Proceedings of the 2nd International Conference of Control, Dynamic Systems, and Robotics
Ottawa, Ontario, Canada, May 7-8 2015
Paper No. 160

Learning Gravitation Compensation on a Simulated Passive Compliant Robot Limb for Multiple Targets and Load Cases

Daniel Basa, Axel Schneider
University of Applied Sciences Bielefeld, Faculty of Engineering and Mathematics, Biomechatronics
Wilhelm-Bertelsmann-Str. 10, Bielefeld, Germany
Bielefeld University, Biomechatronics,
Center of Excellence Cognitive Interaction Technology (CITEC)
Inspiration 1, Bielefeld, Germany
daniel.basa@fh-bielefeld.de, axel.schneider@fh-bielefeld.de

Abstract- Compliance is an important feature for service robots but also for robots in industrial applications to increase human machine interaction safety. While active compliance is online adaptable and easy to control its dynamic response is determined by the sampling rate and the response rate of the controller. In contrast, passive compliance is inherent to the system. It responds naturally fast to any perturbation exerted on the robot independently of the controller. In this paper an extension to the DMP framework is introduced which facilitates the generation of trajectories for passive compliant link drives using reinforcement learning. The compliance of the joint actuators is preserved entirely during motion as well as at the goal position. The proposed approach is evaluated in simulation using a simplification of the common articulated configuration with 2 DOF and passive compliant link drives. Experiments are presented for point-to-point movements and different load case scenarios. The results demonstrate that the proposed approach is capable of generating trajectories for point-to-point movements to trained and untrained goal positions as well as for trained and untrained load cases.

Keywords: reinforcement learning, robot arm, compliance, directed limb movement, gravitation compensation

1 Introduction

Human machine interaction and collaboration is becoming increasingly important with advancing technology not only in service robots but also in industrial applications. Yet the majority of robotic systems is designed using stiff and inelastic joint actuators. While stiff systems can be controlled with proven control theoretical methods, such systems represent a possible threat to humans in contact situations during human machine interaction (Zinn et al. 2004). Hence, they are usually found caged and separated from humans. To create systems for safer human robot interaction a leightweight construction together with a low interface stiffness is beneficial. The latter can be achieved by incorporating elastic properties into the link drive. Here, two approaches and a combination of both are applicable. *Active compliance* uses a controlled virtual elasticity and allows the application of control strategies such as impedance control (Hogan 1985) or the unified passivity-based control framework (Albu-Schaeffer et al. 2007). The compliance of the drive can be adapted on demand to environmental requirements. Instead of using a real elastic element inside a link drive, active compliance simulates compliance by control. For this reason, active compliance can still be dangerous during human machine interaction regarding controller outage, control errors and control frequency. In contrast, *passive compliance* uses a real series elasticity inside the link drives to achieve elastic behaviour. This makes interaction safety independent to the above mentioned control errors. Different actuators incorporating passive compliance have been proposed like the Serial Elastic Actuator (SEA) (Pratt & Williamson 1995), an elastic joint drive for robotics applications based on a sensorized elastomer coupling (Paskarbeit et al. 2013) or a compact soft actuator unit as used in the child humanoid robot "iCub" (Tsagarakis et al. 2009). Alternative actuator types like pneumatic muscles also feature passive compliance as used for instance in the humanoid muscle robot torso called ZAR5 (Boblan & Schulz 2010). Although actuators with inherent series elasticity reduce the risk during human machine interaction, this is achieved at the expense of controllability. Standard feedback based control approaches use feedback from the drive side of the joint to generate motor commands (motor torque) for the gear side of the joint. Such a control loop will inevitably react with an increase of motor torque if the observed drive side is perturbed by an obstacle blocking its path. Elastic properties of passive compliant limbs are canceled out to some extent by such a behaviour. A possible strategy to sustain the elastic properties of passive compliant actuators during operation is presented in this work. The proposed approach uses reinforcement learning to train a trajectory generating process which can be adapted to different goal positions and load scenarios using a mixture approach. The trajectory is generated for the gear side of passive compliant joint actuators in such a way that the drive side of the joint approaches a desired goal state under consideration of boundary conditions along the path. This trajectory generating process is based on an extension of the original *Dynamic Movement Primitives* formulation by Ijspeert et al. (2003, 2013).

