"Automatic Resonance Tuning and Feedforward Learning of Biped Walking using Adaptive Oscillators"

Ronsse, Renaud ; van den Kieboom, Jesse ; Ijspeert, Auke Jan

Abstract
In this contribution, we propose a novel approach to achieve two fundamental features of adaptive control in rhythmic movements: The automatic locking of the controller to the system's resonance frequency, and the progressive transfer from feedback to feedforward control. This is realized by using adaptive oscillators, i.e. mathematical tools capable of synchronizing to a periodic input and learning this input’s main features (frequency, amplitude). In particular, these concepts are illustrated here on a simulated walking task. Automatic resonance tuning emerges from the interaction between the plant (the biped walker), the controller, and the movement phase directly estimated by the adaptive oscillator. Progressive transfer to feedforward control is realized by learning the control pattern (initially provided by the feedback controller) with local filters. In conclusion, this paper illustrates to potential relevance of using adaptive oscillators as central elements in adaptive c...

Document type : Communication à un colloque (Conference Paper)

Référence bibliographique
AUTOMATIC RESONANCE TUNING AND FEEDFORWARD LEARNING OF BIPED WALKING USING ADAPTIVE OSCILLATORS

Renaud Ronsse*,†, Jesse van den Kieboom*, Auke Jan Ijspeert*

* Biorobotics Laboratory – Institute of Bioengineering
École Polytechnique Fédérale de Lausanne, Station 14, 1015 Lausanne, Switzerland
e-mail: jesse.vandenkieboom@epfl.ch.auke.ijspeert@epfl.ch

†Center for Research in Mechatronics – Institute of Mechanics, Materials, and Civil Engineering
Université catholique de Louvain, Place du Levant 2, 1348 Louvain-la-Neuve, Belgium
e-mail: renaud.ronsse@uclouvain.be

Keywords: Adaptive control, Adaptive oscillator, Biped walking, Compliance, Feedforward, Resonance tuning.

Abstract. In this contribution, we propose a novel approach to achieve two fundamental features of adaptive control in rhythmic movements: The automatic locking of the controller to the system’s resonance frequency, and the progressive transfer from feedback to feedforward control. This is realized by using adaptive oscillators, i.e. mathematical tools capable of synchronizing to a periodic input and learning this input’s main features (frequency, amplitude). In particular, these concepts are illustrated here on a simulated walking task. Automatic resonance tuning emerges from the interaction between the plant (the biped walker), the controller, and the movement phase directly estimated by the adaptive oscillator. Progressive transfer to feedforward control is realized by learning the control pattern (initially provided by the feedback controller) with local filters. In conclusion, this paper illustrates to potential relevance of using adaptive oscillators as central elements in adaptive control of rhythmic tasks.
1 Introduction

Most of modern robots are designed to be autonomous and to evolve in a human-friendly environment [15]. These requirements have significant consequences on the design features: These robots have to achieve high energy efficiency, to maximize their operating time, and compliance, to react to the “perturbations” caused by the environment (including the surrounding humans) in a graceful manner.

In the present paper, we propose a new controller for a simulated biped walker, that automatically tunes its parameters to optimize energy efficiency and compliance. Energy efficiency is achieved by resonance tuning, i.e. by reaching the frequency at which the transfer between kinetic and potential energy is optimal. Resonance tuning has been previously emphasized as a key feature in robotics [1] and human modeling [17, 18, 19]. Enhanced compliance is achieved through a principle that is common to a lot of applications in adaptive control [14]: As the task execution progresses, the control transfers from feedback to a mix of feedback and feedforward. Interestingly, this permits to reduce the feedback gains once the system dynamics are (partly) compensated by the feedforward controller. As such, the controlled plant becomes much more compliant. In overall, this principle has also been suggested in the computational motor control framework, as an account to explain how the brain might “learn” to control the body (and the environment) [20].

The central element of our controller is an adaptive (frequency) oscillator [2, 10], which is a tool used in various applications [11, 13]. It consists of an oscillator whose state space is extended to synchronize to a (quasi-)periodic input signal, and to learn the features (frequency, phase, amplitude, ...) of this input. It can be viewed as a dynamical system providing a real-time Fourier decomposition of the input signal [2]. In particular, we illustrate these concepts on a simulated walking biped (Fig.1).

