Simulated Visually-Guided Paw Placement During Quadruped Locomotion

Miguel Oliveira
Department of Industrial Electronics
Engineering School
University of Minho
Guimarães, Portugal
Email: mcampos@dei.uminho.pt

Cristina P Santos
Department of Industrial Electronics
Engineering School
University of Minho
Guimarães, Portugal
Email: cristina@dei.uminho.pt

Manuel Ferreira
Department of Industrial Electronics
Engineering School
University of Minho
Guimarães, Portugal
Email: mjf@dei.uminho.pt

***Abstract*—Autonomous adaptive locomotion over irregular terrain is one important topic in robotics research. In this article, we focus on the development of a quadruped locomotion controller able to generate locomotion and reaching visually acquired markers. The developed controller is modeled as discrete, sensory driven corrections of a basic rhythmic motor pattern for locomotion according to visual information and proprioceptive data, that enables the robot to reach markers and only slightly perturb the locomotion movement. This task involves close-loop control and we will thus particularly focus on the essential issue of modeling the interaction between the central nervous system and the peripheral information in the locomotion context. This issue is crucial for autonomous and adaptive control, and has received little attention so far. Trajectories are online modulated according to these feedback pathways thus achieving paw placement. This modeling is based on the concept of dynamical systems whose intrinsic robustness against perturbations allows for an easy integration of sensory-motor feedback and thus for closed-loop control.**

The system is demonstrated on a simulated quadruped robot which online acquires the visual markers and achieves paw placement while locomotes.

I. INTRODUCTION

Autonomous visually-guided adaptive locomotion over irregular terrain is a very challenging task which is not yet completely solved. Mainly, research in the field addresses the problem of pre-computing desired trajectories [11], [14], [9], and adaptation to unpredicted changes is still an unsatisfactory solved problem.

The work presented in this article is part of a larger project which aims at developing a closed loop control architecture based on dynamical systems for the autonomous generation, modulation and planning of complex motor behaviors for legged robots with many DOFs. We apply autonomous differential equations to model how behaviors related to locomotion are programmed in the oscillatory feedback systems of Central Pattern Generators (CPGs) in the nervous systems.These systems are solved using numerical integration.

Control approaches based on CPGs and nonlinear dynamical systems are widely used in robotics to achieve tasks which involve rhythmic motions including autonomous adaptive dynamic walking over irregular terrain [13], [10], [7], juggling [15], drumming [16], and basis field approaches for limb movements [12].

This dynamical systems approach model for CPGs presents multiple interesting properties comparatively to other methods [9]. These include: low computation cost which is well-suited for real time; the stability properties of the limit cycle behavior (i.e. perturbations are quickly forgotten); intrinsic robustness against small perturbations; the smooth online modulation of trajectories through changes in the dynamical systems parameters and phase-locking between the different oscillators for different DOFs. Further, these systems, once coupled, produce coordinated multidimensional rhythms of motor activity, under the control of simple input signals.

To tackle both the complexity of movement generation and the complexity inherent to the design of dynamical systems, we assume that any movement can be decomposed in simple rhythmic and discrete primitives that we model by simple, stable, dynamical systems. This movement decomposition and the chosen primitives are supported by current neurological and human motor control findings.

As a main application we address the topical issue of robust, adaptive visually-guided quadruped locomotion in unknown, rough terrain. As a first step in this direction, we focus on visually-guided feet placement, that is to develop a controller able to generate quadruped locomotion and to smoothly modulate these trajectories to reach visually acquired markers.

The motor pattern generator (MPG) is implemented as two embedded dynamical discrete and rhythmic systems. The controller is modeled as discrete, sensory driven corrections of a basic rhythmic motor pattern for locomotion according to visual information and proprioceptive data, that enables the robot to reach markers and only slightly perturb the locomotion movement. This task involves close-loop control and we will thus particularly focus on the integration of sensory-motor information in the architecture. Trajectories are online modulated according to these feedback pathways thus achieving paw placement.

We propose a visual system able to accurately recognize and localize a size-known square with a predefined pattern inside. This visual system applies different image processing and image analysis techniques and the overall result is a robust method able to calculate the marker position with a high accuracy.

978-1-4244-4649-0/09/$25.00 ©2009 IEEE

We present results using a simulated ers-7AIBO robot in Webots [5] that show how the developed system successfully recognizes, localizes and reaches markers while locomotes. Different markers are chosen to show different reaching positions. Different markers were also successfully ignored.

