Rich Periodic Motor Skills on Humanoid Robots: Riding the Pedal Racer

Andrej Gams[1,2], Jesse van den Kieboom[1], Massimo Vespignani[1], Luc Guyot[1], Aleš Ude[2] and Auke Ijspeert[1]

Abstract— Just as their discrete counterparts, periodic or rhythmic dynamic motion primitives allow easily modulated and robust motion generation, but for periodic tasks. In this paper we present an approach for modulating periodic dynamic movement primitives based on force feedback, allowing for rich motor behavior and skills. We propose and evaluate the combination of feedback and learned feed-forward terms to fully adapt the motions of a robot in order to achieve a desired force interaction with the environment. For the learning we employ the notion of repetitive control, which can effectively minimize the error of behavior towards a given reference. To demonstrate the approach, we show results of simulated and real world experiments on a compliant humanoid robot COMAN. We show the initial results of utilizing the approach to control a pedal-racer, a demanding balance toy best described as a hybrid between a skateboard and a bicycle.

I. INTRODUCTION

Periodic motions, often termed also as rhythmic motions, appear in many biological systems and range from manipulatory tasks to locomotion. As they are periodic, specific control approaches can be applied. The latter is not only observable in engineering approaches, but also in biological systems. The term *central pattern generator* (CPG) describes neural circuits found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs [1]. CPGs represent fundamental building blocks for the locomotor neural circuits in animals.

The difference of controlling periodic or discrete motions was also investigated in humans. Schaal et. al. [2] compared fMRI measurements during discrete and rhythmic tasks, which showed that higher cortical centers are used for discrete motions and much less for rhythmic tasks, indicating that rhythmic tasks require separate theoretical treatment. Computational models for rhythmic and discrete models motions were also investigated by Ronsse et al. [3] and Degallier & Ijspeert [4].

Many researchers propose that motor control is based on the combination of motor primitives, i.e., that complex movements are generated by combining a finite set of simpler elementary movements [1], applicable also to locomotion [5]. This notion has been extensively applied to control of robotic

tasks; notably with *dynamic movement primitives* (DMPs), first introduced by Ijspeert et al. [6]. DMPs represent one of the approaches for encoding and generating trajectories; other approaches include, for example, Gaussian Mixture Regression [7] and Mixture Models [8], splines and wavelets [9], etc.

While DMPs were originally designed for discrete, finite tasks, they can be effectively applied to periodic tasks [9]–[12]. Just as their discrete counterparts, they use a set of differential equations to compactly represent control policies and at the same time allow adaptation by modifying only a few parameters [12]. The latter can be exploited in several ways, for example for temporal modulation, where the frequency of motion, which determines the behavior of the whole system, is modulated by a single parameter. The frequency of motion can either be set in advance, or can be modified online to autonomously adapt to external, driven systems, for example by the use of adaptive frequency oscillators [13]. Gams et al. [9] have shown how periodic DMPs can be combined with a pool of adaptive oscillators to facilitate a system with combined frequency and waveform learning. On the other hand, Petrič et al. [11] have shown how a single adaptive frequency oscillator and an adaptive Fourier series can be used in combination with periodic DMPs for complex synchronization tasks. Periodic DMPs allow also spatial modulation and generalization from a library of recorded motions [14].

Any kind of periodic interaction task, even simply maintaining contact with a (periodically) moving object, for example the end of a two-person cross-cut saw [15], requires either extremely precise predefined trajectories, or adaptation to specific tasks. Similar applies also to rotating bicycle pedals or periodically applying force on an a surface. If that task is to be executed in an unstructured environment of human daily life, predefining the correct trajectories is practically impossible. Generalization, as one of the possibilities of adapting to the environment, can provide reasonable solutions [14].

The problem we are tackling in this paper is in adapting such periodic trajectories to achieve desired force behavior and rich motors skills, i.e., in the sense of adaptability to different conditions. The novelty of the approach is in enabling online spatial modulation of periodic DMPs by coupling them to the environment using force feedback. Similarly to external limit modulation [9], our approach utilizes coupling at the velocity level of a DMP, but with actual, measured force instead of a virtual one. In order to achieve desired interaction with either the environment or another robot we

combine feedback with a feed-forward term, which we learn in a few periods of the motion. The learning is based on repetitive control (RC), which can be used to achieve near perfect tracking of a given periodic reference [16], [17]. Our proposed approach allows adaptation of periodic trajectories to the state of the environment in order to achieve desired interaction. We apply the approach to the task of operating the pedal racer, a demanding balance toy best described as a hybrid between a skateboard and a bicycle, with a humanoid robot. The robot and the device are presented in Fig. 8. We test our approach both for operating the device with arms and for riding it. Riding the pedal racer is very complex from the balance point of view. We applied a center of mass controller to maintain the balance and used our proposed approach to modulate the trajectories of motion.

