"Real-time estimate of velocity and acceleration of quasi-periodic signals using adaptive oscillators"

Ronsse, Renaud ; De Rossi, Stefano Marco Maria ; Vitiello, Nicola ; Lenzi, Tommaso ; Carrozza, Maria Chiara ; Ijspeert, Auke Jan

Abstract

Estimating the temporal derivatives of a noisy position signal is a ubiquitous problem in industrial and robotics engineering. Here, we propose a new approach to get velocity and acceleration estimates of cyclical/periodic signals near to steady-state regime, by using adaptive oscillators. Our method combines the advantages of introducing no delay, and filtering out the high-frequency noise. We expect this method to be useful in control applications requiring undelayed but smooth estimates of velocity and acceleration (e.g. velocity control, inverse dynamics) of quasiperiodic tasks (e.g. active vibration compensation, robot locomotion, lower-limb movement assistance).

Ronsse, Renaud ; De Rossi, Stefano Marco Maria ; Vitiello, Nicola ; Lenzi, Tommaso ; Carrozza, Maria Chiara ; et al. *Real-time estimate of velocity and acceleration of quasi-periodic signals using adaptive oscillators*. In: *IEEE Transactions on Robotics*, Vol. X, no.X, p. X (X)

DOI : 10.1109/TRO.2013.2240173

Real-time estimate of velocity and acceleration of quasi-periodic signals using adaptive oscillators

Renaud Ronsse, *Member, IEEE,*
Stefano Marco Maria De Rossi, *Student Member, IEEE,* Nicola Vitiello, *Member, IEEE,* Tommaso Lenzi, *Student Member, IEEE,* Maria Chiara Carrozza, *Member, IEEE,* and Auke Jan Ijspeert, *Member, IEEE*

***Abstract*—Estimating the temporal derivatives of a noisy position signal is a ubiquitous problem in industrial and robotics engineering. Here, we propose a new approach to get velocity and acceleration estimates of cyclical/periodic signals near to steady-state regime, by using adaptive oscillators. Our method combines the advantages of introducing no delay, and filtering out the high-frequency noise. We expect this method to be useful in control applications requiring undelayed but smooth estimates of velocity and acceleration (e.g. velocity control, inverse dynamics) of quasi-periodic tasks (e.g. active vibration compensation, robot locomotion, lower-limb movement assistance).**

***Index Terms*—Oscillator, Filtering, Kinematics, Learning and Adaptive Systems, Calibration and Identification**

I. Introduction

Digital encoders are commonly used in industrial and robotics applications to measure angular positions. However their position accuracy is limited by the quantized position measurement, i.e. the number of slits on the encoder disk surface, and the actual regularity of the space between two consecutive slits. This is equivalent to assume that the actual position is polluted by random (white) noise in the measurement.

On top of that, many applications require a good estimate of the signal temporal derivatives, i.e. velocity (e.g. for feedback control [1]) and acceleration (e.g. for inverse dynamic compensation [2]). An approach is to use specific sensors to get these estimates, e.g. tachometers for velocity and accelerometers for acceleration. This however increases the cost and encumbrance of the device, while the signal processing becomes challenging [3]. The alternative is to infer velocity and acceleration from the position reading. The most naive approach is to do direct differentiation of the raw position signal, but this brings very noisy signals, due to the amplification of quantization errors by time-differentiation [3], [4]. Therefore, smart signal processing techniques have to be used, such that an abundant literature has been produced on that topic [3]–[17].

In this paper, we propose a novel approach to estimate the velocity and acceleration (and potentially higher-order derivatives) of a noisy measured position signal by using adaptive oscillators. Adaptive oscillators are tools developed by Righetti et al. [18], [19] for various applications [20]: signal processing (as an alternative to wavelet to perform time-varying frequency analysis), dynamical systems (construction of limit cycles of arbitrary shape) and robotics (e.g. adaptive control of compliant robots [21]). More recently, we paved the way for a new application field for adaptive oscillators, namely in human-robot interactions, including movement assistance [22]–[26] and predictive control [27], [28]. The main property of an adaptive oscillator refers to its capacity to synchronize to an input signal while learning its features (frequency and amplitude/envelope) in dedicated state variables. As such, estimates of temporal derivatives can be obtained from analytical expressions of the estimated signal envelope. This approach is thus specifically tailored for cyclical/periodic signals close to steady-state regime, but combines the nice advantages of filtering out the measurement noise (through the dynamics of the adaptive oscillator) and providing estimates which are, on average, phase-synchronized (i.e. delay-free) with respect to the actual temporal derivatives [19]. This is a critical difference between this approach and classical low-pass filtering, which unavoidably introduces delay.

In the next section, we briefly review the literature on signal derivative estimation. Thereafter, we present our approach and compare it to the most widely used techniques in classical applications, namely Kalman filtering and polynomial interpolation. A preliminary version of the method presented here with an application to robot-aided walking assistance has recently been accepted to a conference [29]. The original contributions of the present paper are to refine the theoretical developments, to provide quantified comparisons of our approach with respect to the state-of-the-art by using benchmark signals, and to provide the theory for the synthesis of an adaptive oscillator, i.e. to tune its gains to achieve a desired bandwidth in feature estimation (see the appendix).

II. State-of-the-art

Existing methods for estimating the time derivatives of an encoder signal fall in two categories, depending whether a model of the system dynamics is available or not. Model-based approaches require a full dynamic model of the system in order to combine the system's inputs and outputs with dynamic predictions. Typical approaches include model-based Kalman filters [16], neural networks [11] or other nonlinear methods [6], [7]. Model-based approaches can provide accurate estimates of derivatives, but are strongly context-dependent, such that their performances significantly rely on the accuracy of the process model.

Model-free methods, conversely, do not make use of dynamic models of the system, and only rely on data processing algorithms. In the rest of this paper, we will focus on this category of estimators. The naive way to obtain a velocity estimate of an input signal is through a digital, filtered derivative estimation. This approach suffers from the well-know trade-off between bandwidth and smoothness [17]: The smoother the estimate, the higher the time (or phase) delay introduced by the filter. Most non-naive approaches can be divided into three main categories [4]: (i) Predictive post-filtering techniques, (ii) linear state space observers, and (iii) indirect methods interpolating the data before performing an exact (continuous) derivative evaluation.

Predictive post-filtering techniques perform filtering on the numerically differentiated signal. Most common approaches are summarized in [8] and include Taylor series expansion, backwards difference expansion, and least squares fitting. All these techniques provide a predictive filtering of the past signal, and then estimate the signal slope based on its derivative. A more recent approach using adaptive windowing was proposed in [10] and showed to be superior to traditional filtering techniques. Importantly, these techniques were developed to get a delay-free estimate of the first derivative only (i.e. velocity), and are therefore not further described in this paper.

State space observers use a linear, time-invariant model as a representation of data and its derivatives. The most common of these approaches is the model-free linear Kalman filter technique [3]. In this case, the process is modeled as a noise-driven chain of n integrators: $\dot{X}(t) = AX(t) + \Gamma\xi(t)$, $y(t) = CX(t) + e(t)$, where:

$$A = \begin{pmatrix} \mathbf{0_{n-1\times 1}} & \mathbf{I_{n-1}} \\ 0 & \mathbf{0_{1\times n-1}} \end{pmatrix}, \qquad \Gamma = \begin{pmatrix} \mathbf{0_{n-1\times 1}} \\ 1 \end{pmatrix},$$

R. Ronsse is with the Centre for Research in Mechatronics, Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (e-mail: renaud.ronsse@uclouvain.be).

S. M. M. De Rossi, N. Vitiello, T. Lenzi, and M. C. Carrozza are with The BioRobotics Institute, Scuola Superiore Sant'Anna, 56025 Pontedera (Pi), Italy (e-mail: {s.derossi,n.vitiello,t.lenzi,carrozza}@sssup.it).

