, we postulate that the second hypothesis is the correct one, in order to perturb as little as possible the general crawling behavior.

Thus, as a starting hypothesis, we assume that the reaching of the mark is achieved by simply adding the discrete and the rhythmic movements. Both modules are thus independent and do not interfere directly with each other. Independence of the primitives at the motor pattern generation level allows one to keep the nice stability and robustness properties of each system. This will be important in the future when introducing sensory-motor feedbacks to the controller. However, at a higher level, the level of decision, the onset of the discrete movement will be constrained by the rhythmic movement and some slight modifications of the inverse kinematics algorithm will be needed.

The general architecture we have implemented is composed of two control layers: the decision and the motor layers, see Figure 2 for a sketch of the general architecture. The motor layer is composed of the discrete and rhythmic motor primitives, which compose the motor pattern generator (MPG) system. Those primitives can be switched on and off by the decision layer according to the external, visual, information (detection and localization of a mark) and proprioceptive data, namely the current state of the joints and the internal model of the robot.

Concerning the decision layer, we model the switching on and off of the discrete MPG in the following intuitive way (see figure 2). First, a discrete movement is initiated only when a mark is reachable. Then, the onset of a discrete movement is not possible at the very beginning of the swing phase to avoid collisions with the body.

To superimpose the motor primitives, we decided to simply add them. Thus, the discrete movement is simply the difference between the desired trajectory, i.e a trajectory reaching the mark, and the current, rhythmic trajectory. However, as we want the hand to reach the mark at the beginning of the stance phase, we are only concerned by the (rhythmic) position of the arm at the initiation of this phase.

We develop more in details the equations of the motor primitives and the inverse kinematic technique in the next

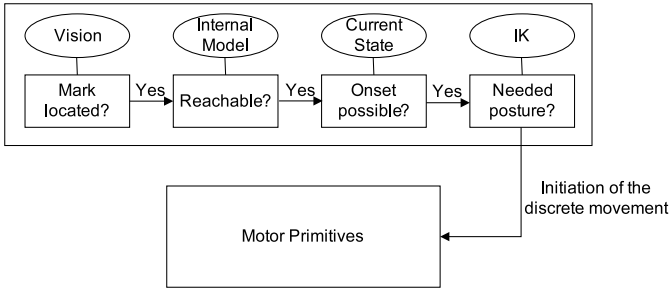


Fig. 2. Implementation of the decision layer. A discrete movement can be initiated when a mark is reachable if the limb is in the swing phase.

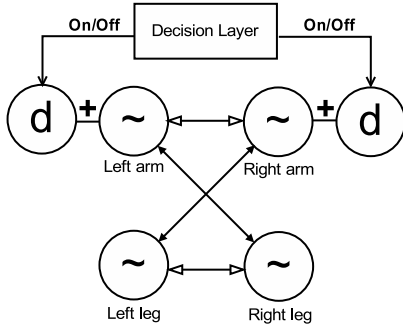


Fig. 3. Architecture of the controller: For each of the four joints the rhythmic primitive is active. The decision layer switch on and off the discrete motor primitives, if they are on, they are added to the corresponding rhythmic primitives. d stands for the discrete motor primitive and the symbol for the rhythmic one. The different arrows between the rhythmic primitives represents different couplings, i.e. the full arrows stand for in phase couplings and the empty ones for anti-phase couplings, see section III for more details.

section.

III. DYNAMICAL SYSTEM

We present here the two motor primitives we will use to generate the rhythmic and the discrete primitives. For the rhythmic primitive, we use a controller for crawling developed by Righetti *et al* [14] in the framework of the RobotCub project. The discrete primitive will be generated using a stable second order system.

A. Rhythmic motor primitive

Analysis of crawling infants have shown that most infants crawl on hands and knees, using a walking trot gait ([15],[16]). This is a gait where diagonally opposed limbs move in synchrony and in anti-phase with the other limbs (trot gait) with a very high duty factor (proportion of the duration of the stance phase over one complete cycle), that is comprised between 50 and 70%. Moreover it has been shown that the speed of locomotion is linearly correlated with the inverse of stance duration and that the swing phase stays almost constant at every speed as for other quadrupeds ([16]).

