



# Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems

Elmar Ruckert<sup>1\*</sup> and Andrea d'Avella<sup>2</sup>

<sup>1</sup> Institute for Theoretical Computer Science, Graz University of Technology, Austria

<sup>2</sup> Laboratory of Neuromotor Physiology, Fondazione Santa Lucia, Rome, Italy

## Edited by:

Martin Giese, University Clinic  
Tuebingen, Germany

## Reviewed by:

Florentin Wörgötter, University  
Goettingen, Germany  
Dominik M. Endres, HIH, CIN,  
BCCN and University of Tübingen,  
Germany  
Tomas Kulvicius, University of  
Goettingen, Germany

## \*Correspondence:

Elmar Ruckert, Institute for  
Theoretical Computer Science,  
Graz University of Technology,  
Inffeldgasse 16b/1, 8010 Graz,  
Austria  
e-mail: ruckert@igi.tugraz.at

A salient feature of human motor skill learning is the ability to exploit similarities across related tasks. In biological motor control, it has been hypothesized that muscle synergies, coherent activations of groups of muscles, allow for exploiting shared knowledge. Recent studies have shown that a rich set of complex motor skills can be generated by a combination of a small number of muscle synergies. In robotics, dynamic movement primitives are commonly used for motor skill learning. This machine learning approach implements a stable attractor system that facilitates learning and it can be used in high-dimensional continuous spaces. However, it does not allow for reusing shared knowledge, i.e., for each task an individual set of parameters has to be learned. We propose a novel movement primitive representation that employs parametrized basis functions, which combines the benefits of muscle synergies and dynamic movement primitives. For each task a superposition of synergies modulates a stable attractor system. This approach leads to a compact representation of multiple motor skills and at the same time enables efficient learning in high-dimensional continuous systems. The movement representation supports discrete and rhythmic movements and in particular includes the dynamic movement primitive approach as a special case. We demonstrate the feasibility of the movement representation in three multi-task learning simulated scenarios. First, the characteristics of the proposed representation are illustrated in a point-mass task. Second, in complex humanoid walking experiments, multiple walking patterns with different step heights are learned robustly and efficiently. Finally, in a multi-directional reaching task simulated with a musculoskeletal model of the human arm, we show how the proposed movement primitives can be used to learn appropriate muscle excitation patterns and to generalize effectively to new reaching skills.

**Keywords:** dynamic movement primitives, muscle synergies, reinforcement learning, motor control, musculoskeletal model

## 1. INTRODUCTION

Reinforcement Learning of motor skills in robotics is considered to be very challenging due to the high-dimensional continuous state and action spaces. In many studies it has been shown that learning can be facilitated by the use of movement primitives (Schaal et al., 2003; Ruckert et al., 2013). Movement primitives are parametrized representations of elementary movements, where typically for each motor skill a small set of parameters is tuned or learned. However, many motor control tasks are related and could be learned more effectively by exploiting shared knowledge.

This is a well-known concept in motor neuroscience, where muscle synergies or coherent activations of groups of muscles (d'Avella et al., 2003; d'Avella and Bizzi, 2005; Bizzi et al., 2008) have been proposed to simplify the control problem of complex musculoskeletal systems. In analyzing muscle activation recordings it has been demonstrated that by combining only few muscle activation patterns multiple task instances of natural motor behaviors, e.g., fast reaching movements of humans (d'Avella

et al., 2006), primate grasping movements (Overduin et al., 2008), or walking patterns of infants, toddlers, and adults (Dominici et al., 2011) could be efficiently modeled. One important finding of these studies is that the dimensionality of the motor control problem can be drastically reduced by reusing common knowledge of related tasks, i.e., grasping objects at different locations using a linear combination of shared muscle synergies. While this has been demonstrated in biological data analysis, only few robotic applications exist that use this shared task knowledge (Chhabra and Jacobs, 2006; Alessandro et al., 2012). These methods demonstrate the advantages of shared synergies in learning robotic tasks. However, different procedures were applied to obtain a parametric description of synergies, i.e., in Chhabra and Jacobs (2006) a variant of non-negative matrix factorization (d'Avella et al., 2003) was used given a set of pre-computed trajectories and in Alessandro et al. (2012) the synergies were extracted from dynamic responses of a robot system with random initialization. In contrast, we propose to learn the synergies representation

in a reinforcement learning framework, where task-specific and task-invariant parameters in a multi-task learning setting are learned simultaneously.

In robotics the most widely used approach for motor skill learning are Dynamic Movement Primitives (DMPs) (Schaal et al., 2003; Ijspeert et al., 2013). This approach uses parametrized dynamical systems to determine a movement trajectory and has several benefits. First, as it is a model-free approach, there is no need to learn the typically non-linear, high-dimensional dynamic forward model of a robot (However, this is not the case when inverse dynamics controller are used to compute the control commands). Second, it provides a linear policy parametrization which can be used for imitation learning and policy search (Kober and Peters, 2011). The complexity of the trajectory can be scaled by the number of parameters (Schaal et al., 2003) and one can adapt meta-parameters of the movement such as the movement speed or the goal state (Pastor et al., 2009; Kober et al., 2010). Finally, the dynamical system is constructed such that the system is stable. This simplifies learning since even without modulating the dynamical system the movement trajectory is always attracted by a known (or learned) goal state. However, this parametrization does not allow for reusing shared knowledge, as proposed by the experimental findings studying complex musculoskeletal systems (d'Avella et al., 2003; Bizzi et al., 2008; d'Avella and Pai, 2010). Thus, typically for each motor task an individual movement parametrization has to be learned.

In this paper we propose to use a superposition of learned basis functions or synergies to modulate the stable attractor system of DMPs. This allows for reusing shared knowledge for learning multiple related tasks simultaneously while preserving the benefits of the dynamical systems, i.e., the stability in learning complex motor behavior. The synergies and their activation in time are learned from scratch in a standard reinforcement learning setup. Note that imitation learning could also be applied to implement an initial guess for the synergies, e.g., by using decomposition strategies discussed in d'Avella and Tresch (2001). However, this is beyond the scope of this paper. Moreover, our approach is like the DMPs applicable to discrete and rhythmic movements and allows for modeling time-varying synergies (d'Avella et al., 2006). We therefore denote our approach *DMPSynergies*. By using for each task a combination of individual, temporally fixed, basis functions DMPs can be modeled as special case of this approach. The benefit of the common prior knowledge is even more drastic when generalizing to new motor tasks given the previously learned basis functions. Thus, for simpler synergies only the weights for the linear combination have to be acquired and for time-varying synergies additionally the time-shift parameters need to be learned. This is demonstrated on a complex walking task and on reaching task using an arm actuated by muscles.

As in previous studies on DMPs (Meier et al., 2011; Mülling et al., 2013) we want to go beyond basic motor skills learning. However, in contrast to those studies that use a library of primitives for sequencing elementary movements (Meier et al., 2011) or mixing basic skills (Mülling et al., 2013), we implement the common shared knowledge among multiple tasks as prior in a hierarchical structure. On the lower level task related parameters, i.e., amplitude scaling weights or time-shift parameters are used

to modulate a linear superposition of learned basis functions, the shared higher level knowledge. This has the promising feature that by combining just a small number of synergies diverse motor skills can be generated.

In the Materials and Methods, we will first briefly introduce DMPs (Schaal et al., 2003; Ijspeert et al., 2013) as we build on this approach. We then extend DMPs to allow for reusing shared task knowledge in the form of parametrized synergies. The advantage of the shared knowledge is evaluated in the Results on three multi-task learning scenarios. First, a simple via-point task is used to demonstrate the characteristics of the proposed representation. Then, rhythmic movements are learned in a dynamic 5-link planar biped walker environment. Finally, a musculoskeletal model of a human arm is used to evaluate our primitives on a muscle actuated system learning discrete reaching movements to multiple targets.

## 2. MATERIALS AND METHODS

### 2.1. DYNAMIC MOVEMENT PRIMITIVES

DMPs generate multi-dimensional trajectories by the use of non-linear differential equations (simple damped spring models) (Schaal et al., 2003). The basic idea is to use for each degree-of-freedom (DoF), or more precisely for each actuator, a globally stable, linear dynamical system of the form

$$\tau \dot{z} = \alpha_z(\beta_z(g - y^*) - z) + f, \quad \tau \dot{y}^* = z, \quad (1)$$

which is modulated by a learnable non-linear function  $f$ . The final position of a movement is denoted by  $g$  and the variables  $y^*$  and  $\dot{y}^*$  represent the desired state in i.e., joint angles and joint velocities. The time constants  $\alpha$  and  $\beta$  are usually pre-defined. The temporal scaling factor  $\tau$  can be used for de- or accelerating the movement execution as needed. Finally  $z$  denotes an internal variable of the dynamical system. For each DoF an individual function  $f$  is used which is different for discrete and rhythmic movements.

For discrete movements the function  $f$  only depends on the phase  $s$ , which is an abstraction of time and was introduced to scale the movement duration (Schaal et al., 2003). The function  $f(s)$  is constructed of the weighted sum of  $N$  Gaussian basis functions  $\Psi_n$

$$f(s) = \frac{\sum_{n=1}^N \Psi_n(s) w_n s}{\sum_{n'=1}^N \Psi_{n'}(s)}, \quad (2)$$

where for discrete movements these Gaussian basis functions are

$$\Psi_n(s) = \exp\left(-\frac{1}{2h_n^2}(s - \mu_n)^2\right), \quad \tau \dot{s} = -\alpha_s s.$$

Only the weights  $w_n$  are parameters of the primitive which can modulate the shape of the movement. The centers or means  $\mu_n \in [0, 1]$  specify at which phase of the movement the basis function becomes active. They are typically equally spaced in the range of  $s$  and not modified during learning. The bandwidth of the basis functions is given by  $h_n^2$  and is typically chosen such that the Gaussians overlap.

For rhythmic movements periodic activation functions are used (Ijspeert et al., 2002). The non-linear function  $f$  reads

$$f(\phi) = \frac{\sum_{n=1}^N \Psi_n(\phi) w_n}{\sum_{n'=1}^N \Psi_{n'}(\phi)}, \quad (3)$$

where the periodic phase angle is denoted by  $\phi \in [0, 2\pi]$ . In Ijspeert et al. (2002) additionally a scalar variable was used to scale the amplitude of the oscillator, which was omitted for simplicity. The basis functions are given by

$$\Psi_n(\phi) = \exp(h_n(\cos(\phi - \mu_n) - 1)), \quad \tau \dot{\phi} = 1,$$

which implement von Mises basis functions. Note that for the periodic basis functions the trajectory in Equation 1 oscillates around the attractor point or goal state  $g$ .

Integrating the dynamical systems in Equation 1 for each DoF results into a desired trajectory  $(\mathbf{y}_t^*, \dot{\mathbf{y}}_t^*)$  of the joint angles. To follow this trajectory, in the most simple case a linear feedback controller is subsequently used to generate appropriate control commands denoted by  $\mathbf{u}_t$ :

$$\mathbf{u}_t = \text{diag}(\mathbf{k}_{\text{pos}})(\mathbf{y}_t^* - \mathbf{y}_t) + \text{diag}(\mathbf{k}_{\text{vel}})(\dot{\mathbf{y}}_t^* - \dot{\mathbf{y}}_t). \quad (4)$$

For each actuator the linear weights  $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_D]$  as well as the control gains  $\mathbf{k}_{\text{pos}}$  and  $\mathbf{k}_{\text{vel}}$  have to be specified, i.e.,  $\boldsymbol{\theta} = [\mathbf{W}, \mathbf{k}_{\text{pos}}, \mathbf{k}_{\text{vel}}]$ . This results into  $ND + 2D$  parameters for the movement representation, where  $D$  denotes the number of actuators or muscles of a system. The simulated trajectory is denoted by  $(\mathbf{y}_t, \dot{\mathbf{y}}_t)$ .

In multi-task learning we want to learn  $k = 1..K$  tasks simultaneously. For very simple tasks, such as the via-point experiments described below, it could be sufficient to adapt the goal state  $g$ . However, this is usually not the case for more complex motor skill learning tasks in robotics. With DMPs typically for each motor skill an individual movement parametrization  $\boldsymbol{\theta}_k$  has to be learned. However, if we assume similarities among these tasks the learning problem could potentially be simplified by reusing shared knowledge. Inspired by experimental findings in biology (d'Avella et al., 2003; Bizzi et al., 2008; d'Avella and Pai, 2010) we extend these DMPs. Only the parametrization for the non-linear function  $f(s)$  for discrete movements or  $f(\phi)$  for rhythmic movement changes. The dynamical system in Equation 1 and the linear feedback controller in Equation 4 remains the same.