A. J. Ijspeert is with the Biorobotics Laboratory, Institute of Bioengineering, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: auke.ijspeert@epfl.ch).

and $C = (1 \ \mathbf{0_{1\times n-1}})$, with $\mathbf{0_{n\times m}}$ being a $n \times m$ zero matrix, $\mathbf{I_n}$ being the square identity matrix of size n, and $\xi(t)$ being white noise, modeling the variability of the last integrator. Finally, $e(t)$ models the encoder's quantization and measurement noise, affecting the measured output $y(t)$, i.e. the position. An observer of this system is constructed as:

$$\dot{\hat{X}}(t) = A\hat{X}(t) + K\left(y(t) - C\hat{X}(t)\right), \tag{1}$$

where K is the vector of the observer gains, whose optimal value is given by Kalman theory. In sum, this approach provides an estimate of the position and its $(n-1)$ first derivatives by using a n-equations system, the variability of the intrinsic noise $\xi(t)$ being the sole open parameter. Practically the estimate of the i-th order temporal derivative is given by the $(i+1)$-th state variable. A more recent implementation, based on an extended Kalman filter taking the encoder quantization into account, was proposed in [30]. Other state-space techniques were proposed in [9], [14], [15]. The approach proposed by [12] is also worth being mentioned, because it specifically targets periodic signals. It is however practically more difficult to implement than the method presented here, due to the large number of open parameters, requiring a partial knowledge of the input frequency content.

Finally, indirect methods are based on the approximation of the signal using interpolation, and analytical derivation of this interpolation to get the derivative estimates. Polynomial interpolation approaches were first presented in [5], and were further developed based on either fixed-time or fixed-position algorithms. Fixed-position algorithms require a very accurate clock and therefore typically more expensive hardware. Fixed-time polynomial interpolation consists in interpolating the n last encoder positions x_k (sampled at time t_k) by a polynomial of order m. This is achieved by solving an equation like $Y_k P_k = X_k$, where:

$$Y_k = \begin{pmatrix} t_{k-n+1}^m & t_{k-n+1}^{m-1} & \dots & 1 \\ \vdots & \vdots & & \vdots \\ t_{k-1}^m & t_{k-1}^{m-1} & \dots & 1 \\ t_k^m & t_k^{m-1} & \dots & 1 \end{pmatrix},$$

$X_k = (x_{k-n+1} \ \dots \ x_{k-1} \, x_k)^T$, and $P_k = (p_m \, p_{m-1} \ \dots \ p_0)^T$ is the coefficient vector, to be determined. If $n > m$, the problem is over-determined and can be solved using the least squares method, i.e. $P = (Y^T Y)^{-1} Y^T X$. Estimates of the signal's time derivatives are obtained by differentiating the estimate $\hat{x}_k$. For example an estimate of the velocity can be obtained from:

$$\hat{\dot{x}}_k = \sum_{i=1}^{m} i p_i t_k^{i-1}. \tag{2}$$

Recently, new techniques to skip some recorded encoder events were proposed to increase the precision of the estimate in fixed-position algorithms (e.g. [4]), for estimation of both velocity and acceleration.

III. Adaptive oscillators

The central element of the estimation method presented in this paper is an adaptive oscillator [18], [19]; i.e. a dynamical system having the capacity to synchronize to a (quasi)-periodic input by learning its features in dedicated state variables. Let's start with the simplest case, i.e. assuming the reference input $\theta_r(t)$ to be a sinusoidal signal:

$$\begin{aligned} \theta_r(t) &= \alpha_r \sin\left(\phi_r(t)\right), \\ \dot{\phi}_r(t) &= \omega_r(t), \end{aligned} \tag{3}$$

where α_r, ϕ_r, and ω_r are the amplitude, phase, and frequency (i.e. the features) of the reference input. An adaptive oscillator for such input is thus a dynamical system whose state variables (namely phase ϕ, frequency ω, and amplitude α) converge to the corresponding input's ones, and whose output $\hat{\theta}(t) = \alpha \sin\left(\phi(t)\right)$ gets synchronized to the reference input (3). The simplest system achieving this behavior is an augmented phase oscillator:

$$\begin{aligned} \dot{\phi}(t) &= \omega(t) + \nu_\phi \frac{F(t)}{\alpha(t)} \cos\phi(t), \\ \dot{\omega}(t) &= \nu_\omega \frac{F(t)}{\alpha(t)} \cos\phi(t), \\ \dot{\alpha}(t) &= \eta F(t) \sin\phi(t), \end{aligned} \tag{4}$$

where ν_ϕ, ν_ω, and η are the learning gains determining the speed of phase synchronization to the error signal $F(t) = \theta_r(t) - \hat{\theta}(t)$. A method for doing the *synthesis* of this oscillator, i.e. to tune these gains to achieve desired performances in learning the features, is provided in the appendix. Note that in (4), the oscillator frequency is a state variable *learning* the frequency of the error signal $F(t)$, instead of doing mere synchronization only. As such, the oscillator has the capacity to constantly adapt its features as a function of the error signal, and to keep these features in memory, i.e. in the state variables. Righetti et al. provided the proof of convergence of $\omega(t)$ toward a stationary frequency for a similar adaptive oscillator as the one presented here [18], while [19] provided further results with time-varying parameters.

Now, if the reference input is periodic but no longer sinusoidal, the same authors further proposed to put several oscillators in parallel in a feedback loop (see the upper part of Fig. 1) to learn the input features (i.e. the frequency spectrum) [31]. Indeed, if the input signal is periodic (such that all frequencies with a non-zero power spectrum are multiple of a fundamental harmonic), an estimate can be derived from its Fourier decomposition:

$$\hat{\theta}(t) = \sum_{i=0}^{K} \alpha_i \sin\left(\phi_i(t)\right) = \sum_{i=0}^{K} \alpha_i \sin\left(i\omega t + \varphi_i\right), \tag{5}$$

where K denotes the number of harmonics kept in the estimate. Righetti et al. [31] established the convergence of $\hat{\theta}(t)$ to $\theta(t)$ by copying the former oscillator for the learning of each of the harmonics:

$$\begin{aligned} \dot{\phi}_i(t) &= i\omega(t) + \nu_\phi \frac{F(t)}{\sum_i \alpha_i(t)} \cos\phi_i(t), \\ \dot{\omega}(t) &= \nu_\omega \frac{F(t)}{\sum_i \alpha_i(t)} \cos\phi_1(t), \\ \dot{\alpha}_i(t) &= \eta F(t) \sin\phi_i(t). \end{aligned} \tag{6}$$

Note that in (6), the 0-th oscillator ($i = 0$) is a simple integrator (assuming $\phi_0(0) = \pi/2$) learning the signal offset, i.e. $\dot{\alpha}_0(t) = \eta F(t)$. From (6), it is visible that steady-state is reached when $F(t) = 0$, i.e. when $\hat{\theta}(t) = \theta(t)$. If $\theta(t)$ is only quasi-stationary — i.e. if the input features (frequency, amplitudes, phases) slowly vary in time — $\hat{\theta}(t)$ will be a low-pass filtered version of $\theta(t)$. Indeed, random noise affecting the input $\theta(t)$ will be filtered out by the dynamics of the adaptive oscillator (6). Importantly, $\hat{\theta}(t)$ and $\theta(t)$ will however be phase-synchronized on average [19].