Quadruped walking control using CPGs exploring sensory feedback integration into the locomotion control has been extensively explored by Hiroshi Kimura and his colleagues. Feet placement has also been extensively studied before but usually accuracy is the main requirement. However, we address paw placement in the framework of dynamical systems with superposition of discrete and rhythmic movements. We build on previous work, where controllers were developed for combining discrete and rhythmic motor primitives in drumming and dancing tasks [16], [17]. In this article, we focus the issue of modeling the interaction between the central nervous system and the peripheral information. This issue is crucial for autonomous and adaptive control, and has received little attention so far. The intrinsic robustness of the dynamical systems approach against perturbations allows for an easy integration of sensory-motor feedback and thus for closed-loop control. The proposed work tries to serve these purposes and focus on the integration of sensory-motor information in the developed dynamical architecture.

[6] proposed a very similar architecture for a quadruped hand placement. Herein, we extend and change the proposed model. First, we effectively introduce sensor-motor feedback through the visual system which modulates the generated trajectories. Second, the controller is slightly changed because the discrete system is embedded onto the rhythmic one. This approach is more consistent with our previous work [16], [17] and we assure that trajectories are in fact generated by attractor solutions.

In this article, we first present the overall system architecture and the set of rules used to integrate the sensory information. In section III, we present the locomotion controller architecture able to generate locomotion and how we integrate sensory feedback onto the architecture. Next, we detail the proposed visual system. In Section IV, we present the simulation results obtained and some limitations of the proposed method. We conclude by discussing the main results we obtained, possible improvements to the system and the work we are currently working on.

II. SYSTEM ARCHITECTURE

Our aim is to propose a control architecture that is able to generate locomotion for a quadruped robot and change the generated trajectories such that a limb may reach a visually acquired marker. These trajectories should be smoothly modulated when simple control parameters change. Further, this controller should be as simple as possible in order to enable the inclusion of other higher controls (herein we address feet placement but see [8] for balance control or [7] for sensory feedback inclusion).

The proposed controller is modeled as discrete, sensory driven corrections of a basic rhythmic motor pattern for locomotion in order to achieve reaching of a marker. The rhythmic movement induces the velocity and step length of the robot and its parameters are kept fixed. The discrete movement specifies the offsets around which the rhythmic movement occurs. However, these offsets change and depend on the visually detected marker, on the current joint values and on the robot internal model. Thus, trajectories generated by this architecture are modulated by sensory feedback.

The overall system architecture is depicted in fig. 1.

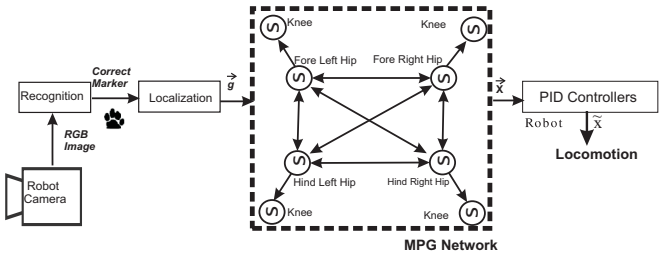

Fig. 1. The overall system architecture.

The MPG is implemented as two embedded dynamical systems and superimposes discrete and rhythmic motor primitives. This enables to independently control these primitives thus keeping the individual intrinsic stability and robustness properties against perturbations and allow for an easy integration of sensor-motor feedback and thus for closed-loop control.

Within the MPG, the discrete system specifies an offset for the rhythmic movement, that enables the robot to reach markers and only slightly perturb the locomotion movement. This offset, g, is given by

$$g_{\mathrm{i}} = g_{\mathrm{i,d}} + (\theta_{\mathrm{i}} - \psi_{\mathrm{i}}), \tag{1}$$

where i are the DOFs; $g_{\mathrm{i,d}}$ is the default offset value for locomotion behavior only; θ_{i} is the joint angle calculated by an inverse kinematics (IK) algorithm corresponding to the marker coordinates localized by the visual system and ψ_{i} is the joint angle corresponding to the end-effector position at the beginning of the stance phase in case no marker had been localized (that is, limb exhibiting rhythmic movement with the default offset value). Note that marker detection occurs during the swing phase of the limb that must reach the marker. Thus, marker localization can be described in the robot internal frame and directly sent to the IK algorithm that calculates the required joint angles for reaching the marker. However, reaching occurs only at the beginning of that limb stance phase.

The final trajectories x_{i} specify the planned joint values needed to generate locomotion and reaching. These are sent online for each DOF and the lower level control is done by PID controllers.

Fig. 2 shows the set of rules and procedures that define if the update of the offset, g_{i}, should occur.