## 2.2. DYNAMIC MOVEMENT PRIMITIVES WITH SHARED SYNERGIES (DMPSynergies)

For learning the  $k$ th task, we propose to use a linear combination of temporal flexible basis functions or synergies to parametrize the non-linear function  $f(s)$  in Equation 2 or for rhythmic movements  $f(\phi)$  in Equation 3:

$$f(s, k) = \sum_{m=1}^M \beta_{m, k} \Lambda(s, \boldsymbol{\theta}_m, \Delta s_{m, k}), \quad (5)$$

$$f(\phi, k) = \sum_{m=1}^M \beta_{m, k} \Omega(\phi, \boldsymbol{\theta}_m, \Delta s_{m, k}), \quad (6)$$

where  $s$  denotes the phase variable which is only used for discrete movements. As with DMPs ( $\Psi_n$  in Equation 2) the functions  $\Lambda(\cdot)$  and  $\Omega(\cdot)$  are different for discrete and rhythmic movements.

All  $K$  tasks share  $m = 1..M$  synergies which are parametrized via the vector  $\boldsymbol{\theta}_m$ . Solely the weights  $\beta_{m, k}$  and the time-shift  $\Delta s_{m, k}$  are individual parameters for each task. The basic concept of the model is sketched in Figure 1 for a one-dimensional discrete movement.

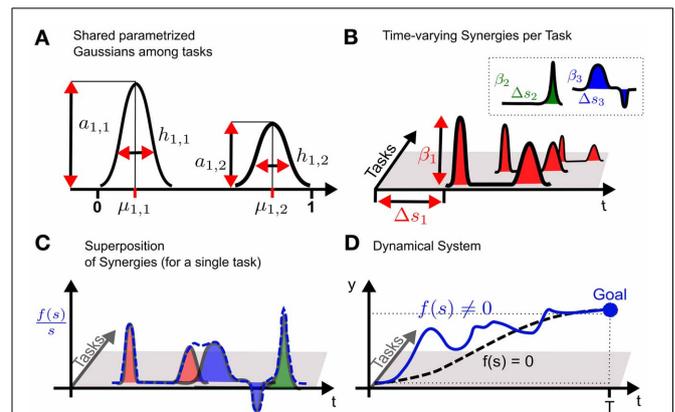
The complexity of each synergy is controlled by the number of Gaussians for discrete movements or by the number of von Mises basis functions for rhythmic patterns. We denote this number by  $N$ , where we parametrize in both cases the amplitude, the mean and the bandwidth. Thus, each synergy is represented by a parameter vector  $\boldsymbol{\theta}_m = [a_{m, 1}, \mu_{m, 1}, h_{m, 1}, \dots, a_{m, N}, \mu_{m, N}, h_{m, N}]$ .

For discrete movements the function  $\Lambda(\cdot)$  reads

$$\Lambda(s, \boldsymbol{\theta}_m, \Delta s_{m, k}) = \sum_{n=1}^N a_{m, n} \exp\left(-\frac{1}{2h_{m, n}}(s - \mu_{m, n} + \Delta s_{m, k})^2\right). \quad (7)$$

For rhythmic movements a superposition of von Mises basis functions is used

$$\begin{aligned} \Omega(\phi, \boldsymbol{\theta}_m, \Delta s_{m, k}) \\ = \sum_{n=1}^N a_{m, n} \exp(h_{m, n} \cos(\phi - \mu_{m, n} + \Delta s_{m, k}) - 1). \end{aligned} \quad (8)$$



**FIGURE 1 | Conceptual idea of using shared synergies in dynamical systems. (A)** A synergy is constructed by a superposition of parametrized Gaussians. The parameters are the amplitude  $a_{m, n}$ , the mean  $\mu_{m, n}$  and the bandwidth  $h_{m, n}$ . In the example two Gaussians ( $n = 1..2$ ) are used to model the first  $m = 1$  synergy. **(B)** For each task only the activation  $\beta_m$  of a synergy is learned. Time-varying synergies additionally implement a time-shift  $\Delta s_m$ . The key concept is that multiple tasks share the same parametrized synergies shown in **(A)**, which represent task related common knowledge. **(C)** For each task the non-linear function  $f(s)$  is given by the weighted sum of the (time-shifted) synergies. Shown is a normalized version of  $f(s)$  to illustrate the effects of the superposition also at the end of the movement, which would usually converge toward zero. **(D)** Finally, the non-linear function  $f(s)$  is used to modulate a dynamical system. The unperturbed system with  $f(s) = 0$  is denoted by the dashed line which is attracted by the goal state that is indicated by the large dot.

DMPs (Schaal et al., 2003) can be modeled as a special case of this formulation. For DMPs using  $n = 1..N$  basis functions the mean  $\mu_{m,n}$  and the bandwidth  $h_{m,n}$  of the basis functions are fixed as discussed in Section 2.1. Solely the  $n = 1..N$  amplitudes or weights  $a_{m,n}$  are learned. By fixing these parameters and by modeling the non-linear function  $f(s)$  for discrete movements or  $f(\phi)$  for rhythmic movements using a single ( $M = 1$ ) synergy our representation can be used to implement DMPs.

### 2.2.1. Multi-dimensional systems

For multi-dimensional systems for each actuator  $d = 1..D$  an individual dynamical system in Equation 1 and hence an individual function  $f(s, k)$  in Equation 5 or  $f(\phi, k)$  in Equation 6 is used (Schaal et al., 2003). The phase variable  $s$  or  $\phi$  is shared among all DoF (Note that  $k = 1..K$  denotes the task.).

Extending our notation for multi-dimensional systems the non-linear function  $f(s, k)$  in Equation 5 can be written as

$$\underbrace{f(s, d, k)}_{1 \times 1} = \sum_{m=1}^M \underbrace{\beta_{m,k,d}}_{1 \times 1} \underbrace{\Lambda(s, \boldsymbol{\theta}_{m,d}, \Delta s_{m,k,d})}_{1 \times 1} s.$$

Depending on the dimension  $d$  different weights  $\beta_{m,k,d}$ , policy vectors  $\boldsymbol{\theta}_{m,d}$  and time-shift parameters  $\Delta s_{m,k,d}$  are used. Note that the policy vector  $\boldsymbol{\theta}_{m,d}$  is task-independent. Interestingly, when implementing additionally dimension-independent policy vectors, i.e.,  $\boldsymbol{\theta}_m$  anechoic mixing coefficients (Giese et al., 2009) can be modeled.

Here, we only discuss discrete movement representations, however, the reformulation procedure applies also for rhythmic movement parametrizations. Let us also define a vector notation of  $\mathbf{f}(s, k)$

$$\underbrace{\mathbf{f}(s, k)}_{1 \times D} = \sum_{m=1}^M \underbrace{\beta_{m,k}}_{1 \times D} \circ \underbrace{\mathbf{w}_m(s, \boldsymbol{\theta}_{m,1..D}, \Delta s_{m,k,d})}_{1 \times D}, \quad (9)$$

where the symbol  $\circ$  denotes the Hadamard product, the element-wise multiplication of vectors. The synergy vectors are specified by

$$\mathbf{w}_m(s, \boldsymbol{\theta}_{m,1..D}, \Delta s_{m,k,d}) = [\Lambda(s, \boldsymbol{\theta}_{m,1}, \Delta s_{m,k,1})s, \Lambda(s, \boldsymbol{\theta}_{m,2}, \Delta s_{m,k,2})s, \dots, \Lambda(s, \boldsymbol{\theta}_{m,D}, \Delta s_{m,k,D})s].$$

This vector notation is used in the following to compare to existing synergies representations (d'Avella et al., 2003, 2006).

### 2.3. MUSCULOSKELETAL MODELS AND MUSCLE SYNERGIES

We also use the proposed movement representation, the DMPsSynergies, to generate muscle excitation patterns. These patterns are applied as input in a forward simulation of a musculoskeletal model. A schematic overview of such a simulation is shown in Figure 2. We briefly discuss all processes involved.

**Muscle synergies for generating muscle excitation patterns** are used as input in forward dynamics simulations. In our simulation experiments we evaluate time-varying synergies (d'Avella et al., 2006), which are a particular instance of the DMPsSynergies, i.e., the weights  $\beta_{m,k}$  and time-shift parameters  $\Delta s_{m,k}$  in Equation 9 are independent of the dimension  $d$ . Thus, for discrete movements in multi-dimensional systems  $f(s, k)$  reads

$$\underbrace{\mathbf{f}(s, k)}_{1 \times D} = \sum_{m=1}^M \underbrace{\beta_{m,k}}_{1 \times 1} \underbrace{\mathbf{w}_m(s + \Delta s_{m,k}, \boldsymbol{\theta}_{m,1..D}, 0)}_{1 \times D}, \quad (10)$$

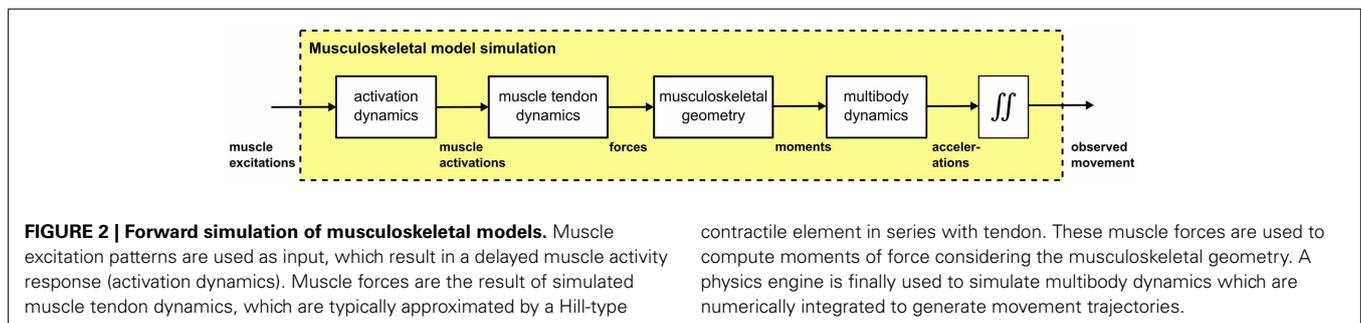
where  $\beta_{m,k}$  is a scalar and the time-shift parameter  $\Delta s_{m,k}$  is directly added to the phase variable  $s$ . This allows for a comparison to e.g., the formulation of time-varying synergies given in d'Avella et al. (2006), where

$$\mathbf{x}(t, k) = \sum_{m=1}^M a_m^k \mathbf{v}_m(t - t_m^k).$$

Shared synergies are represented by time-dependent vectors  $\mathbf{v}_m(t - t_m^k)$ , where in contrast to the proposed DMPsSynergies a minor difference is the sign of the time-shift parameter  $t_m^k$ .

In this formulation of time-varying synergies (d'Avella et al., 2006) only the time-invariant combination coefficients  $a_m^k$  are task-dependent, whereas the vector  $\mathbf{v}_m$  is task-independent. However, by using task, spatial or temporal (in)variant implementations of the mixing coefficients  $a$  or the basis vectors  $\mathbf{v}$  other representations of synergies (d'Avella et al., 2003; Ivanenko et al., 2004; Giese et al., 2009) can be also implemented.

**Activation dynamics** model the effect of the delayed force generating process in muscles, as they are not capable of generating force instantaneously. Typically, for each muscle a first order differential equation is used, i.e.,  $\dot{a} = (f(s, k)^2 - f(s, k)a)/\tau_{\text{rise}} + (f(s, k) - a)/\tau_{\text{fall}}$  (Zajac, 1989).



Here,  $f(s, k)$  denotes the generated muscle excitation signal using e.g., the proposed DMPSynergies. The actual muscle activation is denoted by  $a$ , which is a function of the rise time constant  $\tau_{\text{rise}}$  and the fall time constant  $\tau_{\text{fall}}$ . For our evaluations we implemented  $\tau_{\text{rise}} = 10$  ms and  $\tau_{\text{fall}} = 40$  ms (Winters and Stark, 1985).

**Muscle tendon dynamics** describe the complex and non-linear force generation properties of muscles. For an approximation a variety of models exist (Zajac, 1989). In these models a muscle is approximated by a number of musculotendinous units, each of which is implemented by a Hill-type contractile element in series with tendon. Characteristic properties of muscles are the optimal fiber length  $L_0^M$ , the maximum isometric force  $F_0^M$ , and the muscle pennation angle  $\alpha$ , which are shown in **Table A5** in the appendix for the investigated model of a human arm. The tendon dynamics in this musculoskeletal model were approximated by the muscle model proposed in Schutte et al. (1993).

**Musculoskeletal geometry** represents the path of a muscle from its origin to its insertion that can be implemented as a series of straight-line path segments, which pass through a series of via points (Delp et al., 2007). To simulate how muscles wrap over underlying bone and musculature wrapping surfaces i.e., cylinders, spheres and ellipsoids are implemented, where this model is based on the upper extremity model discussed in Holzbaur et al. (2005). A detailed description of the implemented musculoskeletal geometry is given in the supplement (in form of a simulation model file, .osim).