If the estimated features (namely ω, α_i, φ_i) converged to the actual ones, an estimate of the input's derivatives can be obtained by differentiating (5), i.e.:

$$\widehat{\frac{d^j\theta}{dt^j}}(t) = \sum_{i=1}^{K} \alpha_i(t) (i\omega(t))^j \sin\left(\phi_i(t) + j\frac{\pi}{2}\right). \tag{7}$$

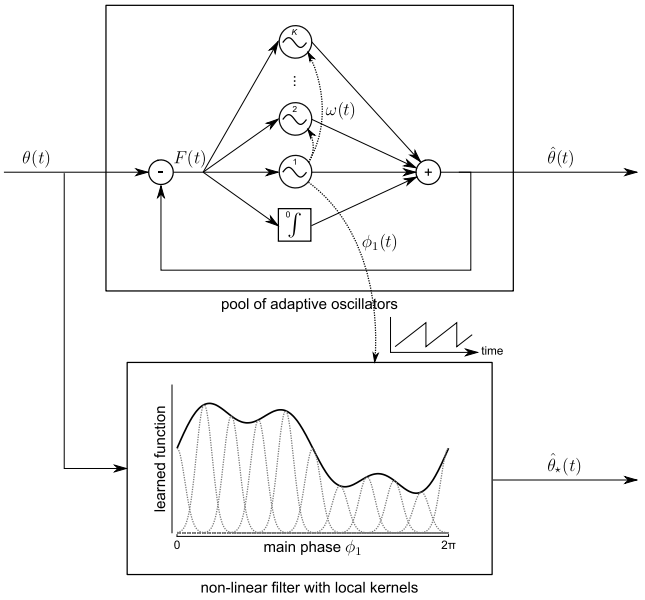

Fig. 1. On-line learning of a periodic but non-sinusoidal input signal $\theta(t)$. The upper block is a pool of adaptive oscillators (6), decomposing the input into a real-time Fourier series. The lower block is a kernel-based non-linear filter, mapping the phase of the main harmonic $\phi_1(t)$ to the input envelope. Adapted from [32]

Again in steady-state (i.e. if the features are stationary and the estimations converged), these filtered estimates are undelayed with respect to the actual ones, in contrast to more classical low-pass filter-based approaches like Kalman filtering.

In [22], [23], [25], we used this approach to estimate the velocity and acceleration of human quasi-sinusoidal elbow movements — i.e. by limiting (5) and (7) to the first harmonic ($K = 1$) — and to provide robot-aided movement assistance based on an inverse dynamic model.

IV. Kernel-based filter

If the input signal $\theta(t)$ possesses a large frequency spectrum, like for instance if it contains a plateau of quasi-constant position, the estimated signal $\hat{\theta}(t)$ can only merely approximate the original one, since the learned signal will be truncated to a finite number of harmonics K. Moreover, this approximation error grows up with the successive time derivatives, which amplify high frequencies, each by a factor ω (see (7)).

To solve this limitation by keeping K reasonably low, we propose to augment the structure of the learning algorithm with a second block, working in the time (or phase) domain [32]. The approach is described in Figure 1. The pool of adaptive oscillators is used only to extract the *phase* of the input signal. i.e. $\phi_1(t)$. This phase is used afterwards as the input of a non-linear filter working in the phase domain.

This filter actually solves a supervised learning problem, where the signal to be learned is approximated as a sum of local models, i.e.:

$$\hat{\theta}_\star(t) = \frac{\sum_{i=1}^{N} \Psi(\varphi_i(t)) w_i}{\sum_{i=1}^{N} \Psi(\varphi_i(t))}, \tag{8}$$

where $\varphi_i(t) = \phi(t) - c_i$, and $\Psi(\varphi) = \exp\left(h(\cos\varphi - 1)\right)$ is a Gaussian-like kernel functions. The parameter h determines the kernel width, and $c_i = \bar{c} + i2\pi/N$ the center of the N kernel functions being summed (equally spaced between 0 and 2π in N steps). These kernel functions are represented by the dotted gray curves in Figure 1. Locally weighted regression corresponds to finding the weight vector w_i which minimizes a quadratic error criterion. Following [32], [33], an on-line version of this learning process can be implemented using incremental regression to determine the weights w_i. This relies on recursive least squares with a forgetting factor of λ, determining the learning dynamics. Given the input $\theta(t)$, w_i is updated by:

$$\begin{aligned} w_i(t_{k+1}) &= w_i(t_k) + \Psi(\varphi_i(t_k)) P_i(t_{k+1}) \left(\theta(t_k) - w_i(t_k)\right), \\ P_i(t_{k+1}) &= \frac{1}{\lambda}\left(P_i(t_k) - \frac{P_i(t_k)^2}{\frac{\lambda}{\Psi(\varphi_i(t_k))} + P_i(t_k)}\right), \end{aligned} \tag{9}$$

where the t_k's are the discrete time steps, and P is the inverse covariance matrix [34]. If $\lambda < 1$, the regression gives more weight to recent data.

An estimate of the temporal derivatives of $\theta(t)$ can again be obtained by differentiating equation (8). The first derivative gives:

$$\dot{\hat{\theta}}_\star(t) = \omega(t)\left(\frac{\sum \Psi'(\varphi_i(t)) w_i}{\sum \Psi(\varphi_i(t))} - \frac{\sum \Psi'(\varphi_i(t)) \sum \Psi(\varphi_i(t)) w_i}{\left(\sum \Psi(\varphi_i(t))\right)^2}\right), \tag{10}$$

where $\sum$ stands for $\sum_{i=1}^{N}$ and the kernel derivative Ψ' is given by:

$$\Psi'(\varphi) = \frac{d\Psi}{d\varphi} = -\Psi(\varphi) h \sin\varphi. \tag{11}$$

The second term of (10) is proportional to $\sum \Psi'$. This sum can be viewed as an approximation of the integral of its argument by the midpoint method. Interestingly, the analytical expression of the error due this approximation can be found in any standard textbook of numerical methods. It is equal to:

$$\begin{aligned} \sum_{i=1}^{N} \Psi'(\varphi_i) &= \frac{N}{2\pi} \sum_{i=1}^{N} \left(\int_{\varphi_i - \frac{\pi}{N}}^{\varphi_i + \frac{\pi}{N}} \Psi'(\varphi) d\varphi - \frac{(2\pi)^3}{24N^3} \Psi'''(\sigma_i) \right) \\ &= \frac{N}{2\pi} \int_0^{2\pi} \Psi'(\varphi) d\varphi - \frac{(2\pi)^2}{24N^2} \sum_{i=1}^{N} \Psi'''(\sigma_i) \\ &= \frac{N}{2\pi} \left(\Psi(2\pi) - \Psi(0)\right) - \frac{(2\pi)^2}{24N^2} \sum_{i=1}^{N} \Psi'''(\sigma_i), \end{aligned}$$

where $\left[\varphi_i - \frac{\pi}{N} \leq \sigma_i \leq \varphi_i + \frac{\pi}{N}\right]$. The first term is equal to 0 since $\Psi(\varphi)$ is a 2π-periodic function. The second term can be made arbitrarily small since it can be showed that:

$$\left|\Psi'''(\varphi)\right| \leq \min\left(3h + 1, h^2 + 1.25\right).$$

Therefore, by choosing the number of kernels N large enough and the kernel width wide enough (i.e. h small enough), the second term of (10) can be neglected with respect to the first one. Then (10) simply reduces to:

$$\dot{\hat{\theta}}_\star(t) = \omega(t) \frac{\sum \Psi'(\varphi_i(t)) w_i}{\sum \Psi(\varphi_i(t))},$$

i.e. to a weighted sum of the derivatives of each local kernel. Similarly, estimates of temporal derivatives of any order j can be obtained from:

$$\widehat{\frac{d^j \theta_\star}{dt^j}}(t) = \omega^j(t) \frac{\sum \frac{d^j \Psi(\varphi_i(t))}{d\varphi^j} w_i}{\sum \Psi(\varphi_i(t))}, \tag{12}$$

if the system is at steady-state, i.e. if $\dot{\omega}(t) = 0$

For acceleration, the second derivative of the local kernel is obtained from (11):

$$\Psi''(\varphi) = -\Psi'(\varphi) h \sin\varphi - \Psi(\varphi) h \cos\varphi. \tag{13}$$

