HAL
open science

Online generation of cyclic leg trajectories synchronized with sensor measurement

Rodolphe Héliot, Bernard Espiau

To cite this version:

Rodolphe Héliot, Bernard Espiau. Online generation of cyclic leg trajectories synchronized with sensor measurement. Robotics and Autonomous Systems, 2008, 56 (5), pp.410-421. 10.1016/j.robot.2007.09.019 . hal-00346644

HAL Id: hal-00346644

https://hal.science/hal-00346644v1

Submitted on 12 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online Generation of Cyclic Leg Trajectories Synchronized with Sensor Measurement

R. Héliot [a,b] B. Espiau [a]

[a] *INRIA Rhône Alpes, 655 av. de l'europe, 38334 Saint Ismier - France.*
Bernard.Espiau@inrialpes.fr

[b] *CEA-LETI, Minatec, 17 rue des Martyrs, 38054 Grenoble - France.*
Rodolphe.Heliot@inrialpes.fr

Abstract

The generation of trajectories for a biped robot is a problem which has been largely studied for several years, and many satisfying off-line solutions exist for steady-state walking in absence of disturbances. The question is a little more complex when the generation of the desired trajectories of joints or links has to be achieved or adapted online, i.e. in real time, for example when it is wished to strongly synchronize these trajectories with an external motion. This is precisely the problem addressed in this paper. Indeed, we consider the case where the "master" motion is measured by an position sensor embedded on a human leg. We propose a method to synchronize the motion of a robot or on other device with respect to the output signal of the sensor. The main goal is to estimate as accurately as possible the current phase along the gait cycle. We use for that purpose a model based on a nonlinear oscillator, to which we associate an observer. Introducing the sensor output in the observer allows us to compute the oscillator phase and to generate a synchronized multilinks trajectory, at a very low computational cost. The paper also presents evaluation results in terms of robustness against parameter estimation errors and velocity changes in the input.

Key words:
Online trajectory generation - Oscillator - Synchronization - Observer - Sensor

1 Introduction

The problem of trajectory generation is classical in robotics. Although there now exists efficient industrial solutions to generate complex trajectories for manipulator robots, allowing for example obstacle avoidance, or energy consumption or cycle time optimization, the problem is still a research one for

some classes of mobile robots. In fact, the realisability of precomputed trajectories may become questionable when it is needed to cope with control requirements or to adapt in real time to unexpected changes. Non-holonomy of wheeled robots, and dynamical walking stability of legged robots are two examples of such difficulties.

In the case of bipedal systems, addressed in this paper, trajectory generation may be splitted in two parts: 1- path planning of the whole system: this problem is close to the one of mobile robots, except when an accurate determination of footprints is required; 2- computation of leg trajectories preserving some invariants in the posture and ensuring a certain form of stability. Focusing on this second issue, it can be seen that several classes of methods can be found in the literature (an overview of bipedal trajectory generation techniques is presented in [1]), among them: computation of optimal trajectories in various spaces, derivation of intrinsic trajectories in passive walking, or use of model predictive control. More precisely, when periodic motions are considered, a frequently used approach consists in mimicking living being's CPGs (Central Pattern Generator), under the form, either of ANNs, or of nonlinear oscillators. Originally designed for very stable systems, like 6- or more legged machines, snake or fish robots, etc... (see for example [2]), this approach has more rarely been used for biped robots [3]. However, the problem of using this class of method, in conjunction with sensory inputs delivered in real-time, has rarely been addressed. Some studies [4,5] report the use of sensory inputs in dynamic neural networks; however, the use of such neural networks does not allow the system designer to **ensure** behavioural properties, such as synchronization of the network with respect to the input. This is precisely the goal of this paper. Indeed, we consider the case where we want to generate cyclic joint trajectories which **have to be synchronous** with the output of a sensor mounted on another link. This is a kind of teleoperation, which can be, for example, very useful in the case of controlling a leg prosthesis or remotely operating a walking robot.

In this paper we will focus on the case of bipedal systems and, therefore, consider the following situation: a sensor, installed on the thigh of a human, provides at each time with an information related to the absolute angular position of the link; the question which arises is then: how to use this single information to generate the full trajectories of another system (a robot or the other leg, for example), in a perfectly synchronous way? We will see later that the problem is addressed by designing a specific conjunction of a nonlinear oscillator and of the related observer. The paper is organized as follows: after having recalled some basic facts on nonlinear oscillators required in the following, we will discuss the choice of the oscillator, and derive the associated nonlinear observer. Then, we will analyze the properties of the designed sys-

tem, and give some simulation and experimental results. Finally, we will draw some plans for the near future.

2 Framework

2.1 Oscillators: some definitions

In this section, we recall some basic facts concerning periodic solutions of ordinary differential equations (ODE). All this section is largely inspired from the book by Pikowsky et al. [6]: "*Synchronization, a universal concept in nonlinear sciences*", part 7.1: phase dynamics. The concepts of phase and isochrones, that are defined there, and will be used in the following.

Let's consider a system of autonomous ordinary differential equations:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}), \qquad \mathbf{x} \in \Re^n \tag{1}$$

and suppose that this system has a stable periodic (with a period T_0) solution $\mathbf{x}_0(t) = \mathbf{x}_0(t + T_0)$. In the phase space (space in which all possible states are represented) this solution is an isolated closed attractive trajectory, called the limit cycle of Eq. (1) (Fig. 1). A classical example of a self-oscillating system is the van der Pol equation:

$$\ddot{x} - \mu(1 - x^2)\dot{x} + \omega_0^2 x = 0 \tag{2}$$

with $\mu > 0$ and $\omega_0 > 0$.

Let's introduce the phase φ as a coordinate along the limit cycle, as a variable which grows uniformly in the direction of the motion and gains 2π during each rotation, thus obeying the equation:

$$\frac{d\varphi}{dt} = \omega_0 \tag{3}$$

where $\omega_0 = 2\pi/T_0$ is the frequency of the self-sustained oscillations.

From Eq. 3 follows an important property of the phase: it is a neutrally stable variable, in the sense that a perturbation in the phase remains constant: it never grows or decays in time.

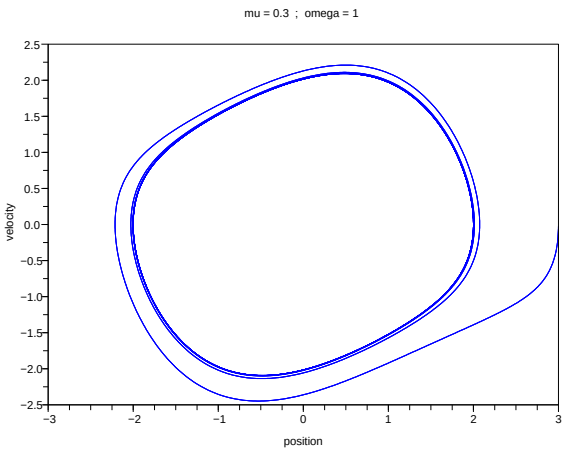

Fig. 1. Limit cycle of the van der Pol oscillator. Here $\mu = 0.3$ and $\omega_0 = 1$.

Consider now the effect of a small external periodic input on the self-sustained oscillations, described by:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}) + \epsilon \mathbf{p}(\mathbf{x}, t) \tag{4}$$

where the input $\epsilon \mathbf{p}(\mathbf{x}, t) = \epsilon \mathbf{p}(\mathbf{x}, t + T)$ has a period T, which is in general different from T_0. The input is proportional to a small parameter ϵ, and below we consider only first-order effects in ϵ. The external perturbation drives the trajectory away from the limit cycle, but because it is small and the cycle is stable, the trajectory only slightly deviates from the original one $\mathbf{x}_0(t)$. Thus perturbations in the directions transverse to the limit cycle are small; contrary to this, the phase perturbations can be large.