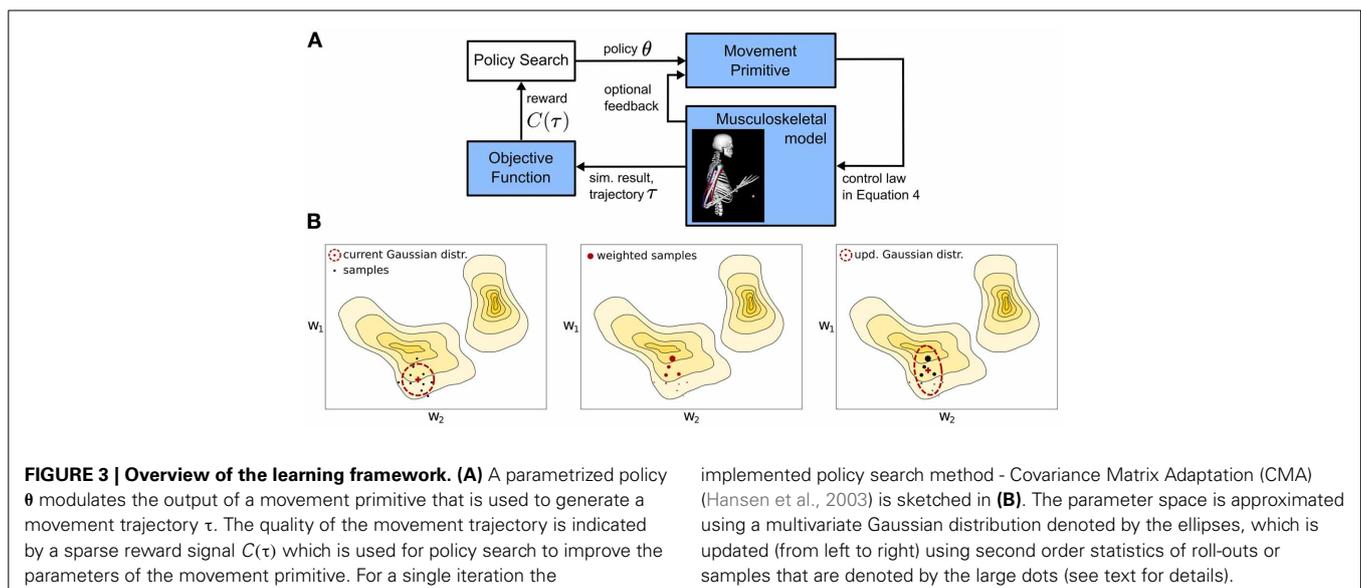
**Multibody dynamics** are simulated by the physics simulation application OpenSim (Delp et al., 2007; Seth et al., 2011). It is an open source software that already implements a variety of muscle models (Zajac, 1989) and a large number musculoskeletal models are freely available. In our experiments the computational time needed to simulate a movement with a duration of e.g., 500 ms takes between 10 and 20 s (OpenSim implements numerical integrators with an adaptive

time step) on a standard computer (3 GHz and 4 GB memory). However, we exploited parallel computing techniques for policy search, which resulted in a gain of factor 10. Alternatively, the muscle dynamics could be approximated via regression methods to speed-up the simulations (Chadwick et al., 2009).

## 2.4. LEARNING WITH MOVEMENT PRIMITIVES

We denote the parametrization of a movement primitive by a policy vector  $\theta$ . A widely used approach in robotics to learn these parameters is episodic reinforcement learning (Kober and Peters, 2011), which is outlined in **Figure 3A**. A policy search method is used to improve the movement primitive's representation  $\theta$  assuming a given objective or reward function  $C(\tau) \in \mathbb{R}^1$ . Throughout this manuscript  $C(\tau)$  denotes a cost value that is equivalent to the negative reward in classical reinforcement learning (Sutton and Barto, 1998). It indicates the quality of an executed movement. A trajectory  $\tau = (\mathbf{y}_{1:T}, \mathbf{u}_{1:T-1})$  is specified by the simulated joint angles  $\mathbf{y}$  and the applied controls (torques)  $\mathbf{u}$ , where  $T$  denotes the number of time steps. We want to find a movement primitive's parameter vector  $\theta^* = \text{argmin}_{\theta} J(\theta)$  which minimizes the expected costs  $J(\theta) = \mathbb{E}[C(\tau)|\theta]$ . We assume that we can evaluate the expected costs  $J(\theta)$  for a given parameter vector  $\theta$  by performing roll-outs (samples) on the real or simulated system. In other words each movement trajectory is quantified by a single scalar reward  $C(\tau)$ , which can be used by an optimization method to improve the best guess of the movement policy  $\theta$ .

For learning or optimizing the policy parameters  $\theta$  a variety of policy search algorithms exist in the motor control literature. Examples are the REINFORCE (Williams, 1992), the episodic Natural Actor Critic (Peters and Schaal, 2008), the Power (Kober and Peters, 2011) or the PI<sup>2</sup> (Theodorou et al., 2010) algorithm, which are reviewed in Kober and Peters (2011). Alternatively, standard optimization tools such as the 2nd order stochastic search methods (Hansen et al., 2003; Wierstra et al.,



2008; Sehnke et al., 2010) can be used for policy search. These machine learning tools make no assumptions on a specific form of a policy and typically have just a single parameter to tune, the initial exploration rate. We therefore use the stochastic search method Covariance Matrix Adaptation (CMA) (Hansen et al., 2003) for learning the policy parameters in our experiments.

Roughly, CMA is an iterative procedure that locally approximates the function  $C(\tau(\theta))$  by a multivariate Gaussian distribution, which is denoted by the ellipse in the sketch in **Figure 3B**. From left to right a single optimization step for a two-dimensional policy vector  $\theta = [w_1, w_2]$  is shown. The colored regions denote the unknown optimization landscape, where solid lines depict equal  $C(\tau)$  values. From the current Gaussian distribution, denoted by the ellipse in the left panel, CMA generates a number of samples, denoted by the black dots, evaluates the samples (the size of the dots in the center panel is proportional to their  $C(\tau)$  values), computes second order statistics of those samples that reduced  $C(\tau)$  and uses these to update the Gaussian search distribution, which is shown in right panel. For algorithmic details we refer to Hansen et al. (2003).

Note that for most interesting robotic tasks the unknown optimization landscape that is also sketched in **Figure 3B** is multimodal and policy search might converge to a local optimum. Thus, the result of learning is sensitive to the initial policy parameters  $\theta$  and for evaluating the convergence rate of different policy search methods multiple initial configurations should be considered (Kober and Peters, 2011). However, in this manuscript we evaluate the characteristics of movement primitive representations and put less emphasis on a particular policy search method. As we will demonstrate in our experiments CMA is robust in terms of converging to “good” solutions given the initial values of the evaluated movement primitive representations listed in the appendix.

In our experiments we compare single task learning with DMPs to learning multiple tasks simultaneously with DMPSynergies. With DMPs for each task  $k = 1..K$  an individual policy vector  $\theta_k$  is learned, where the objective function used in policy search takes the task index as additional argument, i.e.,  $C(\tau, k)$ . For learning multiple tasks simultaneously with DMPSynergies the policy vector  $\theta$  encodes all  $K$  task specific parameters  $\beta_{m,k}$  and  $\Delta s_{m,k}$ , and all shared parameters denoted by  $\theta_m$  in Equation 5 or Equation 6. The objective function is the sum of the individual task dependent costs  $C(\tau) = \sum_{k=1}^K C(\tau, k)$ .

### 3. RESULTS

We evaluated the proposed movement representation, the DMPSynergies, with simulations using three multi-task learning scenarios. A simple via-point task was used to illustrate the characteristics of the proposed movement representation. A challenging robotic learning task was used to generate rhythmic walking patterns for multiple step heights. Discrete reaching movements were learned using a musculoskeletal model of a human arm with eleven muscles.

#### 3.1. VIA-POINT REACHING TASK WITH A SIMPLE TOY MODEL

The goal of this simple multi-task learning problem is to pass through  $k = 1..5$  via-points ( $vp_k \in \{0.2, 0.1, 0, -0.1, -0.2\}$ ), denoted by the large dots in **Figure 4A** and navigate to the goal state  $g$  at 1. We used a point mass system (1 kg), where the state at time  $t$  is given by the position  $y_t$  and the velocity  $\dot{y}_t$ . The applied controls  $u_t$  shown in **Figure 4B** are computed using the linear feedback control law with  $k_{pos} = 400$  and  $k_{vel} = 15$  specified in Equation 4. The finite time horizon is given by  $T = 50$ . For the dynamical system in Equation 1 we used the parameters  $\alpha_z = 2$ ,  $\beta_z = 0.9$  and  $\tau = 0.1$ . Further parameter settings used for policy search are summarized in **Table A1** in the appendix.

This task is specified by the objective function

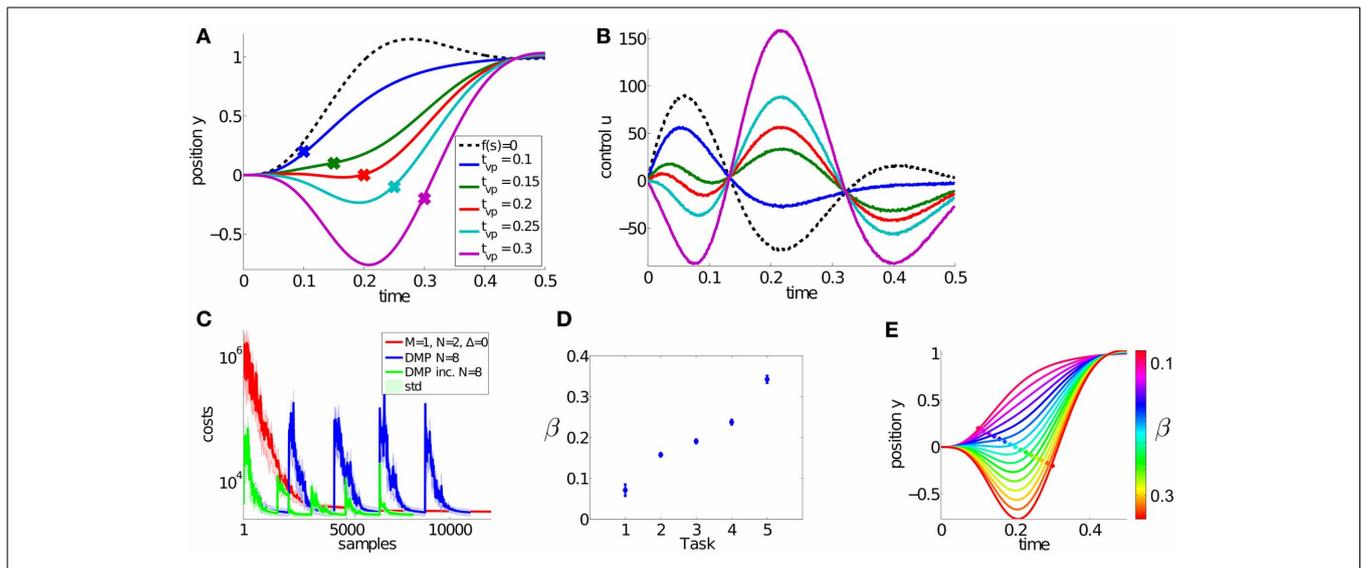
$$C(k) = 10^5 \left( y_{t_{vp_k}} - vp_k \right)^2 + 10^4 \left( \dot{y}_T^2 + 10 (y_T - g)^2 \right) + 5 \cdot 10^{-4} \sum_{t=1}^T u_t.$$

The first two terms punish deviations from the via-point  $vp_k$  and the goal state  $g$ , where  $y_{t_{vp_k}}$  denotes the position of the state at the time index of the  $k$ th via-point. The last term punishes high energy consumption, where  $u_t$  denotes the applied acceleration. Note that for simplicity we did not introduce the variable  $\tau$  denoting the movement trajectory in  $C(\tau, k)$  in Subsection 2.4. We always add a Gaussian noise term with a standard deviation of  $\sigma = 0.5$  to the control action to simulate motor noise.

We used a single synergy ( $M = 1$ ) with  $N = 2$  Gaussians to model the shared prior knowledge. The learning curve is shown in **Figure 4C**, where we compare to single-task learning using DMPs with  $N = 8$  Gaussian basis functions. For the via-point task 8 Gaussians were optimal with respect to the convergence rate, where we evaluated representations using  $N = 2..20$  Gaussians (not shown). Additionally, we compare to an incremental learning setup (DMP inc.) in **Figure 4C**, where the DMP representation is always initialized with the best learned solution from the previous task. On the x-axis the number of samples or trajectory evaluations on the point mass system is plotted. As it can be seen the proposed approach can benefit from the shared knowledge and has a faster overall learning performance.