First, computer vision techniques recognize if the acquired RGB image contains the correct marker. Second, a localization module determines the marker center (X, Y, Z) coordinates in

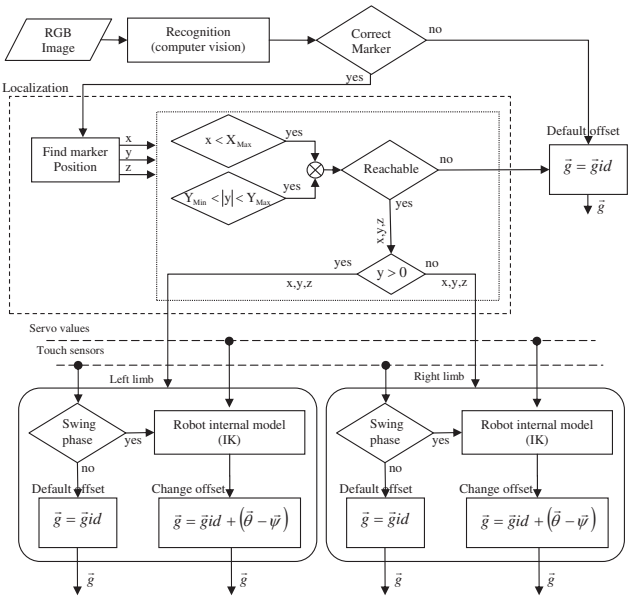

Fig. 2. Update of the offset values, g_i, of the discrete system.

the robot internal frame. The next procedure is to determine if the marker is reachable in the current limb step by one of the fore legs. A marker is considered to be reachable if: (1) it can be reached in the current step ($X < X\_\text{Max}$); and (2) it is a physically feasible place to be reached without making the robot fall ($Y\_\text{Min} < Y < Y\_\text{Max}$).

If the marker is reachable, the offsets g_i of the leg that will reach the marker are updated according to the robot internal model and the actual state of the touch sensors.

Offsets g_i of the reaching limb are set equal to the default values (i.e. values required for locomotion behavior only without reaching) after the marker has been touched during a certain fixed time. This time was set to 3/4 (0.7 s) of the swing of the hind limb that succeeds the limb that reached the marker. This condition was necessary to achieve a better locomotion in terms of balance and to enable the locomotion to normally continue after reaching the marker.

III. LOCOMOTION CONTROLLER

In this section we present our model of the MPG used to generate the trajectories for one DOF. The rhythmic movement is generated by an Hopf oscillator. The discrete primitive is generated by a stable dynamical system such that it integrates visual sensory information and proprioceptive data onto the controller that generates the trajectories.

A. Motor Pattern Generator

1) Rhythmic Movement Generation: Rhythmic movements are generated by the following Hopf oscillator

$$\dot{x}_i = \beta\left(\mu_i - r_i^2\right)(x_i - y_i) - \omega z_i, \quad (2)$$

$$\dot{z}_i = \beta\left(\mu_i - r_i^2\right) z_i + \omega\,(x_i - y_i), \quad (3)$$

where $r_i = \sqrt{(x_i - y_i)^2 + z_i^2}$, amplitude of the oscillations are given by $R = \sqrt{\mu_i}$, ω specify the oscillations frequency (in rad s^{-1}) and relaxation to the limit cycle is given by $\frac{1}{2\beta\mu_i}$.

This Hopf oscillator contains a bifurcation from a stable fixed point at $x_i = y_i$ (when $\mu_i < 0$) to a structurally stable, harmonic limit cycle, for $\mu_i > 0$. The fixed point x_i has an offset given by y_i, which is the state variable of the discrete system. The y variable evolution will be specified and explained in the next subsection.

We apply an Hopf oscillator because it can be completely analytically solved, which facilitates the smooth modulation of the generated trajectories according to changes in the amplitude, goal and frequency parameters. This is interesting for trajectory generation in a robot.

In [16] it was shown how the generated trajectories can easily and smoothly be modulated by modifying on the fly the offset values (y variable).

2) Discrete Movement Generation: It is important that this discrete movement generator applies to the control of a real robot. Thus, the generated movement must be able to: 1) smoothly adapt to the control parameters and 2) allow trajectory modulation through changes in these control parameters. In our case, the discrete system specify the offsets around which the oscillations are generated in the hip and knee joints. Further, the solution of this discrete system must smoothly adapt to variations of a parameter g_i which changes and depends on the visually detected marker, on the current joint values, on the touch sensors and on the internal robot model. Therefore, to generate the discrete movements, we define a nonlinear dynamical system whose solution, given by y_i, is the offset of the output x_i (2).

$$\dot{y}_i = v_i, \quad (4)$$

$$\dot{v}_i = \frac{-b^2}{4}\,(y_i - g_i) - b\,v_i, \quad (5)$$

where speed of convergence is controlled by b to a unique attractive goal g.