A need is then to define the phase variable in such a way that it rotates uniformly according to Eq. 3 not only on the cycle, but in its neighbourhood as well. To this end, we define the so-called *isochrones* in the vicinity of the limit cycle [7]. Observing the dynamical system stroboscopically, with the time interval being exactly the period of the limit cycle T_0, we get a mapping:

$$\mathbf{x}(t) \rightarrow \mathbf{x}(t + T_0) \equiv \Phi(\mathbf{x})$$

This construction is illustrated by the figure 2. Let us choose a point $\mathbf{P}^*$ on the cycle and consider all the points in the vicinity that are attracted to $\mathbf{P}^*$ under the action of Φ. They form a $(M-1)$-dimensional hypersurface I (where M is the dynamical system dimension) called an isochrone, crossing the limit cycle at $\mathbf{P}^*$. An isochrone can be drawn at each point of the limit cycle, thus we can parametrize the hypersurface according to the phase as $I(\phi)$ (see Fig. 2). We now extend the definition of the phase to the vicinity of the limit cycle,

demanding that the phase be constant on each isochrone. In this way, phase can be defined in the neighbourhood of the limit cycle.

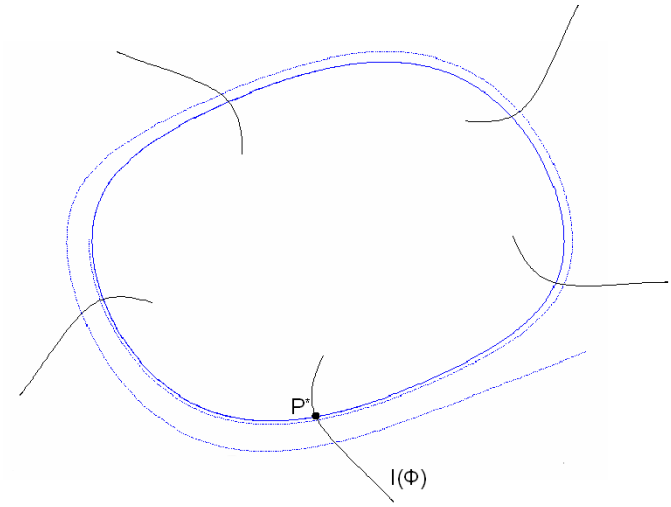

Fig. 2. Isochrones in the vicinity of the limit cycle

2.2 CPG for trajectory generation

Recently, the concept of Central Pattern Generator (CPG) has been used in robotics for online trajectory generation [8–10,3]. The CPG concepts comes from biology [11,12]: it is a small neural network, located at the spinal level, able to generate rhythmic commands for the muscles. It can be divided into two parts: a rhythm generator, and a patterning mechanism [13]. CPGs receive inputs from higher parts of the central nervous system, and also from peripheral afferents; thus, its functioning results from an interaction between central commands and local reflexes (see fig 3).

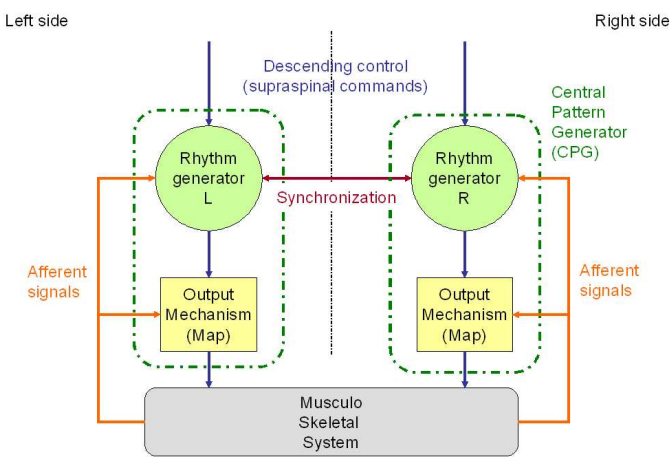

Fig. 3. CPG architecture for movement control in vertebrates

The CPG can be modelled by either a simulated neural network (for example using the Fitzhugh-Nagumo model [14]) or a non-linear oscillator. In both cases, the idea is to encode the desired trajectory in a stable limit cycle.

In robotics, a CPG-based command structure has several advantages for the design of cyclic trajectories: when using a feedback control, the system is robust with respect to small perturbations, thanks to the intrinsic stability of the limit cycle; one can easily modulate the amplitude or the period of the trajectory; it is well suited for feedback integration [15]. Finally, a multidimensional output can be generated for the same low computational price, which is helpful when dealing with robots with numerous Degrees Of Freedom (DOFs), having to exhibit multiple synchronized periodic motions, like walking machines. The control scheme presented in Fig. 4 gives an example of a multi-DOFs command structure.

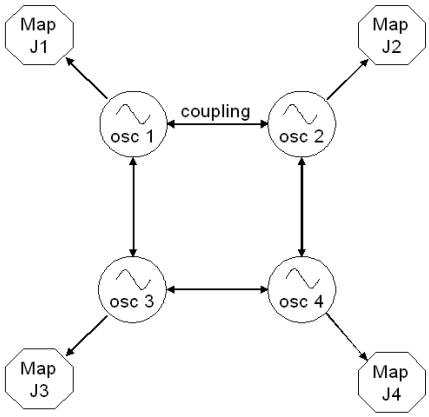

Fig. 4. This schematics gives an example of a joints command structure. Four coupled oscillators (osc 1 .. osc 4) produce rhythm for the output mechanisms (maps) of the four joints J1 .. J4

In this paper, we will focus on the sensory integration problem in the CPG concept: "How can we build a rhythm generator (oscillator) such that we can be *sure* that it will synchronize with a given input ?". The issue here is to provide the designer with tools that could ensure the synchronization property of the system with respect to the input, and with some proofs of robustness of the system. This can be of great importance when the oscillator is embedded in a control scheme.

2.3 Synchronization and observation

The synchronization problem has been recognized of great interest recently [16,6] and a link has been made between synchronization and observation theories, mostly by Nijmeijer and his group [17,18]. The observer theory, coming

from the control theory, has been introduced in the early seventies by Luenberger [19] in the linear case; in the non-linear case, some partial results exist [20].
The idea of observation is to estimate the state variables of a system, only given the inputs and the outputs of the system. Let's consider the system

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(u) \\ y = h(x) \end{cases} \tag{5}$$

and build a copy of Σ with *output injection* (Fig. 5):

$$\Sigma' : \begin{cases} \dot{\hat{x}} = f(\hat{x}) + g(u) + K(\hat{y} - y) \\ \hat{y} = h(\hat{x}) \end{cases} \tag{6}$$

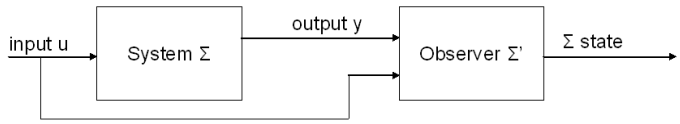

Fig. 5. Observer principle

In the linear case, if the original system Σ is observable (see [21] for a complete description of the observability conditions), and if gain K is correctly set, then the observer state will converge toward the original system state. When the output error $(\hat{y} - y)$ is canceled, the observer state exactly matches Σ's state: the observer is synchronized with the observed system.

