Predictive Gaze Stabilization During Periodic Locomotion Using a Feedforward-Feedback Controller based on Adaptive Frequency Oscillators

Sébastien Gay, Auke Ijspeert and José Santos Victor

Abstract— **In this paper we present an approach to the gaze stabilization problem using Adaptive Frequency Oscillators to learn the frequency, phase and amplitude of the optical flow and generate compensatory commands during robot locomotion. Assuming periodic and nearly sine shaped motion of the robot, the system successfully stabilizes the gaze of the robot, whether the robot itself is moving, or an external object is moving relative to the robot. We present experiments in simulation and with a real robotics setup, the Hoap 3, showing that the system can be successfully applied to gaze stabilization during locomotion, even when the feedback loop is very slow and noisy.**

I. INTRODUCTION

Vision is, for animals and robots, the most versatile sensor to provide information about the surrounding environment. However, vision is most efficient when the image (and thus the gaze) is stable since a moving gaze causes motion blur. Evolved animals use saccades when switching gaze direction to minimize the time during which the image is moving. During locomotion, compensatory movements of the eyes and head aim at minimizing the retina slip. The same issue is present when dealing with robots since most vision processing algorithms reach optimal performance with a stable image.

Head stabilization systems exist in the robotics literature, many of them being based on models of the vestibulo-ocular reflex observed in many vertebrates. These systems typically use a vestibular sensor as main sensory input to excite a leaky integrator. The remaining retinal slip (usually measured by optical flow) is then used to calibrate the gains of this integrator. [14] applies Kawato's Feedback-Error-Learning model [5] to the gaze stabilization problem and extends it with a nonparametric regression network to improve the opto-kinetic response. [6] implements the Recurrent Decorrelation Control model [3] which forms a recurrent network with an artificial brainstem getting as input rotational speeds from the vestibular sensor, and an artificial cerebellum getting input from the brainstem and the retinal slip, and feeding back its output to the brainstem. [9] uses a single neural network excited directly by both the vestibular sensor output and the optical flow from the camera image to estimate the optimal compensatory motor command.

These systems reach very good performance but rely highly on the availability of a fast (typically around 500Hz) vestibular sensor in the head of the robot. Very few approaches tackle the problem of head stabilization specifically during locomotion. The work in [12] relies on a forward kinematics and genetic algorithm to build an internal model of the head motion and compensate for it using a feedforward CPG based controller. This method however relies on offline optimization for the CPG parameters which has to be done for each different gait and is thus not very suitable for gaits changing in time (to cope with environmental specificities for instance).

In this paper we propose a system for stabilizing the head of a legged robot during locomotion, which only relies on optical flow information. Assuming a periodic movement of the head (as is usually the case for legged locomotion), the system uses Adaptive Frequency Oscillators to learn the frequency and phase shift of the optical flow and generate compensatory movements to minimize the head motion. At convergence, the system is mostly feedforward and the feedback signal (the optical flow) is only used to finely tune the parameters of the oscillator. This system is efficient even when using relatively slow cameras ($< 30Hz$) and is predictive in the sense that unlike reactive systems which use the last few sensor values to estimate the amplitude of the compensatory movement at the next step, our controller generates a compensatory signal which is phase locked with the optical flow signal. It effectively tries to predict the future, as the stabilizing commands are generated at a higher frequency than the optical flow. Our system is able to track changes in the movement applied to the robot and adapt its parameters to go back to a stabilized gaze. We show that our system can be used to stabilize the gaze of a moving robot using multiple degrees of freedom in the head. Further, it can be applied to tracking objects of arbitrary shape, colors and textures.

In the following sections we present the system and its properties, then explain the influence of the different open parameters of the system. We explain how to use the system to stabilize the gaze using multiple degrees of freedoms in the head, and show that it can even stabilize the gaze of a robot on a moving object. We show that the system can be applied on legged locomotion (with the Hoap2 humanoid robot walking) and non legged locomotion (with a swimming salamander robot), as long as the movement of the head is periodic and close to a sine wave. We present the system applied on the real Hoap3 robot tracking a periodically moving apple.

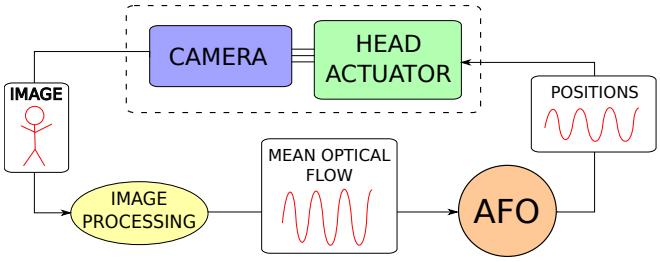


Fig. 1: Outine of a simplified version of the system

II. PRESENTATION OF THE SYSTEM

In this section we present the details of the head stabilization controller. First we present a simplified version of the controller using a standard Hopf Adaptive Frequency Oscillator as first developed by Buchli, Righetti and Ijspeert ([2], [10]), and then show how we adapted it to satisfy the requirements of the head stabilization problem.