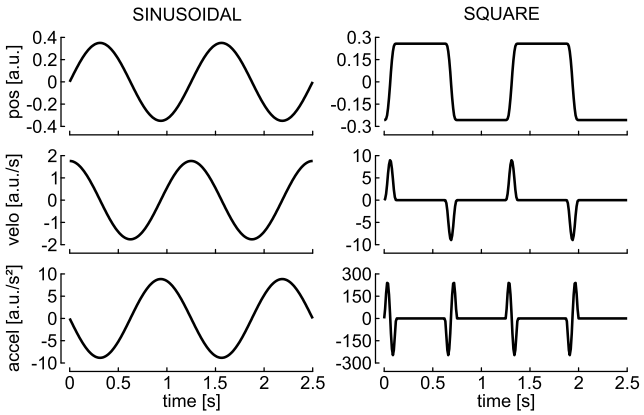

Fig. 2. Benchmark signals (position/velocity/acceleration) used to evaluate the filter performances. Left: a simple sinusoidal signal; right: almost square wave with continuous first, second, and third order derivatives

V. Tests with benchmark signals

To evaluate the method efficiency and compare it to the most standard existing methods, we tested it on benchmark signals for which an analytical expression of the position, velocity, and acceleration can be derived. Two signals were selected, the first having a narrow frequency spectrum, the second having a wide one:

1) A simple sinusoidal signal, such that $\theta(t) = \alpha_r \sin(\omega_r t)$, $\dot{\theta}(t) = \alpha_r \omega_r \cos(\omega_r t)$, and $\ddot{\theta}(t) = -\alpha_r \omega_r^2 \sin(\omega_r t)$, with $\alpha_r = 0.35$ and $\omega_r = 2\pi/T$.
2) A square wave, with smooth transients such that the signal is continuous up to the third order derivative:
   - for $0 \leq \bar{t} < 0.1$, ascending slope: $\theta(t) = \sum_{i=0}^{7} q_i \bar{t}^i$, $\dot{\theta}(t) = 1/T \sum_{i=0}^{6} (i+1) q_{i+1} \bar{t}^i$, and $\ddot{\theta}(t) = 1/T^2 \sum_{i=0}^{5} (i+2)(i+1) q_{i+2} \bar{t}^i$;
   - for $0.1 \leq \bar{t} < 0.5$, plateau: $\theta(t) = -q_0$, $\dot{\theta}(t) = \ddot{\theta}(t) = 0$;
   - for $0.5 \leq \bar{t} < 0.6$, descending slope: same expressions as in the first case, multiplying the q_i by -1;
   - and for $0.6 \leq \bar{t} < 1$, plateau: $\theta(t) = q_0$, $\dot{\theta}(t) = \ddot{\theta}(t) = 0$;

   with $q_0 = -0.257$, $q_1 = q_2 = q_3 = 0$, $q_4 = 18$, $q_5 = -43.2$, $q_6 = 36$, $q_7 = -10.3$, and $\bar{t} = \mathrm{mod}(t/T, 1)$ denotes the normalized time.

These signals are shown in Figure 2, over 2 periods. Both have the same period $T = 1/\omega_r = 1.25$s, the same mean ($\overline{\theta(t)} = 0$) and standard deviation ($\mathrm{std}(\theta(t)) = 0.248$). The figure also shows the velocity and acceleration profiles. The second signal ("square") has larger acceleration peaks, due to a widest frequency spectrum.

To simulate realistic experimental conditions, these two signals were sampled at 100Hz, and corrupted by a random noise of standard deviation equal to 0.0124 (5% of the signal standard deviation, giving rise to a signal-to-noise ratio equal to 20). Four different estimation methods, all being fixed-time, were tested on these two signals:

1) A Kalman filter (1), designed as explained in Section II, with $n = 3$ and the variability of $\xi(t)$ being equal to the standard deviation of the signal's third order derivative (jerk);
2) a filter based on polynomial interpolation (see (2) and Section II), with a polynomial of degree $m = 2$ (parabola), and a buffer size equal to $n = 10$;
3) AdOsc: a pool of adaptive oscillators (Equations (6) were discretized by using finite difference), with the gains tuned from (16), (17), and (18) (see appendix) to get desired frequency and amplitude learning time constants equal to $\tau_\omega = 1.25\text{s} = T$ and $\tau_\alpha = 3.5\text{s} = 2.8T$. Estimates were computed from (7).
4) AdOsc+NLF: a pool of adaptive oscillators (6) with the gains tuned to get $\tau_\omega = \tau_\alpha = 3.5\text{s} = 2.8T$, coupled with a non-linear filter (9). A slower frequency time constant τ_ω as in the 'AdOsc' was selected, to comply with the non-linear filter own dynamics. The filter parameters were set to $\lambda = 0.9985$, $N = 90$, and $h = 90$. Estimates were obtained from (12).

The number of oscillators K was chosen large enough to capture the rich frequency spectrum of the square wave. In the 'AdOsc' case, we selected K as being equal to the number of harmonics having a Fourier transform larger than 1% of the main harmonic, i.e. $K = 8$ in the case of the square wave. In the 'AdOsc+NLF' case, we raised the threshold at 15% of the main harmonic, i.e. $K = 3$ in the case of the square wave. Indeed, less oscillators were required since the signal envelope was learned by the non-linear filter. All methods were implemented in MATLAB (the MatWorks, Natick, MA), using dedicated functions when appropriate (e.g. `kalman` for the Kalman filter design and `polyfit` for the polynomial interpolation).

The estimated accelerations with these four methods are shown in Figure 3, over a representative cycle (1.25s) of steady-state regime. Position and velocity estimates are not shown since they were hardly distinguishable from each other. Since the Kalman filter and the filter based on polynomial interpolation are low-pass filters, they introduced a delay in the signal. As a consequence, the estimations lag behind the actual signals. This is mainly visible for the acceleration. Moreover, the filter based on polynomial interpolation was the noisiest estimator, mainly if the input had a narrow frequency spectrum

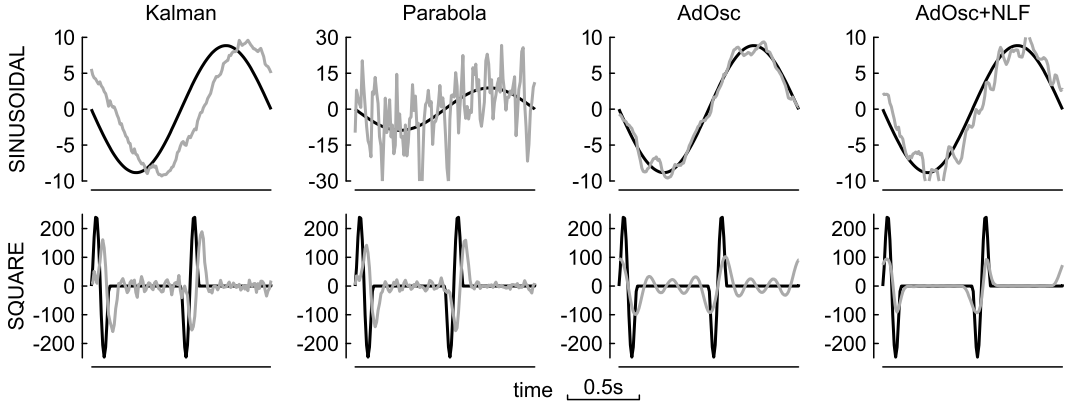

Fig. 3. Estimated acceleration of the benchmark signals over a representative cycle. Top: simple sinusoidal signal; Bottom: almost square wave with continuous first, second, and third order derivatives. The actual acceleration is shown in black, and the estimates obtained with the different methods in gray: Kalman filter (leftmost), polynomial interpolation (middle left), adaptive oscillators only (middle right), and adaptive oscillators + non-linear filter (rightmost)