Force feedback was previously used with periodic DMPs in a table wiping task, where complete trajectories were learned using regression techniques [18]. On the other hand, discrete DMPs were previously modulated based on force feedback using iterative learning control (ILC) [19]. While ILC requires initial conditions reset after every repetition, RC can handle the change in the initial conditions – at the start of every period. Furthermore, the approach of learning the correct motion comes more naturally for periodic motions, where the motion is by itself repeated over periods.

Other DMP coupling approaches used acceleration level feedback on discrete DMPs in combination with learning [20], or a feedback controller [21]. A combination of feed-back and learned feed-forward terms in rhythmic DMPs to stabilize a two-link robotic arm was presented in [22].

In the rest of this paper we first give a summary on periodic DMPs and present the modulation approach in Section II. We argument and present the Repetitive Learning algorithm applied for the learning in Section III. We evaluate the approach with both simulations and real world experiments in Section IV. In Section V we show how the approach can be used for controlling a pedal-racer. Conclusions and a brief discussion follow in Section VI.

II. MODULATING PERIODIC MOVEMENT PRIMITIVES

In the following we provide the basic motion representations and the proposed approach to solving the problem of adapting periodic motions with force feedback.

A. Periodic Motion Primitives

Periodic DMPs have been studied for various tasks [12]. We provide only the basic information, based on the formulation in [14]. For a single degree of freedom (DOF), in our case one of the external task-space coordinates and denoted by y, a periodic DMP is defined by the following system of nonlinear differential equations

$$\dot{z} = \Omega\left(\alpha_z\left(\beta_z\left(g-y\right)-z\right)+f(\phi)\right), \quad (1)$$
$$\dot{y} = \Omega z. \quad (2)$$

$f(\phi)$ is defined as a linear combination of periodic radial basis functions $\Gamma_i(\phi)$

$$f(\phi) = \frac{\sum_{i=1}^{N} w_i \Gamma_i(\phi)}{\sum_{i=1}^{N} \Gamma_i(\phi)} r, \quad (3)$$

$$\Gamma_i(\phi) = \exp\left(h_i\left(\cos\left(\phi-c_i\right)-1\right)\right), \quad (4)$$

where r is the amplitude control paramter, $h_i > 0$ are their widths and c_i are equally spaced between 0 and 2π in N steps. The phase variable ϕ is introduced to avoid explicit dependency on time. The phase is assumed to increase with constant rate

$$\dot{\phi} = \Omega. \quad (5)$$

The parameter Ω denotes the frequency. If parameters $\alpha_z,\ \beta_z,\ > 0$ and $\alpha_z = 4\beta_z$, the system $(1) - (2)$ converges in a critically damped manner to the goal g.

The weights w_i, $i = 1, ..., N$ and N the number of kernel functions, given in the vector $\mathbf{w}$, define the shape of the encoded trajectory. [6] and [14] describe the learning of the weight vector. Multiple DOFs are realized by maintaining separate sets of eqs. $(1) - (2)$, while a single canonical system given by (5) is used to synchronize them by providing a common phase variable.

B. Coupling of periodic DMPs

An example of coupling to achieve spatial modulation of DMPs is to include a simple virtual repulsive force to avoid moving beyond a given limit [9]. In this paper we propose a similar approach, but by including the real, measured force. This is implemented by modifying (2) into

$$\dot{y} = \Omega\left(z + C\right). \quad (6)$$

Here C represents a coupling term, composed of feedback (c_{fb}) and feed-forward (c_{ff}) terms

$$C = c_{\text{fb}} + c_{\text{ff}}. \quad (7)$$

The feedback term is defined by

$$c_{\text{fb}} = k\left(F_{\text{des}} - F_{\text{act}}\right), \quad (8)$$

where k is a positive constant, F_{des} is the desired force trajectory and F_{act} is the real, measured force. The desired force can be any force trajectory. For the case of riding the pedal racer (in simulation) we used a measured force profile (also obtained in simulation), as the reference.

A virtual force can also be used, defined for example by a virtual spring between the position of the robot y_{r1} and an arbitrary object o

$$F_{\text{act,v}} = k_{\text{env}}(o - y_{r1}). \quad (9)$$

with k_{env} defining the stiffness of the virtual spring. The position of another robot can be used for an object, coupling two robots. The virtual force between the positions of two robots (y_{r1} and y_{r2}), with d the desired distance is then

$$F_{\text{act,v}} = k_{\text{env}}(d - (y_{r1} - y_{r2})). \quad (10)$$

The feedback term acts as a P-controller, with the gain k_{env} defining the behavior. If it is high, the robot essentially bounces of on impact, while a low k results in slow adaptation of the robot trajectory.