In this experiment, for each task we fixed the time-shift  $\Delta s_k = 0$  and only learned the  $k = 1..5$  weights  $\beta_k$  in Equation 5 (Note that the synergy index  $m$  was omitted as only a single synergy was used). For each of the  $N = 2$  Gaussians we learned the mean  $\mu$ , the bandwidth  $h$  and the amplitude  $a$  in Equation 7. Thus, in total  $5 + 2 \times 3 = 11$  parameters were learned. In contrast with DMPs 8 Gaussian amplitudes were optimized.

The  $\beta$  values of the DMPSynergies representation for the five via-points are shown in **Figure 4D** for 10 runs. New motor skills can be generated without re-learning via a simple linear interpolation. The resulting trajectories are shown in **Figure 4E**. However, this is only the case in the simple via-point task. For more complex tasks these  $\beta$  values have to be learned.



**FIGURE 4 | Results for the dynamic via-point task.** The goal of this simple multi-task learning problem is to pass through five via-points, denoted by the large dots in (A) and navigate to the target state at 1. The corresponding controls (accelerations) of this dynamical system are shown in (B). These five trajectories are simultaneously learned using DMPSynergies with a single synergy ( $M = 1$ ) represented by  $N = 2$  Gaussians and to an incremental

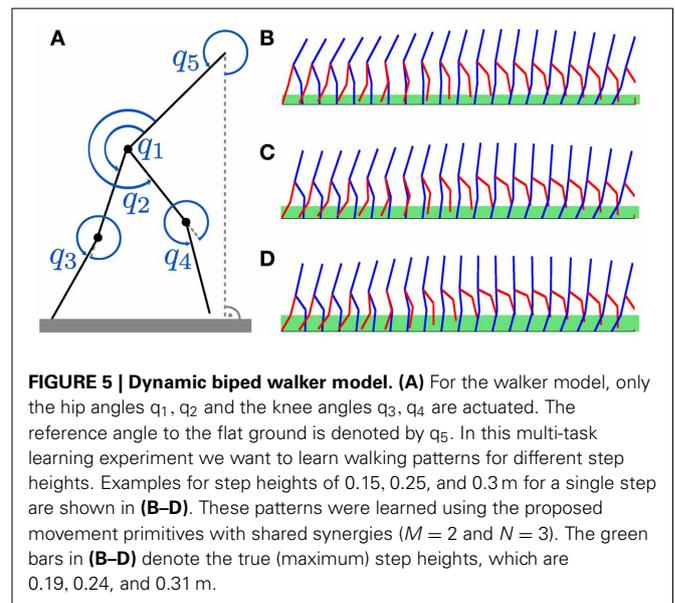
variant of DMPs in (C). For the DMP approaches each task (via-point) has to be learned separately. Thus, the two learning curves have five peaks. In contrast with DMPSynergies we could learn these five tasks at once, which resulted in faster overall convergence. The plot in (D) illustrates the mean and the standard deviation of the learned  $\beta$  values for the DMPSynergy approach. Via interpolating  $\beta$  and by reusing the learned synergy new motor skills can be generated without re-learning. This is illustrated in (E), where  $\beta \in [0.07, 0.34]$ .

**3.2. DYNAMIC BIPED WALKER TASK**

To evaluate the DMPSynergies on a multi-dimensional robotic task we learned multiple walking patterns using a 5 degree-of-freedom (DoF) dynamic biped robot model, which is shown in Figure 5A. We demonstrate that by exploiting the shared knowledge among multiple walking gaits, solutions could be found more robustly and more efficiently in terms of learning speed compared to single task learning. Further, the shared synergies could be used to generalize new skills. The model is only as complex as required to study difficulties like limb coordination, effective underactuation, hybrid dynamics or static instabilities. More details on the design and challenges can be found in Westervelt et al. (2004).

The 10-dimensional state  $\mathbf{q}_t = [q_{1:5}, \dot{q}_{1:5}]$  of the robot is given by the hip angles ( $q_1$  and  $q_2$ ), the knee angles ( $q_3$  and  $q_4$ ), a reference angle to the ground ( $q_5$ ), and the corresponding velocities  $\dot{q}_{1:5}$ . Only the hip and the knee angles are actuated. Thus, 4 dynamical systems in Equation 1 are used to generate desired trajectories for the linear feedback controller in Equation 4. A phase resetting strategy is implemented to facilitate learning (Nakanishi et al., 2004). At each impact of the swing leg the phase  $\phi$  in Equation 8 is set to zero. This increases the stability of the robot as the gait cycle duration is implicitly given by the impact time.

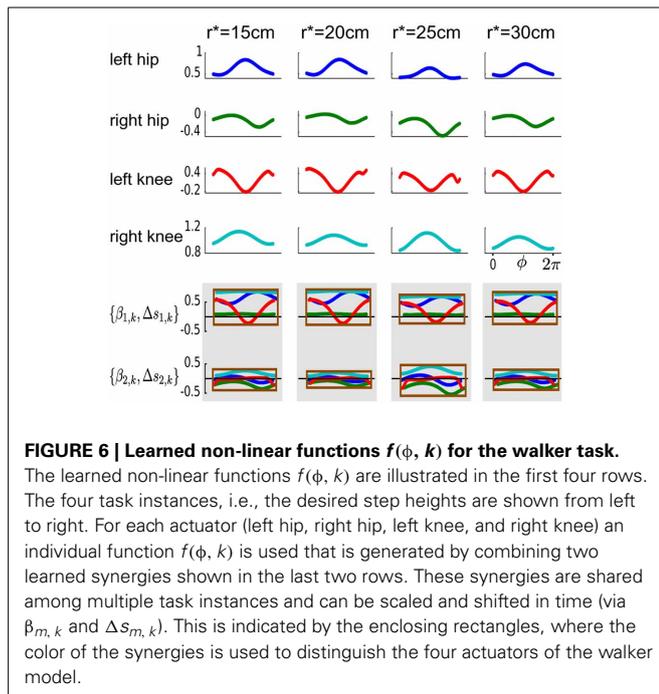
The initial state  $\mathbf{q}_1 \in \mathbb{R}^{10}$ , the goal state  $\mathbf{g}$  and the control gains in Equation 4 were optimized in advance for a desired step height of  $r^* = 0.2$  m to simplify learning. The resulting values are shown in Table A2 in the appendix. For rhythmic movements the goal state  $\mathbf{g} \in \mathbb{R}^5$  models an attractor point which is only specified for joint angles and not for velocities in Equation 1. As for the via-point reaching task, Gaussian noise with  $\sigma = 1$  is added to the



**FIGURE 5 | Dynamic biped walker model.** (A) For the walker model, only the hip angles  $q_1, q_2$  and the knee angles  $q_3, q_4$  are actuated. The reference angle to the flat ground is denoted by  $q_5$ . In this multi-task learning experiment we want to learn walking patterns for different step heights. Examples for step heights of 0.15, 0.25, and 0.3 m for a single step are shown in (B–D). These patterns were learned using the proposed movement primitives with shared synergies ( $M = 2$  and  $N = 3$ ). The green bars in (B–D) denote the true (maximum) step heights, which are 0.19, 0.24, and 0.31 m.

simulated controls. For Equation 1 we used the parameters  $\alpha_z = 2$ ,  $\beta_z = 0.5$  and  $\tau = 0.06$ . The initial parameter values and the applied ranges used for policy search are shown in Table A3 in the appendix.

In this multi-task learning experiment we want to learn walking patterns for different desired step heights:  $r_k^* \in \{0.15, 0.2, 0.25, 0.3\}$  m. Example patterns for step heights of 0.15, 0.25 and 0.3 m are shown in Figures 5B–D, where the green



**FIGURE 6 | Learned non-linear functions  $f(\phi, k)$  for the walker task.**

The learned non-linear functions  $f(\phi, k)$  are illustrated in the first four rows. The four task instances, i.e., the desired step heights are shown from left to right. For each actuator (left hip, right hip, left knee, and right knee) an individual function  $f(\phi, k)$  is used that is generated by combining two learned synergies shown in the last two rows. These synergies are shared among multiple task instances and can be scaled and shifted in time (via  $\beta_{m,k}$  and  $\Delta s_{m,k}$ ). This is indicated by the enclosing rectangles, where the color of the synergies is used to distinguish the four actuators of the walker model.

bars denote the maximum step heights during a single step (0.19, 0.24 and 0.31 m).

The objective function for a single walking task is given by the distance travelled in the sagittal plane, the duration of the simulation and deviations from the desired step height  $r_k^*$  with  $k = 1..4$ :

$$C(k) = -0.6(x_T - x_1) + 0.2(5 - T \cdot \Delta t) + 50 \sum_{i=1}^S (r_i - r_k^*)^2, \quad (11)$$

where  $x$  denotes the x-coordinate of the hip,  $S$  the number of steps and  $r_i$  the maximal step height during the  $i$ th step. We used a time step  $\Delta t = 2$  ms. The time horizon  $T \in [1, 5000]$  is given by the last valid state of the robot, where the biped does not violate the joint angle constraints specified by  $\mathbf{q}_{\min}$  and  $\mathbf{q}_{\max}$  in **Table A2** in the appendix.

With the proposed DMPSynergies the non-linear function  $f(\phi, k)$  in Equation 6 is generated by combining a set of learned synergies that are shared among multiple task instances, i.e., the four ( $k = 1..4$ ) desired step heights. This combination mechanism is illustrated for a representation using  $M = 2$  synergies modeled by  $N = 3$  Gaussians in **Figure 6**. For each actuator (left hip, right hip, left knee, and right knee) an individual function  $f(\phi, k)$  is generated, which is subsequently used to modulate an attractor system shown in Equation 1 to compute the desired movement trajectories. The shared synergies shown in the last two rows in **Figure 6** can be scaled and shifted in time. This is indicated by the enclosing rectangles. Note that the color of the synergies is used to distinguish the four actuators of the walker model.

We evaluated different movement primitive representations with increasing complexity compared to single-task learning using DMPs with  $N = 4$  and  $N = 8$  Gaussians. The average final

**Table 1 | Achieved costs for the walker task, where the standard deviation is denoted by the symbol  $\pm$ .**

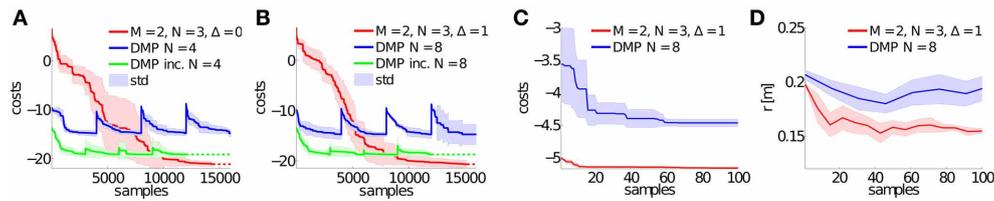
Setup	$\Delta$	$C_{\text{mean}}$	Best learned $r$ [m]	#
$M = 2$	0	$-20.3 \pm 1.2$	0.21, 0.21, 0.27, 0.27	56
$N = 2$	1	$-20.8 \pm 0.9$	0.19, 0.23, 0.28, 0.26	88
$M = 2$	0	$-21.4 \pm 0.4$	0.22, 0.24, 0.23, 0.26	80
$N = 3$	1	$-20.5 \pm 1.4$	0.22, 0.22, 0.26, 0.28	112
$M = 3$	0	$-19.0 \pm 2.3$	0.17, 0.19, 0.2, 0.26	84
$N = 2$	1	$-19.6 \pm 1.9$	0.19, 0.22, 0.19, 0.2	132
DMP $_{N=4}$		$-13.1 \pm 0.8$	0.17, 0.17, 0.23, 0.28	64
DMP $_{N=8}$		$-14.8 \pm 1.4$	0.15, 0.2, 0.23, 0.29	128
DMP inc. $_{N=4}$		$-19.2 \pm 0.6$	0.15, 0.20, 0.22, 0.3	64
DMP inc. $_{N=8}$		$-18.8 \pm 0.5$	0.18, 0.19, 0.31, 0.26	128

In the second column the symbol  $\Delta$  denotes if additionally 16M time-shift variables  $\Delta s$  are learned. The total number of parameters is denoted by the symbol #, where e.g., for the representation in the 1st row  $4M = 8$  task related weights and  $3 \cdot M \cdot N \cdot 4 = 48$  shared parameters were optimized. The best results with the lowest costs denoted by  $C_{\text{mean}}$  are highlighted in gray shading.

costs  $C_{\text{mean}}$  after learning over 10 runs are shown in **Table 1**. In the most simple representation we used  $M = 2$  synergies modeled by  $N = 2$  Gaussians. More complex representations implementing time-varying synergies are denoted by the symbol  $\Delta = 1$  in **Table 1**. Here, additionally the time-shifts  $\Delta s_{1:M}$  were learned for all synergies and all actuators. However, the final learning performance did not outperform the representation with fixed time-shifts (i.e.,  $M = 2$ ,  $N = 3$  and  $\Delta = 0$ :  $-21.4 \pm 0.4$  compared to  $M = 2$ ,  $N = 3$  and  $\Delta = 1$ :  $-20.5 \pm 1.4$ ). This can be also seen in **Figure 7**, where we plot the learning curve for synergies with  $\Delta s_{1:M} = 0$  in **Figure 7A** and the results for time-varying synergies in **Figure 7B**.