![Figure 1: Snapshot of the biped model.](image)

2 Model and controller

The simulated walking biped used in this paper is shown in Fig.1. The morphology and segment masses were set to match those of a standard human. Moreover, the impedance (i.e. stiffness and damping) of the hips, knees, and ankles, were taken from the literature [7]. Sim-
ulations were carried out using Webots (Cyberbotics Ltd., Switzerland) [9], a physics-based simulation environment, with a simulation sampling rate of 2ms. The walker was constrained to move only in the sagittal plane (2D) and its torso was constrained to stay vertical, preventing falling backward/forward.

In the default mode, this walker was controlled only using feedback (Fig.2A with $s_{FF} = off$), and with an external clock providing the cycle phase ($s_\phi = 1, \phi = \text{mod}(\omega_{ref}t, 2\pi)$, where $\omega_{ref}$ is the desired movement frequency). This phase was fed into a lookup table to get the corresponding reference positions of the left and right hip, knee and ankle joints, all grouped into the vector $\Theta_{ref}$. The feedback controller was a simple proportional controller, providing torques that depended on the difference between the reference and actual trajectories:

$$\tau_{FB} = K_p(\Theta_{ref} - \Theta_{act})$$

The gains were set to 2000Nm/rad for the six joints, in order to achieve satisfying trajectory tracking during walking. This makes the controller very stiff and therefore non-compliant. The lookup table with the reference trajectories was constructed from human data collected at the University of Twente\(^1\) on a single human subject with a morphology close to our simulated artifact. These data are shown by solid lines in Fig.2B,C,D, for the left leg.

![Figure 2: A: Block diagram of the system (biped+controller) with the feedback controller (central part), the adaptive frequency oscillator achieving resonance tuning (bottom, dark gray), and the feedforward controller (top, light gray). Reference trajectories of left hip (B), knee (C), and ankle (D) joints for two different forward speeds (solid lines, about 1.2m/s; and dotted lines, about 0.8m/s). Data are aligned on the phase of the hip first harmonic (gray in panel B). Legend: time of heel strike (HS) and toe off (TO).](image)

3 Resonance tuning

The walker model exchanges kinetic and potential energy through pendulum-like motion of the legs and impacts with the ground. To achieve automatic resonance tuning, we used a method inspired by [1]. First, the movement instantaneous phase was extracted using an adaptive oscillator (dark gray in Fig.2B), i.e. a modified phase oscillator which can learn the features of the input signal $\theta_{act}$ using an augmented state-space.

\(^1\)Laboratory of Biomechanical Engineering, www.bw.ctw.utwente.nl.
The first equation is thus simply a phase oscillator with an external periodic output $F(t)$:

$$\dot{\phi}(t) = \omega(t) + \nu F(t) \cos \phi(t),$$

(1)

where, $\phi(t)$ is the oscillator phase, $\omega(t)$ its intrinsic frequency, and $\nu$ the learning parameter determining the speed of phase synchronization to the input, i.e. the teaching signal $F(t)$. In order to learn the frequency of the teaching signal $F(t)$, instead of doing mere synchronization only, the oscillator frequency was turned into a new state variable, integrating the phase update:

$$\dot{\omega}(t) = \nu F(t) \cos \phi(t).$$

(2)

As such, Righetti et al. developed an adaptive oscillator, having the capacity to constantly adapt its intrinsic frequency to the teaching signal frequency, and to keep this input frequency in memory, i.e. in the state variable $\omega(t)$. Proof of convergence is provided in [10].

Let us now assume:

(i) that the input signal follows a sinusoidal pattern:

$$\theta(t) = \alpha_{1,in} \sin (\omega_{in} t) + \alpha_{0,in},$$

where $\alpha_{1,in}$, $\omega_{in}$, and $\alpha_{0,in}$ are the amplitude, frequency, and offset of this input, respectively;

(ii) that the adaptive oscillator (1), (2) is synchronized with this input.