Figure 1 outlines the architecture of the system. Images from the camera to stabilize are processed to obtain a measure of the optical flow using the standard OpenCV [1] implementation of the Lucas Kanade - Shi Tomasi algorithm ([7], [13]). The optical flow signal is fed negatively to an Adaptive Frequency Oscillator which will tune its frequency, amplitude and phase shift so as to generate a signal in phase with its teaching signal (in anti phase with the optical flow), with the correct amplitude to minimize the optical flow. The output of the AFO is then used to control the head of the robot. We use here a slightly modified version of the Hopf AFO in polar coordinates in which we removed the influence of the forcing signal on the radius of the oscillator, to avoid divergence with high coupling terms. The equations of the AFO are given below:

$$\dot{r} = \gamma(1 - r^2)r \quad (1)$$
$$\dot{\phi} = \omega - \sin\phi\epsilon F \quad (2)$$
$$\dot{\omega} = -\sin\phi\kappa F \quad (3)$$
$$x = r\cos\phi \quad (4)$$
$$\dot{\alpha} = -\eta x F \quad (5)$$
$$\theta = \alpha x + O \quad (6)$$

where r is the radius of the oscillator (i.e. the amplitude of its oscillations), ϕ its phase, ω its frequency and θ its output here used to control the position of the head actuator. α here directly defines the amplitude of the oscillations and O their offset. F is an external forcing signal (here the opposite of the mean optical flow). κ and ϵ are scaling factors for the forcing signal. We describe the effect of these scaling factors in Section III. Equations 1 and 2 describe a limit cycle of radius 1. The forcing term in Equation 2 causes the phase to synchronize with that of the forcing signal, while a similar forcing on ω (Equation 3) tunes the frequency to that of the forcing signal. When the oscillations are synchronized (same frequency and same phase) with the forcing signal, the correlation between x and F is maximized and α starts increasing, causing the head of the robot to oscillate in anti phase with the optical flow with increasing amplitude, and thus decreasing the retinal slip, until the flow is null. All the parameters of the generated compensatory signal are effectively learned such that they are conserved if the forcing term F is removed. This is particularly useful to deal with varying camera speeds, communication problems or occlusion.

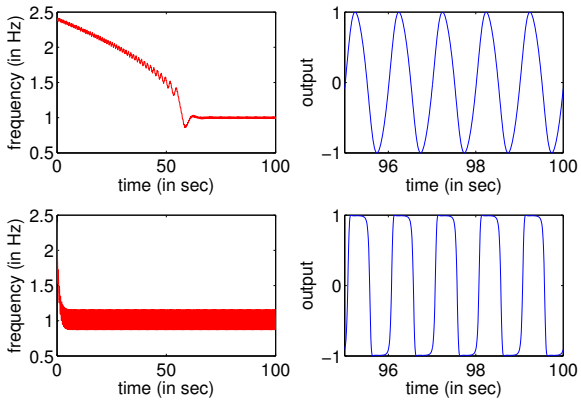


Fig. 2: Evolution of the frequency (left) and shape of the output (right) for small scaling factors $\kappa = \epsilon = 2$ (top) and big scaling factors $\kappa = \epsilon = 50$ when the AFO is forced by the signal $F = sin(2\pi t)$.

Theoretically, this system works by itself. However, the convergence of the frequency is typically slow in the experiments by Buchli and Righetti (a few hundred seconds). This is mainly due to the fact that setting high values to ϵ and κ changes a lot the shape of the oscillations of the AFO as well as issues discussed in Section III. Figure 2 shows how the output of the oscillator is modified when ϵ and κ are increased. When ϵ and κ are high, the shape of the oscillations is highly modified from the original cosine wave. Furthermore, having too high coupling terms, when dealing with head stabilization, would cause divergence of the system. Indeed, since a jerky output as in Figure 2 (bottom) would cause a high optical flow which would in term induce a higher forcing etc.

To solve this problem and obtain fast convergence of the frequency while keeping control on the shape of the oscillations we used two phases for the AFO. The first phase, ϕ_1, is used only to learn the frequency of the forcing signal. The second phase ϕ_2 is the actual phase of the oscillations, and is coupled to the forcing signal for synchronization, with a different coupling term β. Typically we set $\beta << \kappa$ so that the shape of the oscillations is not altered too much. These two phases share the same value for ω, so that the frequency learned is reflected on the oscillations of the head. Note that this system is equivalent to an AFO passing its frequency to a Hopf oscillator, and thus the proofs of convergence in [10] remain valid and the properties of the Hopf oscillator are conserved.

The equations of the final system become :

$$\dot{r} = \gamma(1 - r^2)r \quad (7)$$
$$\dot{\phi_1} = \omega - \sin\phi_1 \epsilon F \quad (8)$$
$$\dot{\phi_2} = \omega - \sin\phi_2 \beta F \quad (9)$$
$$\dot{\omega} = -\sin\phi_1 \kappa F \quad (10)$$
$$x = r\cos\phi_2 \quad (11)$$
$$\dot{\alpha} = -\eta x F \quad (12)$$
$$\theta = \alpha x + O \quad (13)$$

III. PARAMETER TUNING

In this section we study the influence of the parameters κ and ϵ on the convergence of the frequency of the system. Note that β only acts on the second phase ϕ_2 which has no influence on the frequency modulation. Figure 3 shows the results of systematic tests monitoring the convergence time and the error after convergence for varying values of κ and ϵ. The forcing signal used for this experiment was obtained by recording the optical flow when rotating a camera in the air around its pitch axis with a frequency of 2Hz (in the Webots robotics simulator [8]), and normalizing its amplitude. We used eight instances of our oscillator initialized at eight different frequencies uniformly distributed around the desired frequency.