Using the feedback coupling term alone resembles the approach in [21]. It is important to note that we couple the DMP at the velocity level in (2), and not at the acceleration level, i.e., not with providing a coupling term to (1) after $f(\phi)$. This has several advantages. First, this makes feedback modulation highly reactive to the actual force, and low gains can be used. The most important feature of velocity level coupling is in producing lower oscillations in the direction of the force. For details on advantages of velocity-level coupling and stability of coupled DMPs see [19].

If the conditions of the periodic task do not change during the execution, the error of using only a feedback term will not change through the periods of motion, but will continue to repeat. We propose adding a feed-forward term to cancel the error ($e = F_{\mathrm{des}} - F_{\mathrm{act}}$) of tracking the predefined force trajectory. The feed-forward term has to adapt to the specific task by a learning method. We propose using repetitive control (RC), which can achieve near perfect tracking of a given reference if the conditions of the task remain relatively the same [16], [17]. The approach is described in the next section.

To make c_{ff} also a function of the phase, we encode the feed-forward term in the form of weighted kernel functions Γ_i, similar to $f(\phi)$ in (4)

$$c_{\mathrm{ff}}(\phi) = \frac{\sum_{i=1}^{N} v_i \Gamma_i(\phi)}{\sum_{i=1}^{N} \Gamma_i(\phi)}. \tag{11}$$

The target for fitting is the output of the repetitive control (see Fig. 1). We use iterative locally weighted regression (ILWR) with a forgetting factor $\lambda = 0.99$ to learn the weights v_i. See [9] for details on ILWR. The encoding acts as a filter, canceling out some of the noise of the measurements. Additionally, once we stop learning, the same phase drives both the DMP and the coupling, allowing for temporal modulations.

III. REPETITIVE CONTROL

Repetitive control is a control method that uses previous experience to design a new control signal and is thus categorized as *learning-type control* [16]. It is mainly used in continuous processes for tracking or rejecting periodic exogenous signals and in most cases, the period of the exogenous signal is known. In this paper we use what is known as a *plug-in type repetitive control*, which incorporates both feed-forward and feedback terms. The structure of such controllers is presented in the top left part of Fig. 1, denoted by the solid border. Figure 1 also shows how RC is combined with a DMP, and how a virtual force can be utilized, shown bottom right and denoted by the dashed border.

Note that the schematic is written for a discrete-time system, where z^{-1} represents a delay of one time sample and z^{-T} of one period. In this paper we also only tackle constant frequencies.
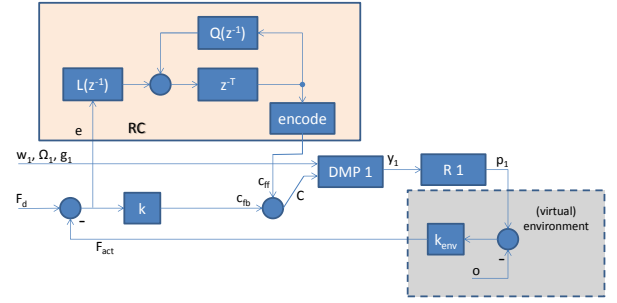

Fig. 1. Schematic of the RC in combination with the DMP and the robot (R1). Contact with the environment provides the force F_{act}. The force can be the real, measured force, or a virtual force calculated with (9).

The feed-forward term is composed of two parts, known as the Q and the L filter, which are defined as [16]

$$L(z^{-1}) = k_{RC}, \tag{12}$$
$$Q(z^{-1}) = \epsilon(\alpha_1 z + \alpha_0 + \alpha_1 z^{-1}), \tag{13}$$
$$2\alpha_1 + \alpha_0 = 1. \tag{14}$$

Here $\alpha_i > 0$, $1 \geq \epsilon > 0$ and $k_{RC} > 0$. Effectively, the feed-forward term uses the error signal from the previous period (see z^{-T} in Fig. 1) and applies it to the control signal. The values of k_{RC}, just as the value of k in the feedback term, has to be determined before hand. In our experiments we determined the values empirically.

The robustness of the system is determined through the gain of the Q-filter. Stability of RC is a wide and complex topic. As discussed in a survey by Cuiyan et al. [17], different types of RC control require different design and synthesis methods. In general, the selection of control parameters involves a tradeoff between steady state accuracy, robustness and transient response of the system [17]. Furthermore, there is also the phenomenon of apparent convergence, where the system apparently converges, but after some time diverges [23]. Multiplying the output of the Q filter with $\epsilon < 1$ will increase the robustness of the system, but also the steady-state error. In our simulated experiments we used $\epsilon = 0.95$ and in our real-word experiments we used $\epsilon = 0.7$.