B. Controller Architecture

Each DOF is controlled by one generic MPG. In order to ensure phase-locked synchronization between the different DOFs of the robot, we bilaterally couple the Hopf oscillators of the hips MPGs, those couplings being illustrated by right-left arrows on fig. 1 and unilaterally couple each hip MPG to the corresponding Knee MPG. This is achieved by modifying (2) and (3) of all the hips DOFs as follows:

$$\begin{bmatrix}\dot{x}_{i[1]}\\ \dot{z}_{i[1]}\end{bmatrix} = \begin{bmatrix}\beta\mu_i & \omega\\ -\omega & \beta\mu i\end{bmatrix}\begin{bmatrix}x_{i[1]} - y_{i[1]}\\ z_{i[1]}\end{bmatrix} - \beta r_{i[1]}^2\begin{bmatrix}x_{i[1]} - y_{i[1]}\\ z_{i[1]}\end{bmatrix} + \sum_{j\neq i}\mathbf{R}(\theta_{i[1]}^{j[1]})\begin{bmatrix}x_{j[1]} - y_{j[1]}\\ z_{j[1]}\end{bmatrix}$$

For the knee joints, we modify (2) and(3) as follows:

$$\begin{bmatrix}\dot{x}_{i[3]}\\ \dot{z}_{i[3]}\end{bmatrix} = \begin{bmatrix}\beta\mu i & \omega\\ -\omega & \beta\mu i\end{bmatrix}\begin{bmatrix}x_{i[3]} - y_{i[3]}\\ z_{i[3]}\end{bmatrix} - \beta r_{i[3]}^2\begin{bmatrix}x_{i[3]} - y_{i[3]}\\ z_{i[3]}\end{bmatrix} + \frac{1}{2}\mathbf{R}(\psi_{i[3]}^{j[1]})\begin{bmatrix}x_{j[1]} - y_{j[1]}\\ z_{j[1]}\end{bmatrix}$$

where $r_i[k]$ is the norm of vector $(x_i[k], z_i[k])^T$ (k = 1,3). The linear terms are rotated onto each other by the rotation matrices

TABLE I
PHASE DIFFERENCES BETWEEN HIP OSCILLATORS (i[1]) FOR A WALKING GAIT.

	$\theta_{FLL-FRL}$	$\theta_{FLL-HLL}$	$\theta_{FLL-HRL}$	$\theta_{FRL-HLL}$	$\theta_{FRL-HRL}$	$\theta_{HLL-HRL}$
(°)	-180	-270	-90	-90	90	180

$\mathbf{R}(\theta_{i[1]}^{j[1]})$ and $\mathbf{R}(\psi_{i[3]}^{j[1]})$, where $\theta_{i[1]}^{j[1]}$ is the desired relative phase among the i[1]'s and j[1]'s MPGs and $\psi_{i[3]}^{j[1]}$ is the desired relative phase among the i[3]'s and j[1]'s MPGs (i, j = FLL, FRL, HLL, HRL). In our case, we set these values according to table I, which defines the phases required for performing a walking gait (we exploit the fact that $\mathbf{R}(\theta) = \mathbf{R}^{-1}(-\theta)$). The $\psi_{i[3]}^{j[1]}$ were all set to -90^o. Due to the properties of this type of coupling among oscillators, the generated trajectories are always smooth and thus potentially useful for trajectory generation in a robot.

A current limitation of the proposed locomotion controller is that the movement of the end-effectors does not have the ideal shape. This movement should not be oscillatory and should be different during stance and swing phases. This is mainly due to the fact that a MPG generates both hip and knee trajectories. Currently, we are addressing this problem by generating knee trajectories not by a MPG but by a set of rules which define the best shape for the end-effector movement. Inclusion of feedback loops for robustness and independent control of swing and stance duration [7], [6], are presently being take into consideration.

IV. VISUAL SYSTEM

In order to choose which marker to use, meaning which vision cue to detect and track, there are several issues to be considered, such as: the clutter environment degree; the processing time; the illumination drifts; the accuracy of the position and pose estimation of the marker.

The spectrum of techniques for object tracking, a crucial research issue in robot vision especially for the applications where the environment is in continuous changing and uncontrolled, has been increasing over the past decade. The most common approaches are based mainly on the detection of one of the following cues: edges, color and texture [1], [2]. Presently, techniques for texture detection still demand a high processing time, and for that reason this cue was not considered in this work.