In term of kinematics, the hip and shoulder have a simple pattern of single flexion (during swing) and extension (during stance). For infants with the highest duty factor, it also seems that the hip and shoulder joints slow down during the swing phase of contralateral limbs ([11]). The movement of the

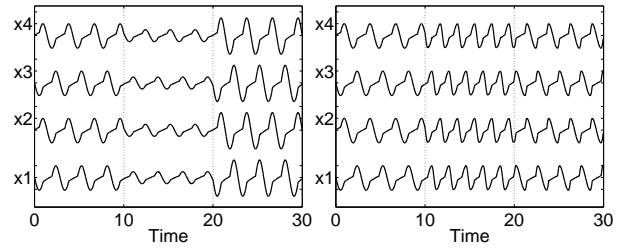


Fig. 4. Trajectories obtained for the crawling. On the left, we modulate the amplitude and on the right, the frequency.

elbow is also rather simple, with a flexion during swing phase to avoid contact with the ground.

To reproduce these characteristics, specific coupled oscillators for the trajectory generation of the hips and shoulders were designed. Their general equations are

$$\dot{x}_i = y_i \quad (1)$$

$$\dot{y}_i = \alpha_i y_i (K_i (\mu^2 - x_i^2) - y_i^2) - K_i x_i - c_1 y_j + c_2 y_l \quad (2)$$

with

$$K_i = \frac{k_{stance}}{(e^{-by_i} + 1)(e^{-ky_j} + 1)} + \frac{k_{swing}}{e^{by_i} + 1} \quad (3)$$

$$\alpha_i = \nu(1 + \beta e^{-\sigma x_i^2}) \quad (4)$$

and where $i=1,\dots,4$ denotes the i th oscillator, j the oscillator controlling the contralateral limb and k the diagonally opposed oscillator. c_1 and c_2 are positive coupling constants ensuring a trot gait, figure 3 depicts those couplings. k_{stance} and k_{swing} control the duration of the stance and the swing respectively. K_i is defined such that it switches between k_{stance} and k_{swing} and that when a limb initiates a swing phase, the opposite limb will slow down its movement. b and k are positive constants that control the speed of switch and slow down respectively. α_i is a positive constant for energy damping. Typical trajectories obtained using this controller are shown in figure 4.

Righetti *et al.* ([11]) have proved that this system was generating a stable walking trot gait. We will refer to this set of equations as the rhythmic motor primitives in the rest of the article.

The advantage of using dynamical system is that parameters can be smoothly modulated in order to change the characteristics of the movement. For instance, the speed of the movement can be modulated online by changing the stance duration, i.e. k_{stance} in our controller. The step length may also be modulated by modifying the amplitude μ of the movement. Examples of such modulations are shown on figure 4.

B. Discrete motor primitive

To generate the discrete movements, we use a simple nonlinear system with a unique attractor, i.e. a system of the form

$$\dot{u}_i = v_i \quad (5)$$

$$\dot{v}_i = \frac{-\gamma^2}{4} (u_i - g_i) - \gamma v_i \quad (6)$$

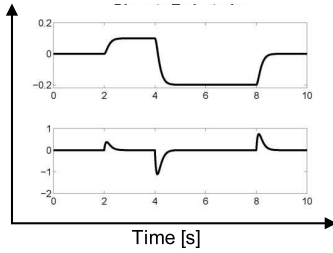


Fig. 5. Position (UP) and velocity (DOWN) profile of the discrete trajectories. Variations of parameter g through time induce trajectories converging to the value of g . We set $g=0.1$ during the interval [2, 4], $g=-0.2$ during the interval [4, 8] and $g=0$ otherwise.

The system has a unique attractor g and the speed of convergence is controlled by the parameter γ . A discrete movement is produced every time g is changed as depicted in figure 5. In the following, we will refer to this system as the discrete motor primitive.

Once again, it is beneficial to use a differential equation to model the motor primitive, notably because the solution of the system will smoothly adapt to variation of the parameter g . In our case, it will be of crucial importance since g correspond, up to a constant, to the angles determined by the inverse kinematics algorithm which are not fixed, but change with the displacement of the shoulder relatively to the mark during the crawling movement.

We have chosen a second order system¹ rather than a simpler first order equation in order to have a continuous velocity (rather than having a step-type curve of velocity).

In applications, attention must be paid to the limitation of the speed which increases with the difference $u-g$. This latter value should thus be bounded.

IV. APPLICATION OF THE MODEL TO ICUB

iCub is a 54 dofs baby robot developed by the RobotCub Consortium. As the robot is still under construction, we will use the mobile robot simulator Webots ([12]) in order to test our model. We control the robot in position.

We use three dofs in each limb, two in the shoulder/hip and one in the elbow/knee. During crawling, the shoulder and the hip flexion/extension joints are controlled by the rhythmic motor primitive introduced in section III. The shoulder abduction/adduction and the elbow joints are kept fixed apart from the beginning of the swing phase where they are in flexion to avoid collisions with the ground. The hip abduction/adduction and knee joints are kept fixed throughout the cycle. We do not use the rotational joints of the hips and shoulders.