In the non-linear case, there is no general result concerning the observer existence. However, it is sometimes possible to build a non-linear observer, when the error dynamics is feedback linearizable. To achieve this, the system has to belong to the *Lur'e* class [22], in which the non-linearity f is a function of the output only:

$$\Sigma : \begin{cases} \dot{x} = Ax + f(y,t) + Bu \\ y = Cx \end{cases} \quad x \in \Re^n, u \in \Re^m, y \in \Re^p \tag{7}$$

The observer is then given by:

$$\Sigma' : \begin{cases} \dot{\hat{x}} = A\hat{x} + f(y,t) + Bu + K\left(\hat{y} - y\right) \\ \hat{y} = C\hat{x} \end{cases} \tag{8}$$

and the error dynamics can be linearized:

$$\begin{aligned} \text{with} \quad e &= \hat{x} - x, \\ \dot{e} &= \dot{\hat{x}} - \dot{x} \\ &= A\hat{x} + f(y,t) + Bu + K(\hat{y} - y) - Ax - f(y,t) - Bu \\ &= (A + KC)e \end{aligned} \tag{9}$$

2.4 Synthesis : our approach

Our aim is to build an oscillator which will synchronize with a given cyclic sensory input. If we want to use the observer theory, we need to have a model of the system. However, such a model is not often available. This leads us to propose the following method, in two steps (Fig. 6):

(1) build a system as a phenomenological model, that simulates the sensor measurements
(2) build an observer of this system, in which are injected the actual sensor measurements

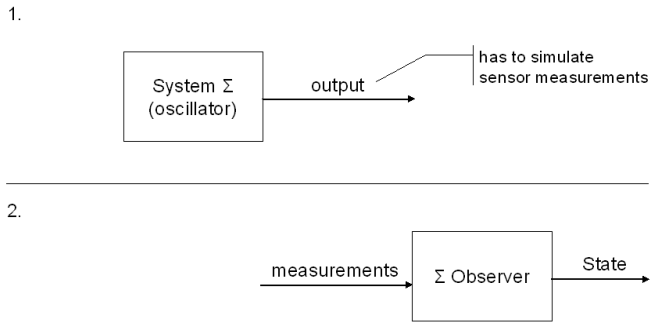

Fig. 6. Schematics of the method

To simulate a cyclic sensor measurement, one could choose a linear system as a model, and provide it with a cyclic input u (for example, a sinusoidal input); in that case the linear system is shaping the input so as its output simulates the given measurement. A problem arises then: there is a need for providing with a cyclic input which has to be synchronous with the measurement. This

problem has been explored in [23], where a network of adaptive coupled Hopf oscillators can dynamically adapt to any periodic signal. Unfortunately, since the proposed solution uses an external forcing, it adapts to frequency changes too slowly for our application.

To solve this problem, we chose a non-linear oscillator as a phenomenological model for our sensor measurement: it can generate a **cyclic** signal **without input**. Consequently, there is no need for an adapted input. This approach, using an observer of a non-linear oscillator, presents as major advantage the fact that the behaviour of the observer can be **proved** to converge toward the given cyclic input: the synchronization is ensured. The convergence speed can be set through the gain K in Eq. 8.

3 Methods

In the previous sections, we have presented the general background of our approach. We will now focus on a particular class of applications, in the framework of bipedal locomotion. The addressed question is the following:let us install a motion sensor on the leg of a human, and consider another walking device, the motion of which has to be hardly synchronized with the one of the equipped leg during steady-state walking ; this can be for example an anthropomorphic robot to be teleoperated or the other leg of an hemiplegic subject assisted by FES ; the problem is now to generate in real time from the sensor output a signal allowing to synchronize the trajectories of the "slave" links with the one of the "master" human leg. For that purpose, we will instantiate the approach previously described by selecting an adapted oscillator and synthetizing the related observer.

3.1 Which oscillator and why ?

3.1.1 What could be the right form of an oscillator?

To some extent, and under the assumption of rigidity of the links, a bipedal walking system can be modelled as a tree-structured n-link mechanical system free in space. Its dynamics can therefore be described through a Lagrange equation:

$$M(q)\ddot{q} + N(q,\dot{q}) + G(q) = -B(\dot{q}) + \Gamma + \lambda^T C(q) \tag{10}$$

where q is the set of joint coordinates, $q \in \Re^n \otimes SE(3)$ [1], M is a symmetric positive definite mass matrix, N gathers coriolis and centrifugal forces, B is the friction term, G the gravity vector, Γ the actuation input, and $\lambda^T C(q)$ are the constraints of ground contacts, which are unilateral and time-varying. $C(q)$ is the jacobian matrix of the coordinates of the contact points and λ is the array of related Lagrange multipliers. The underlying contact model is rigid and based on a LCP (Linear Complementary Problem) approach. In the absence of constraints, friction and control, the right-hand side of the equation is equal to zero, and the system is conservative with mechanical energy as first integral, the continual exchange between kinetics energy and potential energy producing a periodic motion.

Let us now consider a single coordinate q_a, which can be for example the thigh angle. Starting from the unforced version of eq. (10), we can express its dynamics as:

$$H(q)\ddot{q}_a + F(q, \dot{q}) + T(q) = 0 \tag{11}$$

By considering all other variables than q_a as external inputs or disturbances, we can, roughly, see this equation as the one of a nonlinear second order system with perturbations and a kind of potential function. Furthermore, the behaviour of q_a is periodic. This incitates to research the oscillator preferably within the class of modified and disturbed second-order nonlinear mass-spring systems.

3.1.2 Is the concept of limit cycle licit for human walking?

As seen previously, the natural behaviour of a mechanical robotics system without dissipative terms and other inputs is an oscillator with constant energy. Nevertheless, this does not correspond with the idea of an attractive limit cycle which underlies the oscillator-based approach. To justify this point of view, we have to refer to another class of mechanical systems: the passive walking machines. Indeed, let us consider the case of a planar compass, walking above a slope, with instantaneous and inelastic step transition, as addressed in [24] and several others [25,26]. It then can be shown that, for a given slope, such a system exhibits a limit walking cycle (see Fig. 7), with a rather large basin of attraction. This behaviour can be compared to the concept of "natural gait" or "comfort gait" which is spontaneously reached and followed by a human in steady state walking, and which corresponds to a minimum of the

[1] this means that the configuration space of the system is the product of the space spanned by the joints and of the 6-dimensional position/orientation of a given link in the space

metabolic energy consumption with respect to distance.

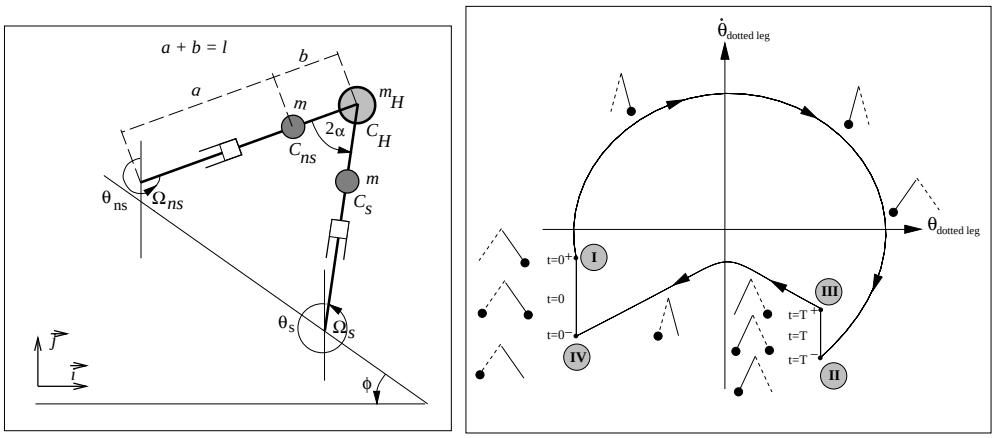

Fig. 7. Mechanical model of a passive walking machine, and its stable limit cycle. Numbers II and III corresponds to the instantaneous transition ad the end of the swing and IV to I ad the end of the stance. Figure derived from [25].