For objects tracking avoiding the problems that are present in clutter environments, namely drift in light conditions and presence of an uncontrolled number of colors, the most common approaches use specific markers and use edges cues [3]. The author uses a size-known square marker for fast tracking and for high accuracy of the position and pose estimation of the markers. This algorithm considers only one parameter to adapt the system to different light conditions, which can be determined automatically with a generic illumination calibration technique.

A. Tracking Vision Module

To obtain a high degree of robustness and at the same time real-time constrains the approach followed in our work is based on the [3] using the size-known square marker of Fig. 3.

The tracking module is responsible for accurately determine the position of the marker (Xm, Ym, Zm) and the type of marker. This module makes extensively use of computer vision techniques for edge detection, pattern recognition and camera position and pose estimation. Fig. 4 shows the architecture of the tracking vision module. At this stage, the system is searching for a specified marker and the camera is already calibrated, meaning the intrinsic and extrinsic parameters are known.

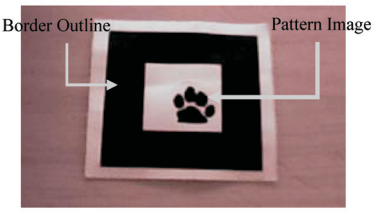

Fig. 3. Marker used in our work.

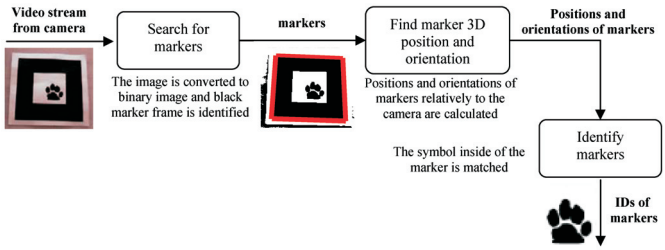

Fig. 4. Tracking vision module architecture. Adapted from [4].

B. Search for Markers

This procedure looks for blobs that can be fitted by four line segments. Since the pattern is a black and white image, the first step consists on a thresholding procedure. For each blob identified in the image the outline contours are extracted and a four straight lines fitting procedure is applied to identify the candidates of markers. The equation parameters of the lines and the intersection of the lines (vertex) are used in the pose estimation procedure.

C. Pattern Matching

After the detection of the border outline the system is able to identify the pattern image. For that a template matching is performed with patterns previously specified by the user.

The pattern images are 16x16 pixels in gray level at different rotations: 0°, 90°, 180° and 270°. Fig. 5 shows the store patterns for the marker used.

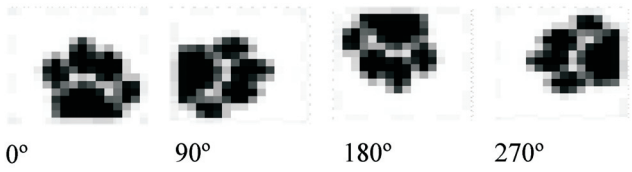

Fig. 5. Stored patterns for the marker used.

D. Position and Pose estimation of Markers

In order to determine the position and the pose estimation of the markers a set of coordinates must be specified. Fig. 6 presents those used in this work following Kato work [3].

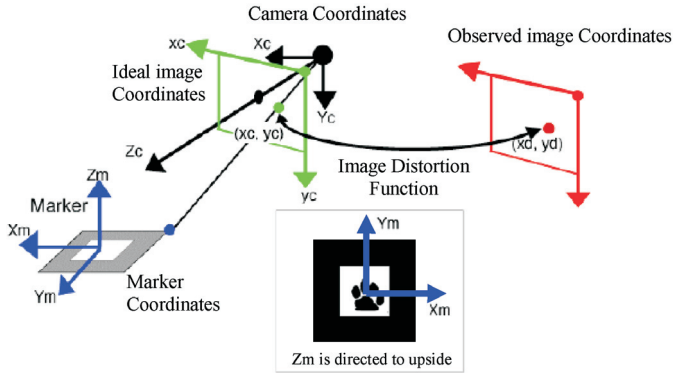

Fig. 6. System coordinates. Adapted from [4].