The contribution of this paper is in showing how we can combine DMPs with their modulation and disturbance rejection properties, and a learning framework of RC. While RC alone could be applied to the problems at hand, the combination with DMPs allows more robust and adaptive behavior and modifications with a small set of parameters.

IV. EVALUATION

We evaluated the algorithm in both simulation and in real-world experiments. In the first experiment we simulated a task where the robot has to maintain contact with a periodically moving object (under external actuation) while applying a constant, predefined referential force. The scheme in Fig. 1 was used for simulated experiments and perfect tracking of the robot was assumed ($R1 = 1$, see Fig. 1). We used (9) to simulate the forces.
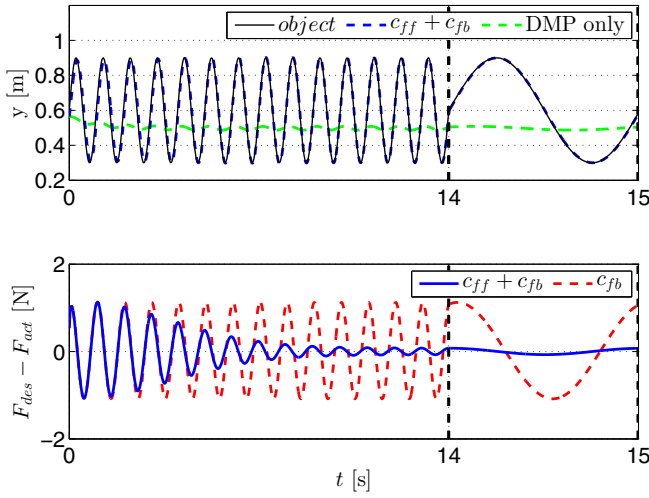
Fig. 2. Simulated results of using the proposed RC learning to reduce the error of force tracking for the task of maintaining contact with a periodically moving object. The trajectory of the object (black), the trajectory with coupling (dashed blue) and the original DMP trajectory (green dashed) in the top plot. The error of force tracking when using the feedback coupling only (red dashed) and when using both feedback and feed-forward couplings (blue). Using both couplings reduces the error about 25 times. The final second of the plot is zoomed in.
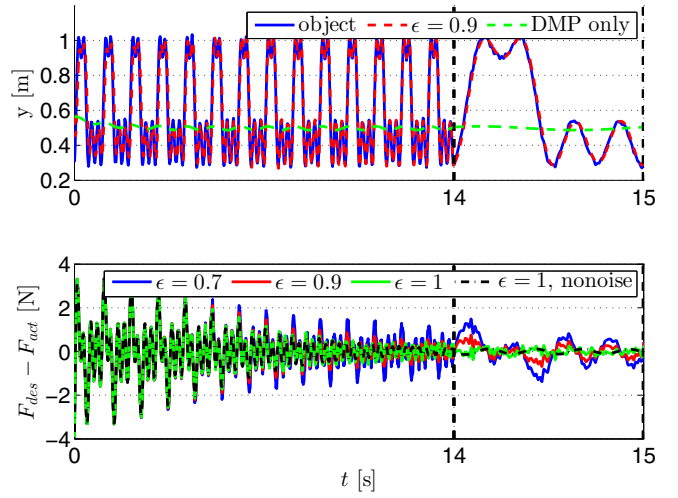

Fig. 3. Simulated results of the effect of ϵ and noise on using the proposed RC learning to reduce the error of force tracking for the task of maintaining contact with a periodically moving object. The trajectory of the object (black), the trajectory with coupling and $\epsilon = 0.9$ (dashed red) and the original DMP trajectory (green dashed) in the top plot. The error of force tracking with different ϵ values in the bottom plot. Steady-state error is lowest when there is no noise and highest $\epsilon = 1$, shown in black dash-dot line. The final second of the plot is zoomed in.

Figure 2 shows the results for the task of maintaining desired contact with an object performing a simple sinusoidal trajectory. In the top plot we can see the original, uncoupled DMP trajectory (green dashed), which starts at a randomly chosen position close to the goal ($g = 0.5$) and close to the trajectory of the periodically moving object (black). Also shown is the trajectory of the DMP with both feedback and feed-forward coupling (dashed blue). The initial guess for the motion (green dashed), which was here predefined, but could have been acquired autonomously through generalization, has the correct frequency but too small amplitude. The bottom plot of Fig. 2 shows the error of the desired force for two scenarios: using only the feedback coupling $C = c_{\text{fb}}$ (red dotted); and both feedback and feed-forward coupling $C = c_{\text{fb}} + c_{\text{ff}}$ (blue solid). We can see that the error is reduced approximately 25 times when also the feed-forward term is used.