As a consequence, an observer of $\theta(t)$ can be obtained by:

$$\hat{\theta}(t) = \alpha_1(t) \sin \phi(t) + \alpha_0(t),$$

(3)

where $\alpha_1(t)$, $\phi(t)$, and $\alpha_0(t)$ are supposed to converge to the corresponding input variables. Righetti et al. [12] showed that this convergence is guaranteed by using the difference between the input $\theta(t)$ and the filtered (or estimated) input $\hat{\theta}(t)$ as teaching signal: $F(t) = \theta(t) - \hat{\theta}(t)$, and by implementing the following integrators for learning the amplitude and offset:

$$\dot{\alpha}_0(t) = \eta F(t), \quad \dot{\alpha}_1(t) = \eta F(t) \sin \phi(t),$$

(4)

where $\eta$ is the integrator gain. Again, (2) and (4) reach steady-state when $F(t) = 0$, i.e. when $\hat{\theta}(t) = \theta(t)$. If $\theta(t)$ is only quasi-sinusoidal, i.e. if $\alpha_{1,in}$, $\omega_{in}$, and $\alpha_{0,in}$ slowly vary in time, $\hat{\theta}(t)$ will be a low-pass filtered version of $\theta(t)$, but, importantly, both will still be phase-synchronized on average. This is a critical difference between this approach and classical low-pass filtering, which unavoidably introduces delay.

In sum, the adaptive oscillator presented above provides a continuous estimate of the input signal features: phase(1), frequency (2), amplitude, and offset (4). It further provides a filtered version of the input (3) which has, on average, the same phase as the input (see the example in Fig.3).

Importantly, if the input signal is periodic but non-sinusoidal, Righetti et al. [12] proposed to extend the method explained above by putting several oscillators in parallel (see Fig.4). As such, each of these oscillators should learn one frequency component of the input signal, providing therefore a kind of real-time Fourier decomposition. We slightly adapted the equations of [12]
by assuming that the input signal was periodic. Therefore, only the main frequency had to be learned, the others being multiples of it. Concretely, (1), (2), and (4) were changed to:

\[
\begin{align*}
\dot{\phi}_i(t) &= i\omega(t) + \nu F(t) \cos \phi_i(t), \\
\dot{\omega}(t) &= \nu F(t) \cos \phi_1(t), \\
\dot{\alpha}_i(t) &= \eta F(t) \sin \phi_i(t),
\end{align*}
\]

(5)

with \( F(t) = \theta(t) - \hat{\theta}(t) \), \( \hat{\theta}(t) = \sum_{i=0}^{K} \alpha_i(t) \sin \phi_i(t) \), and \( i \in [0, K] \) are the \( K + 1 \) parallel oscillators. Note that, in (5), the 0th oscillator is still a simple integrator, learning the input offset, with \( \phi_0(t) = \phi_0(0) = \pi/2 \).

Figure 4: Online learning of a periodic but non-sinusoidal input signal \( \theta(t) \). The block is a pool of adaptive oscillators (1), (4), decomposing the input into a real-time Fourier series. The phase of the main harmonic \( \phi_1(t) \) is extracted from the first oscillator.

To get an estimate of the movement phase \( \hat{\phi} \), we plugged the position of both hips \( \Theta_{\text{act,hip}} \) as input to two adaptive oscillators similar to Fig.4. Since both hips are supposed to move along the same envelope, we forced to two oscillators to reach consensus on their estimated frequency \( \omega \) and amplitudes \( \alpha_i \) through averaging. Then, the estimated phase \( \hat{\phi} \) was obtained from the phase of the first oscillator \( (\phi_1(t)) \) of the left hip, arbitrarily.

By replacing the external phase by the estimated one \( \hat{\phi} \) (i.e. \( s_\phi = 2 \) in Fig.2B), Buchli et al. [1] showed, and demonstrated on a simple example, that the system naturally converges to its nearest resonance frequency.
To illustrate that this mechanism was effective in our biped walker as well, we performed the following simulation: Initially, the system was forced to follow the reference trajectory at a frequency $\omega_{\text{ref}}$ which is above resonance. After 30s, the external clock was replaced by the phase estimated from the actual trajectory of the hips $\hat{\phi}$ ($\omega_{\phi} = 1 \rightarrow 2$), such that the system converged to the resonance frequency. Two minutes later, the lookup table was changed to follow the kinematics corresponding to a slower forward speed (dotted lines in Fig.2). 2.5 minutes later, the lookup table was changed back to the one corresponding to the fastest forward speed (solid lines in Fig.2). The oscillator parameters were equal to $\nu = 15$, $\eta = 0.5$, and $K = 3$.