The convergence time $\mathcal{T}_c(S)$ and error after convergence $\mathcal{E}_c(S)$ of a signal S (here the optical flow) to a desired value s are defined as follows:

$$\mathcal{T}_c(S) = \min(t), \forall t > \mathcal{T}_c(S), |S(t) - s| < \lambda \quad (14)$$
$$\mathcal{E}_c(S) = \frac{1}{\mathcal{T}_f(S) - \mathcal{T}_c(S)} \int_{\mathcal{T}_c(S)}^{\mathcal{T}_f(S)} |S(t) - s| dt \quad (15)$$

where $\mathcal{T}_f(S)$ is the final time of the signal S and λ is a chosen small value (in this study we used $\lambda = 0.25$). In clear, $\mathcal{T}_c(S)$ is defined as the minimum time after which the signal S stays bounded in a neighborhood of a desired value s, and $\mathcal{E}_c(S)$ as the mean of the instantaneous distance between $S(t)$ and s after $\mathcal{T}_c(S)$. These two quantities are then averaged over the eight oscillators.

The error after convergence $\mathcal{E}_c(S)$ (Figure **??**, right) is basically proportional to κ, although it sightly decreases when epsilon is increased for a given value of κ. The convergence time $\mathcal{T}_c(S)$ (left) decreases monotonically with κ and meets a minimum for a specific value of ϵ which depends on the value of κ. This minimum is however less visible when κ increases.

Figure 4 shows the evolution of the frequency of the system for characteristic values of κ and ϵ and for different initial frequencies. For small values of κ and ϵ (Figure 4a), the convergence takes a long time, especially for initial frequencies far away from the frequency of the forcing signal, while the remaining oscillations after convergence have very small amplitude. When ϵ and κ are increased (Figures 4b and 4c), the convergence time decreases but the oscillations after convergence amplify. Increasing only ϵ (Figure 4d) has a smoothing effect on the convergence. The AFOs with initial frequency far away from that of the forcing signal converge faster, while the others converge more slowly. Increasing κ while keeping ϵ low (Figure 4e) causes the convergence to be very jerky, and increases the amplitude of the remaining oscillations at convergence compared to when both parameters are set high (Figure 4c). Figure 4f shows an example of a compromise between convergence speed and error after convergence.

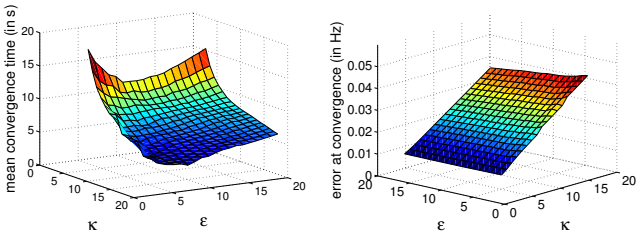


Fig. 3: Convergence time $\mathcal{T}_c(S)$ (left) and error after convergence $\mathcal{E}_c(S)$ (right) for different values of κ and ϵ when the system is forced by a normalized optical flow signal of frequency 2Hz.

This study will serve as a reference to choose the values of these parameters depending on the application, i.e. whether convergence speed or precision at convergence is more critical, but also depending of whether we can have a good estimate of the frequency of the head movement (in which case we can afford to set lower values for κ while still converging fast enough). Typically during locomotion and especially for statically stable gaits, the frequency of the head motion is nearly that of the controlled robot motion. In Section V we show that in the case of the salamander robot swimming and the Hoap2 robot walking, this is not true for the pitch axis.

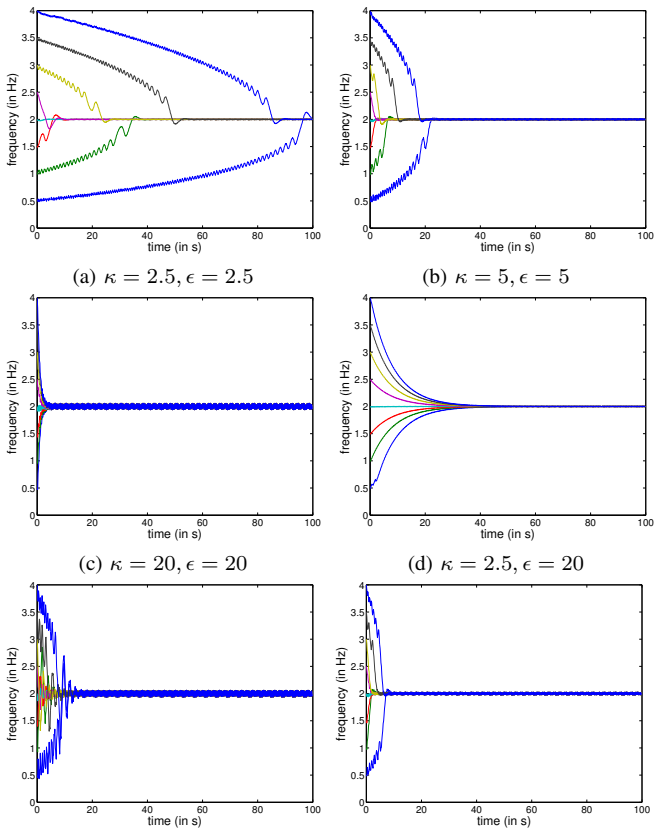
IV. EXTENSION TO MULTIPLE AXIS STABILIZATION

So far we have only considered one oscillator, for a single degree of freedom. However the system is fairly easy to extend to multiple degrees of freedom for the head. Typically one would use one AFO per degree of freedom. The only constraint here is finding the right forcing signal for each AFO.

To result in a successful head stabilization, the forcing signal for one degree of freedom should:

- have the same frequency as the motion of the robot around this axis.
- decrease towards zero when the head is stabilized around this axis.
- have zero mean.