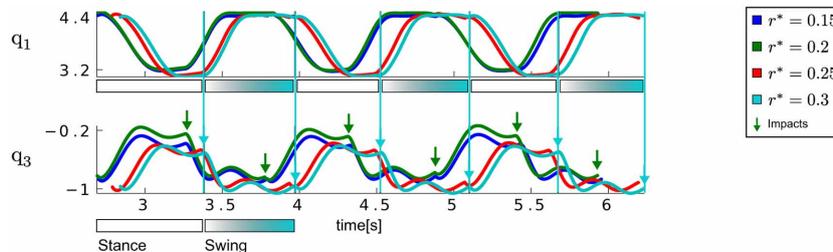
The average final cost value of the DMP representation is higher (i.e., DMP $_{N=8}$ :  $-14.8 \pm 1.4$ ) compared to the best costs achieved with shared synergies ( $M = 2$ ,  $N = 3$  and  $\Delta = 0$ :  $-21.4 \pm 0.4$ ). This holds also for an incremental learning setup (e.g., DMP inc. $_{N=4}$ :  $-19.2 \pm 0.6$ ), where DMPs were initialized with the best result from the previous task.

The joint angle trajectories of the left hip and knee joint for the DMPSynergy representation using  $M = 2$  synergies modeled by  $N = 3$  Gaussians and  $\Delta = 1$  are illustrated in **Figure 8**. The average step heights were  $r \in \{0.22, 0.22, 0.26, 0.28\}$ , which do not match the desired step heights  $r^* \in \{0.15, 0.2, 0.25, 0.3\}$ . The reason for this is that the objective function in Equation 11 is designed to prefer correct multi-step walking movements over exact matches of the step heights since learning to walk is already a complex learning problem (approximately 90% of the costs are determined by the travelled distance and only 5% are caused by the distance to the desired step heights). However, for the different desired step heights the shape of the trajectories as well as the moment of the impact vary. The moments of impact are denoted by arrows in **Figure 8**.



**FIGURE 7 | Learning curves for the biped walker task.** This figure illustrates the learning performance over 10 runs of the proposed approach using  $M = 2$  synergies with  $N = 3$  Gaussian basis functions. In **(A)** the time-shift variables  $\Delta s$  are not learned and set to zero. Whereas, in **(B)** also these  $\Delta s$  variables are adapted during learning. We compare to the dynamic movement primitives (DMP) with  $N = 4$  Gaussians in **(A)** and to DMPs with  $N = 8$  Gaussians in **(B)**. *DMP inc.*

denotes an incremental learning setup, where DMPs were initialized with the best result from the previous task. Generalizing to a new step height ( $r^* = 0.1$  m) is shown in **(C)**, where we applied the best learned policy for DMPsSynergies from **(B)** and only optimized the weights  $\beta_{1:2}$  for the two (fixed) synergies. The corresponding average step height over all steps is shown in **(D)**. We compare to DMPs with  $N = 8$  Gaussians.



**FIGURE 8 | Results for the biped walker task.** This figure illustrates the (initially) left hip angle denoted by  $q_1$  and the left knee angle ( $q_3$ ) for the multi-task learning scenario. Shown are the best learned trajectories using the proposed approach (with  $M = 2$ ,  $N = 3$  and  $\Delta = 1$ ) for the desired step heights of  $r^* \in \{0.15, 0.2, 0.25, 0.3\}$ . The

true step heights of the learned walking patterns are  $0.22 \pm 0.07, 0.22 \pm 0.08, 0.26 \pm 0.08, 0.28 \pm 0.08$ . The points in time of the ground contacts are denoted by large arrows for desired step heights of 0.25 m and 0.3 m. For the later additionally the duration of the stance and the swing phases are illustrated by large boxes.

While generalizing to new motor skills was straightforward for the simple via-point task, for the walking tasks a linear interpolation turns out to be ineffective. We therefore demonstrate in **Figures 7C,D** how a new walking pattern for a desired step height of  $r^* = 0.1$  m can be learned by reusing the previously learned prior knowledge (taking the best solution for  $r^* = 0.25$  m) for  $M = 2$  synergies modeled by  $N = 3$  Gaussians and  $\Delta = 1$ . Only the weights  $\beta_{1:M}$  are optimized in this experiment, keeping the learned time-shifts fixed. The costs in **Figure 7C** and the average step height  $r$  in **Figure 7D** demonstrate the advantage of using a fixed prior, where we compare to DMPs with  $N = 8$  Gaussians.

### 3.3. MULTI-DIRECTIONAL REACHING TASK WITH A MUSCULOSKELETAL MODEL OF THE HUMAN ARM

A simplified model of a human arm based on the model by Holzbaur et al. (2005) was used to learn six reaching tasks simultaneously. The shoulder and the elbow joint were modeled by hinge joints. Thus, only movements in the sagittal plane were possible. The initial arm configuration and the six target locations (with a distance of 15 cm to a marker placed on the radial stylium) are shown in **Figure 9A**. A learned example movement is illustrated in **Figure 9B**, where the cylinders, the spheres and the ellipsoids denote wrapping surfaces discussed in Subsection 2.3.

We focused on fast reaching movements of 500 ms duration ( $T = 500$  and  $\Delta t = 1$  ms) that can be implemented in an

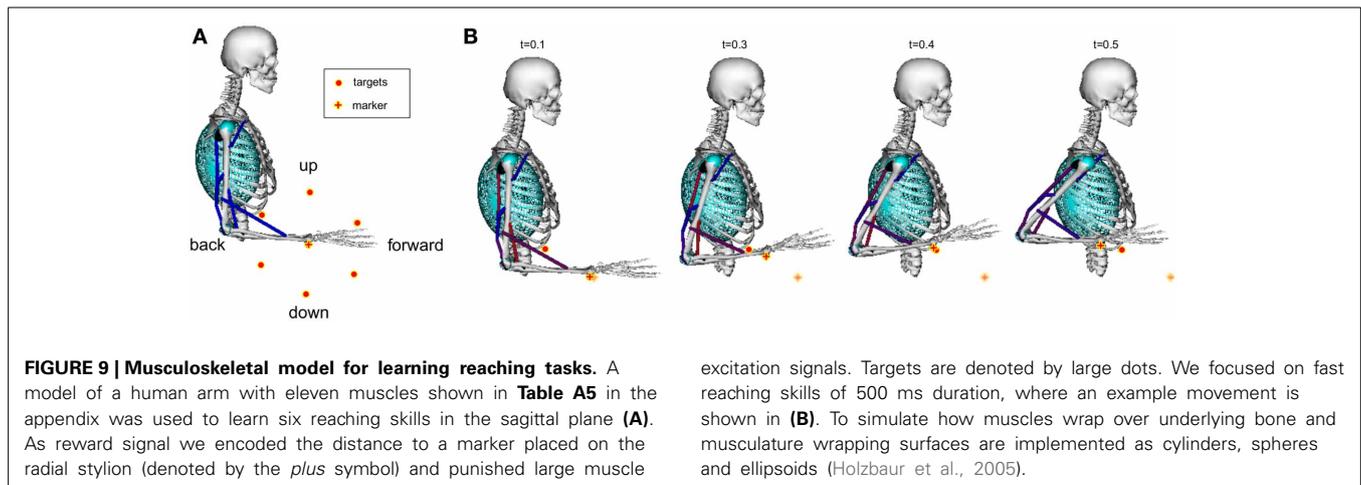
open-loop control scheme. Note that with our approach also closed-loop systems with feedback could be implemented, as discussed below. Thus, the learnable non-linear function  $f(s, k)$  in Equation 10 is directly used as input to the system in form of muscle excitation patterns. The parameter settings for learning are shown in **Table A4** in the appendix.

For learning the reaching tasks we evaluated the Euclidean distance of a marker  $\mathbf{v}_k(t)$  placed on the radial stylium to a given target  $\mathbf{g}_k$ , where  $k = 1..6$  denotes the task index. Additionally, large muscle excitations signals are punished:

$$C(k) = \sum_{k=1}^6 3 \cdot \frac{1}{T} \sum_{t=1}^T \|\mathbf{g}_k - \mathbf{v}_k(t)\| + 10^{-3} \int_{s=0}^1 \mathbf{f}(s, k)^T \mathbf{f}(s, k) ds, \quad (12)$$

where  $\|\cdot\|$  denotes the Euclidean distance between the marker  $\mathbf{v}_k(t)$  and the target  $\mathbf{g}_k$  at time  $t$ .

We evaluated five movement representations, defined in Equation 10, with an increasing number of shared synergies, i.e.,  $M = \{1, 2, 3, 4, 5\}$ . Each synergy is represented by a single ( $N = 1$ ) Gaussian. For each target and for each synergy the task-specific parameters  $\beta_{k,m}$  and  $\Delta_{s,k,m}$  are learned. The number of task-specific and the number of task-invariant or shared parameters is shown in **Table 2**.



excitation signals. Targets are denoted by large dots. We focused on fast reaching skills of 500 ms duration, where an example movement is shown in (**B**). To simulate how muscles wrap over underlying bone and musculature wrapping surfaces are implemented as cylinders, spheres and ellipsoids (Holzbaur et al., 2005).

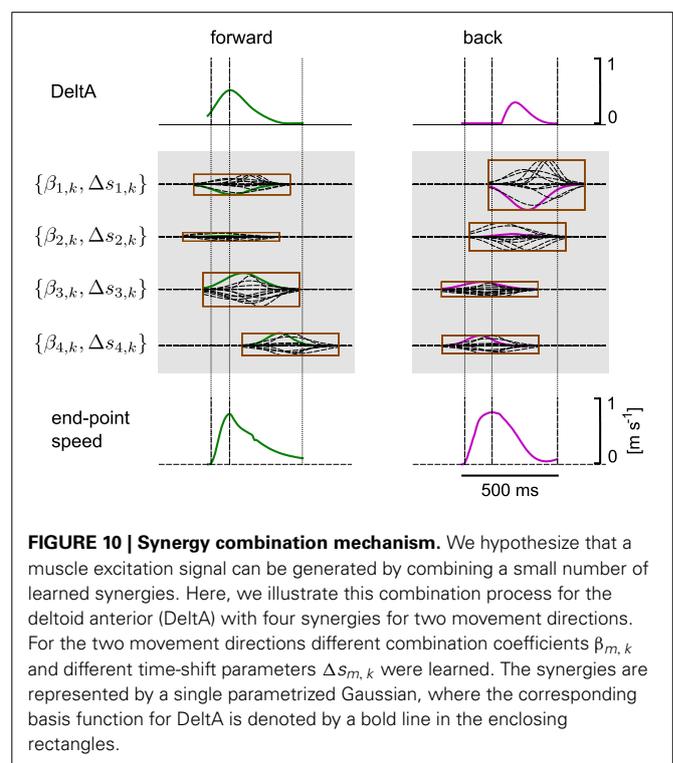
**Table 2 | Details of the evaluated parametrizations and achieved costs for the reaching task.**

No. of synergies	$C_{mean}$	$\#_K$	$\#_M$	Total
$M = 1$	$2.38 \pm 0.05$	12	33	45
$M = 2$	$1.52 \pm 0.12$	24	66	90
$M = 3$	$1.15 \pm 0.12$	36	99	135
$M = 4$	$1.15 \pm 0.05$	48	132	180
$M = 5$	$1.17 \pm 0.05$	60	165	225