The steady-state error depends on the value of ϵ in (13). A higher ϵ, i.e., $\epsilon \approx 1$, reduces the steady state error, but also reduces the robustness of the system. To demonstrate the effect of varying ϵ, Fig. 3 shows the results for the same experiment of maintaining constant contact with a periodically moving object, but with a considerably more complex waveform of object motion. Additionally, the signal of the position of the robot is noisy, with a maximal noise amplitude of 1 cm. Such noise is unrealistically high for the estimation of position through forward kinematics, but could be present if a vision system is used to estimate the position of the object. The top plot of Fig. 3 shows the waveform of the object motion (blue solid), while the bottom plot shows the effect of ϵ on steady-state error. Steady-state error is lowest when there is no noise (black dash-dot) and $\epsilon = 1$.

The results overlap considerably in the transient part, i. e., from the beginning.

Increasing ϵ decreases the robustness of the learning algorithm, which might cause it to diverge, even after apparent convergence. This is not uncommon [23] and different methods of canceling out this phenomena exist; the easiest in reducing ϵ, reducing the gain of the L filter, see (13), or by cutting the learning after some time. The latter involves some heuristics in determining the time to cut the learning; observing that the error remains below a threshold for more than one period can be used as a criterion. Figure 4 shows the occurrence of apparent convergence, where the system still diverges after some time.

A. Bimanual Coupling

We tested the approach in a real-world scenario where we used the arms of a humanoid robot to rotate a pedal racer fixed to a solid structure by its axes. Real measured force was used to modulate the trajectories of motion.

We used the 4 DOF arms of the compliant humanoid robot named COMAN [24]. COMAN approximates the size of a 4
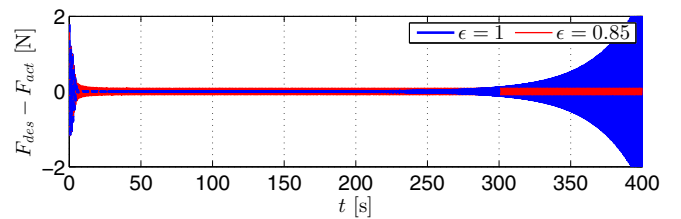

Fig. 4. The phenomena of apparent convergence appears when $\epsilon = 1$ was used, but not for $\epsilon = 0.85$.

Fig. 5. Image sequence showing approximately one period of rotating the vertically mounted pedal racer with the arms of the COMAN robot after the learning, when the contact with the pedals is constantly kept. The attached video shows the complete experiment. The tape on the wheels simulates contact with the ground and keeps them at the same velocity, preventing the device from locking up, which happens if the front and the back wheels do not rotate together.
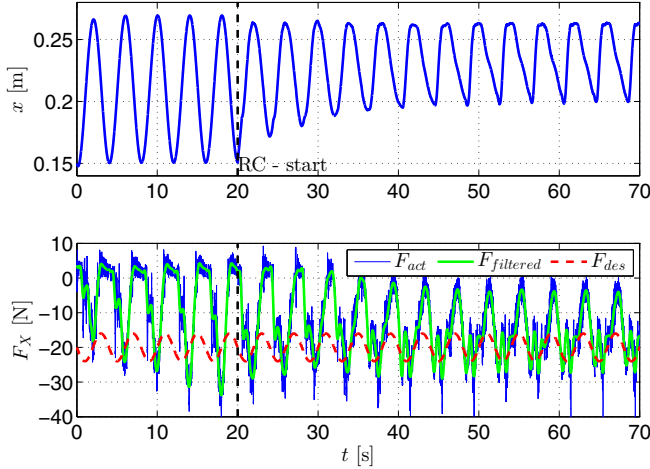

Fig. 6. Results of modifying the arm trajectories for a cooperative bimanual task of rotating the pedal racer with arms. The top plot gives the trajectory of the arm with x denoting the distance from the body to the tip of the arm. The bottom plot gives the estimate of the measured force (blue), filtered force (green) and desired force (red dashed).

year old child, having 945mm from the foot to the center of the neck, and 312mm between the centers of the shoulders. It weights 31.2kg, out of which the legs and the waist module weigh 18.5kg. The complete robot has 23 DOF; each leg has 6: 3 at the hip, 1 at the knee and 2 at the ankle. The trunk is composed of a 3 DOF waist and the body, while each arm has currently 4 DOF, i.e. 3 in the shoulder and 1 in the elbow. Passive compliance based on series elastic actuation (SEA) is added to 14 of the 23 DOF including all flexion/extension DOF of the legs, the flexion/extension of the shoulders and elbows and the shoulder abduction/adduction. The robot is presented in Fig. 8.