In conclusion, it appears that searching for a oscillator of second-order type which exhibits a limit cycle is a reasonable way of modelling the periodic behaviour of a link of the human leg when walking steadily.

3.1.3 Structure choice and parameters setting

From the previous sections, we know that we have to choose a nonlinear oscillator which belongs to the Lur'e class (so that it can be observable, see section 2.3), and which is derived from a second-order mass-spring system. Two common oscillators fit these requirements: the van der Pol oscillator and the Rayleigh oscillator, which are very similar.
Let us select the van der Pol equation to represent the sensor signal obtained when walking. However, a basic van der Pol oscillator is unable to fit immediately any kind of periodic motion, and we need to modify it according to the considered application. In the present case, bipedal walking, a natural idea is to use a representation of the leg "position": for example, we could use a measurement of a joint coordinate, such as the knee angle. Another possibility is to consider the absolute orientation of one link; indeed, because of the used sensor technology (see section 4.1), the thigh inclination with regard to the vertical is the easiest variable to access. During human gait, this inclination presents an asymmetrical shape, with a ascending phase shorter than the descending one. The van der Pol equation provides with symmetrical signals; some examples already exist in the literature [27,28] where the van der Pol equation is modified in order to get asymmetrical signals. In the same spirit, we may therefore insert a new term in the equation:

$$\ddot{x} - \mu(1\text{- } \mathbf{b}\ \mathbf{x} - x^2)\dot{x} + \omega_0^2 x = 0 \tag{12}$$

with $b > 0$.

The idea is to modify the damping coefficient $\mu(1 - bx - x^2)$ so as it is different when $x < 0$ or $x > 0$. In that way, the output of the modified van der Pol oscillator won't be symmetrical anymore: for a given $|x|$ when $x < 0$, $|\mu(1 - bx - x^2)|$ is higher than when $x > 0$.

Once the structure of the nonlinear oscillator is chosen, we have to find the best parameters μ, b, and ω_0 so that the trajectory of the limit cycle of this oscillator will fit the sensor measurement. This measurement comes from a real walking experiment, where a inclination sensor is placed on the thigh of the subject. We write this identification problem as a least squares one: minimizing the error between the measurements and the output of the oscillator:

$$\begin{cases} \min\limits_{\mu, b, \omega_0, x_s^i} \quad \sum\limits_{i=1}^{m} (x_s^i - x_m^i)^2 \\ \\ \ddot{x}_s^i - \mu(1 - bx_s^i - x_s^{i^2})\dot{x}_s^i + \omega_0^2 x_s^i = 0 \end{cases} \tag{13}$$

where x_m^i are the discretized sensor measurements (for example, over one given cycle), and x_s^i are the simulated oscillator outputs, thus following the dynamics of Eq. (12).

One can notice that this problem is similar to an optimal control problem, that can be solved using a direct method [29]. We include the discrete output of the oscillator in the parameters to optimize, and we add constraints on them following the dynamical model of the oscillator. The discretization of this problem leads to a "nonlinear programming" problem which has been solved using a constrained sequential quadratic programming solver (Feasible Sequential Quadratic Programming: FSQP [30]). Practically, this method gave good results: we obtained a very good match between the measurements and the oscillator output (see fig. 8).

3.2 Observer design

In this section we apply the general observer theory presented in section 2.3 to a specific case: we want now to build an observer of the dynamical system described by the modified van der Pol equation (12), which can be written as:

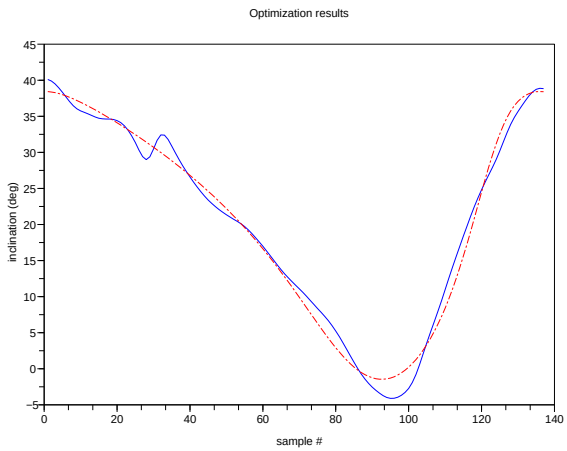

Fig. 8. Comparison of the sensor measurement cycle (solid line) with the optimized oscillator output (dotted line)

$$\Sigma : \begin{cases} \dot{x_1} = x_2 \\ \dot{x_2} = \mu(1 - bx_1 - x_1^2)x_2 - \omega_0^2 x_1 \\ y = x_1 \end{cases} \tag{14}$$

We want to have access to the internal variables x_1 and x_2 of the oscillator described by Eq. 14. To achieve this, we can only use its output y, that will become the observer input. One can notice in Eq. 14 that its output y directly provides us with the internal state variable x_1. Thus, we can build a *reduced* observer, which means that the dimension of the observer is lower than the dimension of the original system to be observed. In our case, the dimension of the original system is 2, and the dimension of the observer will be 1. The reduced observer will in some sense tend to track a transformation of the state variable to be estimated, namely x_2. In the same spirit as [17] does, we introduce the variable z:

$$z \; = \; x_2 \; + \; k_1 y \; + \; k_2 y^2 \; + \; k_3 y^3 \tag{15}$$

which satisfies the following equation:

$$
\begin{aligned}
\dot{z} &= \dot{x_2} + k_1\dot{y} + 2k_2\dot{y}y + 3k_3\dot{y}y^2 \\
\dot{z} &= \dot{x_2} + k_1\dot{x_1} + 2k_2\dot{x_1}x_1 + 3k_3\dot{x_1}x_1^2 \\
\dot{z} &= \dot{x_2} + k_1x_2 + 2k_2x_2x_1 + 3k_3x_2x_1^2 \\
\dot{z} &= \mu(1 - bx_1 - x_1^2)x_2 - \omega_0^2x_1 + k_1x_2 + 2k_2x_2x_1 + 3k_3x_2x_1^2 \\
\dot{z} &= (\mu + k_1)x_2 + (2k_2 - \mu b)x_1x_2 + (3k_3 - \mu)x_1^2x_2 - \omega_0^2\ x_1
\end{aligned}
\tag{16}
$$

We want to cancel the non-linear terms (this is also called feedback linearization); so, setting:

$$
\begin{aligned}
\mu + k_1 = -1 &\ \rightarrow\ k_1 = -\mu - 1 \\
2k_2 - \mu b = 0 &\ \rightarrow\ k_2 = \mu b/2 \\
3k_3 - \mu = 0 &\ \rightarrow\ k_3 = \mu/3
\end{aligned}
\tag{17}
$$

we get:

$$
\begin{aligned}
\dot{z} &= -x_2 - \omega_0^2x_1 \\
\dot{z} &= -(z - k_1y - k_2y^2 - k_3y^3) - \omega_0^2x_1 \\
\dot{z} &= -z + (k_1 - \omega_0^2)y + k_2y^2 + k_3y^3
\end{aligned}
\tag{18}
$$

