The Role of Coupling Terms in Variable Impedance Policies Learning

Florian Winter, Matteo Saveriano and Dongheui Lee

Abstract— **Enabling a robot to perform assistive tasks in everyday human environments requires adaptation capabilities to compensate for unknown physical interaction forces. Impedance defines the reaction behavior to such contacts, hence robust and safe interaction may be facilitated by founding suitable impedance gains. This paper proposes a novel reinforcement learning approach to simultaneously learn trajectories and impedance behaviors. A modified version of dynamic movement primitives is used to compactly encode skills as a mixture of dynamical systems. The resulting algorithm learns impedance behaviors considering couplings across motor control variables to allow a better exploitation of the dynamic capabilities of the robot. A simulated comparison with state-of-the-art approaches demonstrates the effectiveness of the proposed approach.**

I. Introduction

Novel assistive robotics applications require efficient physical interaction with unknown environments and human beings in a safe and robust manner. Having a set of fixed impedance gains it is probably not sufficient to adapt the robot's behavior to different interaction scenarios.

A possible solution to find proper impedance gains is offered by the framework of Reinforcement Learning (RL) [1]. A valuable asset of RL is the availability of model-free algorithms that do not require any model knowledge of the robot nor the environment. This is a desirable property for learning interaction tasks where good contact models are hard to derive [2]. Due to the high-dimensional nature of robot control problems, the robotics community focused on policy-search RL methods [3]–[5], which search for solutions in a smaller policy space that contains all possible policies representable by a certain parametrization.

Recent developments in RL have brought out two different policy-search algorithms, namely Policy learning by Weighting Exploration with the Returns (PoWER) [3] and Policy Improvement with Path Integrals (PI^2) [4], which are considered state-of-the-art RL algorithms in robotics [5]. The empirical comparison between PoWER and PI^2 in [4] resulted in compatible performance of both algorithms. However, in case of PI^2, the immediate cost in the reward function can be chosen arbitrary. Hence, boolean cost functions can be used to encode whether a task has been succeeded or not [2]. Although it cannot be concluded from these findings that PI^2 outperforms PoWER, a case may be made for a simpler realization and a broader scope of applicability of PI^2.

Policy parametrization is a crucial aspect to reduce the search space and to guarantee a rapid convergence. Human demonstrations can be used to initialize position [6] and impedance [7] policy parameters. In [2], PI^2 is applied to simultaneously learn motion trajectories and variable impedance gains using Dynamic Movement Primitives (DMPs) [8] as policy representation. The impedance behavior of each joint is described by a DMP and a diagonal stiffness matrix is learned, neglecting the interdependency among different DoFs. Learning synergies and couplings in motor control helps to better exploit dynamics capabilities of human limbs [9], [10]. The benefits of considering off-diagonal (coupling) terms in gain matrices are highlighted in [11] with a robot weightlifting experiment. To reduce the number of parameters, in [12] impedance tasks are efficiently encoded as Correlated DMPs [13]. Full stiffness matrices (*coordination* matrices) are associated to each primitive, allowing to learn synergies across different motion variables.

We propose a novel RL algorithm to learn variable impedance behaviors explicitly considering synergies among DoFs. The proposed *Coordination Policy Improvement with Path Integral (C-PI^2)* combines the flexibility of PI^2, that allows arbitrary reward functions, with the possibility, offered by Correlated DMPs, to learn couplings between DoFs. Hence, rather than gain scheduling for each DoF, full stiffness matrices are learned by robot self practice. The proposed approach outperform PI^2 (and slightly PoWER) in terms of learning speed, in tasks for which couplings among DoFs cannot be neglected. In contrast to [12], where PoWER is used to update the policy, our approach does not impose any restriction on the immediate cost of the reward function.