The task of the algorithm was to maintain sinusoidal force trajectories on a vertically fixed pedal racer, as presented in Fig. 5. The experiment served to show the applicability of the approach in a real-world cooperative scenario, where the cooperation was between the two robot arms. It also served as a preparation for the task of actually operating the pedal racer by standing on it.

The initial trajectories of motion, encoded by DMPs were accurate for the up-down (z) direction, because the arms could not brace against any part of the pedal racer in

the z direction. Therefore only friction could be used to produce up-down force on the device. The forward-backward (x) motion was a sinusoidal with an estimated amplitude, deliberately set not to maintain contact with the device at all times. The robot was standing straight in front of the pedal racer with no balance control. Therefore, a person had to hold the robot (by the neck) so that it produced forces on the pedal racer and did not tip over when the arms made contact. Joint-torque sensors were used to estimate the force on the end effector. Forces were first estimated for pedaling in the air. These forces were deducted from the estimates during the experiment to estimate the force of contact. The measurements were extremely noisy.

Figure 6 shows the results for the left arm of the robot. The results for the right arm were similar, but in counter-phase. The experiment is presented also in the attached video. The learning was started after 20 s. We can see from the top plot that the arm trajectory changed. In fact, it maintained constant contact with the pedal racer when the learning was stopped. On the bottom plot we see that the measured force curve did change its shape, but not completely to the desired one. This is a direct consequence of three major reasons. The first is that the force measurement was only an estimate, valid for static conditions only. The second is that the robot was held by a person and therefore the periods were not completely repeatable. The third is in setting learning for greater robustness with $\epsilon = 0.7$ in order to cope with the first two reasons. The results show that the trajectories were modified to maintain constant contact with the device and that while not achieving perfect tracking, a general shape of the force profile was still observed.

V. PEDAL RACER

We tested the proposed approach on the task of operating a pedal racer by standing on it. The task is harder than it seems even for people, and that is also how the device is marketed. The pedal racer moves forward if the pedals are moved in a circular fashion. Contact has to be maintained with both pedals, which imposes kinematical constraints. In one period of motion, the operator has to twice pass through singular configurations of the device by applying forces forward-backward using only friction to brace against the device. The experimental scenario demanded the robot to optimize the initial estimated trajectory of motion over several periods of learning in order to achieve the desired force profile.

We evaluated our approach in simulation using the Webots dynamic simulator [25], and we conducted preliminary experiments on the real robot.

A. Balance Control

In order to have the robot maintain balance, we implemented a hierarchial control structure with several tasks. We used iterative inverse kinematics algorithms to control the center-of-mass (COM) of the robot and the position/orientation of the two feet. Both were implemented as primary tasks by stacking the COM Jacobian and the Jacobian of one foot with respect to the other in an augmented
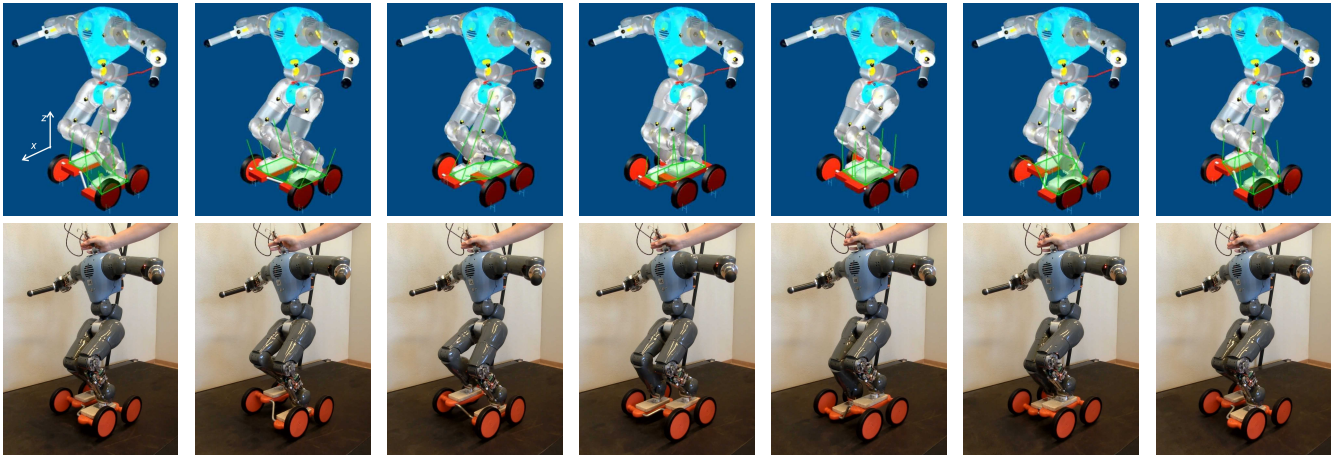
Fig. 8. In the top row an image sequence of one period of virtual COMAN steady-state pedaling on a virtual pedal racer in Webots dynamic simulator, after learning. The bottom row shows the real COMAN robot pedaling on the device. Note that the robot was helped to maintain balance at least once per period.
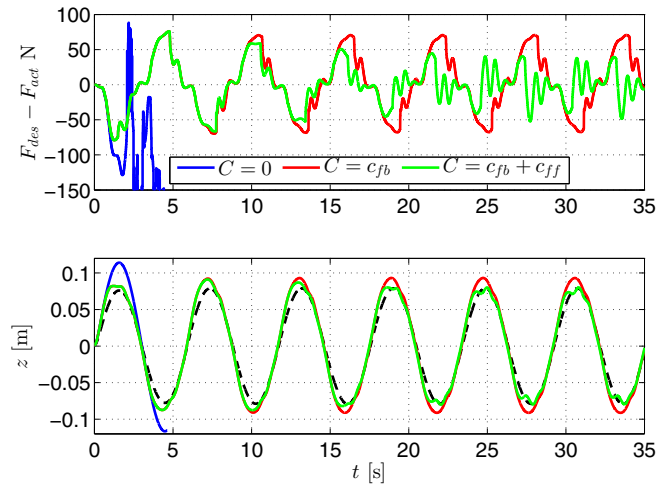