2 Learning Movement Patterns

Dynamic Movement Primitives (DMPs) as suggested by Ijspeert et al. (2002) are pattern generators for point-to-point and rhythmic movements. For point-to-point movements DMPs can be described as point attractive systems, specified by start state, goal state and movement time, which is the time in which the start state has to be transformed into the goal state. The trajectory connecting start and goal state can be of an arbitrary shape defined by a policy whose parameters can be learned. This policy is represented by a non-linear function.

Since the introduction of DMPs by Ijspeert et al. (2002) improvements, extensions and simplifications have been introduced. Kober et al. (2010) proposed a system that allows to change the target velocity of the movement while maintaining the overall duration and shape in order to hit a moving target. Mülling et al. (2013) refined this system and introduced "Modified Motor Primitives for Striking Movements" together with a framework that enables a robot to learn basic cooperative table tennis using a mixture of movement primitives.

The proposed extension to the DMP framework in this text is based on the latest DMP formulation by Ijspeert et al. (2013) and Schaal et al. (2007).

2.1 Dynamic Movements Primitives

DMPs are based on a 2nd order dynamical system, analog to a spring-damper system. This 2nd order system connects the fixed goal state and the moved degree of freedom, e.g. a rotatory link drive and constructs a point attractive system with the goal state as the attractor as depicted in Fig. 1(a). The shape of the trajectory transforming the start state into the goal state can be manipulated by a non-linear function f_1 which is add to the 2nd order dynamical system:

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z)+f_1, \quad \tau \dot{y} = z \tag{1}$$

with τ being a temporal scaling factor, g the goal state, α_z representing the damping constant of the system and the product of α_z and β_z representing the spring constant. Position, velocity and acceleration of the moved degree of freedom are given by $q = y$, $\dot{q} = \frac{1}{\tau}z$ and $\ddot{q} = \frac{1}{\tau}\dot{z}$. The parameters α_z and β_z must be chosen such that the spring-damper system is critically damped under the assumption $f_1 = 0$. To avoid

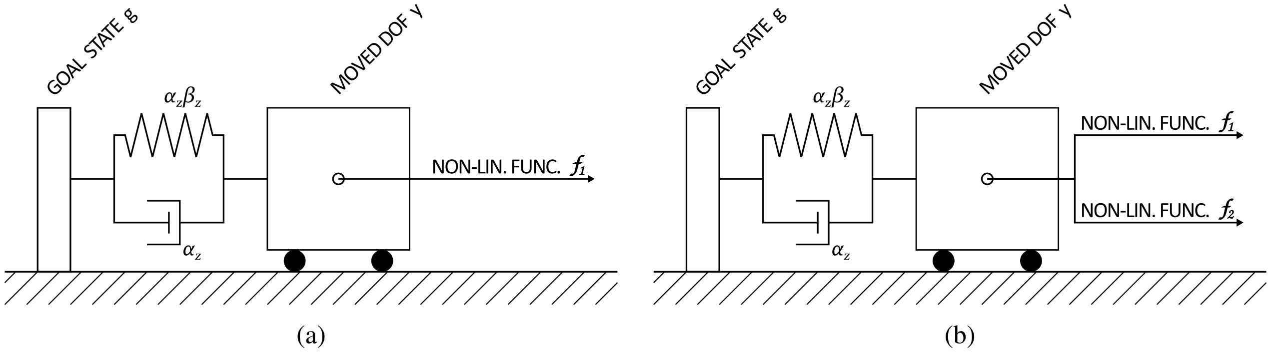


Fig. 1: Graphical interpretation of the 2nd order dynamical system of (a) a *Dynamic Movement Primitive* (DMP) as described by Ijspeert et al. (2013) with the attached non-linear function f_1. The system connects the fixed goal state (e.g. target position of rotational link drive) with the moved degree of freedom (e.g. current position of a rotational link drive). The moved DOF is pulled towards the goal state by the 2nd order dynamical system. The non-linear function f_1 is used to alters the monotonic movement of the DOF to produce a more complex trajectory until it fades out at the end of the movement. (b) the reformulation of the DMP framework proposed in this work. A further non-linear function f_2 is attached to the moved DOF. The 2nd non-linear function does not fade out at the end of the movement. Instead it is designed to reach a constant non-zero value to compensate the influence of gravitation near the end of the movement when the first non-linear function fades out.