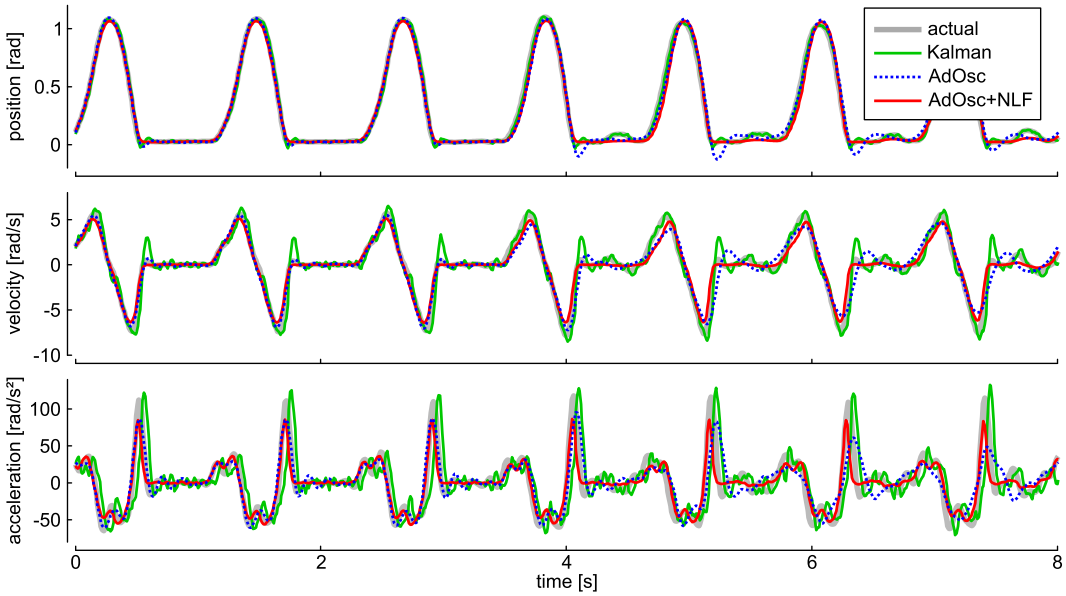

Fig. 4. Knee kinematics (position/velocity/acceleration) used to evaluate the filter performances. The actual reference signals are shown in gray, and the estimates obtained using different methods in color: Kalman filter (green), adaptive oscillators (dotted blue), and adaptive oscillators + non-linear filter (red)

TABLE I
RESULTS OF THE ESTIMATES OF THE BENCHMARK SIGNALS

		Kalman	poly	AdOsc	AdOsc +NLF
SINUSOIDAL					
Vel.	error [a.u./s]	0.303	0.405	**0.030**	**0.030**
	correlation	0.990	0.929	**1.000**	**1.000**
Acc.	error [a.u./s^2]	4.12	8.73	**0.87**	1.56
	correlation	0.730	0.486	**0.984**	0.956
CPU time [a.u.]		**0.03**	14.17	1.64	2.13
SQUARE					
Vel.	error [a.u./s]	1.312	1.067	0.787	**0.450**
	correlation	0.699	0.805	0.892	**0.929**
Acc.	error [a.u./s^2]	56.5	50.4	40.8	**23.6**
	correlation	-0.258	-0.218	0.596	**0.813**
CPU time [a.u.]		**0.03**	14.11	1.52	2.01

a.u.: arbitrary unit

(small velocity/acceleration). The pool of adaptive oscillators behaved very well when the input had a narrow frequency spectrum (sinusoidal signal), the behavior resulting from a perfect synchronization with the input. However, when the input spectrum was wider, the loss of the $> K$ harmonics resulted in the well-known Gibbs phenomenon, i.e. small oscillations during the plateaued epochs. This was improved by adding a non-linear filter in the time-domain (AdOsc+NLF). Note that all methods underestimated the acceleration peaks for the square signal.

To further quantify these results, we computed the average absolute error between the estimated and actual signals, i.e. $\overline{\| \bullet(t) - \hat{\bullet}(t) \|}$; and their correlation, for the velocity and acceleration of both benchmarks. The results are given in Table I.

As already pointed out, the pool of adaptive oscillators provided the best estimates when the input spectrum was narrow, while the addition of the non-linear kernel filter proved to be effective when the input spectrum was wider. The method based on polynomial interpolation provided noisy estimates, mainly for high order derivatives, and when the velocity was low [3]. Note that the delay caused both by the Kalman filter and the polynomial filter caused the acceleration correlation to be negative (the estimated positive peak was sync to the actual negative peak) for the square signal. Table I also gives the computation time required for each method. The polynomial filter had the poorest computational efficiency, since it required the largest amount of basic operations. In contrast, the Kalman filter was very cheap (it basically required two matrix multiplications and one addition each time step). The new proposed method was in between, with a factor < 100 with respect to Kalman filter when the non-linear filter was added.

VI. TEST WITH A REAL "PERIODIC" SIGNAL

We further put our method to test using a real signal, corresponding to the kinematics of the right knee during a walking task on a treadmill (data published in [26]). This signal was acquired by the encoder of the LOPES training device [35], [36], in a condition where the LOPES was controlled to provide some assistance to the hips. The main interest in this test is to check the method accuracy for a signal that is not stationary. Therefore, we computed the same metrics as for the benchmark signals, but focusing on epochs were the treadmill speed changed, forcing the walker to change the gait cadence. In this case, the reference signal used for derivatives was obtained through a zero-phase forward-backward filtered differentiation (Butterworth, 3-rd order, 10 Hz).

Figure 4 shows the estimation results during a representative 8s epoch (including a treadmill speed transient around $t = 3$s). For the sake of clarity, we did not show the results obtained with the polynomial filter, since the tests with the benchmarks signals showed its performance to be equal or lower than the Kalman filter, with a higher computational cost. Similar conclusions as for the benchmark signals can be emphasized: (i) all filters performed equally well on the position signal, except maybe the one based on the pool of adaptive oscillators only (AdOsc) which had difficulties to capture abrupt transients (with a rich frequency spectrum); (ii) the Kalman filter was noisier, and introduced a delay (mainly visible in the acceleration peak); and (iii) the non-linear filter mainly served to capture the high frequency components (the plateaus were more flat as with the two other methods) and to adapt more rapidly to a frequency change.

Figure 5 shows the actual instantaneous movement frequency (the inverse of the cycle duration, multiplied by 2π) of the same data as

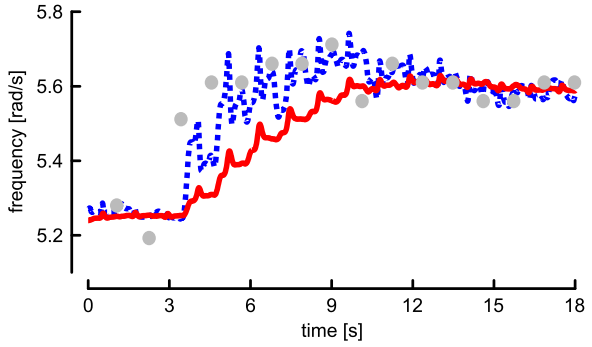

Fig. 5. Estimated movement instantaneous frequency $\omega(t)$ (6). Dotted blue: adaptive oscillators only 'AdOsc'; red: adaptive oscillators + non-linear filter 'AdOsc+NLF'. Gray dots represent an off-line estimate of the actual instantaneous frequency

TABLE II
RESULTS OF THE ESTIMATES OF THE KNEE KINEMATICS

		Kalman	AdOsc	AdOsc +NLF
Vel.	error [rad/s]	0.507	0.317	**0.273**
	correlation	0.965	0.985	**0.990**
Acc.	error [rad/s^2]	18.19	8.98	**6.89**
	correlation	0.599	0.918	**0.949**
CPU time [a.u.]		**0.13**	6.42	8.58

presented in Figure 4 (same $t = 0$s, but longer time horizon, for better readability). It also shows the frequency $\omega(t)$ (6) estimated by AdOsc ($\tau_\omega = 1.25$) and AdOsc+NLF ($\tau_\omega = 3.5$). It shows that both provided a very good and reactive estimate of the movement frequency, in line with the desired time constants.