The frequency estimated by the adaptive oscillator, and therefore determining the estimated phase $\hat{\phi}$, is shown in Fig.5\(^2\). Although both sets of kinematic data are not so different from each other, the system converged to two very different resonance frequencies. Interestingly, the resonance frequency corresponding to the fastest forward speed was above the other one, in agreement with the data.

### 4 Feedforward learning

Automatic learning of a feedforward controller was done using a method inspired by Dynamical Motor Primitives for imitation learning [4, 5]. This approach can be formulated as a supervised learning problem:

$$
\tau_{\text{FF}}(t) = \frac{\sum \Psi_i(\phi_1(t))w_i}{\sum \Psi_i(\phi_1(t))},
$$

where $\tau_{\text{FF}}(t)$ is the feedforward torque acting at one joint\(^3\), and

$$
\Psi_i(\phi(t)) = \exp \left( h(\cos (\phi(t) - c_i) - 1) \right)
$$

is a set of $N$ Gaussian-like kernel functions, acting like local filters. The parameter $h$ determines their width, and $c_i$ their center (equally spaced between 0 and $2\pi$ in $N$ steps). This algorithm

\(^2\)A movie of this experiment is available at biorob2.epfl.ch/utils/movieplayer.php?id=169.

\(^3\)Actually, this should be a vector with six elements, each corresponding to one of the actuated joints. For simplicity, we will make all derivations for a single joint and assume the system is copied five times for the other joints.
then constructs a series of local mappings of the input as a function of the phase $\phi_1(t)$, and an estimate of the input from a weighted sum of these mappings.

Following [4, 16], an on-line version of this learning process can be implemented using incremental regression, which is done with the use of recursive least squares with a forgetting factor of $\lambda$, to determine the weights $w_i$. Given the input $\tau[k]$ (where $\tau[k] = \tau(t_k)$ at $t_k$, the time of the $k^{th}$ sample), $w_i$ is updated by:

$$
w_i[k+1] = w_i[k] + k_{\text{learn}} \Psi_i[k] P_i[k+1] (\tau[k] - w_i[k]),
$$

$$
P_i[k+1] = \frac{1}{\lambda} \left( P_i[k] - \frac{P_i[k]^2}{\Psi_i[k] + P_i[k]} \right),
$$

where $k_{\text{learn}}$ is the global learning gain (somehow redundant with the forgetting factor $\lambda$), and $P$ is the inverse covariance matrix [8]. If $\lambda < 1$, the regression gives more weight to recent data.

As shown in Fig.2A, the principle is the following: The filters construct the feedforward signal to progressively match with the applied torque, using the phase $\phi$ as input. The phase can be provided either by the external clock or by the adaptive oscillator, as previously.

We performed another simulation using the same walker model. As previously, the walker was first started with the external clock ($s_{\phi} = 1$), then switched to the phase provided by the adaptive oscillator ($s_{\phi} = 2$, after 10s in this case), to achieve resonance tuning. During the next 30s, the feedforward controller was trained ($k_{\text{learn}} = 1$) but not injected into the system ($s_{FF} = \text{off}$). Then, simultaneously, the feedforward torque was switched on ($s_{FF} = \text{on}$), the feedback gains were divided by 10 ($K_p = 200\,\text{Nm/rad}$ for all joints), and the learning was stopped ($k_{\text{learn}} = 0$). The simulation ran for another 60s in this condition before finishing. Other parameters were equal to $\lambda = 0.999$, $N = 90$, and $h = 5N$.

![Figure 6: Torques at the left hip joint during feedforward learning ($10 < t \leq 40$) and feedforward activation ($t > 40$). The gray curve shows the feedback torque $\tau_{FB}$, and the black curve shows the feedforward torque $\tau_{FF}$ (dotted during the learning phase, i.e. when not applied).](image)

Results are shown in Fig.6 for the left hip torques$^4$. The figure shows that the kernel filter correctly learned the feedback torque during the initial part of the simulation. Then, once the feedback was switched on, the gains can be divided by one order of magnitude while the gait

$^4$A movie of this experiment is available at [biorob2.epfl.ch/utils/movieplayer.php?id=170](http://biorob2.epfl.ch/utils/movieplayer.php?id=170).
remained perfectly stable, giving rise to a much smaller feedback torque and a more compliant controller. Similar results can be observed for the five other joints.