explicit time dependency which is convenient to reach a high robustness of movements to perturbations and interactions with the environment an additional dynamical system called *canonical system* (Ijspeert et al. 2002) is introduced. This system is used to represent phases of the movement:

$$\tau \dot{x} = -\alpha_x x \tag{2}$$

x is called the phase variable and represents time in the framework of DMPs. Similar to time, x is monotonic, but decreases and saturates at 0. τ is the same time constant as in eq. (1) and is used to scale the progression of x according to the duration of movement T. α_x defines the progression of x and should be chosen such that $x \approx 0$ at time T. The non-linear function f_1 is usually defined as a function approximator and depends on the phase variable x, the goal state g and the start state y_0:

$$f_1(x, g, y_0) = \frac{\sum_{i=1}^{N} \psi_i w_i x}{\sum_{i=1}^{N} \psi_i}(g - y_0) \tag{3}$$

$$\psi_i = e^{-h_i(x-c_i)^2} \tag{4}$$

where N is the number of Gaussian basis functions used and ψ_i is a Gaussian basis function with center c_i and bandwidth h_i. The start state is defined by y_0. Due to the factor x in eq. (3), the non-linear function f_1 fades out at the end of the movement. The remaining 2nd order dynamical system guarantees the convergence of the moved degree of freedom to the desired goal state. The amplitude term $g - y_0$ is used to generalize the trajectory so that any start-goal state combination within the workspace of the DMP can be used.
The shape of the non-linear function f_1 is defined by the weights w_i, which scale the amplitude of the Gaussian kernels. If the desired shape of the trajectory is known the weights $\mathbf{w}$ can be learned using locally weighted learning techniques (LWL) such as Locally Weighted Regression (Atkeson et al. 1997) or Locally

Weighted Projection Regression (Vijayakumar & Schaal 2000). To adapt f_1 to a trajectory with an unknown shape reinforcement learning techniques like PI^2, Contextual REPS (Kupcsik et al. 2013) and Parametrized Skills Framework (Da Silva et al. 2012) can be used.

2.2 Modified Motor Primitive for gravitation compensation

In general, the trajectory generated by DMPs is tracked by a feedback based controller e.g. a PID controller from classical control theory using feedback from the drive side of the actuator. Since the actuators output is controlled the impact of gravitation on the actuator plays a subordinated role and does not necessarily need to be considered explicitly during control. Simultaneously the effective elasticity of the compliant actuator is reduced by such a control approach. To maintain the full potential of passive compliant actuators it is necessary to decouple the controller from the feedback of the drive side. This can be achieved by generating a trajectory for the gear side of a compliant actuator instead of generating it for the drive side. Tracking of the trajectory can then be realized by a feedback based controller using feedback of the gear side (see Fig. 2(a)).

The DMP framework utilizes two processes to generate trajectories, the 2nd order dynamical system and the attached non-linear function f_1. The general movement direction of the system is determined by the dynamical system which can be influenced by the spring and damping constants α_z and β_z, the temporal scaling factor τ, the start state y_0 and the goal state g. Any deviation from this general movement direction (complex shape) is generated by f_1 which can be modified by the weights $\mathbf{w}$. Due to the canonical system, f_1 converges to zero when approaching the end of the movement time T. Therefore, without further measures, the drive side of a compliant link drive will not reach the desired target due to gravitational forces.

To generate a trajectory for the gear side which transforms the start state of the drive side to a desired goal state it is possible to alter the goal state of the DMP. Mülling et al. (2013) and Kober et al. (2010) have proposed solutions to dynamically alter the goal state. However, compliant actuators introduce a further dynamics component to the system and an increase of the degrees of freedom used in a robotic limb turns the influence of gravitation on a compliant actuator to a high dimensional problem. Since the non-linear function f_1 vanishes at the end of the movement a simple alternation of the goal state might not be sufficient. The approach proposed in this work is an extension to the DMP framework using its existing structure. It introduces a second non-linear function f_2 to compensate the effects of the gravitation on the link drive as depicted in 1(b). This non-linear function is added to the DMP formulation from equation (1):

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z)+f_1+f_2, \quad \tau \dot{y} = z \tag{5}$$