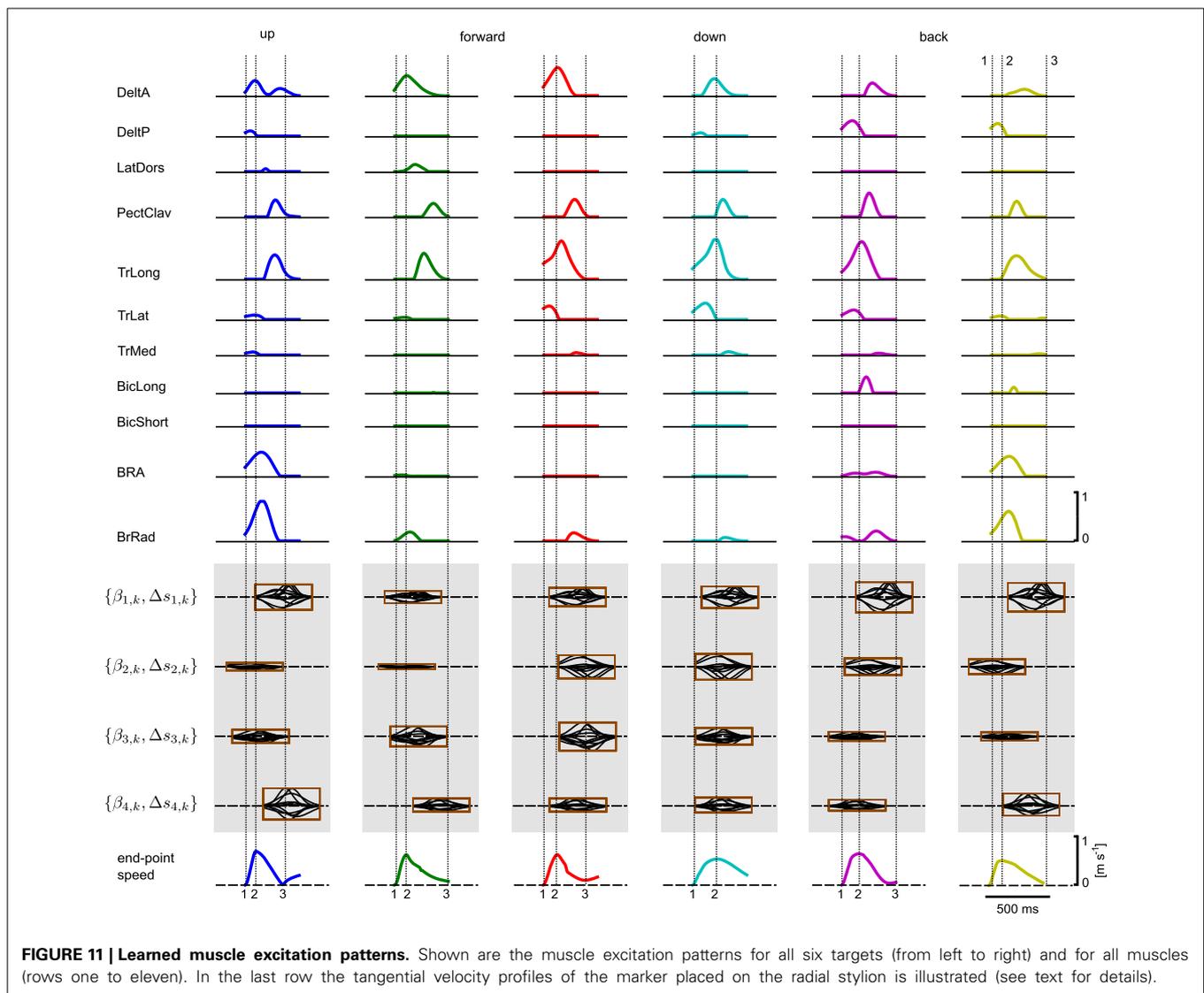
$M$  denotes the number of implemented synergies,  $C_{mean}$  the final cost values, and the symbol  $\pm$  the standard deviation. We use  $\#_K$  to denote the number of task-specific parameters ( $6 \cdot M$ ) and  $\#_M$  to denote the number of task-invariant or shared parameters ( $3 \cdot 11 \cdot M$ ).

We hypothesized that a muscle excitation signal can be generated by combining a small number of learned synergies. An example for the anterior deltoid muscle (DeltA) is shown in **Figure 10** for two movement directions. Here, DMPSynergies with  $M = 4$  synergies were used to generate the muscle excitation patterns. The muscle excitation patterns for all six movement directions and all eleven muscles are shown in **Figure 11**. Two observations can be made: first, as our objective function in Equation 12 punishes large muscle excitation signals a sparse representation of multiple motor skills is learned. Second, the learned muscle patterns partially show the typical triphasic behavior of human movement (Angel, 1975; Hallett et al., 1975; Berardelli et al., 1996; Chiovetto et al., 2010), where individual muscles (e.g., DeltA, PectClav and BRA in the first column in **Figure 11**) become activated at the onset of the movement, shortly before the velocity peak to decelerate, and finally, multiple muscles co-contract at the target location. These three events are denoted by the labels 1, 2, and 3 in the last row in **Figure 11**, where a threshold of  $2 \text{ cm s}^{-1}$  was used to determine the movement onset and the termination of the movement.

Comparing all five movement representations ( $M = \{1, 2, 3, 4, 5\}$ ), we found that at least three synergies were necessary to accomplish all reaching tasks. This is shown in **Figure 12A**, where with only one ( $M = 1$ ) or two synergies ( $M = 2$ ) not all targets can be reached. Shown are the marker trajectories of three independent learning sessions (out of ten



runs). Note that similar findings were obtained in analyzing human arm reaching movements, where four to five synergies were observed (d'Avella et al., 2006). The corresponding learning curves for all five movement representations are shown in **Figure 12B**, where the parametrizations with  $M = 3.5$  synergies perform equal. This is also reflected in the final costs shown in **Table 2** (rows 3..5). As an example the marker trajectories and the tangential velocity profiles for the representation using  $M = 4$  synergies are illustrated in **Figure 12C**. As we evaluated an open-loop control scheme these marker trajectories did not exactly terminate at the target location (after the limited number of episodes for learning). However, by increasing the number of episodes or by adding feedback the terminal accuracy could be improved.



**FIGURE 11 | Learned muscle excitation patterns.** Shown are the muscle excitation patterns for all six targets (from left to right) and for all muscles (rows one to eleven). In the last row the tangential velocity profiles of the marker placed on the radial stylium is illustrated (see text for details).

For testing the generalization ability of DMPSynergies we rotated all six targets by 30 degrees and only re-learned the task-specific coefficients, i.e., the mixing coefficients  $\beta_{m,k}$  and the time-shift parameters  $\Delta s_{m,k}$ . Interim solutions with a movement representation implementing  $M = 4$  synergies are shown in **Figure 13A**. Note that, as we evaluated an open-loop controller, the rotated targets were unknown to the controller. Solely the objective function in Equation 12 quantifies deviations from the targets. After 15 episodes a first trend toward the new targets was visible, however, most of the trajectories (three learned solutions are illustrated) ended at the original targets. The corresponding learning curves for DMPSynergies with three ( $M = 3$ ) and four ( $M = 4$ ) synergies are shown in **Figure 13B**. The learning curve for the unperturbed scenario from the previous experiment is denoted by the dashed line ( $M = 4$  orig.). Note that in both - the unperturbed and the perturbed experiments  $K = 6$  reaching movements were learned, which demonstrates the benefit of the shared learned knowledge when generalizing new skills. For a comparison the blue line denoted by DMP  $N = 4$  illustrates

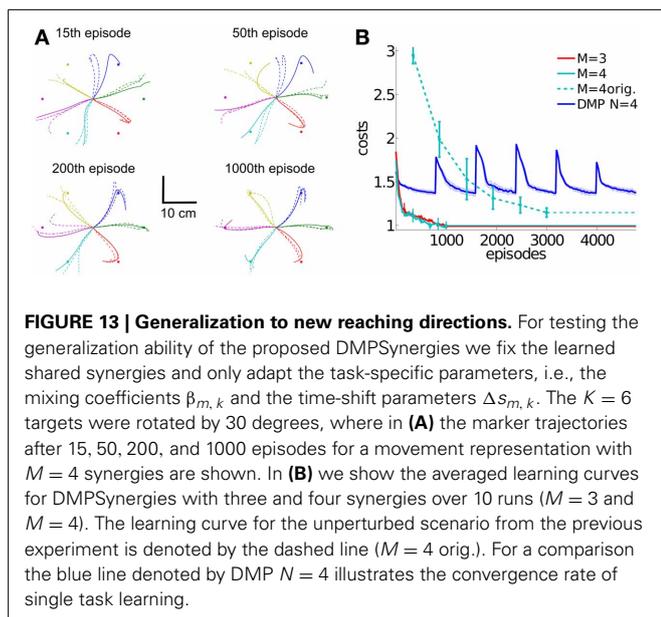
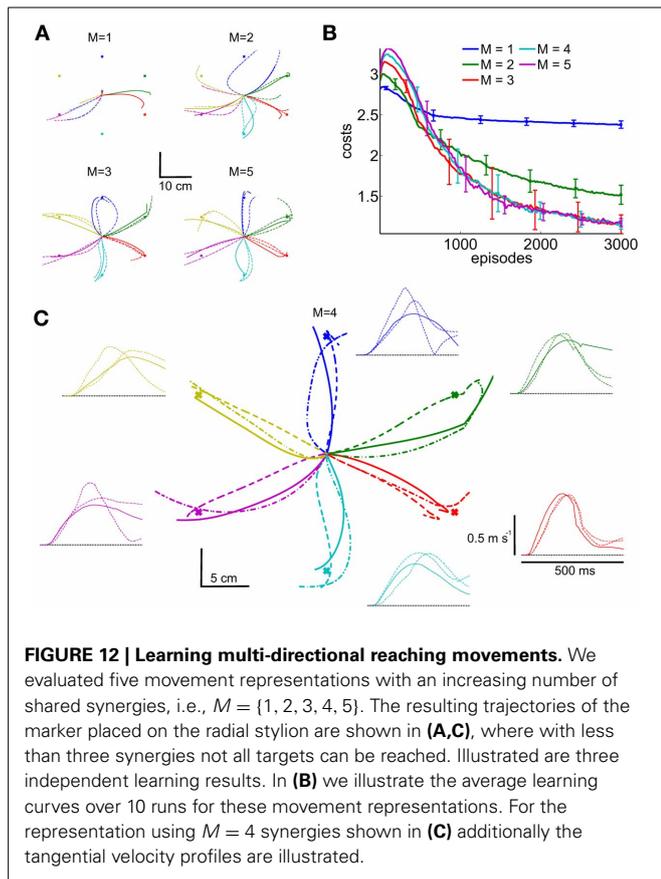
the convergence rate of single task learning with DMPs, where DMPSynergies ( $M = 4$  orig.) can compete in terms of learning speed.

## 4. DISCUSSION

We proposed a movement representation based on learned parametrized synergies (DMPSynergies) that can be linearly combined and shifted in time. These learned synergies are shared among multiple task instances significantly facilitating learning of motor control policies. This was demonstrated on simulated robotic and on musculoskeletal systems. Below we discuss the significance and the implication of our findings with respect to robotics and biological motor control.

### 4.1. EXPLOITING SHARED SYNERGIES FOR MOTOR SKILL LEARNING IN ROBOTICS

For motor skill learning in robotics a common strategy is to use parametrized elementary movements or movement primitives (Kober and Peters, 2011). In this paper we proposed



a generalization of the most widely used movement primitive representation in robotics, dynamic movement primitives (DMPs) (Schaal et al., 2003; Ijspeert et al., 2013). DMPs evaluate parametrized dynamical systems to generate trajectories. The dynamical system is constructed such that the system is

stable. This movement representation has many advantages. It is a model-free approach, partially explaining its popularity in robotics as model learning in high-dimensional stochastic robotics systems is challenging. Further, its stable attractor system facilitates learning and DMPs can represent both rhythmic and discrete movements. Meta parameters can be used for adapting the movement speed or the goal state. Finally, the movement representation depends linearly on the policy parametrization, i.e., the learnable function  $f$  depends linearly on the parameters  $\theta$  of the movement primitive:  $f(s) = \Phi(s)^T \theta$ , where  $s$  is the time or phase variable. As a result, imitation learning for DMPs is straightforward, as this can simply be done by performing linear regression (Schaal et al., 2003). However, for each task  $k$  an individual set of parameters  $\theta_k$  has to be learned, which unnecessarily complicates learning of a large number of related motor skills. In contrast we proposed a generalization that allows for reusing shared knowledge among multiple related motor skills, i.e., the parameter vector  $\theta$  is task-invariant.

In particular, we replaced the non-linear modulation function  $f(\cdot)$  in DMPs by a hierarchical function approximator. On the lower level task related parameters (amplitude scaling weights and time-shift parameters) are used to modulate a linear superposition of basis functions. These basis functions encode shared higher level knowledge and are modeled by a mixture of Gaussians. With the proposed DMPSynergies representation discrete and rhythmic movements can be generated. By using Gaussians at the higher level DMPs can be implemented as special case. However, the DMPSynergies can compete with DMPs in terms of learning efficiency while allowing for learning multiple motor skills simultaneously.