To reach the target, we developed an inverse kinematics (IK) algorithm to get the angles values for each joint. This algorithm is active only when the mark is reachable by one of the hand, as defined in section II, and the distance to from the hand to the mark is calculated so that the closest hand is always used to reach it.

¹The system can be turned into a second order equation by simply replacing v by \dot{u} and \dot{v} by \ddot{u} in the second equation.

Moreover, once the mark has been touched, it is deactivated to avoid multiple stepping on the same mark. Finally, to avoid collisions with its own body, since the arm is very closed to the torso, we deactivate the IK at the very beginning of the swing phase.

Since we use two dofs in each shoulder and one in each elbow, the solution of the inverse kinematics is made unique by specifying the sign of the elbow angle. In future works, we may use the third dof of the shoulder (the rotation dof) to achieve more natural postures or to increase the ability to support the body weight of the arm.

To take in account the rhythmic movement of the joints, we subtract to the IK's angles the target angles of the rhythmic movement at the beginning of the stance phase. Doing so, the hand will be on the mark at the beginning of the stance only and thus will only slightly perturb the normal crawling movement.

An example is given on figure 6. At the beginning, the movement is purely rhythmic (black plain line), then after circa 4 seconds, a discrete movement (clear plain line) is initiated to reach a angle of 1.5 radians. However, since the rhythmic position is known to be 1 radian at the beginning of the stance, time when the mark has to be reached, the discrete movement should only compensate the difference of 0.5 radians. Again, after 8 seconds, an angle of 0.5 radians is demanded, and the discrete movement has to be negative in order to reduce the position of joint at the beginning of the stance.

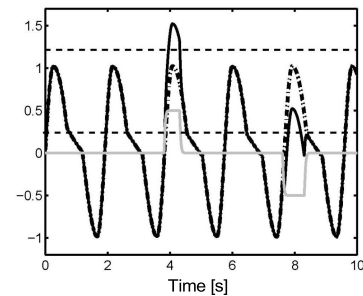


Fig. 6. Illustration of the inverse kinematics. The discrete movement is represented by the clear plain line, the purely rhythmic by the black dash-dot line and the sum of them by the black plain line. Discrete movement is initiated at $t = 3.7$ and $t = 7.8$ to attain an angle of 1.5 and 0.5 rad respectively.

It can be easily seen on the graph that as soon as the perturbation induced by the discrete movement is removed, the trajectory resume to the purely rhythmic trajectory (dash-dot line).

Note that the parameters of the rhythmic movement are kept fixed during the simulation. The only parameters we control here are the switching on and off of the discrete primitive and the target of the discrete movement, i.e. the parameter g of the discrete motor primitive.

V. RESULTS

We present here the results obtained using the physic based simulator Webots. As our aim is to develop the minimal set of criteria to make to enable hands placement, we do not expect

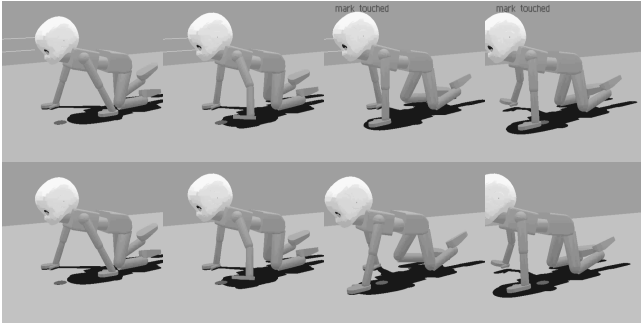


Fig. 7. Snapshots of the simulations. The baby robot has to reach the mark in front of his left arm (dark circle). First line shows the results with our controller and second line the results obtained when the robot crawl only .

it to work wherever the mark is. Moreover, it is sometimes simply physically infeasible to place one's hand on a mark, while keeping the same speed of locomotion, without falling.

As developed before, the IK is not active in the beginning of the swing phase to avoid collisions with the body and the mark is considered as having been touched after half a second of continuous contact with the hand of the robot, this representing half of the swing.

As a first experiment we asked the robot to reach only one mark with its left arm. Then, we placed two marks successively, so that it has to reach the first one with its left arm and the second one with the right arm immediately afterward.

A. First experiment: Reaching a single mark while crawling

To determine the behavior of the system, we start by a simple experiment. The robot is crawling and a mark is placed on the ground. When the mark is reachable, the discrete motor primitive is switch on so that the robot will place is hand on the mark. Coordinates of the mark are provided to the robot for the inverse kinematics.