II. Proposed Approach

A. Correlated Dynamic Movements Primitives (C-DMPs)

Consider that M demonstrations of a task are given as a set of N positions $\boldsymbol{x}_t$, velocities $\dot{\boldsymbol{x}}_t$ and accelerations $\ddot{\boldsymbol{x}}_t$ in joint or Cartesian space. C-DMPs [13] represent training data $\{\{\boldsymbol{x}_t, \dot{\boldsymbol{x}}_t, \ddot{\boldsymbol{x}}_t\}_{t=1}^{N}\}_{i=1}^{M}$ as a mixture of P spring-damper (PD like) dynamical systems $\ddot{\boldsymbol{x}} = \sum_{j=1}^{P} h_{t,j} \left[\boldsymbol{K}_j^{\mathcal{P}}(\boldsymbol{\mu}_j^{\mathcal{X}} - \boldsymbol{x}_t) - \kappa^{\mathcal{V}}\dot{\boldsymbol{x}}_t\right]$ with attractor vectors $\boldsymbol{\mu}_j^{\mathcal{X}}$, full stiffness matrices (i.e. coordination matrices) $\boldsymbol{K}_j^{\mathcal{P}}$ and scalar damping gain $\kappa^{\mathcal{V}}$. The set $\{\boldsymbol{\mu}_j^{\mathcal{X}}, \boldsymbol{K}_j^{\mathcal{P}}\}_{j=1}^{P}$ represents the learnable parameters.

To reproduce a desired path $\boldsymbol{x}_{t_i,d}, \dot{\boldsymbol{x}}_{t_i,d}, \ddot{\boldsymbol{x}}_{t_i,d}$ (discretized into N time steps t_i with $i = 0, 1, \ldots, N-1$), the desired position can be computed by summarizing the weighted attractor points over all basis functions $\boldsymbol{x}_{t_i,d} = \sum_{j=1}^{P} h_{t_i,j}\boldsymbol{\mu}_j^{\mathcal{X}}$, where $h_{t_i,j} = \frac{\psi_{t_i,j}}{\sum_{l=1}^{P}\psi_{t_i,l}}$ and $\psi_{t_i,j} = \mathcal{N}(t_i; \mu_j^{\mathcal{T}}, \Sigma_j^{\mathcal{T}})$ are composed of equally in time distributed Gaussians with means $\mu_j^{\mathcal{T}}$ and variances $\Sigma_j^{\mathcal{T}}$. The $\psi_{t_i,j}$ are activated by

Authors are with Chair of Automatic Control Engineering, Technische Universität München, Munich, Germany {`florian.winter`, `matteo.saveriano`, `dhlee`}`@tum.de`.

$t_i = -\frac{\ln(\nu_{t_i})}{\alpha_\nu \tau}$, where α_ν and τ determine the movement duration. ν_t is set to $\nu_t = 1$ to initiate the movement and then converges to zero. The temporal varying stiffness matrix $\boldsymbol{K}^{\mathcal{P}}_{t_i}$ is computed as $\boldsymbol{K}^{\mathcal{P}}_{t_i} = \sum_{j=1}^{P} h_{t_i,j} \boldsymbol{K}^{\mathcal{P}}_j$, to generate the PD motor command

$$\ddot{\boldsymbol{x}}_{t_i,d} = \boldsymbol{K}^{\mathcal{P}}_{t_i}(\boldsymbol{x}_{t_i,d} - \boldsymbol{x}_{t_i}) - \kappa^{\mathcal{V}} \dot{\boldsymbol{x}}_{t_i} \qquad (1)$$

B. Coordination Policy Improvement with Path Integrals

In order to additionally learn the couplings between motor control variables, the PI^2 algorithm must be viewed from a different perspective. A trajectory can be learned by parametrizing the attractors $\boldsymbol{x}_{t_i,d}$ in the policy form

$$\boldsymbol{x}_{t_i,d} = \sum_{j=1}^{P} h_{t_i,j}(\boldsymbol{\mu}^{\mathcal{X}}_j + \boldsymbol{\epsilon}^{\mathcal{X}}_{t_i,j}) = \sum_{j=1}^{P} h_{t_i,j}(\boldsymbol{\theta}^{\mathcal{X}}_j + \boldsymbol{\epsilon}^{\mathcal{X}}_{t_i,j}) \quad (2)$$

with parameter vectors $\boldsymbol{\theta}^{\mathcal{X}}_j$ and exploration noise vectors $\boldsymbol{\epsilon}^{\mathcal{X}}_{t_i,j}$. To learn full stiffness matrices, the policy $\boldsymbol{K}^{\mathcal{P}}_{t_i}$ can be parametrized in the policy form