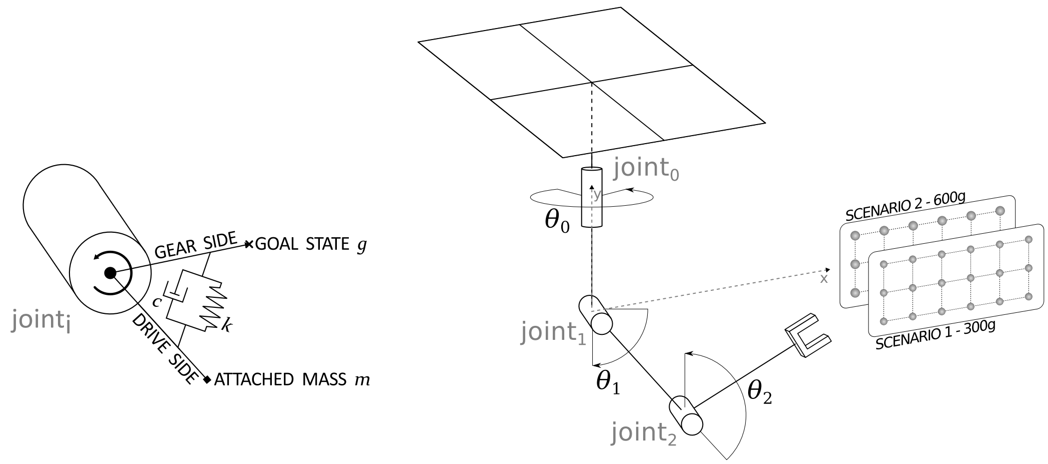
The definition of the second non-linear function

$$f_2(x) = \frac{\sum_{i=1}^{M} \psi_i w_i}{\sum_{i=1}^{M} \psi_i} \tag{6}$$

$$\psi_i = e^{-h_i(x-c_i)^2} \tag{7}$$

follows the definition of f_1 in equation (3) except that it does not use the phase variable x as multiplicative term. Hence the non-linear function does not fade out at the end of the movement. To produce a quasi constant output at the end of the movement, the bandwidths of the Gaussian basis functions have to be altered accordingly. The non-linear functions f_1 and f_2 should be learned independently of one another. This ensures that f_1 adapts to the trajectory in an optimal way and f_2 only has to compensate the short comings of f_1. In the simplest case this is only a constant output at the end of the movement at time T.

Since the influence of the gravitation does not scale with the distance between start state and goal state, f_2 does not generalize. Nonetheless, a quasi-generalization can be reached using a mixture approach. For this,



(a) (b)

Fig. 2: (a) Visual representations of a physical connection between gear side and drive side consisting of spring and damper in parallel inside a compliant joint. All following joints and the system load are attached to the gear side. This is depicted by the mass m. (b)Articulated configuration of a robot limb mounted in a hanging position. The wrist joint is negligible because the impact of gravitation on the other joints (*joint$_0$* to *joint$_2$*) is only insignificantly influenced by the posture of the wrist. Due to its orientation gravitation has no impact on *joint$_0$* either, hence it can be neglected too. The remaining robot limb has two DOF. The origin of the reference coordinate system can be located in the center of *joint$_1$*. The later on trained goal states and load case scenarios are indicated by the two cards in the right side of the figure.

an equally spaced grid is established inside the workspace of the end effector. Each grid node represents a goal state and load case and is associated with a non-linear function f_{2_k} which was trained for this specific goal state and load case, with $k = 1 \dots K$ and K being the number of grid nodes. Since the first non-linear function f_1 still generalizes, only the 2nd non-linear function needs to be linked to the respective grid node. To generate trajectories to unlearned targets, non-linear functions associated with the appropriate grid nodes are mixed based on mixing coefficients $\mathbf{b}$. These mixing coefficients can be determined using linear interpolation or e.g. linear regression. The appropriate mixing strategy depends on the distance between the grid nodes. The reformulation of DMP framework utilizing the mixing approach is defined as follows:

$$\tau\dot{z} = \alpha_z(\beta_z(g-y)-z) + f_1 + \mathbf{b}\mathbf{f}_2, \quad \tau\dot{y} = z \tag{8}$$

where $\mathbf{b} = (b_1, b_2, \dots b_K)$ is a vector of mixing coefficients, $\mathbf{f}_2 = (f_{2_1}, f_{2_2}, \dots, f_{2_K})^T$ is a vector of non-linear functions.