So, z dynamics is fully described by its single eigenvalue, that has to be negative to ensure the convergence of the observer. Here this eigenvalue equals to -1, once k_1 is set to $-\mu - 1$. Finally:

$$\Sigma' : \begin{cases} \dot{z} &= -z + (k_1 - \omega_0^2)y + k_2 y^2 + k_3 y^3 \\ \hat{x}_1 &= y \\ \hat{x}_2 &= z - k_1 y - k_2 y^2 - k_3 y^3 \end{cases} \tag{19}$$

3.3 Trajectory generation

By injecting a measurement input y in the observer (19), we get an estimation of the two state variables $\hat{x}_1$ and $\hat{x}_2$. Since this observer is also an (forced) oscillator, we can compute its phase. This can be easily done using the isochrones defined in section 2.1. Isochrones can be computed, using the free nonlinear oscillator equation (12), in two different ways. The first idea, analytical, is to write the oscillator equation in polar coordinates (R, θ), define the phase φ such as it grows uniformly, and compute the lines of constant phase on the (R, θ) plane. Unfortunately, it is not possible to derive analytically the equation of isochrones exepted in very specific cases. The second idea is to obtain them by simulation; first, let's assess the free oscillator period T_0. Then, for each point $\mathbf{x}_i$ on the phase plane which is in the vicinity of the limit cycle, simulate its trajectory under oscillator dynamics during a time nT_0, with n being an integer large enough such that the distance from the point $\mathbf{x}_i(nT_0)$ to the limit cycle is small. In that case, the original point $\mathbf{x}_i$ has the same phase as $\mathbf{x}_i(nT_0)$, which is known, since it is on the limit cycle.

Finally, let's say that the (cyclic) trajectory we want to generate is parametrized by its phase: we thus have a trajectory pattern $T(\varphi)$, for $\varphi \in [0, 2\pi]$.

In summary, the online computation scheme for trajectory generation is the following:

(1) inject the sensor measurement y in the adapted observer
(2) from the observer state variables x_i, compute the phase φ of the oscillator
(3) provide with the command trajectory: $C = T(\varphi)$
(4) follow the computed trajectory with a dynamic controller (ex: PID or computed torque)

and is embedded in the global scheme presented in Fig. 9:

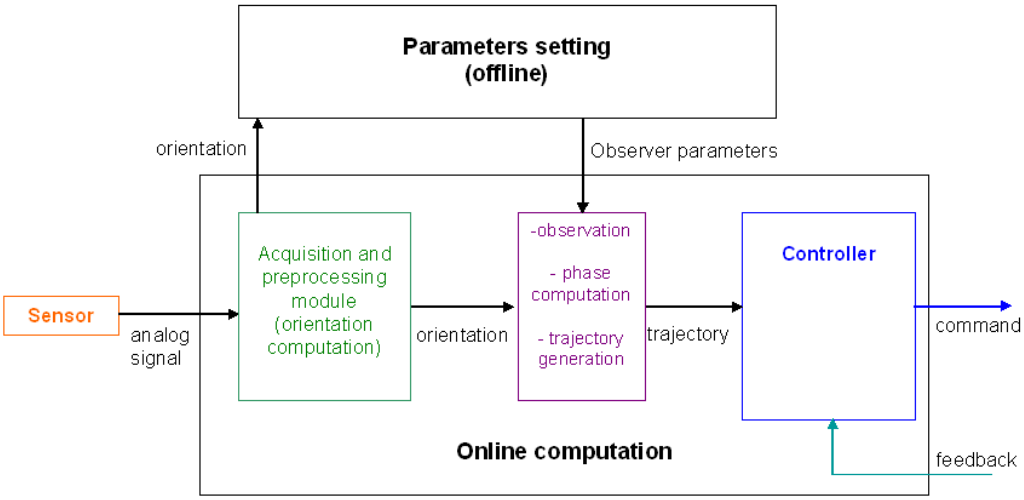

Fig. 9. Overall schematics of our method. The trajectory is computed online in a synchronous way according to the sensor signal used as input (left). This trajectory is then followed by the dynamic controller, that uses feedback from the controlled segments and joints.

4 Implementation and Results

4.1 Angle estimation

We estimate the thigh inclination with a micro-sensor, developed by CEA-LETI (Grenoble, France), which associates 3 accelerometers and 3 magnetometers in a minimal volume (see Fig. 10). This attitude sensor is able, through the associated processing algorithms, to reconstruct the orientation in space of the segment to which it is attached (the reader may refer to [31] for further information concerning this sensor). Figure 11 shows an example of thigh inclination estimation during human gait, which will be used as input in the following.

4.2 Results

To validate our method, we recorded sensor signals during human gait, in order to generate a synchronous command. This was achieved in a first time offline (in simulation but using real measurements, in order to have an insight on generated trajectory). Figure 12 shows three consecutive measurement cycles, together with the reference limit cycle: although the estimated state variables of the oscillator do not always stay on the limit cycle, they remain close to it.

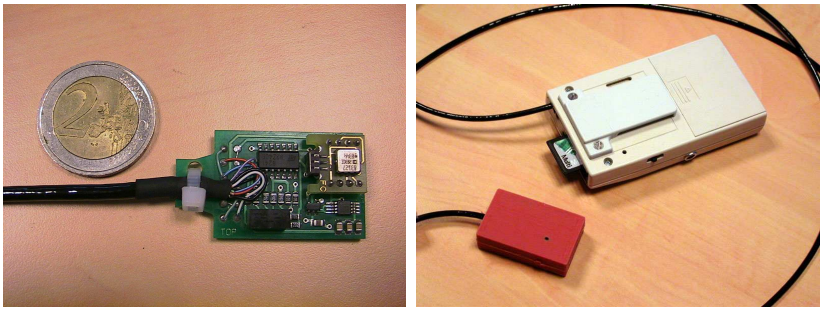

Fig. 10. CEA-LETI attitude sensor. Left: Sensor size, compared to a coin. Board is then embedded in a silicon-like material. Right: Final view of the sensor, with its datalogger.

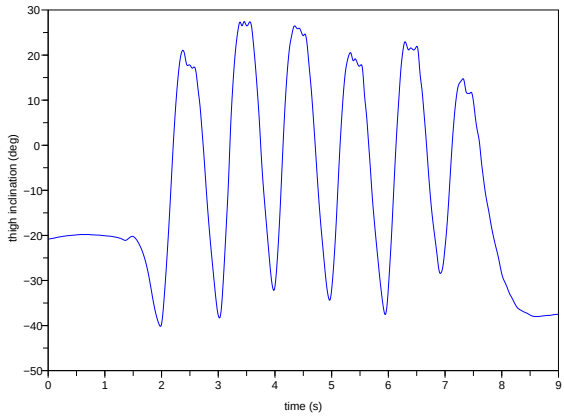

Fig. 11. Thigh inclination (regarding to the vertical) estimation during human gait, using the attitude sensor. Subject is walking at approximately 1.2 m/s.

Some of the discrepancy might result from the impact when the leg hits the ground.