Note however that the forcing signal does not need to be an estimate of the head rotation speed around the considered



(a) $\kappa = 2.5, \epsilon = 2.5$ (b) $\kappa = 5, \epsilon = 5$

(c) $\kappa = 20, \epsilon = 20$ (d) $\kappa = 2.5, \epsilon = 20$

(e) $\kappa = 20, \epsilon = 2.5$ (f) $\kappa = 8.7, \epsilon = 7.8$

Fig. 4: Evolution of the frequency of the system for typical values of κ and ϵ when the system is forced by a normalized optical flow signal of frequency 2Hz and for AFOs initialized with various initial frequencies.

axis in any way. In this paper we typically use the optical flow since it is the most basic information provided by a camera, but we could just as well use for instance the position of an object around the center of the image, or the position and orientation of the horizon.

To extend the system to three axis (pitch, roll, yaw, as commonly defined in aviation) stabilization, we use the following forcing term for the corresponding AFOs:

- For the pitch: the mean of the y coordinate of the flow vectors of the whole image.
- For the yaw: the mean of the x coordinate of the flow vectors of the whole image.
- For the roll: the difference between the mean of the y coordinate of the flow vectors of the left quarter and the right quarter of the image.

These three forcing terms are applied negatively to the AFOs, so that at convergence, the oscillators are in anti phase with the optical flow. The following equations formalize the forcing for the pitch, roll and yaw axis (respectively F_p, F_r and F_y)

$$F_p = -\frac{1}{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathcal{F}_{ij}^{y} \quad (16)$$

$$F_y = -\frac{1}{K}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathcal{F}_{ij}^{x} \quad (17)$$

$$F_r = -\left(\frac{1}{K_l}\sum_{i=1}^{m}\sum_{j=1}^{\frac{n}{4}}\mathcal{F}_{ij}^{y} - \frac{1}{K_r}\sum_{i=1}^{m}\sum_{j=\frac{3n}{4}}^{n}\mathcal{F}_{ij}^{y}\right) \quad (18)$$

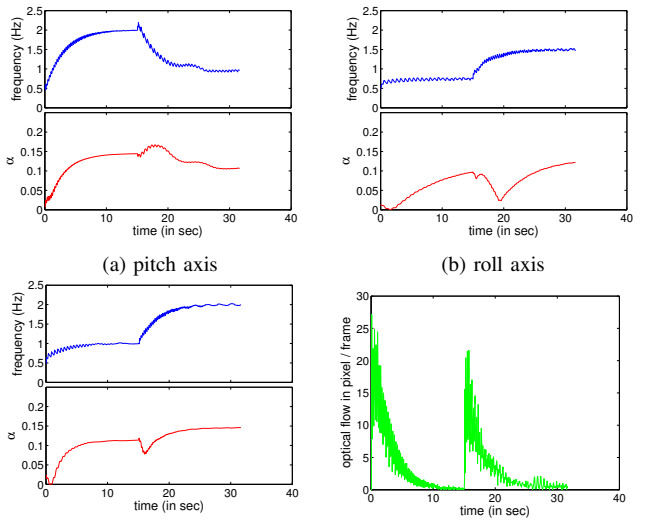
where K is the number of non zero flow vectors in the whole image, K_l and K_r are the numbers of non zero flow vectors in the left and right quarters of the image, m and n are the dimensions of the image, and F_{ij}^{x} and F_{ij}^{y} are the x and y components of the optical flow vector computed at position (i, j).

Note that these three forcing terms do not give a direct measure of the rotation speed of the head around each axis. This is not needed by our system. The forcing terms used for each axis need however to satisfy the two conditions given earlier. In our case, this implies that the pitch axis of the head moves the image approximately along its y axis, the yaw along its x axis, and that the roll rotates the image around its center. In the case of a head with two cameras on each side for instance, the forcing for the roll axis F_r may not work as it is. It could be adapted by taking the difference of the flow of the left part of the left camera image and the right part of the right camera image.

V. RESULTS

In this section we present results of the system actually applied to the head stabilization problem. All the experiments described below have been carried out using Webots [8], a physics based simulator for robotics. The camera is a simulated pinhole camera with a field of view of $45°$ and providing an image of 320 x 240 pixels at 20Hz. (which is below standard for robotics cameras). The reader is advised to refer to the video attached to this paper for a better insight of the following experiments.

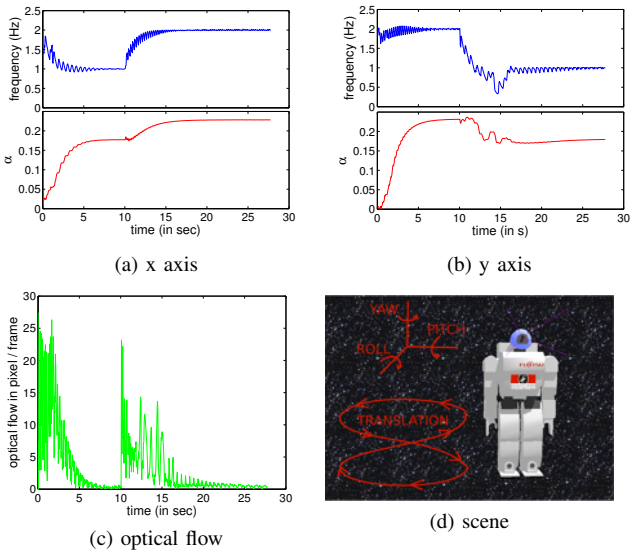
Figure 5 shows the evolution of the frequency and the amplitude of the system when a robot (here the Fujitsu Hoap 2 humanoid robot) is rotated in the air with sine waves of different frequencies for the pitch, roll and yaw axis (see Figure 6d). One instance of our oscillator is used per degree of freedom with different forcing signals as explained in Section IV. To demonstrate the self tuning ability of the system, the frequency of the motion for the pitch axis is set arbitrarily to 2Hz, for the roll 0.75Hz and for the yaw 1Hz. At t=15s, the frequencies are switched to: pitch axis: 1Hz, roll axis: 1.5Hz, yaw axis: 2Hz. The AFO is initialized with a frequency of 0.5 Hz.
The frequency of the AFOs controlling each actuator of the head quickly converges to that of the motions applied to the robot and the amplitude starts increasing until the optical flow is minimum. When the frequencies of motion are