Fig. 7. Simulated results of operating the pedal racer with the robot standing on it. In the top plot the error of the desired force: no coupling (blue), feedback coupling (red), feedback and feed-forward coupling (green). We can see that the error of the force is considerably reduced when both feedback and feed-forward couplings are used. In the bottom plot the relative distance between the feet: first approximation (blue), feedback coupling (red), feedback and feed-forward couplings (green). Ideal trajectory is dashed black.

Jacobian. The control sets the COM to remain in the center between the feet. For details on the implementation of the controller refer to [26].

B. Results

We conducted several experiments in simulation. The task was designed to maintain a desired periodic force profile on the feet of the robot. Each foot of the robot is equipped with a 6-DOF force-torque sensor.

In the manner of the approach in [21], we used the measured forces of a successful execution as the reference, obtained in simulation by maintaining the balance using a COM controller and providing the exact kinematical motion to the feet of the robot, which was perfectly aligned to the device. Our task was designed to adapt the vertical motion of

the robots feet, starting from a reasonable first approximation (off by 4 cm). The approximation alone results in the robot tipping over. The phase relation between the horizontal and vertical movements was preset.

The plots in Fig. 7 show errors of force tracking for three scenarios: using no coupling (blue); using only feedback coupling (red) and; using both feedback and feed-forward coupling (green). The bottom plot shows the relative position of the feet for the same three scenarios and in the same colors. No coupling results in the robot tipping over. Feedback-only coupling results in higher errors and different trajectories than both feedback and feed-forward coupling. The bottom plot shows also the trajectory of referential, i. e., predefined pedaling (black-dotted).

Fig. 8 shows in the top row an image sequence of steady-state simulated pedaling after learning, executed in Webots dynamic simulator. The bottom plot shows the results on a real robot. The robot was pedaling on a treadmill; the frequency of the pedaling was calculated to keep the robot and the pedal racer in place. Both experiments are presented also in the accompanying video. Note that the robot had to be helped to maintain balance at certain times, at least once per period. The reason is in the discrepancy between the real robot and the model used to estimate the position of the center of mass, and in the compliance of the pedal racer, which would visibly bend under load.

C. Discussion of the results

It is important to note that we designed the task to show the applicability of the proposed approach for a demanding task. While we show the potential of the proposed approach, other, possibly fundamentally different approaches approach might prove equally effective, perhaps more. The issue at hand is in the referential forces. Since the task of riding the pedal racer is demanding – the robot has to be stable, only friction is used to produce forces forward-backward, singular configurations have to be overcome when the feet are the most apart in the up-down direction – we mimicked the approach by Pastor et al. [21] and used a successful execution to acquire

the referential force trajectory. A successful execution is relatively easy to come by in simulation, but achieving the same on the real robot is complex in maintaining the balance of the robot. The latter is crucial for force measurements. It is evident in Fig. 8 that we had to intervene by hand to maintain balance at certain points of the period of motion.