The transformation from the marker coordinates to the camera coordinates is given by (6).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} & V_{3x3} & & W_{3x1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = T_{cm} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (6)$$

Observing the image coordinates of the four marker vertices it is possible, by means of geometrical calculations, to estimate the T_{cm} matrix using the relation of (7), and minimizing the error (8) described by iterative optimization.

$$\begin{bmatrix} h\,\hat{x}_{ci} \\ h\,\hat{y}_{ci} \\ h \end{bmatrix} = P\; T_{cm} \begin{bmatrix} X_{mi} \\ Y_{mi} \\ Z_{mi} \\ 1 \end{bmatrix}, i = 1,2,3,4, \quad (7)$$

$$err = \frac{1}{4} \sum_{i=1,2,3,4} \{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\} \quad (8)$$

where P is the camera perspective projection.

E. Discussion

The vision tracking module has been developed to work in the AIBO simulator of cyberbotics [5] and in the AIBO memory stick. For that, an image processing framework (camera calibration, morphological and spatial filters, image binarization, blob analysis and pattern recognition techniques), independent of the platform, was developed. This framework was also integrated with the new versions of webots simulator from cyberbotics.

The low spatial resolution of the patterns templates decreases the number of patterns to use and impedes the detection of missing details in the patterns. One of the main advantage is the low processing time to perform the matching: around 20ms for the controller under OPEN-R. The best patterns are those that are asymmetric and do not have fine detail on them [4].

Concerning the thresholding procedure the threshold value used to convert the acquired image to a binary image was 100. However, to counterbalance poor lighting, which would affect the desaturation of the image, as it could result in areas that are actually white being assumed as black, an illumination calibration procedure can be used and this value can be determined on-line.

V. RESULTS

In this section, we describe experiments done in a simulated ers-7 AIBO robot using Webots [5]. This simulator is based on ODE, an open source physics engine for simulating 3D rigid body dynamics. The model of the AIBO is as close to the real robot as the simulation enable us to be. Thus, we simulate the exact number of DOFs, mass distributions and the visual system.

The ers-7 AIBO dog robot is a 18 DOFs quadruped robot made by Sony. The locomotion controller generates the joint angles of the hip and knee joints in the sagittal plane, that is 8 DOFs of the robot, 2 DOFs in each leg. Flap joint angles were simply set according to the returned values from the IK algorithm. Only walk gait is generated and tested.

The AIBO has a camera built into its head. The neck joints, which position the AIBO head, have been moved to values such that the camera is able to acquire a marker within the current step of the locomotion. The other DOFs are not used for the moment, and remain fixed to an appropriately chosen value during the experiments.

At each sensorial cycle, sensory information is acquired, dynamic equations are calculated and integrated thus specifying servo positions. The dynamics of the CPGs are numerically integrated using the Euler method with a fixed time step of 1 ms. Parameters were chosen in order to respect feasibility of the experiment and are given in table II. We recorded the actual trajectories from the joints incremental encoders $\vec{\tilde{x}}$ and the planned trajectories $\vec{x}$.

A. First Experiment

In a first attempt to verify the proposed approach and its integration with the visual system, two different marks are

TABLE II
PARAMETER VALUES USED IN THE EXPERIMENTS.

	β	ω (rad s^{-1})	μ_i	$\frac{1}{2\beta\mu_i}$ (s)
FLS	0.1	2.044	6.25	0.8
FRS	0.1	2.044	6.25	0.8
HLS	0.025	2.044	25	0.8
HRS	0.025	2.044	25	0.8
FLK	0.011	2.044	56.25	0.8
FRK	0.011	2.044	56.25	0.8
HLK	0.051	2.044	12.25	0.8
HRK	0.051	2.044	12.25	0.8

placed in spots slightly apart from the path the robot would have in case no perturbations arose. The first mark is the good mark (Fig. 3), meaning the one the visual system should correctly recognize and localize. The second mark should not be recognized by the visual system. The first mark is positioned slightly to the right of the robot at $(X,Y,Z) = (169.5,-78.1,-148.8)$ (mm) coordinates (in the internal robot frame), which is a physically feasible place to be reached by the robot fore right paw, but obliges some additional movement of the flap joint.

Snapshots of the AIBO robot while locomotes and successfully reaching the first mark and ignoring the second one are depicted in Fig. 7. The visual system successfully detected the first mark and returned its localization at $(X,Y,Z) = (169.5,-74.1,-143)$ (mm) (in the internal robot frame), meaning an error quite small.

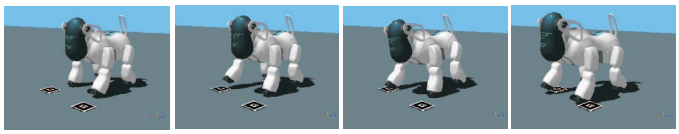

Fig. 7. Snapshots of the experiment: The robot successfully reaches the correct mark to the right of its front right limb and ignores the second mark while locomotes. Time increases from left to right.