(a) pitch axis (b) roll axis

(c) yaw axis (d) optical flow

Fig. 5: Evolution of the frequency and the amplitude (α) of the oscillator when the robot is rotated in the air around the three axis pitch, roll and yaw, by sine waves of different frequencies. For the pitch axis: 2Hz, for the roll axis: 0.75Hz and for the yaw axis: 1Hz. At t = 15s the frequencies are switched to: pitch axis: 1Hz, roll axis: 1.5Hz, yaw axis: 2Hz. The AFO is initialized with a frequency of 0.5 Hz. Figure 5d shows the evolution of the norm of the mean optical flow vector over time.



(a) x axis (b) y axis

(c) optical flow (d) scene

Fig. 6: Evolution of the frequency and the amplitude (α) of the oscillator when the robot is translated in the air along the x and y axis by sine waves of different frequencies. For the x axis: 1Hz and for the y axis: 2Hz. At t = 10s the frequencies are switched to: x axis: 2Hz, y axis: 1Hz. The AFO is initialized with a frequency of 2 Hz. Figure 6c shows the evolution of the norm of the mean optical flow vector over time.

suddenly altered, the system tracks the change of frequency and recovers until the optical flow is minimal again. The resulting flow after convergence is reduced to less than 1 pixel/frame both times, in about 10 seconds.

One of the main benefits of our system compared to those expecting rotational speed from an artificial vestibular system is that it does not assume that the movement to compensate is a rotation. It actually works as well for pure translations. Figure 6 describes a similar experiment as the previous one, but this time with the robot translated along the x and y axis (the y axis here is the vertical, while the x axis is sideways with respect to the robot) with sine waves of different frequencies: for the x axis 1Hz, for the y axis: 2Hz. At t = 10s, the frequencies are switched to 2Hz for the x axis and 1Hz for the y axis. Again the system converges quickly leading to the optical flow being almost completely suppressed. After the switch in frequency, the system recovers and goes back to nearly perfect stabilization.

As explained in Section II, our system generates oscillations whose shape can be slightly modified by the forcing signal, but remains close to a sine. Figure 7 shows the behavior of the system when the robot is rotated around its pitch axis with waves of different shapes. For every shape, the system manages to learn the frequency of the optical flow signal. It also manages to reduce the optical flow, leading to a more stabilized gaze than without the system. However, the further the shape of the rotation is from a sine, the worst the performance, as expected.

Figure 8 shows the performance of the system when the robot is rotated around its pitch axis with a chirp ($\sin(2\pi(\omega_0 + kt)t)$), first with a relatively slow changing frequency, and then with a much faster changing one. When the frequency of the robot rotation is changing slowly, the system is able to track these changes fast enough to enable good gaze stabilization. When the frequency of the movement is changing faster, the system still tracks it but not fast enough to lead to optimal performance stabilization.

Our system relying only on visual cues, it can also be used to stabilize the gaze of the robot on periodically moving objects of arbitrary shapes, colors etc. Figure 9 shows results of the system applied to the Hoap 2 robot tracking a sphere (the moon) being translated with a sine wave along the x and y axis (vertical and sideways). The robot is not moved in this experiment. The frequencies of the motion of the sphere along the x and y axis are set respectively to 2Hz and 1Hz. At t=15s, the frequencies are switched to 1Hz for the x axis and 2Hz for the y axis. The system is able to stabilize the gaze of the robot on the object almost perfectly, and tracks changes in the movement of the object. The result is the object staying almost exactly in the center of the image after convergence (about 5s).

We now show the system applied to actual robot locomotion. Figure 10 shows the evolution of the frequency and the amplitude of the oscillators controlling the pitch, roll and yaw axis of the head of a simulated salamander robot

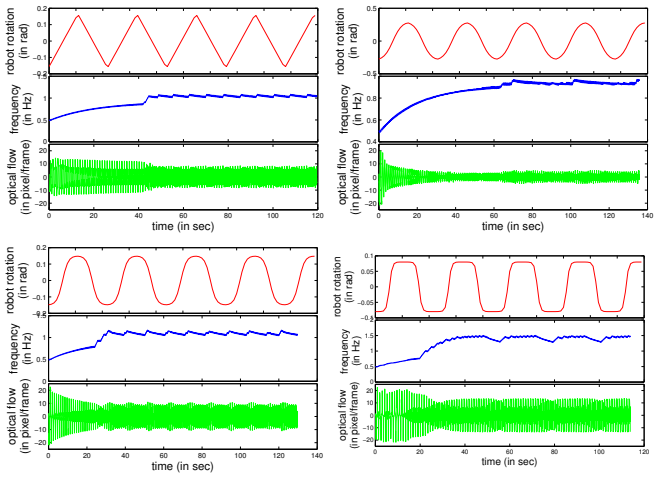


Fig. 7: Evolution of the frequency (middle) of the head stabilizing oscillator and the optical flow (bottom) when the robot is rotated around its pitch axis with waves of different shapes (top), from near triangle to nearly step functions.