$$\boldsymbol{K}^{\mathcal{P}}_{t_i} = \sum_{j=1}^{P} h_{t_i,j}(\boldsymbol{K}^{\mathcal{P}}_j + \boldsymbol{\Xi}^{\mathcal{P}}_{t_i,j}) = \sum_{j=1}^{P} h_{t_i,j}(\boldsymbol{\Theta}^{\mathcal{P}}_j + \boldsymbol{\Xi}^{\mathcal{P}}_{t_i,j}) \quad (3)$$

with parameter matrices $\boldsymbol{\Theta}^{\mathcal{P}}_j$ and exploration noise matrices $\boldsymbol{\Xi}^{\mathcal{P}}_{t_i,j}$. The parameters $\boldsymbol{\theta}^{\mathcal{X}}_j = \boldsymbol{\mu}^{\mathcal{X}}_j$ and $\boldsymbol{\Theta}^{\mathcal{P}}_j = \boldsymbol{K}^{\mathcal{P}}_j$ form the policy output $\ddot{\boldsymbol{x}}_{t_i,d}$ in equation (1), which can be interpreted as the motor command for the C-PI^2 algorithm. As a consequence, the immediate cost can be expressed [4] as $r_{t_i} = q_{t_i} + \frac{1}{2}\ddot{\boldsymbol{x}}^T_{t_i,d}\boldsymbol{R}\ddot{\boldsymbol{x}}_{t_i,d}$ with an arbitrary, state-dependent cost function q_{t_i} and a quadratic control weight matrix $\boldsymbol{R}$.

The exploration vector for the trajectory in equation (2) is drawn from a zero-mean Gaussian distribution $\boldsymbol{\epsilon}^{\mathcal{X}}_{t_i,j} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\mathcal{X}}_j)$ with covariance $\boldsymbol{\Sigma}^{\mathcal{X}}_j$ for each basis function. Similarly, the exploration matrix in equation (3) is drawn from $\boldsymbol{\Xi}^{\mathcal{P}}_{t_i,j} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\mathcal{P}}_j)$, despite the covariance matrices $\boldsymbol{\Sigma}^{\mathcal{P}}_j$ have to be chosen to guarantee that the stiffness matrix $\boldsymbol{K}^{\mathcal{P}}_{t_i}$ in (3) is symmetric and positive semidefinite (SPS). Recalling that the sum of two SPS matrices is a SPS matrix, a valid full stiffness matrix $\boldsymbol{K}^{\mathcal{P}}_{t_i}$ is obtained if the exploration matrices are all SPS. These properties are retained in C-PI^2, since a weighted averaging over SPS matrices is a SPS matrix.

The generalized cost term from PI^2 is [4]

$$S(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} + \frac{1}{2}\sum_{j=i+1}^{N-1} \ddot{\boldsymbol{x}}^T_{t_j,d}\boldsymbol{R}\ddot{\boldsymbol{x}}_{t_j,d} \qquad (4)$$

where ϕ_{t_N} is the terminal cost. The generalized cost of each rollout path defines a probability of a path $\boldsymbol{\tau}^k_i$ as $P(\boldsymbol{\tau}^k_i) = \frac{E_S(\boldsymbol{\tau}^k_i)}{\sum_{k=1}^{K} E_S(\boldsymbol{\tau}^k_i)}$ with automatic sensitivity regulation term $E_S(\boldsymbol{\tau}^k_i) = \exp\left(-h_\lambda \frac{S(\boldsymbol{\tau}^k_i) - \min S(\boldsymbol{\tau}^k_i)}{\max S(\boldsymbol{\tau}^k_i) - \min S(\boldsymbol{\tau}^k_i)}\right)$ that maximizes the discrimination between experienced paths for every time step i with sensitivity regulation constant h_λ [4]. Probability-weighted averaging over K rollouts yields the trajectory and stiffness parameter updates at each time step $\delta\boldsymbol{\theta}^{\mathcal{X}}_{t_i,j} = \sum_{k=1}^{K} P(\boldsymbol{\tau}^k_i)\boldsymbol{\epsilon}^{\mathcal{X},k}_{t_i,j}$ and $\delta\boldsymbol{\Theta}^{\mathcal{P}}_{t_i,j} = \sum_{k=1}^{K} P(\boldsymbol{\tau}^k_i)\boldsymbol{\Xi}^{\mathcal{P},k}_{t_i,j}$. Temporal weighted averaging over N time steps $\delta\boldsymbol{\theta}^{\mathcal{X}}_j = \frac{\sum_{i=0}^{N-1}(N-i)\psi_{t_i,j}\delta\boldsymbol{\theta}^{\mathcal{X}}_{t_i,j}}{\sum_{i=0}^{N-1}\psi_{t_i,j}(N-i)}$ and $\delta\boldsymbol{\Theta}^{\mathcal{P}}_j = \frac{\sum_{i=0}^{N-1}(N-i)\psi_{t_i,j}\delta\boldsymbol{\Theta}^{\mathcal{P}}_{t_i,j}}{\sum_{i=0}^{N-1}\psi_{t_i,j}(N-i)}$, leads eventually to parameter updates