This was demonstrated in two robotic multi-task learning scenarios, where we showed that, with the DMPSynergies, good policies could be found more reliably, i.e., local minima with high cost values were more often avoided, more efficiently (fewer samples were needed), and new skills could be generalized by exploiting the previously learned shared knowledge. A simple via-point task was used to demonstrate the characteristics of the approach, where the proposed movement representation could be used to generalize new movement trajectories by applying a linear interpolation on the synergy's weights  $\beta$ . In a second robotic task, a biped walker task, the hierarchical representation was used to learn walking patterns with multiple step heights. In this complex reinforcement learning task, it was shown that better solutions were found more reliably by exploiting the learned shared knowledge, which is a strong feature of a movement representation. While also with the classical DMP approach high quality movement gaits were learned, on average the achieved costs were higher compared to the proposed hierarchical synergies representation, i.e.,  $-19.2 \pm 0.6$  for DMPs with 4 Gaussians (and with incremental learning) compared to  $-21.4 \pm 0.4$  when using  $M = 2$  synergies with  $N = 3$  Gaussians (where the time-shift parameters were fixed and set to zero,  $\Delta = 0$ ). In this experiment 10,000 samples were needed to learn 4 walking gaits simultaneously, where the DMPSynergies approach can compete with DMPs (15,000 samples). Additionally, we demonstrated in a generalization experiment that walking patterns for an unknown step height ( $r^* = 0.1$  m) could be learned

with 100 samples by exploiting the previously learned prior knowledge.

While DMPs (Schaal et al., 2003; Ijspeert et al., 2013) are most closely related to our shared synergies approach, there exist a few other approaches (Chhabra and Jacobs, 2006; Alessandro et al., 2012) also implement shared knowledge. In Chhabra and Jacobs (2006) a variant of non-negative matrix factorization (d'Avella et al., 2003) was used to compute the synergies given a set of trajectories created by applying stochastic optimal control methods (Li and Todorov, 2004). In Alessandro et al. (2012) an exploration phase was introduced to compute the dynamic responses of a robot system with random initialization. After a reduction phase, where a small number of proto-tasks were executed, a reduced set of dynamic responses was used to compute the synergies matrix by solving a linear system of equations. We proposed an alternative for learning the synergies and their combination parameters, where all unknowns are learned in a reinforcement learning setting from a single sparse reward signal. Moreover, for robotic tasks we embed the synergies approach in stable dynamical systems like in DMPs. This combines the benefits of DMPs and muscle synergies, namely the efficient learning ability of DMPs in high-dimensional systems and the hierarchical representation of movements that can be used for multi-task learning.

As with DMPs the complexity of the DMPSynergies representation can be scaled by the number of combined synergies or the number of implemented Gaussians modeling these synergies. However, as the trajectories generated with our representation depend non-linearly on the policy parameters (in contrast to DMPs) more sophisticated decomposition strategies like for example d'Avella and Tresch (2001); Chiovetto et al. (2013) are needed for imitation learning. With such approaches the extracted synergies could be implemented as initial solutions in our learning framework.

## 4.2. LEARNED SHARED SYNERGIES FOR BIOLOGICAL MOVEMENT GENERATION

The idea of reusing shared knowledge for movement generation is a well-known concept in biological motor control. Muscle activation patterns recorded during multiple task instances of natural motor behavior, i.e., fast reaching movements of humans (d'Avella et al., 2006), primate grasping movements (Overduin et al., 2008), or walking patterns (Dominici et al., 2011), could be efficiently modeled by combining only few muscle activation patterns. In particular, time-varying muscle synergies (d'Avella et al., 2003; Bizzi et al., 2008) were proposed to be a compact representation of muscle activation patterns. The key idea of this approach is that muscle activation patterns are linear sums of simpler, elemental functions or synergies. Each muscle synergy can be shifted in time and scaled with a linear factor to construct a large variety of activation patterns. In this manuscript we proposed a generative model to represent and learn time-varying synergies (d'Avella et al., 2006).

The proposed framework allows for studying the concept of muscle synergies from a generative perspective in contrast to the analytical approach, where muscle synergies are identified from observed data. Applying such a generative approach to

a musculoskeletal model, we could provide a proof-of-concept of the feasibility of a low-dimensional controller based on shared synergies and a demonstration of its learning efficiency. Moreover, we could ask different question, i.e., how does performance scale with the complexity of the movement representation, how sparse is the encoding of the muscle patterns to solve particular tasks, and how well does the learned representation generalize to new movements? We addressed these questions in a multi-directional reaching task, where we investigated a musculoskeletal model of the upper limb with 11 muscles. Motor skills for 6 reaching directions were learned within 3000 episodes and by exploiting the learned shared synergies movements for rotated target directions can be generalized 3 times faster (Figure 13). We found that a minimum of three synergies were necessary to solve the task (Figure 12B). In our objective function large muscle excitation signals were punished, which resulted in a sparse representation of muscle excitation patterns. This sparse representation illustrated in Figure 11 shows similarities to observed electromyographic activity recorded in related human reaching tasks (d'Avella et al., 2006), i.e., triphasic muscle patterns, where some of the muscles contributed at the movement onset, some at point of the maximum tangential velocity, and some at the end of the movement to co-contract. However, sensor feedback might be an important modulation signal to make this effect more pronounced.

The model was designed to capture salient features of human musculoskeletal system, such as muscle activation dynamics, Hill-type musculotendinous units, realistic geometry. However, to reduce the computational effort needed to simulate a movement we made a few simplifying assumptions. First, a limited number of muscles (11) were implemented, where simplified wrapping objects and muscle paths were modeled. Further, we implemented the shoulder and the elbow joint as hinge joints. Thus, only reaching movements in the sagittal plane could be performed. Finally, we focused on fast reaching movements in an open-loop control scheme. This was a valid assumption for comparing to human data for fast reaching movements (d'Avella et al., 2006). However, our proposed learning and control framework also allows for implementing closed-loop controllers, i.e., when introducing an inverse kinematics model  $\Psi \in \mathbb{R}^{D \times 3}$  in Equation 1, i.e.,  $\tau \dot{\mathbf{z}} = \Psi(\alpha_z(\beta_z(\mathbf{g} - \mathbf{y}^*) - \mathbf{z})) + \mathbf{f}$ , where  $D$  denotes the number of actuators and we assumed that the goal state  $\mathbf{g}$  lives in a three-dimensional Cartesian space. The inverse kinematics model  $\Psi$  maps the feedback error signal into the muscle pattern space and modulates the learned muscle excitation basis  $\mathbf{f} \in \mathbb{R}^D$ . With such closed-loop systems we might better understand the contribution of feedback to muscle control in biological movement generation (Lockhart and Ting, 2007).

Musculoskeletal models have been used before to investigate movement generation with muscle synergies (Berniker et al., 2009; Neptune et al., 2009; McKay and Ting, 2012). Berniker and colleagues used model-order reduction techniques to identify synergies as a low-dimensional representation of a non-linear system's input/output dynamics and optimal control to find the activations of these synergies necessary to produce a range of movements. They found that such a set of synergies was capable of producing effective control of reaching movements with

a musculoskeletal model of a frog limb and that it was possible to build a relatively simple controller whose overall performance was close to that of the system's full-dimensional non-linear controller. Neptune and colleagues generated muscle-actuated forward dynamics simulations of normal walking using muscle synergies identified from human experimental data using non-negative matrix factorization as the muscle control inputs. The simulation indicated that a simple neural control strategy involving five muscle synergies was sufficient to perform the basic sub-tasks of walking. McKay and Ting, studying an unrestrained balance task in cats, used a static quadrupedal musculoskeletal model of standing balance to identify patterns of muscle activity that produced forces and moments at the center of mass (CoM) necessary to maintain balance in response to postural perturbations. CoM control could be accomplished with a small number of muscle synergies identified from experimental data, suggesting that muscle synergies can achieve similar kinetics to the optimal solution, but with increased control effort compared to individual muscle control. In line with these simulation studies, we also found that a small number of muscle synergies was sufficient to perform multiple reaching tasks in a forward dynamic simulation of a musculoskeletal model. However, we did not use experimental data or model-order reduction techniques to identify muscle synergies. In our framework, both synergy structural parameters and synergy combination parameters were found with reinforcement learning, supporting the generality of the solutions identified. Moreover, we were able to test the generalization ability of the synergies in the same framework by optimizing only the task-specific synergy combination parameters.

The proposed reinforcement learning framework with movement primitives relates to optimal control approaches in the biological motor control literature (Delp et al., 2007; Erdemir et al., 2007). In these simulation studies muscle patterns are parametrized by e.g., bang-bang (on-off) controls, constant control values, or control vectors approximated with polynomials [see Table 2 in Erdemir et al. (2007) for an overview of different control strategies]. However, to the best of our knowledge none of these approaches implemented shared synergies as control signal representation for learning multiple task instances simultaneously. Even with complex representations, e.g., with  $M = 5$  synergies learning 225 parameters converged within 3000 episodes, which is a promising feature of the proposed approach for studies on more complex musculoskeletal models.

In this manuscript we demonstrated how time-varying synergies (d'Avella et al., 2006) can be implemented and learned from

scratch. Interestingly, by adding an additional constraint on the movement representation, i.e., by using a single policy vector for all actuators anechoic mixing coefficients (Giese et al., 2009) can be implemented. However, in general any synergy representation such as synchronous synergies (Ivanenko et al., 2004; Dominici et al., 2011) used for locomotion can be learned. Thus, we do not argue for a particular synergy representation. Our representation was motivated to extend the widely used DMPs (Schaal et al., 2003) for exploiting shared task-invariant knowledge for motor skill learning in robotics.

## 5. CONCLUSION

We proposed a movement primitive representation implementing shared knowledge in form of learned synergies. The representation competes with the state-of-the-art, it can implement DMPs (Schaal et al., 2003) as a special case, and it allows for an efficient generalization to new skills. Importantly, shared knowledge simplifies policy search in high-dimensional spaces, which was demonstrated in a dynamic biped walking task. Further, the proposed learned synergies are a compact representation of high-dimensional muscle excitation patterns, which allows us to implement reinforcement learning in musculoskeletal systems. In such frameworks muscle patterns are learned from scratch using a sparse reward signal, where we could investigate how muscles and muscle synergies contribute to a specific task, how complex a task-invariant representation must be, and how well the learned synergies generalize to changes in the environment. In a multi-directional arm reaching experiment we provided first insights to these questions. In future research the proposed movement generation and learning framework will be used to study feedback signals and feedback delays, imitation learning from biological data and the effect of simulated muscle synergies.

## ACKNOWLEDGMENTS

This paper was written under partial support by the European Union project FP7-248311 (AMARSI) and project IST-2007-216886 (PASCAL2). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <http://www.frontiersin.org/journal/10.3389/fncom.2013.00138/abstract>

## REFERENCES

- Alessandro, C., Carbajal, J., and d'Avella, A. (2012). "Synthesis and adaptation of effective motor synergies for the solution of reaching tasks," in *From Animals to Animats (SAB 2012)*, Vol. 7426 *Lecture Notes in Computer Science*, eds T. Ziemke, C. Balkenius, and J. Hallam (Denmark: Odense), 33–43. doi: 10.1007/978-3-642-33093-3\_4
- Angel, R. W. (1975). Electromyographic patterns during ballistic movement of normal and spastic limbs. *Brain Res.* 99, 387–392. doi: 10.1016/0006-8993(75)90042-6
- Berardelli, A., Hallett, M., Rothwell, J. C., Agostino, R., Manfredi, M., Thompson, P. D., et al. (1996). Single-joint rapid arm movements in normal subjects and in patients with motor disorders. *Brain* 119, 661–674. doi: 10.1093/brain/119.2.661
- Berniker, M., Jarc, A., Bizzi, E., and Tresch, M. C. (2009). Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics. *Proc. Natl. Acad. Sci. U.S.A.* 106, 7601–7606. doi: 10.1073/pnas.0901512106
- Bizzi, E., Cheung, V., d'Avella, A., Saltiel, P., and Tresch, M. (2008). Combining modules for movement. *Brain Res. Rev.* 57, 125–133. doi: 10.1016/j.brainresrev.2007.08.004
- Chadwick, E., Blana, D., van den Bogert, A., and Kirsch, R. (2009). A real-time, 3-d musculoskeletal model for dynamic simulation of arm movements. *Biomed. Eng. IEEE Trans.* 56, 941–948. doi: 10.1109/TBME.2008.2005946