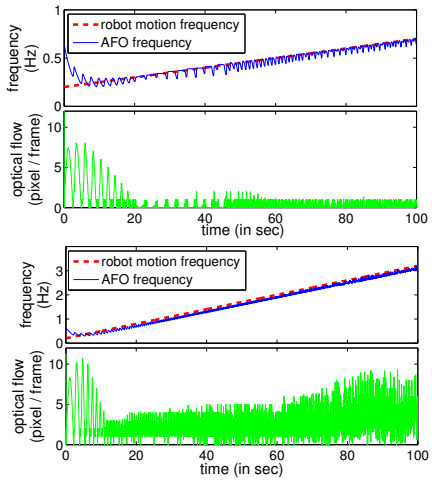
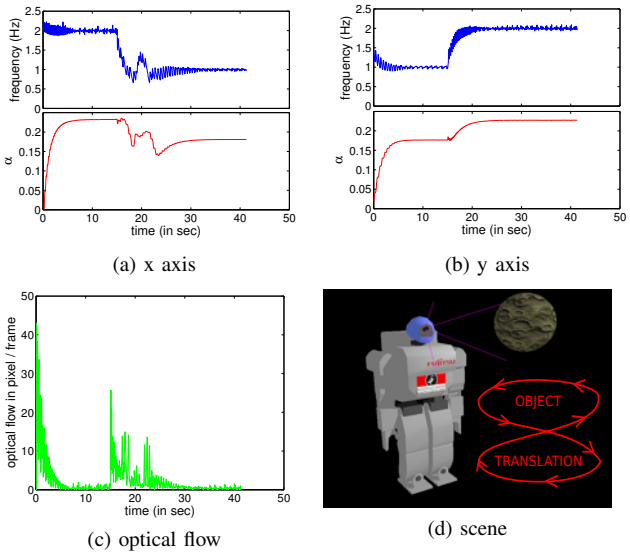


Fig. 8: Evolution of the frequency of the head stabilizing oscillator and the optical flow when the robot is rotated around its pitch axis with a a sine wave with frequency increasing in time. Top: slow changing frequency, Bottom: fast changing frequency.

swimming. The salamander robot ([4]) is a modular 12DoF robot controlled with coupled oscillators (central pattern generators), and is capable to switch from walking to swimming. For this experiment, it is swimming by generating a traveling wave along its body whose frequency and amplitude can be modulated. The frequency of this wave is initially set to 1Hz. At t=30s, the frequency is switched to 1.5Hz. The frequency of the oscillators is initialized to 0.5Hz. Again the system successfully stabilizes the head of the robot along the two axis, and tracks the change of frequency of the motion. The remaining optical flow after convergence is due partially to the forward motion of the robot, as shown in Figure 10d.

In the case of the salamander swimming, we could have initialized the frequency of the head stabilizing oscillator to the frequency of the motion control (we did not to



(a) x axis (b) y axis

(c) optical flow (d) scene

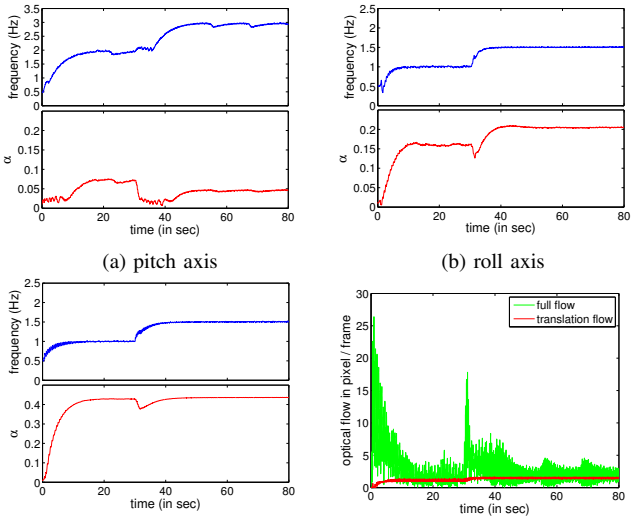
Fig. 9: Evolution of the frequency and the amplitude (α) of the oscillator when an object (here a sphere) is translated in the air along the x and y axis by sine waves of different frequencies. For the x axis: 2Hz and for the y axis: 1Hz. At t = 15s the frequencies are switched to: x axis: 1Hz, y axis: 2Hz. The AFO is initialized with a frequency of 2 Hz. Figure 9c shows the evolution of the norm of the mean optical flow vector over time.

demonstrate the tuning abilities of the system). Note however that the frequency of the motion of the head around the pitch axis is twice that of the general motion of the robot. The head is diving in the water at each half period of the traveling wave controlling the robot. This particularity is highly related to the gait used here and is very difficult to predict a priori (it would need complex modeling of the fluid dynamics). Our system however learns the correct frequency for this axis without the need of any modeling.

Figure 11 shows snapshots of the salamander swimming, with and without the head stabilization system enabled.

Figure 12 shows a similar experiment as above with the Hoap2 walking. Three axis stabilization is used in the same way as for the other experiments. The robot is controlled using the default gait provided by Fujitsu. The frequency of the motion is not altered for this experiment, since the gait is only stable with the precomputed parameters. The frequency of the AFOs is initialized to 2Hz (different from that of the motion). Figure 12e shows the shape of the robot motion at the base of the head, for each axis. Even though this motion is quite far from a sine wave, the parameters of the system converge and gaze stabilization reaches decent performance, with the optical flow after convergence reduced to less than 7 pixels/frame. Note that, as in the case of the salamander robot swimming, the frequency of the motion around the pitch axis is different to that of the other axis.