Figure 7 shows some typical snapshots of the robot iCub successfully reaching a mark while crawling². On the first line, the baby robot has to reach a mark with his left hand, while on the second line we show snapshots of the same simulation, taken at the same times, but when only the rhythmic motor primitive is active (no reaching). One can see the clear difference between both movements.

Using our controller, the rhythmic movement is only slightly perturbed so as to allow the crawling movement to continue after the mark has been reached. The target and the actual trajectories of those snapshots are shown on figure 8 for solely crawling (UP) and for reaching while crawling (DOWN).

Our results have been quite satisfactory, especially relatively to the simplicity of the system. However, we noticed that deactivating the mark after it has been touched during a certain time (half a second here) was a necessary condition to permit the balancing of the body.

²Note that we selected the most pertinent snapshots of the movement, at irregular intervals. The complete sequences are available on the movie presented together with this article.

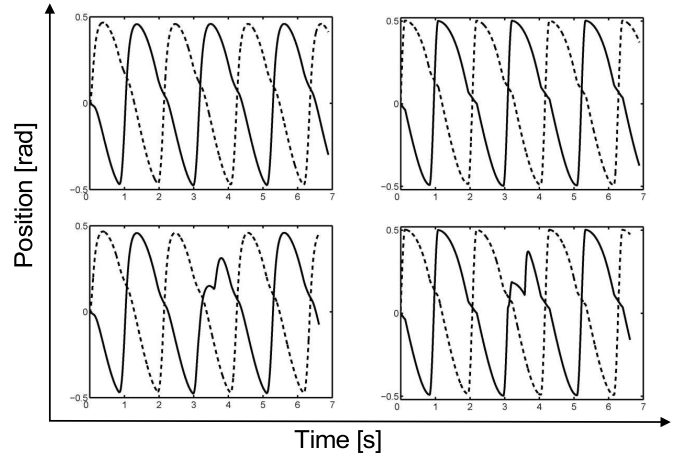


Fig. 8. Trajectories of the left arm (dash-dotted line) and of the right arm (plain line), obtained during the successful simulation illustrated on figure 7. UP: Crawling only, DOWN: Crawling and Reaching. LEFT: Actual trajectories, RIGHT: Target trajectories.

Indeed, keeping the arm fixed on the mark during the whole stance often stopped the progression of the body, making the robot fall. Snapshots of such a situation are shown in figure 9. Intuitively, the robot falls if the arm is in such a position that it cannot support the weight of the upper body and particularly of the head. On the second picture of the snapshots of fig 9, we see that the right arm of robot is too close to the knees at the beginning of the swing of the left arm, leading the robot to topple on his head. A simple ZMP control could be used to know whether the position is safe or not for the robot. In the latter case, the robot should not try to reach the mark.

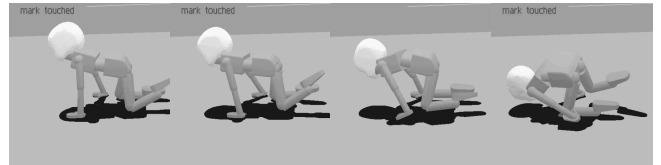


Fig. 9. Snapshots of a situation where the robot falls.

B. Second Experiment: Reaching two marks successively while crawling

To be efficient, the controller must be fast enough so that the robot can recover quickly from the physical perturbation induced by the reaching movement. Thus performed a second experiment, where two marks are placed on the ground so that the robot will have to reach the first mark and the then the second one immediately afterward. Snapshots of the results, and the corresponding trajectories, are shown in figure 10 and 11 respectively.

Again, supporting results have been obtained. Even if the crawling movement was sometimes disturbed, the robot usually finally achieve to reach the marks and to continue crawling afterward.

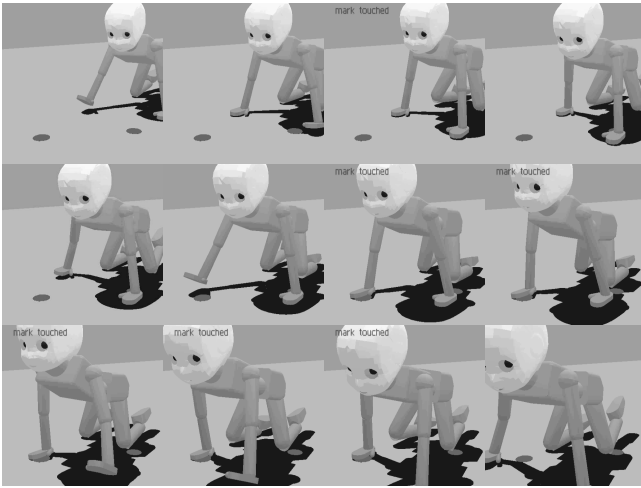


Fig. 10. Snapshots of a situation where the robot has to reach two marks.