Since the robot is standing on the pedal racer, any instability or perturbation will directly affect the forces. The robot boasts SEAs and is therefore slightly wobbly even when standing on solid ground. Any oscillation of the springs introduces forces which have an effect on the trajectory and also learning, adding to the acceleration of the robot, which results in additional forces, possibly leading to self-excitement of the trajectories. Some of this is evident also in Fig. 7.

VI. CONCLUSIONS AND FUTURE WORK

While thus far mostly restricted to the kinematic domain, coupling the DMPs with force feedback extends their domain to include dynamic tasks. The combination of periodic DMPs, force-feedback and learned feed-forward coupling allows for generation and execution of previously unattainable motions, resulting in rich and adaptable periodic motor skills of a robot.

We used repetitive control, which can cope with the change of initial conditions of the start of a period, making it suitable for periodic tasks. The plug-in type RC with a feedback term can also cope with noisy and less-than-perfect repetitions through periods of motion. In the paper we only briefly touched the issue of the stability of the algorithm, relying on empirically set values to achieve and present initial results. In the future we will derive explicit stability criteria, just as was derived for discrete tasks in [19].

The complex real-world task of operating the pedal racer shows the potential of the proposed approach. In our scenario the trajectory of required motion to pedal on the device was only estimated and then optimized for the task through several periods of learning. Even though the final, learned trajectory resulted in operating the device, showing that a learning/adaptation approach is needed, it still leaves room for improvement of robotic pedal-racing in future work.

REFERENCES

[1] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642 – 653, 2008, robotics and Neuroscience.

[2] S. Schaal, D. Sternad, R. Osu, and M. Kawato, "Rhythmic movement is not discrete," *Nature Neuroscience*, no. 10, pp. 1137–1144, 2004.

[3] R. Ronsse, D. Sternad, and P. Lefèvre, "A computational model for rhythmic and discrete movements in uni- and bimanual coordination," *Neural Computation*, vol. 21, no. 5, pp. 1335 – 13 370, 2009.

[4] S. Degallier and A. Ijspeert, "Modeling discrete and rhythmic movements through motor primitives: a review," *Biological Cybernetics*, vol. 103, no. 4, pp. 319–338, 2010.

[5] N. Hogan and D. Sternad, "Dynamic primitives in the control of locomotion," *Frontiers in Computational Neuroscience*, vol. 7, no. 71, 2013.

[6] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 2, Washington, DC, 2002, pp. 1398–1403.

[7] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[8] S. Khansari-Zadeh and A. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010, pp. 2676–2683.

[9] 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, no. 1, pp. 3–23, 2009.

[10] D. Pongas, A. Billard, and S. Schaal, "Rapid Synchronization and Accurate Phase-Locking of Rhythmic Motor Primitives," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 2911–2916.

[11] T. Petrič, A. Gams, A. J. Ijspeert, and L. Žlajpah, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1775–1788, 2011.

[12] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[13] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D*, vol. 216, no. 2, pp. 269–281, 2006.

[14] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[15] L. Peternel, T. Petrič, E. Oztop, and J. Babič, "Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach," *Autonomous Robots*, pp. 1–14, 2013.

[16] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589 – 1600, 2009.

[17] L. Cuiyan, Z. Dongchun, and Z. Xianyi, "A survey of repetitive control," in *2004 IEEE/RSJ International Conference on Robots and Systems (IROS)*, vol. 2, 2004, pp. 1160–1166.

[18] A. Gams, M. Do, A. Ude, T. Asfour, and R. Dillmann, "On-line periodic movement and force-profile learning for adaptation to new surfaces," in *2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Nashville, TN, 2010, pp. 560–565.

[19] A. Gams, B. Nemec, L. Žlajpah, M. Waechter, A. Ijspeert, T. Asfour, and A. Ude, "Modulation of motor primitives using force feedback: Interaction with the environment and bimanual tasks," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 5629–5635.

[20] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, 2013.

[21] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, 2011, pp. 365–371.

[22] N. Gopalan, M. P. Deisenroth, and J. Peters, "Feedback Error Learning for Rhythmic Motor Primitives," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[23] R. W. Longman and Y.-C. Huang, "The phenomenon of apparent convergence followed by divergence in learning and repetitive control," *Intelligent Automation & Soft Computing*, vol. 8, no. 2, pp. 107–128, 2002.

[24] L. Colasanto, N. Tsagarakis, and D. Caldwell, "A compact model for the compliant humanoid robot coman," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, 2012, pp. 688–694.

[25] Webots, "http://www.cyberbotics.com," 2012, commercial Mobile Robot Simulation Software. [Online]. Available: http://www.cyberbotics.com

[26] A. Gams, J. van den Kieboom, F. Dzeladini, and A. Ijspeert, "Stable real-time full body motion imitation on the coman humanoid robot," in *Proceedings of the 22nd International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, 2013.