- Chhabra, M., and Jacobs, R. A. (2006). Properties of synergies arising from a theory of optimal motor behavior. *Neural Comput.* 18, 2320–2342. doi: 10.1162/neco.2006.18.10.2320
- Chiovetto, E., Berret, B., and Pozzo, T. (2010). Tri-dimensional and triphasic muscle organization of whole-body pointing movements. *Neuroscience* 170, 1223–1238. doi: 10.1016/j.neuroscience.2010.07.006
- Chiovetto, E., d'Avella, A., and Giese, M. (2013). "A unifying framework for the identification of kinematic and electromyographic motor primitives," in *Abstracts of Neural Control of Movement Conference (NCM 2013)*, Puerto Rico.
- d'Avella, A., and Bizzi, E. (2005). Shared and specific muscle synergies in natural motor behaviors. *Proc. Natl. Acad. Sci. U.S.A.* 102, 3076–3081. doi: 10.1073/pnas.0500199102
- d'Avella, A., and Pai, D. K. (2010). Modularity for sensorimotor control: evidence and a new prediction. *J. Motor Behav.* 42, 361–369. doi: 10.1080/00222895.2010.526453
- d'Avella, A., Portone, A., Fernandez, L., and Lacquaniti, F. (2006). Control of fast-reaching movements by muscle synergy combinations. *J. Neurosci.* 26, 7791–7810. doi: 10.1523/JNEUROSCI.0830-06.2006
- d'Avella, A., Saltiel, P., and Bizzi, E. (2003). Combinations of muscle synergies in the construction of a natural motor behavior. *Nat. Neurosci.* 6, 300–308. doi: 10.1038/nn1010
- d'Avella, A., and Tresch, M. C. (2001). "Modularity in the motor system: decomposition of muscle patterns as combinations of time-varying synergies," in *Advances of Neural Information Processing Systems (NIPS 2001)*, eds T. G. Dietterich, S. Becker, and Z. Ghahramani (Vancouver, BC: MIT Press), 141–148.
- Delp, S., Anderson, F., Arnold, A., Loan, P., Habib, A., John, C., et al. (2007). Opensim: open-source software to create and analyze dynamic simulations of movement. *Biomed. Eng. IEEE Trans.* 54, 1940–1950. doi: 10.1109/TBME.2007.901024
- Dominici, N., Ivanenko, Y. P., Cappellini, G., d'Avella, A., Mondì, V., Cicchese, M., et al. (2011). Locomotor primitives in newborn babies and their development. *Science* 334, 997–999. doi: 10.1126/science.1210617
- Erdemir, A., McLean, S., Herzog, W., and van den Bogert, A. J. (2007). Model-based estimation of muscle forces exerted during movements. *Clin. Biomech.* 22, 131–154. doi: 10.1016/j.clinbiomech.2006.09.005
- Garner, B., and Pandy, M. (2001). Musculoskeletal model of the upper limb based on the visible human male dataset. *Comput. Methods Biomech. Biomed. Eng.* 4, 93–126. doi: 10.1080/10255840008908000
- Giese, M. A., Mukovskiy, A., Park, A.-N., Omlor, L., and Slotine, J.-J. E. (2009). "Real-time synthesis of body movements based on learned primitives," in *Statistical and Geometrical Approaches to Visual Motion Analysis*, Vol. 5604 *Lecture Notes in Computer Science*, eds D. Cremers, B. Rosenhahn, A. L. Yuille, and F. R. Schmidt (Berlin, Heidelberg, Springer), 107–127. doi: 10.1007/978-3-642-03061-1\_6
- Hallett, M., Shahani, B. T., and Young, R. R. (1975). EMG analysis of stereotyped voluntary movements in man. *J. Neurol. Neurosurg. Psychiatry* 38, 1154–1162. doi: 10.1136/jnnp.38.12.1154
- Hansen, N., Muller, S., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* 11, 1–18. doi: 10.1162/106365603321828970
- Holzbaur, K. R., Murray, W. M., and Delp, S. L. (2005). A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Ann. Biomed. Eng.* 33, 829–840. doi: 10.1007/s10439-005-3320-7
- Ijspeert, A., Nakanishi, J., Pastor, P., Hoffmann, H., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput.* 25, 328–373. doi: 10.1162/NECO\_a\_00393
- Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2002). "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, eds S. Becker, S. Thrun, and K. Obermayer (Vancouver, BC: MIT Press), 1523–1530.
- Ivanenko, Y. P., Poppele, R. E., and Lacquaniti, F. (2004). Five basic muscle activation patterns account for muscle activity during human locomotion. *J. Physiol.* 556, 267–282. doi: 10.1113/jphysiol.2003.057174
- Kober, J., Oztop, E., and Peters, J. (2010). "Reinforcement learning to adjust robot movements to new situations," in *Proceedings of the 2010 Robotics: Science and Systems Conference (RSS 2010)*, (Zaragoza), 2650–2655.
- Kober, J., and Peters, J. (2011). Policy search for motor primitives in robotics. *Mach. Learn.* 84, 171–203. doi: 10.1007/s10994-010-5223-6
- Li, W., and Todorov, E. (2004). "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2004)*, (Setúbal), 222–229.
- Lockhart, D. B., and Ting, L. H. (2007). Optimal sensorimotor transformations for balance. *Nat. Neurosci.* 10, 1329–1336. doi: 10.1038/nn1986
- McKay, J. L., and Ting, L. H. (2012). Optimization of muscle activity for task-level goals predicts complex changes in limb forces across biomechanical contexts. *PLoS Comput. Biol.* 8:e1002465. doi: 10.1371/journal.pcbi.1002465
- Meier, F., Theodorou, E., Stulp, F., and Schaal, S. (2011). "Movement segmentation using a primitive library," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)* (San Francisco, CA), 3407–3412.
- Mülling, K., Kober, J., Kroemer, O., and Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *Int. J. Robot. Res.* 32, 263–279. doi: 10.1177/0278364912472380
- Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robot. Auton. Syst.* 47, 79–91. doi: 10.1016/j.robot.2004.03.003
- Neptune, R. R., Clark, D. J., and Kautz, S. A. (2009). Modular control of human walking: a simulation study. *J. Biomech.* 42, 1282–1287. doi: 10.1016/j.jbiomech.2009.03.009
- Overduin, A., d'Avella, A., Roh, J., and Bizzi, E. (2008). Modulation of muscle synergy recruitment in primate grasping. *J. Neurosci.* 28, 880–892. doi: 10.1523/JNEUROSCI.2869-07.2008
- Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). "Learning and generalization of motor skills by learning from demonstration," in *International Conference on Robotics and Automation (ICRA 2009)*, (Kobe). doi: 10.1109/ROBOT.2009.5152385
- Peters, J., and Schaal, S. (2008). Reinforcement learning of motor skills with policy gradients. *Neural Netw.* 21, 682–697. doi: 10.1016/j.neunet.2008.02.003
- Rückert, E. A., Neumann, G., Toussaint, M., and Maass, W. (2013). Learned graphical models for probabilistic planning provide a new class of movement primitives. *Front. Comput. Neurosci.* 6:97. doi: 10.3389/fncom.2012.00097
- Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. J. (2003). "Learning movement primitives," in *International Symposium on Robotics Research (ISRR 2003)*, (Lucerne), 561–572.
- Schutte, L., Rodgers, M., Zajac, F., and Glaser, R. (1993). Improving the efficacy of electrical stimulation-induced leg cycle ergometry: an analysis based on a dynamic musculoskeletal model. *Rehabil. Eng. IEEE Trans.* 1, 109–125. doi: 10.1109/86.242425
- Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., and Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Netw.* 23, 551–559. doi: 10.1016/j.neunet.2009.12.004
- Seth, A., Sherman, M., Reinbolt, J. A., and Delp, S. L. (2011). "Opensim: a musculoskeletal modeling and simulation framework for in silico investigations and exchange," in *Procedia International Union of Theoretical and Applied Mathematics (IUTAM 2011)*, 2, 212–232. doi: 10.1016/j.piutam.2011.04.021
- Sutton, R., and Barto, A. (1998). *Reinforcement Learning: An Introduction*. Boston, MA: MIT Press.
- Theodorou, E., Buchli, J., and Schaal, S. (2010). "Reinforcement learning of motor skills in high dimensions: a path integral approach," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, (Anchorage, AL), 2397–2403.
- Westervelt, E. R., Buche, G., and Grizzle, J. W. (2004). Experimental validation of a framework for the design of controllers that induce stable walking in planar bipeds. *Int. J. Robot. Res.* 23, 559–582. doi: 10.1177/0278364904044410
- Wierstra, D., Schaul, T., Peters, J., and Schmidhuber, J. (2008). "Episodic reinforcement learning by logistic reward-weighted regression," in

- Proceedings of the 18th International Conference on Artificial Neural Networks, Part I, (ICANN 2008)*, (Berlin), 407–416.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8, 229–256. doi: 10.1007/BF00992696
- Winters, J. M., and Stark, L. (1985). Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. *Biomed. Eng. IEEE Trans.* BME-32, 826–839. doi: 10.1109/TBME.1985.325498
- Zajac, F. E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit. Rev. Biomed. Eng.* 17, 359–411.
- Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Received: 27 May 2013; paper pending published: 04 July 2013; accepted: 25 September 2013; published online: 17 October 2013.
- Citation: Rückert E and d'Avella A (2013) Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems. *Front. Comput. Neurosci.* 7:138. doi: 10.3389/fncom.2013.00138
- This article was submitted to the journal *Frontiers in Computational Neuroscience*.
- Copyright © 2013 Rückert and d'Avella. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

## APPENDIX

For the via-point task the parameter settings for learning are shown in **Table A1**. Initial parameter values and parameter settings for policy search for the biped walker task are shown in **Table A2** and in **Table A3**. In **Table A4** we list the learning settings for the multi-directional reaching task using a musculoskeletal model of a human arm. The implemented muscles and their characteristic parameters are shown in **Table A5**.

**Table A1 | Parameter settings for the discrete via-point task.**

Parameter	Range	Initial value
$a$ , ( $w$ )	$\in [-5, 5]$	$a^0 = 0$
$\mu$ , ( $\mu$ )	$\in [0, 1]$	$\mu^0$ spaced linearly in $[0, 1]$
$h$ , ( $h$ )	$\in [0.01, 1]$	$h^0 = 0.1$
$\beta$	$\in [0, 100]$	$\beta^0 = 1$

*In brackets are the variable names for the dynamic movement primitives, which are used for comparison.*

**Table A2 | Biped walker setting of pre-optimized quantities.**

Variable	Value
$q_1$	$[3.5, 4.4, -0.07, -0.5, -0.7, -1.5, -0.6, -1.0, 0.3, -0.4]$
$g$	$[2.8, 4.5, -0.3, -1.8]$
$k_{\text{pos}}$	$[632.5, 885.6, 463.4, 643.7]$
$k_{\text{vel}}$	$[14.5, 13.2, 38.6, 40.6]$
$q_{\text{min}}$	$[2.8, 2.8, -2.6, -2.6, -1.04]$
$q_{\text{max}}$	$[4.7, 4.7, 0, 0, 1.0]$

**Table A3 | Policy search parameter settings for the rhythmic walking task.**

Parameter	Range	Initial value
$a$ , ( $w$ )	$\in [-2, 2]$	$a^0 = 0$
$\mu$ , ( $\mu$ )	$\in [0.8\mu^0, 1.2\mu^0]$	$\mu^0$ spaced linearly in $[0, 2\pi]$
$h$ , ( $h$ )	$\in [0.1, 10]$	$h^0 = 1$
$\beta$	$\in [0, 2]$	$\beta^0 = 1$
$\Delta s$	$\in [0, 0.5]$	$\Delta s^0 = 0$

*In brackets are the variable names for the dynamic movement primitives.*

**Table A4 | Parameter settings for the multi-directional reaching task.**

Parameter	Range	Initial value
$a$	$\in [-1, 1]$	$a^0 = 0.5$
$\mu$	$\in [0.3, 0.7]$	$\mu^0$ spaced linearly in $[0.3, 0.7]$
$h$	$\in 8[10^{-5}, 10^{-3}]$	$h^0 = 8 \cdot 10^{-4}$
$\beta$	$\in [0, 1]$	$\beta^0 = 1$
$\Delta s$	$\in [-0.2, 0.2]$	$\Delta s^0 = 0$

**Table A5 | Characteristic muscle parameters of an upper limb model taken from Garner and Pandy (2001).**

	$L_0^M$ [cm]	$L_s^T$ [cm]	$F_0^M$ [N]	$\alpha$ [rad]
Anterior deltoid (DeltA)	14.68	<b>9.3</b> (1.64)	277.48	0
Posterior deltoid (DeltP)	17.02	5.93	567.15	0
Latissimus dorsi-thoracic (LatDors)	34.87	14.75	173.43	0
Pectoralis major-clav (PectClav)	22.65	0.45	342.46	0
Triceps brachii-long (TrLong)	15.24	<b>25.05</b> (19.05)	629.21	0.26
Triceps brachii-lateral (TrLat)	6.17	19.64	1268.87	0.26
Triceps brachii-medial (TrMed)	4.9	<b>18.19</b> (12.19)	619.67	0.26
Biceps-long (BicLong)	15.36	<b>32.93</b> (22.93)	392.91	0.17
Biceps-short (BicShort)	13.07	<b>26.98</b> (22.98)	461.76	0.17
Brachialis (BRA)	10.28	<b>9.75</b> (1.75)	853	0.17
Brachioradialis (BrRad)	27.03	6.04	101.58	0.08

The tendon slack length is denoted by  $L_s^T$ , the maximum isometric force by  $F_0^M$ , the optimal fiber length by  $L_0^M$ , and the muscle pennation angle by  $\alpha$ . To increase the reachable space, we adapted the tendon slack length  $L_s^T$  of a small number of muscles (bold numbers vs. the original values in brackets).