Each couple $(\hat{x}_1; \hat{x}_2)$ of estimated state variables can be plotted in the phase space. We know from section 2.1 (see also Fig. 2) that, using the isochrones, a phase variable can be defined for each point in the phase space. Thus, combining the observer-based estimation of the state variables with the isochrones-based estimation of phase, we get a good phase estimation (i.e. monotonous, and quasi-piecewise linear), synchronous to the input signal. Finally, we can generate a command parametrized by this phase variable. Without loss of generality, this command can be the desired joint coordinates of a poly-articulated system. Such a system has several Degrees Of Freedom (DOFs) which have to be well coordinated, in particular in the case of walking systems. A phase parametrization of the desired trajectory on each joint ensures the coordination of the overall movement: if all joints are parameterized with the same phase, there are perfectly coordinated. An example of generated trajectory for one DOF (knee joint angle, in sagittal plane) is shown in Fig. 13. In this

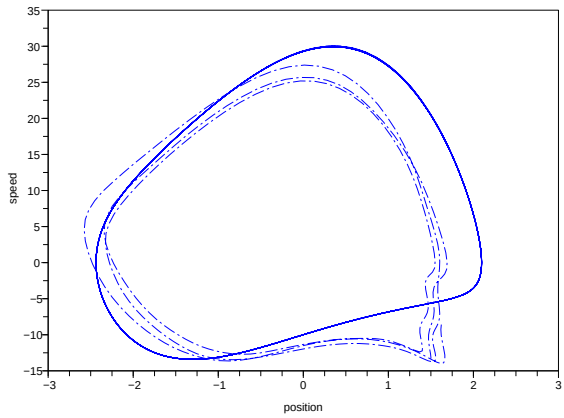

Fig. 12. Estimated state variables of the observer during three consecutive gait cycles (dashed line), compared to the reference limit cycle (solid line)

example we use a reference pattern for the knee trajectory, derived from [32]. This pattern gives the knee joint angle corresponding to each phase value (from 0 % to 100 % along the gait cycle). Thus, knowing the phase variable, we have immediate access to the knee joint angular position.

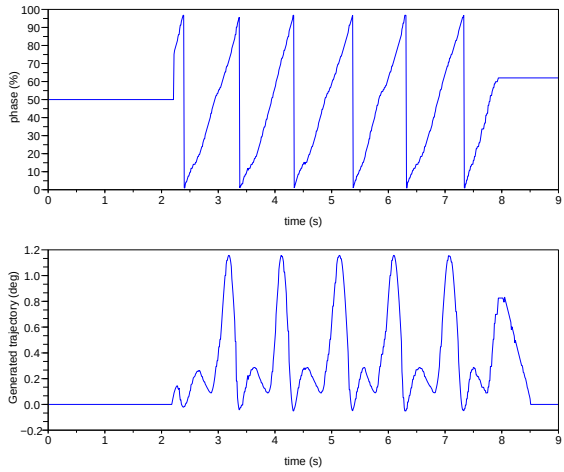

Fig. 13. Phase (top) and trajectory generated for the knee joint (bottom).

4.3 Experiment: Robot teleoperation

By installing the sensor on a human leg, we observe the thigh angle and compute online a biped robot control, such that the robot "follows" the human gait, in a synchronous way. This is done by first generating a desired trajectory for each active DOF of the robot (in our case: ankle, knee and hip sagittal angles on both legs), and then following this trajectory with a PID controller. Such experiments were conducted on the BIP robot (Fig. 14, [33]), the robot

being suspended to the ceiling. The sampling frequency was 100 Hz, and we were able to check that the online computation of the trajectory took less than 1 ms. We thus fully validated the online trajectory generation based on sensor measurement. Although the trajectory generation can adapt to a quite wide range of gaits (see later section 5.3), the robot itself has a limited speed range (joints maximal velocities and accelerations do not allow it to follow fast gaits). For this reason, the online experiments were conducted at a rather low speed (about 0.9 m/s).

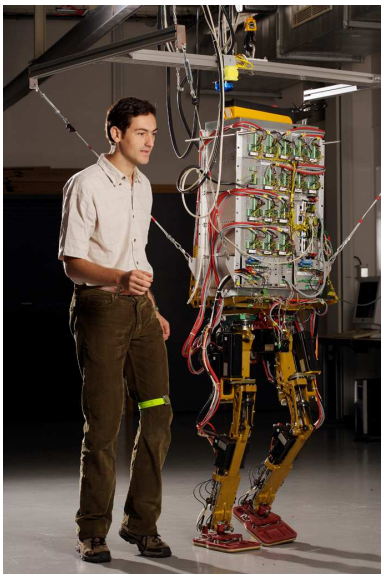

Fig. 14. Teleoperation experiment with the BIP robot

5 Evaluation of the method

In this paper, we consider the use of a nonlinear oscillator in the framework of interaction with human, through actual sensor inputs. It is therefore necessary to assess the practical efficiency of the method. We will consider in the following three important issues:

- the theoretical behavior of the modified oscillator in terms of periodic solutions
- the robustness of the approach with respect to errors in parameters
- the ability of the method to track changes in the input dynamics, which allows in particular to cope with transient walking stages

5.1 Some comments about experimental results

Let us first note that in the case of a practical experiment, as teleoperating a robot with a sensor placed on a real person's thigh, it is impossible to have access to the "actual" intrinsic phase of the observed system (the human). For this reason, is it difficult to evaluate the performance and robustness of our approach with respect to real values that are unknown. This is why quantitative results were computed through simulation, while the presented experimental ones are qualitative, since they cannot be evaluated more accurately.
Also, what we want to assess here is the trajectory generation part, not the trajectory following part through the PID controller of the robot. For this reason, the following results are based on theory and simulation (using real measurements), but not on real-time experiments, that wouldn't be useful to assess the properties addressed here.

5.2 Robustness

5.2.1 Oscillator properties

Since we modified the van der Pol oscillator, it is important to study its properties. We start again from Eq. (12), which can be written:

$$\begin{aligned} \dot{x} &= y = f_1(x, y) \\ \dot{y} &= \mu(1 - bx - x^2)y - \omega_0^2 x = f_2(x, y) \end{aligned} \tag{20}$$

with μ, b, and ω all > 0.

A classical way to study a dynamical system is to analyze its behaviour around its fixed points, if they exist. In the present case:

$$\begin{aligned} \dot{x} = 0 &\rightarrow y = 0 \\ \dot{y} = 0 &\rightarrow x = 0 \end{aligned} \tag{21}$$

There is thus a single fixed point $(x, y) = (0, 0)$. Let's study is stability through its Jacobian:

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \\ \mu(-2x-b)y - \omega_0^2 & \mu(1-x^2-bx) \end{pmatrix} \tag{22}$$

which gives, in $(0,0)$:

$$J(0,0) = \begin{pmatrix} 0 & 1 \\ \\ -\omega_0^2 & \mu \end{pmatrix} \tag{23}$$

The Jacobian trace determines the characteristics of the fixed point. Here, as $Tr\,(J(0,0)) = \mu > 0$, $(0,0)$ is a repulsive point. Let's express the eigenvalues:

$$\begin{aligned} \lambda^2 - \mu\lambda + \omega_0^2 = 0 \\ \\ \Delta = \mu^2 - 4\,\omega_0^2 \end{aligned} \tag{24}$$

And so, using the Poincaré - Bendixson Theorem [7]:

- if $\mu < 2\omega_0$ then $\Delta < 0$, and $(0,0)$ is a repulsive source
- if $\mu > 2\omega_0$ then $\Delta > 0$, and $(0,0)$ is an unstable node

In our experiments, the optimized parameters (through the FSQP method, see section 3.1.3) were: $\mu = 2.03$, $b = 2.29$, $\omega = 5.34$, which corresponds to a standard adult non-pathological gait. We thus had $\mu = 2.03 < 2\,\omega_0 = 10.68$, far from the bifurcation.

5.2.2 Parameters sensitivity

We now focus on the consequences of an error in parameters estimation. This is an important issue, since in real experiments sensor measurements cannot

be exactly the same for a trial to another, even from a cycle to another. Practically, the estimated parameters are always slightly different from what they should be.