We performed the object tracking experiment with the real Hoap3 robot, which has embedded cameras in its head.



(a) pitch axis (b) roll axis (c) yaw axis (d) optical flow

Fig. 10: Evolution of the frequency and the amplitude (α) of the oscillator when the salamander robot swimming. The frequency of swimming is initially 1Hz and at t = 30s the frequency is switched to 1.5Hz. The AFO is initialized with a frequency of 0.5Hz. Figure 10d shows the evolution of the norm of the mean optical flow vector over time as well as the flow due to the forward motion of the robot

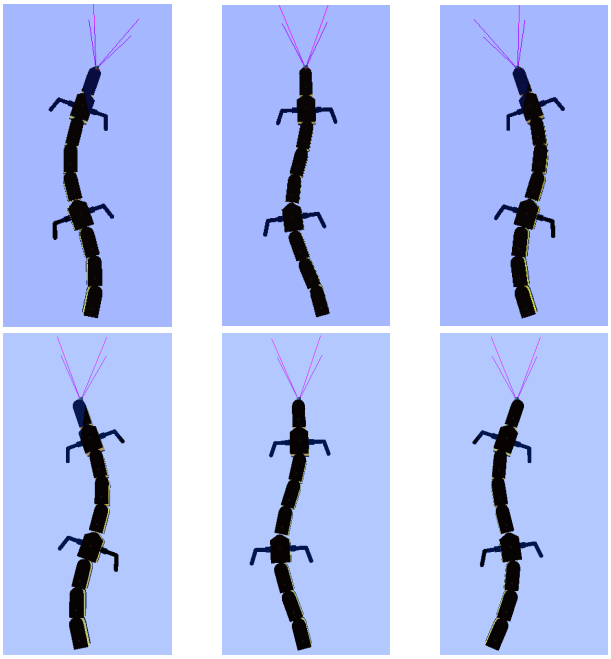
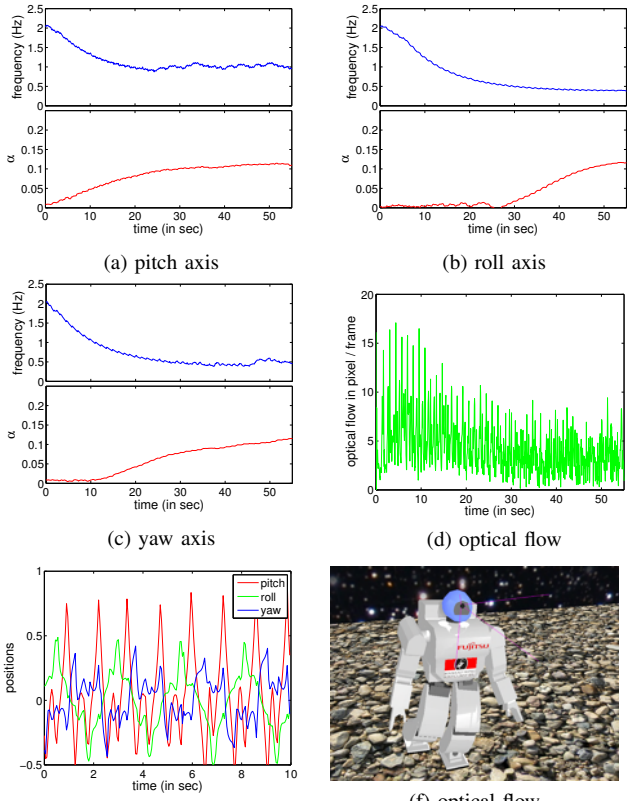


Fig. 11: Snapshots of the salamander robot swimming without head stabilization (top) and with head stabilization (bottom). When the head stabilization system is enabled, the gaze (highlighted by the purple camera frustums) always points in the direction of motion.



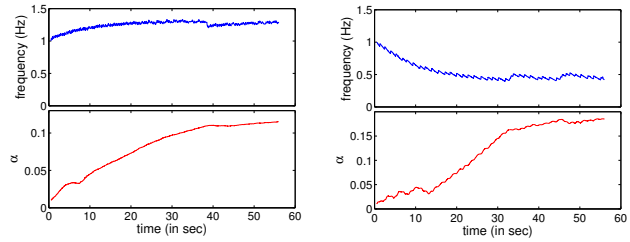
(a) pitch axis (b) roll axis (c) yaw axis (d) optical flow (e) head base rotation (f) optical flow

Fig. 12: Evolution of the frequency and the amplitude (α) of the oscillator when the Hoap2 robot is walking. The AFO is initialized with a frequency of 2Hz. Figure 12d shows the evolution of the norm of the mean optical flow vector over time. Figure 12e shows the shape of the rotation speed of the base of the head.

An apple was attached to a spring, allowing it to swing horizontally and vertically, with different frequencies. We used here the exact same system as in the experiments in simulation. Taking into account the framerate of the camera, the computation time of the optical flow and communication delays, we can provide our oscillator with visual forcing at a frequency of about 10Hz. Figure 13 shows the evolution of the frequency and amplitude of the two axis controlling the head. Here, the optical flow was not a good measure of the performance of the system, due to the high noise even after stabilization (see attached video). Instead we used simple blob tracking to compute the position of the apple in the image frame (Figure 13c). Even with such a slow and noisy optical flow, the system is able to stabilize the object around the center of the image. Around t=30s, the stabilization around the yaw axis gets worse for a couple of seconds, but quickly recovers. Note that the frequency of the apple motion is not perfectly constant here due to the natural damping of the spring and the air friction.