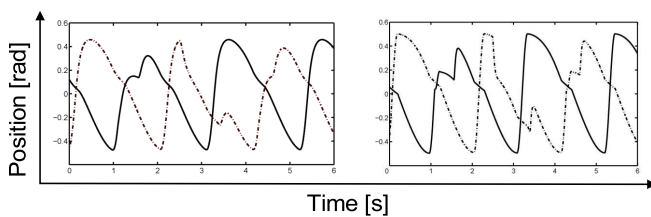


Fig. 11. Target and actual trajectories of the left arm (dash-dotted line) and the right arm (plain line) obtained during the successful simulation illustrated in fig. 10. LEFT: Actual trajectories RIGHT: Target trajectories.

VI. CONCLUSION

We have presented a controller for hands placement during crawling. The model we presented is very simple and does not take into account yet any sensory-motor feedback except the current states of the system in the decision layer.

Nevertheless, we obtained stimulating results, even if a systematical study of the obtained trajectories relatively to the position of the mark shall be performed in the future. Two experiments with successful results were presented, one where the robot has to place its hands on one mark and another one where there are two marks on the bounce.

We noticed that fixing the hand on the mark during the whole swing sometimes lead the robot to fall, however this was avoided by deactivating the mark after it had been touched for more than half the duration of the swing. In the future we would also like to improve our inverse kinematics algorithm by introducing the third joint of the shoulder in order to maximize the support abilities of the arm during the stance phase.

As our aim was to tackle the problem, no planning strategies as modification of speed or step length several steps in advance have been considered. However, as dynamical systems can be modulated online (as shown in section III), and the modulation being smooth, sensory feedback may be added easily to the motor primitives. In particular, strategies to keep the balance of the body may be developed, as moving the other arm to keep the body balance for instance.

Thus, planning using sensory data can be used to modulate the trajectories and improve the adaptability to unknown variations of the environment. For instance, some planning techniques such as adaptation of speed several steps before the reaching might be considered.

Finally analysis of kinetic data of crawling babies reaching target may allow to validate or to show the drawbacks of the method. A deeper study of feet placement during locomotion may also give interesting insights on the coordination of rhythmic and discrete movements under stability constraints.

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] N.G. Tsakarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, J. Santos-Victor, A.J. Ijspeert, M.C. Carrozza, and D.G. Caldwell. iCub - The Design and Realization of an Open Humanoid Platform for Cognitive and Neuroscience Research. *Journal of Advanced Robotics, Special Issue on Robotic platforms for Research in Neuroscience*, 21(10), 2007. In press.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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 *Neurons, Networks, and Motor Behavior*, 2001.
- [7] 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 International Conference on Humanoid Robots (HUMANOIDS06)*, 2006.
- [8] L. Righetti, J. Buchli, and A.J. Ijspeert. From dynamic hebbian learning for oscillators to adaptive central pattern generators. In *Proceedings of 3rd International Symposium on Adaptive Motion in Animals and Machines – AMAM 2005*, page 45. Verlag ISLE, Ilmenau, 2005. Abstract.
- [9] M.C. Tresch, Ph. Saltiel, and E. Bizzi. The construction of movement by the spinal cord. *Nature Neuroscience*, 2:162–167, 1999.
- [10] E. Bizzi, A. D'Avella, P. Saltiel, and M. Tresch. Modular organization of spinal motor systems. *Neuroscientist*, 8(5):437–442, 2002.
- [11] 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.
- [12] O. Michel. Webots tm: Professional mobile robot simulation. *International Journal of Advanced Robotic System*, 1:39–42, 2004.
- [13] J.K. Hodgins and M.H. Raibert. Adjusting step length for rough terrain locomotion. *IEEE Transactions on Robotics and Automation*, YEAR = 1991, VOLUME = 7, NUMBER = 3, PAGES = 289–291,.
- [14] L. Righetti, J. Buchli, and A.J. Ijspeert. Dynamic hebbian learning in adaptive frequency oscillators. *Physica D*, 216(2):269–281, 2006.
- [15] K.E. Adolph, B. Vereijken, and A.B. Denny. Learning to crawl. *Child Development*, 69:1299–1312, 1998.
- [16] L. Righetti, A. Nylén, K. Rosander, and A.J. Ijspeert. Kinematics of crawling in infants. In *Submitted to the Journal of Experimental Biology*.