Fig. 8 depicts the relevant variables for the robot fore left limb of the snapshots illustrated in Fig. 7. Top panel shows the planned $\vec{x}$ (dashed line) and actual $\vec{\tilde{x}}$ trajectories (solid line) for the hip joint. The three bottom panels illustrate the end-effector position in the robot internal frame. Swing and stance phase are also identified.

At $t = 5.37$s, the mark is correctly recognized and localized and the IK algorithm returns the required joint angles to reach the mark at the beginning of the stance phase. The planned trajectories for each joint are changed accordingly and the end-effector position is as expected at $t = 6.15$s (time when the mark is reached). This mark is deactivated at $t = 6.85$s.

B. Second Experiment

A second more complex experiment is attempted. The robot must reach two successive marks with its fore right and left paws. The first and second marks are placed at $(X,Y,Z) = (195.8,-128.1,-150.2)$ (mm) $(X,Y,Z) = (188.9,68.7,148.5)$ (mm) coordinates (in the internal robot frame), which are

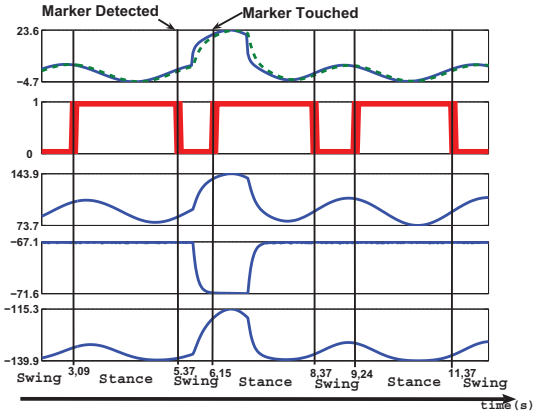

Fig. 8. Variables for the robot fore right limb of the snapshots illustrated in Fig. 7. Up panel: $\vec{x}$ (dashed line) and $\vec{\tilde{x}}$ (solid line) (rad); Middle panel: Touch sensor; 3 Bottom panels: x,y,z (mm) end-effector coordinates in the robot internal frame.

physically feasible places to be reached by the robot fore right and left paws, respectively.

Fig. 9. Snapshots of the second experiment: The robot successfully reaches both markers.

In Fig. 9 we can see a set of snapshots of the experiment, where the robot successfully reaches the two marks. The corresponding trajectories and relevant variables are shown in Fig. 10 and 11 for right and left limbs, respectively.

VI. CONCLUSION

In this article, we have presented a locomotion controller that generates quadruped locomotion and modifies online the generated trajectories to reach visually acquired markers for paw placement. Trajectories are online modulated by modifying on the fly some control parameters according to the visual acquired information and proprioceptive data. The controller superimposes discrete and rhythmic movement primitives.

Our main contribution was to visually recognize and localize a marker and to easily integrate this information onto a controller that is able to modulate and generate locomotion

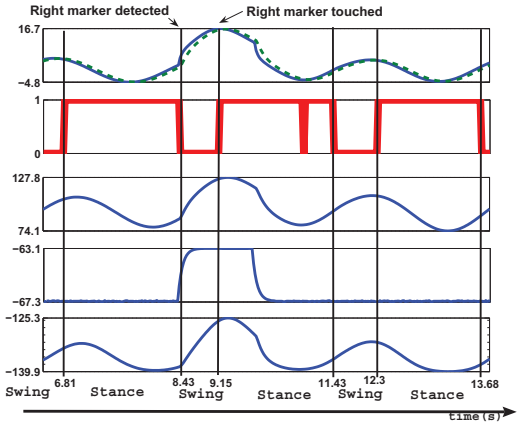

Fig. 10. Similar to Fig. 8 but for snapshots illustrated in Fig. 9.

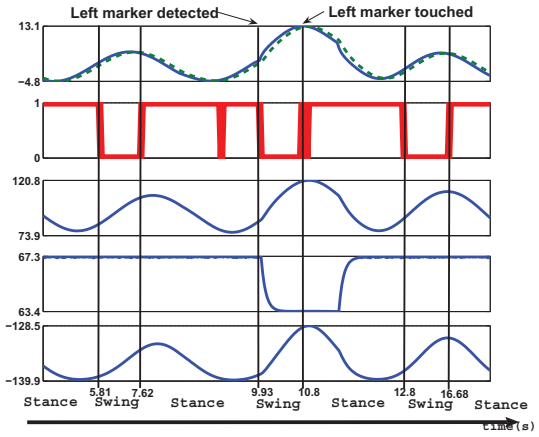

Fig. 11. Similar to Fig. 10 but for fore left limb.

trajectories. The visual system uses different image processing and image analysis techniques. The marker is a size-known square with a predefined pattern inside. These two marker characteristics allow the calculation of its position with a high accuracy even with a single camera and with a low resolution image, in a low processing time. Other main advantage of the system is its robustness even when the illumination suffers some drifts and the environment is complex.