A third simulation was finally performed trying to combine all the ingredients mentioned above, i.e. resonance tuning, progressive switching from feedback to feedforward, and continuous adaptation of the feedforward controller (this last point was missing in the second simulation, since the learning had to be switched off to maintain stability). In the third simulation, we however managed to combine all three elements by allowing different adaptation laws for the different joints. The simulation started with a scenario very similar to the second simulation, i.e. initialization for the first 10 seconds, and convergence to the resonance frequency with feedback only for a few more seconds (20 in this case), with learning of the feedforward patterns \( k_{\text{learn}} = 1 \). Then, at \( t = 30 \) s, the following changes were applied to the joints:

- At the hip, the feedforward torque was switched on \( (s_{\text{FF}} = \text{on}) \), the feedback gains were divided by 100 (i.e. two orders of magnitude, \( K_p = 20 \) Nm/rad), and the learning was maintained, yet with a smaller rate \( k_{\text{learn}} = 0.2 \).

- At the knee, the feedforward torque was switched on \( (s_{\text{FF}} = \text{on}) \), the feedback gains were not changed \( (K_p = 2000 \) Nm/rad), and the learning was maintained, yet with a much smaller rate \( k_{\text{learn}} = 0.02 \).

- At the ankle, nothing was changed, i.e. no feedforward torque, and feedback gains equal to \( K_p = 2000 \) Nm/rad.

Results are shown in Fig. 7. The passive torques (i.e. provided by the joint passive impedance \([7]\)) did not change whatever the mode of control (feedback/feedforward), revealing the the kinematic pattern stayed very similar. However, when the feedforward controller was switched on (i.e. at \( t = 30 \) s), a strong shift was recorded in the hip joint from purely feedback control to almost purely feedforward control (indeed, the feedback gain was divided by 100). This effect was also visible in the knee, yet to a smaller extent, and not in the ankle, since no switching mechanism was implemented in that joint.

5 Conclusion

This contribution proposed a new controller for achieving automatic resonance tuning and feedforward learning in the context of rhythmic movements. In particular, we illustrated this approach on a walker model. This controller was based on an adaptive oscillator, providing an on-line estimate of the movement phase which was used both to induce automatic resonance tuning, and to transfer the control from feedback to feedforward by learning the torque patterns with local filters.

Two interesting findings were observed in our simulations:

- The observed change in resonance frequency might suggest that humans adapt the joint kinematics to tune their resonance frequency according to the desired forward speed. This is coherent with the idea that humans perform rhythmic movements at the limb resonance frequency, and potentially adapt the limb dynamics to tune this frequency to the desired movement frequency \([17, 18, 19]\). The new finding here is that also the movement kinematics could be adapted accordingly.

- To achieve both resonance tuning and constant adaptation of the feedforward controller, it was needed to provide different adaptation mechanisms to the different joints. In particular, the hip were eventually controlled almost only with feedfoward, while the ankles
Figure 7: Mean of the absolute torques at the left joints (hip, knee, ankle) during feedforward learning \((0 < t \leq 30)\) and feedforward activation \((t > 30)\). Absolute torques are averaged over each cycles. The black curve shows the feedback torque \(\tau_{FB}\), and the gray curve shows the feedforward torque \(\tau_{FF}\). The dotted gray curve is the torque generated by the joint impedance (i.e. passive torque).

remained feedback-controlled. The knees lied in between. This picture is quite coherent with the proximo-distal gradient identified for birds: Running experiments led to the conclusion that proximal leg muscles are mainly controlled by central inputs (i.e. feedforward) while distal leg muscles are governed by reflex inputs due to higher proprioceptive feedback gains [3].

This paper contributes to demonstrate the potential interest of adaptive oscillators as central elements in adaptive control of rhythmic tasks. However, deeper investigations to support this claim would necessitate the development of a solid theoretical framework, which is somehow still missing for non-linear oscillators [6]. In this particular example, future work will be directed to model improvements to induce more efficient and realistic walking, for example by relaxing the walker constraints, and removing the lookup table to implement a biologically inspired controller.
Acknowledgments

This work was supported by the EU within the EVRYON Collaborative Project STREP (FP7-ICT-2007-3-231451). We are grateful to Bram Koopman, Wietse van Dijk, and Herman van der Kooij for sharing their walking data with us.

REFERENCES