So let's consider the case where our estimation of parameter μ is different from its "true" value: $\hat{\mu} = \mu + \epsilon$. Then:

$$\begin{array}{lcl}
\hat{\mu} + k_1 = -1 & \rightarrow & k_1 = -\hat{\mu} - 1 \quad \rightarrow \quad \mu + k_1 = -1 - \epsilon \\
2k_2 - \hat{\mu}b = 0 & \rightarrow & k_2 = \hat{\mu}b/2 \quad \rightarrow \quad 2k_2 - \mu b = -\epsilon b \\
3k_3 - \hat{\mu} = 0 & \rightarrow & k_3 = \hat{\mu}/3 \quad \rightarrow \quad 3k_3 - \mu = -\epsilon
\end{array}$$

and:

$$\dot{z} = (-1-\epsilon)x_2 - \epsilon b x_1 x_2 - \epsilon x_1^2 x_2 - \omega_0^2 x_1$$

$$\begin{aligned}
\dot{z} &= (-1-\epsilon)(z - k_1 y - k_2 y^2 - k_3 y^3) \\
&\quad -\epsilon b y(z - k_1 y - k_2 y^2 - k_3 y^3) \\
&\quad -\epsilon y^2(z - k_1 y - k_2 y^2 - k_3 y^3) \\
&\quad -\omega_0^2 y
\end{aligned}$$

$$\begin{aligned}
\dot{z} &= [-1 - \epsilon(1 + by + y^2)] \cdot z \\
&\quad + [(1+\epsilon)k_1 - \omega_0^2] \cdot y \\
&\quad + [(1+\epsilon)k_2 + \epsilon b k_1] \cdot y^2 \\
&\quad + [(1+\epsilon)k_3 + \epsilon b k_2 + \epsilon k_1] \cdot y^3 \\
&\quad + [+\epsilon b k_3 + \epsilon k_2] \cdot y^4 \\
&\quad + \epsilon k_3 \cdot y^5
\end{aligned}$$

In the oscillator dynamics, the error can be thus expressed by:

$$
\begin{aligned}
e = \dot{z}_{\hat{\mu}} - \dot{z}_{\mu} = & -\epsilon(1 + by + y^2) \cdot z \\
& +\epsilon k_1 \cdot y \\
& +\epsilon \left[k_2 + bk_1\right] \cdot y^2 \\
& +\epsilon \left[k_3 + bk_2 + k_1\right] \cdot y^3 \\
& +\epsilon \left[bk_3 + k_2\right] \cdot y^4 \\
& +\epsilon k_3 \cdot y^5
\end{aligned}
$$

It thus appears that the error is linear with respect to the parameter estimation error ϵ. A similar demonstration can be held for each parameter.

We also tested in simulation the effects of a wrong parameter estimation. We compared the observer output $\hat{y}$ when parameters have error with respect to the nominal output. Figure 15 presents the error in $\hat{y}$ estimation with an estimation error on μ, b, or ω_0.

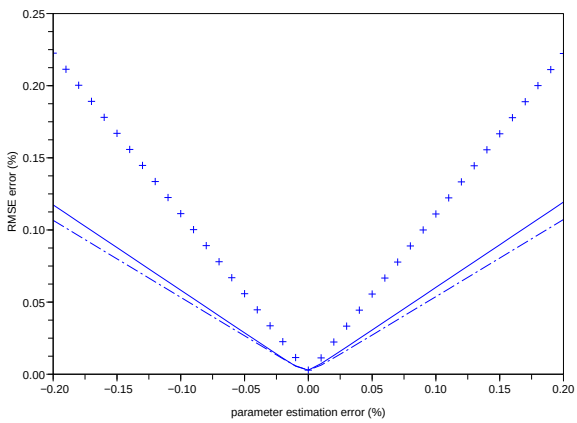

Fig. 15. The 3 lines show the output error (in % RMSE) with respect to the 3 parameters; solid line stands for b estimation error, dotted line for μ, crossed line for ω

Two remarks can be made: first, the output error is linear with respect to the parameter error, whatever the parameter is. Second, the minimum output estimation error is found for 0% parameter error; this result was of course expected, and means that the parameters have been well estimated.

5.3 Transitions

We simulated the effects of a quasi-step input frequency change. The results show very good adaptation: the generated output follows frequency change without distinguishable disturbance (see Fig. 16).

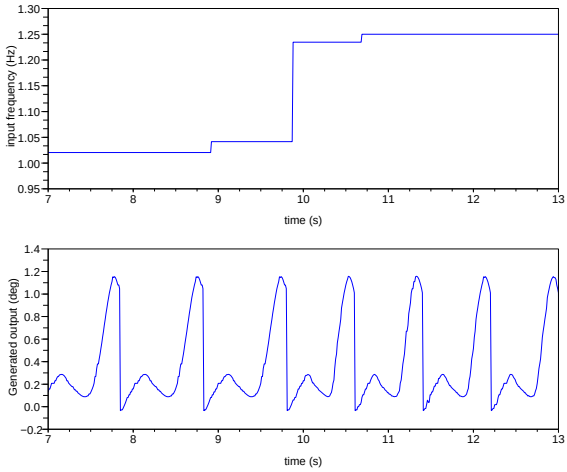

Fig. 16. Frequency change adaptation. Top: Input signal frequency; bottom: generated command

6 Future work

6.1 Aimed application

A first application we consider is hemiplegic stroke patients' gait rehabilitation. Hemiplegia induces a situation where a vertical half of patient's body sensory-motor pathways do not function properly. A direct consequence is that one side of patient is weak or paralyzed, thus severely impairing walking. Training of hemiplegia is likely to promote changes in neural circuits based on excitability and plasticity cortical properties. Some patients re-learn how to walk without assistance. Functional Electrical Stimulation (FES) allows for controlling artificially movements in patients with motor disability by applying electrical stimuli to impaired muscles through surface or implanted electrodes. FES has been shown to be a valuable method for training stroke patients in early phase of hemiplegia to improve recovery of walking skills. In this framework, the timing of muscle stimulation sequences is critical. Today, the existing FES systems usually provide with fixed stimulation patterns, over one gait cycle, parameterized and pre-programmed off-line [34]. The triggering of stimulation sequences is often achieved manually by the clinician assisting

the patient. Some other systems using embedded sensors [35] provide with several triggers corresponding to different gait phases (heel strike, toe off, ...); a specific stimulation sequence is then applied during each gait phase. The problem is that if the patient walks faster or slower than the programmed sequence, the movements on the healthy and the paretic leg might not be well coordinated. A critical issue is the command of this stimulation: there is a need to apply the proper stimulation amplitude at the **proper instant**. This problem can be seen as a teleoperation issue: observing the healthy leg, the goal is to apply an adapted command on the affected one. Applied to that case, our method could be used to adapt stimulation parameters (amplitude, pulse width, frequency) according to the computed phase. The command applied would thus be continuously adapted to the ongoing movement, in an online way.

6.2 Perspectives

One possible development of our method is an extension to the multidimensional case: how to integrate several inputs in our scheme? One idea is to build an oscillator with multiple outputs, each one of them simulating one measurement, and then synthesize a single observer with multidimensional input. Another possibility is to build as many adapted oscillators as sensor measurements, and then fusing the several phase estimations.
Also, it would be of great interest to study the behaviour of our system in the case of uneven terrain (slope, stairs, ...). If the input signal is too different from the original reference, it may loose its synchronization properties. In that case, different oscillator-observers could be prepared, each one corresponding to a given walking task; we would thus build a kind of "filter bank", switching from one to another if necessary.