We presented successful results for different experiments in which a simulated ers-7 AIBO robot must position its fore right and left paws onto markers with predefined characteristics. Different markers were also tested and successfully ignored.

Despite the good results, there is still lots of work to do in order to completely generate adaptive locomotion able to achieve feet placement. These include but are not restricted to: inclusion of feedback loops to independently control swing and stance durations [7]; accurate feet placement; predictive adjustment of locomotion including speed and/or step length control in advance and balance considerations. It is our believe and motivation that the dynamical systems framework has the properties that make it suitable to generate more complex behavior able to adapt to the surrounding environment. As previously stated, the current locomotion controller is not the ideal to generate the correct end-effector movement and current work is being done is this direction.

We are currently extending this work in order to achieve the generation and switch among different gaits according to the sensorial information; to integrate other sensory-motor feedback loops for robust and stable locomotion; to achieve head stabilization for image acquisition and combining with previous work for posture and balance control [8]. We are also improving the current visual system such that the marker can be detected in advance to enable for movement planning in anticipation.

REFERENCES

[1] M. Pressigout and E. Marchand, "Real-time planar structure tracking for visual servoing: a contour and texture approach", *in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IROS'05, Alberta, Canada, 2005.

[2] G.R. Bradski, "Computer vision face tracking as a component of a perceptual user interface", *in Workshop on Applications of Computer Vision*, Princeton, NJ, 1998, pp. 214-219.

[3] H. Kato and M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", *in Proc. IWAR '99*, San Francisco, CA, 1999, pp.85-94.

[4] H.Kato, ArToolKit tutorials, ARToolworks, Inc., Seattle, WA, USA, URL http://www.hitl.washington.edu/artoolkit/tutorials.htm, 2007

[5] O. Michel. Webots tm: Professional mobile robot simulation, *International Journal of Advanced Robotic System*, vol 1, 2004, pp. 39-42.

[6] Sarah Degallier, Ludovic Righetti, and Auke Ijspeert. "Hand placement during quadruped locomotion in a humanoid robot: A dynamical system approach." *In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*,California, 2007,pp. 2047-2052.

[7] Ludovic Righetti and Auke Jan Ijspeert. "Pattern generators with sensory feedback for the control of quadruped locomotion". *IEEE International Conference on Robotics and Automation ICRA 2008*, California, 2008.

[8] Castro, Luiz, Santos, C P; Oliveira, M and Ijspeert, A, "Postural Control on a Quadruped Robot Using Lateral Titl: a Dynamical System Approach", *European Robotics Symposium EUROS 2008*, Prague, 2008.

[9] J Pratt, C Chew, A Torres, P Dilworth, G Pratt, Virtual Model Control: An intuitive approach for bipedal locomotion, The Int. J. of Robotics Research, 2001, 20 (2), 129-143

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

[11] A Ijspeert, J Nakanishi, S Schaal, Learning attractor landscapes for learning motor primitives, Advances in Neural Information Processing Systems 15, 2003, 1547-1554

[12] S Giszter, F Mussa-Ivaldi, E Bizzi. Convergent force fields organized in the frog's spinal cord. J. of Neuroscience, 13:467–491, 1993

[13] G Taga, Emergence of bipedal locomotion through entrainment among the neuro-musculo-skeletal system and the environment, Physica D, 1994, 75 (1-3), 190-208

[14] R Blickhan. The spring-mass model for running and hopping. J. Biomechanics, 22(11–12):1217–1227, 1989

[15] M Bühler, S Koditscheck. Planning and control of a juggling robot. Int J of Robotics Research, 13(2), 101-118, 1994

[16] S Degallier, C P. Santos, L Righetti and A Ijspeert – Movement Generation using Dynamical Systems: a Drumming Humanoid Robot. In Humanoids'06 IEEE-RAS International Conference on Humanoids Robots, Genova, Italy, December 4-6 2006

[17] Santos, C P; Ferreira, M; Oliveira, M; Pires, A; Dégallier, S and Ijspeert, A -Choreography generation for a quadruped robot using dynamical systems, Autonomous Robots (submitted)