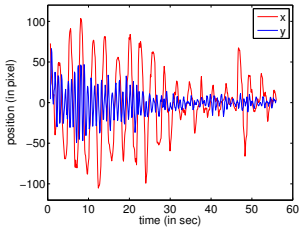
VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a novel approach to the head stabilization problem. Our system uses only visual cues,

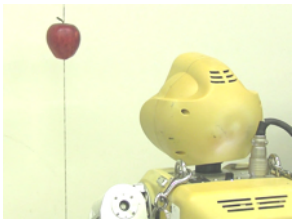


(a) head pitch

(b) head yaw



(c) apple position



(d) scene

Fig. 13: Evolution of the frequency and the amplitude (α) of the oscillator when an apple is oscillating in front of the real Hoap3 robot. The apple is attached to a spring allowing it to swing horizontally and vertically.

here optical flow, to stabilize the head of a robot subject to periodic motion, typically during locomotion. The system tries to predicts the motion of the robot, by learning the frequency, phase and amplitude of the optical flow. All the learning is done online, and embedded into the dynamics of the designed oscillator such that changes in the parameters of the motion are tracked by the system. We showed that our system can be applied to stabilize the gaze when the robot is being moved, or when it is tracking a periodically moving object. We also showed that the system can successfully stabilize the head of the robot during actual locomotion, with a biped and an anguilliform robot, without the need for fast sensors. We demonstrated the performance of the system on a real robotics setup. More experiments on real robot should follow. Another planned experiment would consist in stabilizing an actuated camera attached to a real human or animal during locomotion.

The main limitation of the current system is the single shape of the output of the oscillator. We assume that the necessary compensatory motion to stabilize the head close to a sine wave. This is not always the case during locomotion. The shape of the oscillations is slightly modulated by the feedback term, but being able to generate the exact right oscillation patterns would surely increase the performance of our system. Future work should include learning the whole shape of the robot motion. This could be done for instance by deducing the shape of the compensatory signals from the optical flow, and adding a dynamical filter to our oscillator (for instance of a combination of sine waves with different frequencies and amplitudes), or by designing an adaptive Gaussian mixture filter. Using a pool of coupled Adaptive Frequency Oscillators as done in [11] would be another solution to generate more complex shapes for the head motion.

The approach described in this paper uses only vision as sensory feedback. This is a big advantage of the system since it can be applied to a wide variety of robots which do not necessarily have a large set of sensors. However, this is not a limitation of the system, and one could imagine fusing information from more sensors, depending on the application. A simple way to fuse information from a vestibular system and cameras, for instance, could be to use two different forcing signals for our oscillator. The forcing in Equation 8 would come from the vestibular system while the one in Equation 9 would come from visual cues. This would increase the speed and smoothness of the convergence of the frequency, while keeping the head motion phase locked with the optical flow.

Finally, let us note that only the parameters of the nominal gait of the robot are learned. Fast changes of the head motion pattern are not stabilized. For some applications, this could be a downside of the system. However, during locomotion, fast changes of optical flow after head stabilization could be a sign that some unexpected events are occurring, e.g. the robot loosing balance, and this information could be used to trigger a response in a higher level controller.

REFERENCES

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] Jonas Buchli and AJ Ijspeert. Self-organized adaptive legged locomotion in a compliant quadruped robot. *Autonomous Robots*, pages 331–347, 2008.

[3] Paul Dean, John Porrill, and James V Stone. Decorrelation control by the cerebellum achieves oculomotor plant compensation in simulated vestibulo-ocular reflex. *Proceedings. Biological sciences / The Royal Society*, 269(1503):1895–904, September 2002.

[4] Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science (New York, N.Y.)*, 315(5817):1416–20, March 2007.

[5] M. Kawato. Feedback-error-learning neural network for supervised motor learning. *Advanced neural computers*, 6(3):365–372, 1990.

[6] a Lenz, T Balakrishnan, a G Pipe, and C Melhuish. An adaptive gaze stabilization controller inspired by the vestibulo-ocular reflex. *Bioinspiration & Biomimetics*, 3(3):035001, September 2008.

[7] BD Lucas and T Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, number x, pages 674–679, 1981.

[8] O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.

[9] F Panerai, G Metta, and G Sandini. Learning visual stabilization reflexes in robots with moving eyes. *Neurocomputing*, 48(1-4):323–337, October 2002.

[10] L Righetti, J Buchli, and a Ijspeert. Dynamic Hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena*, 216(2):269–281, April 2006.

[11] Ludovic Righetti and A.J. Ijspeert. Programmable central pattern generators: an application to biped locomotion control. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, number May, pages 1585–1590. IEEE, 2006.

[12] Cristina P Santos, Miguel Oliveira, Ana Maria a. C. Rocha, and Lino Costa. Head motion stabilization during quadruped robot locomotion: Combining dynamical systems and a genetic algorithm. In *2009 IEEE International Conference on Robotics and Automation*, pages 2294–2299. Ieee, May 2009.

[13] Jianbo Shi and Carlo Tomasi. Good Features to Track. In *Conference on Computer Vision and Pattern Recognition*, number June, 1994.

[14] T Shibata and S Schaal. Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks. *Science And Technology*, 14:201–216, 2001.