7 Conclusion

The core of this paper can be seen as a contribution to the problem of periodic motion synchronization, with focus on steady-state walking. The considered application was a kind of teleoperation of an anthropomorphic robot by a human leg. More precisely, we have proposed a method for synchronizing in real time a robot trajectory generator with an external leg motion, through a single phase signal generated from an embedded attitude sensor. We assessed the robustness properties of this method, and demonstrated that it was suitable for human gait observation. In some sense, the approach can also be seen as an online time-scaling method. Since the focus of this work was on the synchronization, we considered neither the problem of dynamical control nor the one

of gait stability. The fact that the robot was suspended helped us to evaluate the quality of the synchronization in itself without being disturbed by other aspects. Nevertheless, stability is a major issue in biped walking and we can't avoid it if we want to explore possible extensions of this work. Let us recall that the fundamental problem in synthesizing and controlling a stable dynamical walk is the compatibility between the existence of preset trajectories and the online adaptation required to avoid falling in presence of disturbances. A classical way of solving the problem is to leave some "degrees of freedom" to the controller: for example, we can dedicate the control of some joints to stabilization exclusively, or, better, specify trajectories depending on a few set of parameters, which are computed as dynamically stable at the nominal value, and the parameters of which can be modified on line by the controller to preserve the stability. The time scale can be such a parameter. Of course, in the framework of teleoperation, it is expected that in case of hard disturbance, the stability will be a priority and the quality of master/slave synchronization will strongly decrease. In that case, a feedback from the slave to the master would be useful, leading to a kind of bilateral synchronization. More generally, the use of nonlinear oscillators as periodic trajectory generators for dynamically stable biped walking has not yet been studied. A way to address the problem might be to connect to the very promising works of Kajita [36] and Wieber [37], who use a predictive control approach to compute on line the parameters which maintains the Center of Pressure in a given area.

References

[1] C. Azevedo, B. Amblard, B. Espiau, and C. Assaiante, "Functional bases of bipedal locomotion: what do humans can tell to robots ?" *Technical Report 5450, INRIA*, 2004.

[2] A. Ijspeert, "A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander," *Biological Cybernetics*, vol. 84, no. 5, pp. 331–348, 2001.

[3] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng, "An empirical exploration of a neural oscillator for biped locomotion control," in *proc. Intl. Conf. on Robotics and Automation (ICRA2004)*, 2002.

[4] G. Taga, "A model of the neuro-musculo-skeletal system for human locomotion," *Biological Cybernetics*, vol. 73, no. 2, pp. 97–111, 1995.

[5] C. Paul, M. Bellotti, S. Jezernik, and A. Curt, "Development of a human neuro-musculo-skeletal model for investigation of spinal cord injury," *Biological Cybernetics*, vol. 93, no. 3, pp. 153–170, 2005.

[6] A. Pikovsky, R. Rosenblumn, and J. Kurths, *Synchronization, a universal concept in nonlinear sciences.* Cambridge University press, 2001.

[7] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields.* Springer-Verlag, 1990.

[8] M. M. Williamson, "Neural control of rhythmic arm movements," *Neural Networks*, vol. 11, no. 7-8, pp. 1379–1394, 1998.

[9] L. Righetti and I. A.J., "Programmable central pattern generators: an application to biped locomotion control," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006, conference.

[10] G. Taga, "Nonlinear dynamics of the human motor control," in *proc. Adaptive Motion in Animals and Machines (AMAM2000)*, 2000.

[11] S. Grillner, "Neurobiological bases of tythmic motor acts in vertebrates," *Science*, vol. 228, pp. 143–149, 1985.

[12] A. Cohen, S. Rossignol, and S. Grillner, *Neural control of rhyhtmic movements in vertebrates.* Wiley, 1988.

[13] S. D. Prentice, A. E. Patla, and D. A. Stacey, "Simple artificial neural network models can generate basic muscle activity patterns for human locomotion at different speeds," *Experimental Brain Research*, vol. 123, no. 4, pp. 474–480, 1998.

[14] K. Matsuoka, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biological Cybernetics*, vol. 52, no. 6, pp. 367–376, 1985.

[15] H. Dong, M. Zhao, J. Zhang, and N. Zhang, "Cpg-based adaptive dynamic control of a quadruped robot with sensory feedback," in *proc. Intl. Conf. on Climbing and Walking Robots (CLAWAR)*, 2006, pp. 77–81.

[16] T. L. Carroll and L. M. Pecora, "Synchronizing chaotic circuits," *IEEE Trans. Circuits and Syst.*, vol. 38, pp. 453–456, 1991.

[17] H. Nijmeijer and I. M. Y. Mareels, "An observer looks at synchronization," *IEEE Transactions on Circuits and Systems - Part I - Fundamental Theory and Applications*, vol. 44, no. 10, pp. 882–890, 1997.

[18] I. I. Blekhman, A. L. Fradkov, H. Nijmeijer, and A. Y. Pogromsky, "On self-synchronization and controlled synchronization," *Systems and Control Letters*, vol. 31, no. 5, pp. 299–306, 1997.

[19] D. Luenberger, "An introduction to observers," *IEEE trans. Automatic Control*, vol. 16, no. 6, pp. 596–602, 1971.

[20] A. Isidori, *Nonlinear Control Systems.* Springer, 1995.

[21] T. Kailath, *Linear Systems.* Prentice-Hall, 1980.

[22] A. I. Lur'e and V. N. Postnikov, "On the theory of stability of control systems," *Applied mathematics and mechanics*, vol. 8, 1944.

[23] L. Righetti, J. Buchli, and A. Ijspeert, "From dynamic hebbian learning for oscillators to adaptive central pattern generators," in *proc. Adaptive Motion in Animals and Machines (AMAM2005)*, 2005.

[24] T. McGeer, "Passive dynamic walking," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[25] A. Goswami, B. Thuilot, and B. Espiau, "A study of the passive gait of a compass-like biped robot: Symmetry and chaos," *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1282–1301, 1998.

[26] S. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.

[27] D. Linkens, "The method of harmonic balance applied to coupled asymmetrical van der pol oscillators for intestinal modelling," *Bulletin of Mathematical Biology*, vol. 41, pp. 573–589, 1978.

[28] K. Grudzinski and J. Zebrowski, "Modeling cardiac pacemakers with relaxation oscillators," *Physica A*, vol. 336, pp. 153–162, 2004.

[29] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, 1997.

[30] C. Lawrence, J. Zhou, and A. Tits, "User's guide for cfsqp version 2.5 : A c code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints," Electrical Engineering Department and Institute for Systems Research, University of Maryland, Tech. Rep., 1997.

[31] S. Bonnet and R. Heliot, "A magnetometer-based approach for studying human movements," *IEEE Trans. Biomedical Engineering*, vol. 54, pp. 1353–1355, 2007.

[32] D. Winter, *The biomechanics and motor control of human Gait.* University of waterloo press, 1987.

[33] C. Azevedo and the Bip team, "Control architecture and alghorithms of the anthropomorphic biped bip2000." in *Proc. of International Symposium on Mobile Climbing and Walking Robots (CLAWAR)*, 2000.

[34] J. Burridge, M. Haugland, B. Larsen, N. Svaneborg, H. Iversen, P. Brogger, R. Pickering, and T. Sinkjaer, "Long-term follow-up of patients using the actigait implanted drop-foot stimulator," in *Proc. 10th IFES conference Montreal, Canada.*, 2005.

[35] I. Pappas, T. Keller, and S. Mangold, "A reliable, gyroscope based gait phase detection sensor embedded in a shoe insole," in *Proceedings of IEEE Sensors 002. First IEEE International Conference on Sensors*, 2002, pp. 1085–8.

[36] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[37] P. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *Int. Conf. on Humanoid Robotics*, 2006.