$$\boldsymbol{\theta}^{\mathcal{X}}_j \leftarrow \boldsymbol{\theta}^{\mathcal{X}}_j + \delta\boldsymbol{\theta}^{\mathcal{X}}_j, \quad \boldsymbol{\Theta}^{\mathcal{P}}_j \leftarrow \boldsymbol{\Theta}^{\mathcal{P}}_j + \delta\boldsymbol{\Theta}^{\mathcal{P}}_j \qquad (5)$$

individually performed for each basis function $j = 1, .., P$.

III. Simulation Results

A. Tuning Parameters Selection

The number of re-used rollouts for learning in C-PI^2 (PI^2) is set to $\sigma = 5$ and the number of rollouts per epoch used for parameter updating is set to $K = 10$. Thus, C-PI^2 (PI^2) updates after the first 10 rollouts and then after 5 more rollouts using these 5 new rollouts and the best 5 of the latter epoch. On the other hand, PoWER updates after each rollout using the best $\sigma = 5$ performed rollouts.

In order to refine and adapt the policy, the parameter space must be explored by perturbing the parameters through exploration noise $\boldsymbol{\epsilon}_t$. The exploration noise is sampled from a zero-mean Gaussian distribution at the beginning of a rollout and kept constant during the entire rollout. To increase the exploitation of the learned information, the exploration noise magnitude ξ is decreased over the number of updates ϑ by multiplying it with a decay parameter γ^ϑ. We set $\gamma^\vartheta = 0.99^\vartheta$ for C-PI^2 (PI^2) and $\gamma^\vartheta = 0.99^{\vartheta/(K-\sigma-1)}$ for PoWER.

B. Via-points task

The objective of this experiment is to learn to traverse two given positions before reaching the goal position. We consider the 2D point mass system $\ddot{\boldsymbol{x}}_t = \frac{1}{m}(\boldsymbol{u}_t - d\dot{\boldsymbol{x}}_t)$ with point mass $m = 1$, damping constant $d = 1$ and motor command $\boldsymbol{u}_t = m\ddot{\boldsymbol{x}}_{t,d} + d\dot{\boldsymbol{x}}_t$. The desired acceleration $\ddot{\boldsymbol{x}}_{t,d}$ in $\boldsymbol{u}_t$ is generated with C-DMPs for C-PI^2 and PoWER, with DMPs for PI^2. Training data are generated using minimum jerk trajectories.

The cost function is chosen to force the movement of the point mass to pass through two intermediate via-points $\boldsymbol{p}_1 = [0.4\ 0.2]^T$ and $\boldsymbol{p}_2 = [0.6\ 0.8]^T$. The cost function for PI^2 and C-PI^2 algorithm is $r_t = w_1\,\delta(t-0.2)\|\boldsymbol{p}_1 - \boldsymbol{x}_t\|^2 + w_2\,\delta(t-0.4)\|\boldsymbol{p}_2 - \boldsymbol{x}_t\|^2$, where $w_1 = w_2 = 1\mathrm{e}10$. The Dirac delta function $\delta(\bullet)$ pushes the point mass to traverse $\boldsymbol{p}_1$ at $t = 0.2s$ and $\boldsymbol{p}_2$ at $t = 0.4s$. To have an equivalent optimization problem with PoWER, we use the reward function $\tilde{r}_t = \delta(t-0.2)\exp\left(-\frac{w_1}{\lambda}\|\boldsymbol{p}_1 - \boldsymbol{x}_t\|^2\right) + \delta(t-0.4)\exp\left(-\frac{w_2}{\lambda}\|\boldsymbol{p}_2 - \boldsymbol{x}_t\|^2\right)$ with $w_1 = w_2 = 1\mathrm{e}10$. The magnitudes of trajectory and stiffness exploration noises are respectively set to $\xi^{\mathcal{X}} = 0.03$ and $\xi^{\mathcal{P}} = 10$ for C-DMPs and $\xi^{\mathcal{X}} = 20$ and $\xi^P = 10$ for gain-scheduling DMPs, which yields a similar exploration in motor command space.