To further quantify these results, the same table as for the benchmark signals is given in Table II. The metrics (errors and correlations) were computed over a period of 14 minutes, including 6 transitions between 3 different treadmill speeds. The table confirms the trends discussed from Figure 4: Better estimations with our method as with the Kalman filter (mainly for the acceleration), and a small improvement if the non-linear filter is added, pending a small increase of the computational cost (by about 25%). The Kalman filter is still by far the cheapest method regarding the computational cost.

VII. CONCLUSION

The paper provided a new method for getting a real-time estimate of the temporal derivatives (mainly velocity and acceleration) of noisy position signals, e.g. obtained through an incremental encoder. This method works only for cyclical/periodic signals near to steady-state regime, but has the paramount advantage to provide estimates which are both filtered (i.e. the high-frequency, noise-driven part of the signal is filtered out), and delay-free (as opposed to classical low-pass filters). This is achieved by synchronizing the input signal to a (pool of) adaptive oscillator(s), whose convergence is proved in [18], [19]. As such, the signal features (frequency, amplitude/envelope) are low-pass filtered, rather than the signal itself. If the input signal has a wide frequency spectrum, the addition of a non-linear kernel filter in the phase domain proved to be efficient to avoid signal distortion (Gibbs effect) in the plateaued epochs. This non-linear filter basically allowed to decrease the number of oscillators (3 vs. 8 in the example above), such that the overall extra computational payload was marginal. The efficiency of our approach vs. classical filtering techniques was further illustrated on benchmark and real (non-stationary) signals.

In conclusion, we expect our method to be useful for applications requiring accurate and predictive control of (quasi-)periodic movements, like active vibration compensation [37], and locomotion-related tasks in robotics, e.g. animaloid and humanoid locomotion, human locomotion assistance/rehabilitation, etc.. We already paved the way in this last area, by studying the method mentioned here for assistance of upper- [22], [23], [25] and lower-limb [29] movements, and to improve the transparency of a lower-limb exoskeleton [28].

APPENDIX
SYNTHESIS OF THE EXTENDED PHASE OSCILLATOR

In this appendix, we derive the synthesis rules of the simple extended phase oscillator (4) that was proposed to learn the features of a sinusoidal input (3) while synchronizing its output to it. More precisely, this appendix establishes how to tune the oscillator gains, namely ν_ϕ, ν_ω, and η, to make the oscillator amplitude α and frequency ω converging to the corresponding references (i.e. α_r and ω_r) with desired time constants.

In the case of a sinusoidal input and a single adaptive oscillator, the error signal is equal to

$$\begin{aligned} F(t) &= \theta_r(t) - \hat{\theta}(t) = \alpha_r(t)\sin\phi_r(t) - \alpha(t)\sin\phi(t) \\ &= (\alpha_r(t) - \alpha(t))\sin\phi_r(t) \\ &\quad + 2\alpha(t)\sin\left(\frac{\phi_r(t)-\phi(t)}{2}\right)\cos\left(\frac{\phi_r(t)+\phi(t)}{2}\right). \end{aligned}$$

Now, assuming that the system is close to steady-state, such that $2\sin\left(\frac{\phi_r-\phi}{2}\right) \simeq (\phi_r - \phi)$ and $\cos\left(\frac{\phi_r+\phi}{2}\right) \simeq \cos\phi$, (3), (4) can be written in the following state-space form:

$$\dot{x} = A(\phi,\alpha)x + B_\omega\omega_r + B_\alpha(\phi,\alpha)\alpha_r, \tag{14}$$

with (explicit time-dependence is withdrawn for clarity) $x^T = (\phi_r, \phi, \omega, \alpha)$ being the state vector,

$$A(\phi,\alpha) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \nu_\phi c_\phi^2 & -\nu_\phi c_\phi^2 & 1 & \frac{-\nu_\phi}{\alpha}s_\phi c_\phi \\ \nu_\omega c_\phi^2 & -\nu_\omega c_\phi^2 & 0 & \frac{-\nu_\omega}{\alpha}s_\phi c_\phi \\ \eta\alpha s_\phi c_\phi & -\eta\alpha s_\phi c_\phi & 0 & -\eta s_\phi^2 \end{pmatrix},$$

$B_\omega^T = (1,0,0,0)$, $B_\alpha^T(\phi,\alpha) = (0, \frac{\nu_\phi}{\alpha}s_\phi c_\phi, \frac{\nu_\omega}{\alpha}s_\phi c_\phi, \eta s_\phi^2)$, $s_\phi = \sin\phi$, and $c_\phi = \cos\phi$. The system (14) can then be viewed as an "input-output" system, whose inputs are the reference features (α_r and ω_r) and outputs are the estimated ones (α and ω). With constant inputs, the system time constants can be determined by calculating the step responses of (14). A few more derivations are required to use standard tools of linear systems theory to derive an explicit analytical form of these time constants.

First, we again make the hypothesis that the system stays close enough to steady-state to assume, for a constant input frequency ω_r: $A(\phi,\alpha) \simeq A(\phi_r,\alpha_r) = A(\omega_r t,\alpha_r)$ and $B_\alpha(\phi,\alpha) \simeq B_\alpha(\phi_r,\alpha_r) = B_\alpha(\omega_r t,\alpha_r)$. Then, we rely on the averaging technique [38] to derive an approximation of the solution of (14) from the solution of its averaged system:

$$\dot{x} = \bar{A}(\alpha_r)x + B_\omega\omega_r + \bar{B}_\alpha(\alpha_r)\alpha_r, \tag{15}$$

where

$$\begin{aligned} \bar{A}(\alpha_r) &= \frac{1}{2\pi}\int_0^{2\pi} A(\phi_r,\alpha_r)d\phi_r \\ &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{\nu_\phi}{2} & -\frac{\nu_\phi}{2} & 1 & 0 \\ \frac{\nu_\omega}{2} & -\frac{\nu_\omega}{2} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\eta}{2} \end{pmatrix} = \bar{A}, \end{aligned}$$

is the averaged state matrix. Similarly, $\bar{B}_\alpha^T(\alpha_r) = (0,0,0,\frac{\eta}{2}) = \bar{B}_\alpha^T$. This averaged system is thus linear, and its input-output transfer functions can be derived.

The amplitude transfer function is equal to [39]:

$$\frac{\alpha(s)}{\alpha_r(s)} = (0,0,0,1)(s\mathbf{I_4} - \bar{A})^{-1}\bar{B}_\alpha = \frac{\eta/2}{s+\eta/2}.$$

It has a static gain equal to 1 and a single pole in $-\eta/2$. Therefore, its step response is $(1-e^{-t/\tau_\alpha})$ with a time constant equal to:

$$\tau_\alpha = 2/\eta. \tag{16}$$

Interestingly, this time constant only depends on η, the larger the gain, the faster the time constant.

Similarly, the frequency transfer function is equal to:

$$\frac{\omega(s)}{\omega_r(s)} = (0,0,1,0)(s\mathbf{I_4} - \bar{A})^{-1}B_\omega = \frac{\nu_\omega/2}{s^2+\nu_\phi/2+\nu_\omega/2}.$$

It also has a static gain equal to 1, this time with two poles. By choosing:

$$\nu_\phi = 22\sqrt{\nu_\omega/20} = \sqrt{24.2\nu_\omega}, \tag{17}$$

this transfer function simplifies to:

$$\begin{aligned}\frac{\omega(s)}{\omega_r(s)} &= \frac{\nu_\omega/2}{(s+\sqrt{5\nu_\omega})(s+\sqrt{0.05\nu_\omega})} \\ &\simeq \frac{\sqrt{0.05\nu_\omega}}{s+\sqrt{0.05\nu_\omega}},\end{aligned}$$

because the neglected pole is 10 times faster than the other. Therefore, its approximated step response is equal to $(1-e^{-t/\tau_\omega})$ with a time constant equal to:

$$\tau_\omega = \sqrt{20/\nu_\omega}. \tag{18}$$

Again, this time constant only depends on a single gain ν_ω, with a faster time constant corresponding to a larger gain. Similarly, it can be established that the two dynamics are entirely decoupled since $\alpha(s)/\omega_r(s) = \omega(s)/\alpha_r(s) = 0$.

In sum, we established three relationships — namely (16), (17), and (18) — to tune the gains of (4) in order to achieve the desired time constants τ_α and τ_ω in the learning dynamics of the input features. The following simulations quantify the approximation error due to the steady-state assumption and averaging.

Figure 6 shows 6 seconds of a simulation of a single adaptive oscillator (4) receiving a sinusoidal input (black). The gains were tuned to get $\tau_\alpha = \tau_\omega = 0.5T$, where T is the initial cycle period. Here, $T = 2\pi/\omega = 1$s. At $t = 2$, the input frequency switches from $\omega_r = 2\pi$ to $\omega_r = 3\pi$ and the input amplitude from $\alpha_r = 1$ to $\alpha_r = 1.5$, simultaneously. The corresponding oscillator output and state variables are showed in red. The dotted blue curves show the solution of the first approximated system (14), assuming small deviations from steady-state. The magenta curve shows the solution of the second approximation, i.e. after averaging (15) and simplifying the fast frequency pole due to (17). These results validate the proposed synthesis, since both approximation are very close to the actual output.

We further reproduced this simulation for a large range of desired time constants and for independent amplitude and frequency steps. An exponential curve of the form $(1-e^{-t/\tau})$ was fit (to the least-square sense) on the system output to get an estimate of the actual time constant, and to compute the relative error with respect to the one predicted by (16) or (18). Figure 7 shows this relative error for the actual amplitude time constant τ_α, and for a large range of desired τ_α and τ_ω (between $0.2T$ and $5T$). This relative error is only marginally dependent on the desired τ_α (i.e. on η). For $\tau_\omega > 1.5$, it is bounded by 5%. It grows up for smaller desired τ_ω, likely because the close-to-steady-state assumption is no longer valid, and because the mutual influence between the frequency and amplitude dynamics can no longer be considered as negligible (as assumed in the averaging).

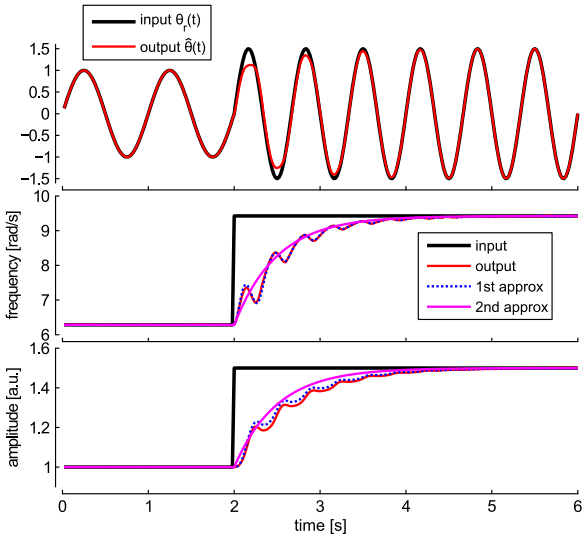

Fig. 6. Simulation results for a single adaptive oscillator subject to a simultaneous input frequency and amplitude change. Black: input signal; red: oscillator output and state variables; blue: first approximation (14); magenta: second approximation, i.e. averaged system (15)

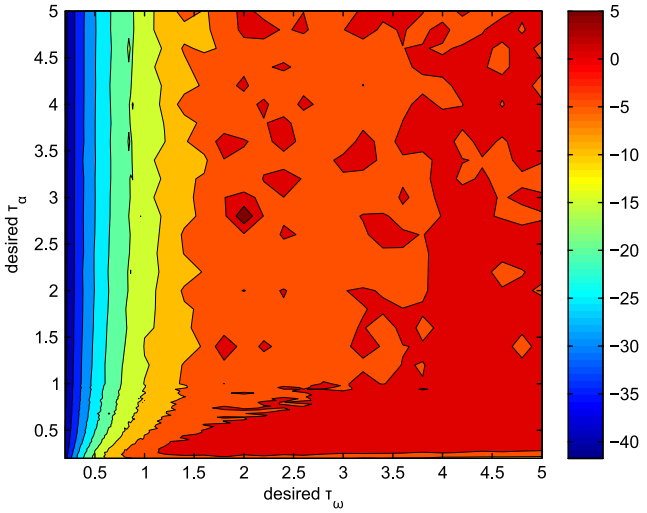

Fig. 7. Relative error (in % of τ_α) between the desired and the actual time constant of the ampltiude learning τ_α, as a function of desired time constants τ_α and τ_ω. The oscillators gains were tuned following (16), (17), (18)

Results for τ_ω are even better (no shown). The relative error stays close to -8% as soon as the desired frequency time constant τ_ω is larger than 0.3. This consistent negative error is likely due to the influence of the neglected fast pole. Finally, larger relative errors appear for the frequency time constants for very small gains (i.e. for large desired time constants τ_α and τ_ω, typically $> 5T$), again because the simplifying assumptions are no longer valid.

In conclusion, these simulations validate the synthesis rules (16), (17), and (18) in a large range of desired time constants, typically $0.2T \leq \tau_\alpha \leq 5T$ and $0.2T \leq \tau_\omega \leq 5T$.

ACKNOWLEDGMENT

The authors were funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements num. 231451 (EVRYON) and 287894 (CYBERLEGs). R.

Ronsse received funding from UCLouvain (internationalization grant, IMMC11/13) and F.R.S.-FNRS ("Crédit aux Chercheurs", 1.5025.12).

REFERENCES

[1] H. G. Sage, M. F. De Mathelin, and E. Ostertag, "Robust control of robot manipulators: A survey," *International Journal of Control*, vol. 72, no. 16, pp. 1498–1522, 1999.

[2] B. Koopman, E. H. F. van Asseldonk, R. Ronsse, W. van Dijk, and H. van der Kooij, "Rendering potential wearable robot designs with the lopes gait trainer," in *Proc. IEEE Int Rehabilitation Robotics (ICORR) Conf*, 2011, pp. 1–6.

[3] P. Belanger, P. Dobrovolny, A. Helmy, and X. Zhang, "Estimation of angular velocity and acceleration from shaft-encoder measurements," *Int J Robot Res*, vol. 17, no. 11, pp. 1225–1233, 1998.

[4] R. Merry, M. van de Molengraft, and M. Steinbuch, "Velocity and acceleration estimation for optical incremental encoders," *Mechatronics*, vol. 20, no. 1, pp. 20–26, 2010, special Issue on "Servo Control for Data Storage and Precision Systems", from 17th IFAC World Congress 2008.

[5] K. Saito, K. Kamiyama, T. Ohmae, and T. Matsuda, "A microprocessor-controlled speed regulator with instantaneous speed estimation for motor drives," *Industrial Electronics, IEEE Transactions on*, vol. 35, no. 1, pp. 95–99, 1988.

[6] S. Nicosia and P. Tomei, "Robot control by using only joint position measurements," *Automatic Control, IEEE Transactions on*, vol. 35, no. 9, pp. 1058–1061, 1990.