The cost over 2005 rollouts (400 updates of PI^2 and C-PI^2) of PoWER, PI^2 and C-PI^2 is depicted in Fig. 1. To achieve the task (see Fig. 2) the novel C-PI^2 algorithm needs only 120 rollouts, whereas PI^2 converges after about 1500 rollouts. Even though PoWER uses the same policy representation as C-PI^2, it converges roughly after 200 rollouts, while the final cost is approximately the same (see Fig. 1).

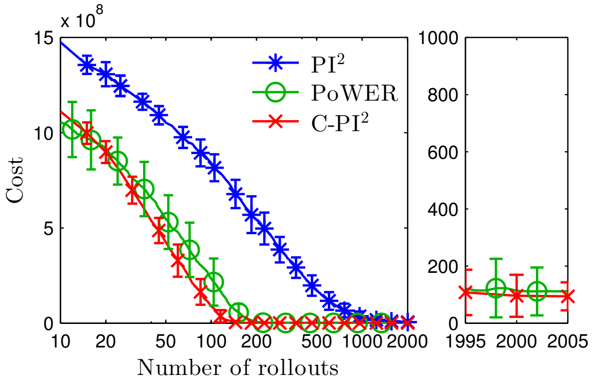

Fig. 1. The noiseless costs (averaged over 10 experiments) of PI^2, PoWER and C-PI^2 are plotted on a semilogarithmic scale over 2005 rollouts with standard deviation error bars. The plot on the right side zooms in to show the final cost during the last 20 rollouts.

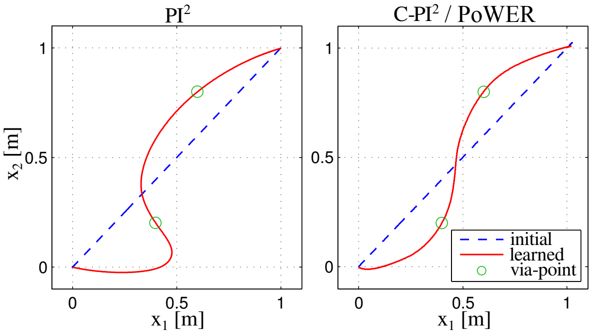

Fig. 2. Trajectory of the point mass. (Left) Trajectory learned with PI^2 after 1500 rollouts. (Right) Trajectory learned with C-PI^2 and PoWER after respectively 120 and 200 rollouts.

C. Via-gains task

Inspired by the previous via-points task, this experiment shows how it is possible to learn specified stiffness gains at certain time instants (via-gains) using C-PI^2. Learning different impedance behaviors in different directions is useful, for example, in assembly tasks (peg in the hole). The cost function is chosen as $r_t = w_1\,\delta(t-0.2)\|\boldsymbol{K}_1 - \boldsymbol{K}_t^{\mathcal{P}}\|^2 + w_2\,\delta(t-0.4)\|\boldsymbol{K}_2 - \boldsymbol{K}_t^{\mathcal{P}}\|^2$ to have $\boldsymbol{K}_t^{\mathcal{P}} = \boldsymbol{K}_1$ at $t = 0.2s$ and $\boldsymbol{K}_t^{\mathcal{P}} = \boldsymbol{K}_2$ at $t = 0.4s$. The variable stiffness matrix $\boldsymbol{K}_t^{\mathcal{P}}$, learned with C-PI^2 (after 210 rollouts), is depicted in Fig. 3. The result is obtained with $w_1 = w_2 = 1e5$, $\xi^{\mathcal{X}} = 0.03$, $\xi^{\mathcal{P}} = 10$ and

$$\boldsymbol{K}_1 = \begin{bmatrix} 250 & 20 \\ 20 & 150 \end{bmatrix} \quad \text{and} \quad \boldsymbol{K}_2 = \begin{bmatrix} 150 & -20 \\ -20 & 250 \end{bmatrix} \tag{6}$$

The initial stiffness gains are learned considering the variability in the training data as in [14]. Note that PI^2 cannot be adopted in this task since it learns a gain for each DoF, while similar results can be achieved using PoWER.

Examples in this section show that learning the couplings across dimensions may enhance the learning speed. Although the performance of PoWER and C-PI^2 are similar, C-PI^2 has still the advantage of allowing arbitrary cost functions. We want to underline that the proposed evaluation is not a comparison between the general frameworks of PI^2 and PoWER. However, this comparison shows the benefits of learning full stiffness matrices when the coupling among DoFs is not negligible.

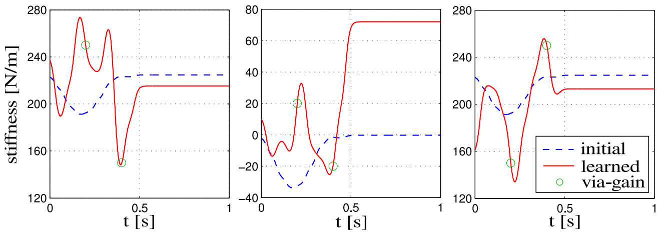

Fig. 3. Variable stiffness gains in the via-gains experiment. Gains $K_{1,1}^{\mathcal{P}}$ (left), $K_{1,2}^{\mathcal{P}} = K_{2,1}^{\mathcal{P}}$ (middle) and $K_{2,2}^{\mathcal{P}}$ (right) reach the specified set points at $t = 0.2s$ and $t = 0.4s$ (green circles).

IV. Conclusions

A novel RL algorithm, namely C-PI^2, has been proposed to simultaneously learn trajectories and variable impedance behaviors. The proposed approach improves the so-called PI^2 algorithm, giving the possibility to learn couplings between DoFs. C-PI^2 uses C-DMPs, the same policy representation of the PoWER algorithm. While PoWER imposes restrictions on the reward function, in C-PI^2 the reward function can be arbitrary. A comparison on simulated tasks shows that C-PI^2 converges faster than state-of-the-art RL approaches.

References

[1] R. Sutton and A. Barto, *Reinforcement learning: an introduction*, ser. A Bradford book. MIT Press, 1998.

[2] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, “Learning variable impedance control,” *International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, 2011.

[3] J. Kober and J. Peters, “Learning motor primitives for robotics,” in *Intl Conf. on Robotics and Automation*, pp. 2112–2118.

[4] E. Theodorou, J. Buchli, and S. Schaal, “A generalized path integral control approach to reinforcement learning,” *Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, December 2010.

[5] P. Kormushev, S. Calinon, and D. Caldwell, “Reinforcement learning in robotics: applications and real-world challenges,” *Robotics*, vol. 2, no. 3, pp. 122–148, 2013.

[6] M. Saveriano, A. Sangik, and D. Lee, “Incremental kinesthetic teaching of end-effector and null-space motion primitives,” in *Int. Conf. on Rob. and Aut.*, 2015, pp. 3570–3575.

[7] M. Saveriano and D. Lee, “Learning motion and impedance behaviors from human demonstrations,” in *URAI*, 2014, pp. 368–373.

[8] 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.

[9] T. Flash and N. Hogan, “The coordination of the arm movements: An experimentally confirmed mathematical model,” *Neurology*, vol. 5, no. 7, pp. 1688–1703, 1985.

[10] E. Todorov and M. Jordan, “Optimal feedback control as a theory of motor coordination,” *Nat. Neurosc.*, vol. 5, pp. 1226–1235, 2002.

[11] M. Rosenstein, A. Barto, and R. Van Emmerik, “Learning at the level of synergies for a robot weightlifter,” *Robotics and Autonomous Systems*, vol. 54, no. 8, pp. 706–717, 2006.

[12] P. Kormushev, S. Calinon, and D. Caldwell, “Robot motor skill coordination with EM-based reinforcement learning,” in *Intl Conf. on Intelligent Robots and Systems*, 2010, pp. 3232–3237.

[13] S. Calinon, F. D’halluin, D. Caldwell, and A. Billard, “Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework,” in *Intl Conf. on Hum. Rob.*, 2009, pp. 582–588.

[14] S. Calinon, I. Sardellitti, and D. Caldwell, “Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies,” in *Intl Conf. on Int. Rob. and Sys.*, 2010, pp. 249–254.