[7] C. Canudas de Wit and N. Fixot, "Robot control via robust estimated state feedback," *Automatic Control, IEEE Transactions on*, vol. 36, no. 12, pp. 1497–1501, 1991.

[8] R. H. Brown, S. C. Schneider, and M. G. Mulligan, "Analysis of algorithms for velocity estimation from discrete position versus time data," *Industrial Electronics, IEEE Transactions on*, vol. 39, no. 1, pp. 11–19, 1992.

[9] S. P. Chan, "Velocity estimation for robot manipulators using neural network," *Journal of Intelligent & Robotic Systems*, vol. 23, pp. 147–163, 1998.

[10] F. Janabi-Sharifi, V. Hayward, and C.-S. J. Chen, "Discrete-time adaptive windowing for velocity estimation," *Control Systems Technology, IEEE Transactions on*, vol. 8, no. 6, pp. 1003–1009, 2000.

[11] S.-H. Kim, T.-S. Park, J.-Y. Yoo, and G.-T. Park, "Speed-sensorless vector control of an induction motor using neural network speed estimation," *Industrial Electronics, IEEE Transactions on*, vol. 48, no. 3, pp. 609–614, 2001.

[12] R. Marino and G. Santosuosso, "Nonlinear observers of time derivatives from noisy measurements of periodic signals," in *Nonlinear control in the year 2000 volume 2*, ser. Lecture Notes in Control and Information Sciences, A. Isidori, F. Lamnabhi-Lagarrigue, and W. Respondek, Eds. Springer Berlin / Heidelberg, 2001, vol. 259, pp. 123–135, 10.1007/BFb0110296.

[13] A. Tilli and M. Montanari, "A low-noise estimator of angular speed and acceleration from shaft encoder measurements," *AUTOMATIKA: Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 42, no. 3-4, pp. 169–176, 2001.

[14] L. Kovudhikulrungsri and T. Koseki, "Precise speed estimation from a low-resolution encoder by dual-sampling-rate observer," *IEEE/ASME Trans Mechatronics*, vol. 11, no. 6, pp. 661–670, 2006.

[15] Y. Su, C. Zheng, P. Mueller, and B. Duan, "A simple improved velocity estimation for low-speed regions based on position measurements only," *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 937 –942, sep. 2006.

[16] M. Barut, S. Bogosyan, and M. Gokasan, "Speed-sensorless estimation for induction motors using extended kalman filters," *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 1, pp. 272–280, 2007.

[17] L. Bascetta, G. Magnani, and P. Rocco, "Velocity estimation: Assessing the performance of non-model-based techniques," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 424 –433, mar. 2009.

[18] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D*, vol. 216, pp. 269–281, 2006.

[19] J. Buchli, L. Righetti, and A. J. Ijspeert, "Frequency analysis with coupled nonlinear oscillators," *Physica D*, vol. 237, pp. 1705–1718, 2008.

[20] L. Righetti, J. Buchli, and A. J. Ijspeert, "Adaptive frequency oscillators and applications," *The Open Cybernetics and Systemics Journal*, vol. 3, pp. 64–69, 2009.

[21] J. Buchli, F. Iida, and A. J. Ijspeert, "Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 9–15, 2006, pp. 3903–3909.

[22] R. Ronsse, N. Vitiello, T. Lenzi, J. van den Kieboom, M. C. Carrozza, and A. J. Ijspeert, "Adaptive oscillators with human-in-the-loop: Proof of concept for assistance and rehabilitation," in *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, 2010, pp. 668–674.

[23] ——, "Human-robot synchrony: Flexible assistance using adaptive oscillators," *IEEE Trans Biomed Eng*, vol. 58, no. 4, pp. 1001–1012, 2011.

[24] R. Ronsse, B. Koopman, N. Vitiello, T. Lenzi, S. M. M. De Rossi, J. van den Kieboom, E. van Asseldonk, M. C. Carrozza, H. van der Kooij, and A. J. Ijspeert, "Oscillator-based walking assistance: A model-free approach," in *Proc. IEEE Int Rehabilitation Robotics (ICORR) Conf*, 2011, pp. 1–6.

[25] M. D. Rinderknecht, F. A. Delaloye, A. Crespi, R. Ronsse, and A. J. Ijspeert, "Assistance using adaptive oscillators: Robustness to errors in the identification of the limb parameters," in *Proc. IEEE Int Rehabilitation Robotics (ICORR) Conf*, 2011, pp. 1–6.

[26] R. Ronsse, T. Lenzi, N. Vitiello, B. Koopman, E. van Asseldonk, S. M. M. De Rossi, J. van den Kieboom, H. van der Kooij, M. C. Carrozza, and A. J. Ijspeert, "Oscillator-based assistance of cyclical movements: model-based and model-free approaches." *Med Biol Eng Comput*, vol. 49, no. 10, pp. 1173–1185, Oct 2011.

[27] R. Ronsse, J. van den Kieboom, and A. J. Ijspeert, "Automatic resonance tuning and feedforward learning of biped walking using adaptive oscillators," in *Multibody Dynamics 2011, ECCOMAS Thematic Conference*, J.-C. Samin and P. Fisette, Eds., Brussels, Belgium, July 2011.

[28] W. van Dijk, B. Koopman, R. Ronsse, and H. van der Kooij, "Feed-forward support of human walking," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS and EMBS International Conference on*, 2012, pp. 1955–1960.

[29] R. Ronsse, S. M. M. De Rossi, N. Vitiello, T. Lenzi, B. Koopman, H. van der Kooij, M. C. Carrozza, and A. J. Ijspeert, "Real-time estimate of period derivatives using adaptive oscillators: Application to impedance-based walking assistance," in *Intelligent Robots and Systems, 2012. (IROS 2012). 2012 IEEE/RSJ International Conference on*, 2012.

[30] J. Corres and P. Gil, "Instantaneous speed and disturbance torque observer using nonlinearity cancellation of shaft encoder," in *Proc. IEEE 33rd Annual Power Electronics Specialists Conf. pesc 02*, vol. 2, 2002, pp. 540–545.

[31] L. Righetti and A. J. Ijspeert, "Programmable central pattern generators: an application to biped locomotion control," in *Proc. IEEE International Conference on Robotics and Automation ICRA 2006*, May 15–19, 2006, pp. 1585–1590.

[32] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Autonomous Robots*, vol. 27, pp. 3–23, 2009.

[33] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Comput*, vol. 10, no. 8, pp. 2047–2084, Nov 1998.

[34] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, ser. The MIT Press Signal Processing, Optimization, and Control Series, A. S. Willsky, Ed. Cambridge, Massachusetts: The MIT Press, 1983.

[35] E. H. F. van Asseldonk, J. F. Veneman, R. Ekkelenkamp, J. H. Buurke, F. C. T. van der Helm, and H. van der Kooij, "The effects on kinematics and muscle activity of walking in a robotic gait trainer during zero-force control," *IEEE Trans Neural Syst Rehabil Eng*, vol. 16, no. 4, pp. 360–370, 2008.

[36] H. Vallery, J. Veneman, E. van Asseldonk, R. Ekkelenkamp, M. Buss, and H. van Der Kooij, "Compliant actuation of rehabilitation robots," *IEEE Robot Autom Mag*, vol. 15, no. 3, pp. 60–69, Sep. 2008.

[37] S. Pigg and M. Bodson, "Adaptive algorithms for the rejection of sinusoidal disturbances acting on unknown plants," *IEEE Trans Control Syst Technol*, vol. 18, no. 4, pp. 822–836, 2010.

[38] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[39] K. J. Åström and R. M. Murray, *Feedback Systems. An Introduction for Scientists and Engineers.*, see www.cds.caltech.edu/~murray/books/AM05/wiki/index.php, Ed. Princeton University Press, 2008.