3 Evaluation

In section 2 an extension to the DMP framework by Ijspeert et al. (2013) was introduced which allows the generation of trajectories for passive compliant actuators while maintaining their elastic properties and to reduce the risk during human machine interaction. Furthermore, the extension was designed to cope with gravitational influences appearing as a consequence of the described trajectory generation process. In the following the experimental setup is described and the results are evaluated.

3.1 Experimental Setup

The proposed approach is evaluated on a simulated robotic limb with passive compliant joint actuators. The series elasticity inside the joints is modelled as a critically damped linear spring-damper system with a spring constant of $k = 10$ and a damping constant of $c = 6$. The robotic limb has two DOF and is based on a simplification of the common articulated configuration with six DOF as depicted in Fig. 2(b). Three of these six DOF are used by the wrist attached to the end of the limb and have a negligible influence on the impact of gravitation on the remaining joints. Furthermore, there is no gravitational influence on the first link drive $joint_0$ if mounted in a standing or hanging position. Based on these simplifications the robot limb could be reduced to 2 DOF. Regardless of the simplifications made, the proposed approach can be scaled to higher dimensional problems easily due to the underlying DMP framework.

The segment connecting $joint_1$ and $joint_2$ has a length of 265 mm and the segment connecting $joint_2$ and the end effector has a length of 275 mm. The segments and the end effector are modelled as massless. The origin of the coordinate system of the robot limb is located at the center of the joint axis of $joint_1$. For the experiments two differend loads of 300 g and 600 g were attached to end effector. The joint angles of $joint_1$ and $joint_2$ are restricted to $\theta_1 = [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\theta_2 = [0, \frac{3\pi}{4}]$.

The grid of 18 goal states in three rows (3×6 goal states) for the mixture of the non-linear functions f_{2_k} was positioned on the same horizontal level as $joint_1$. The first node of the first row at (0.25 m, 0.05 m) and the sixth node of the third row at (0.5 m, -0.05 m). The grid nodes were equally distributed with an inter node distance of 50 mm.

For the learning of the non-linear functions f_1 and f_{2_k} the reinforcement learning technique PI^2 was applied.

3.2 Evaluation in Simulation

To evaluate the proposed approach the learning of the non-linear functions was organized in two consecutive phases. At first a single non-linear function f_1 was selected from 20 training trials. Each training trail was performed with 100 PI^2 updates. Afterwards, the non-linear functions f_{2_k} were trained for each grid node and each load case using the same procedure as with the first non-linear function. In total 36 non-linear functions were trained. DMPs were initialized using spring and damping constants $\alpha_z = 25$ and $\beta_z = 12.5$ and a movement time of 2 s. Training as well as evaluation experiments were started from a hanging posture of the robot limb at rest with $\theta_1 = -\frac{\pi}{2}$, $\dot{\theta}_1 = 0$, $\theta_2 = 0$, $\dot{\theta}_2 = 0$ and the end effector at $p_{ee} = [0\,\mathrm{m}, 0.537\,\mathrm{m}]$ and $\dot{p}_{ee} = [0\,\frac{\mathrm{m}}{\mathrm{s}}, 0\,\frac{\mathrm{m}}{\mathrm{s}}]$.

For the evaluation the grid of goal states was resampled with a resolution of 10 mm and the load cases were linearly interpolated at a regular interval of 75 g (300 g, 375 g, 450 g, 525 g and 600 g). For each sampled goal state and each load case a trajectory was generated using a mixture of the eight nearest neighbours in x, y and z (load) direction as depicted in Fig. 3(f). The mixing coefficients **b** where determined using trilinear interpolation.

The Figures 3(a-e) show the distance-to-target error surfaces for each load case. It can be seen that a mixture of non-linear functions can generate trajectories to untrained targets inside a grid of trained goal states. Furthermore, the comparison of individual error surfaces shows that the proposed approach not only works for untrained goal states but also for untrained load cases.

4 Conclusion

In this paper, an extension to the DMP framework was proposed, that enables the generation of trajectories for passive compliant link drives while maintaining their elastic properties completely. This goal was achieved by introducing a further non-linear function to the DMP framework. This additional non-linear function can be trained with the same methods already used with DMPs. By the introduction of a mixture approach trajectories can be generated for untrained targets within a grid of trained goal states.

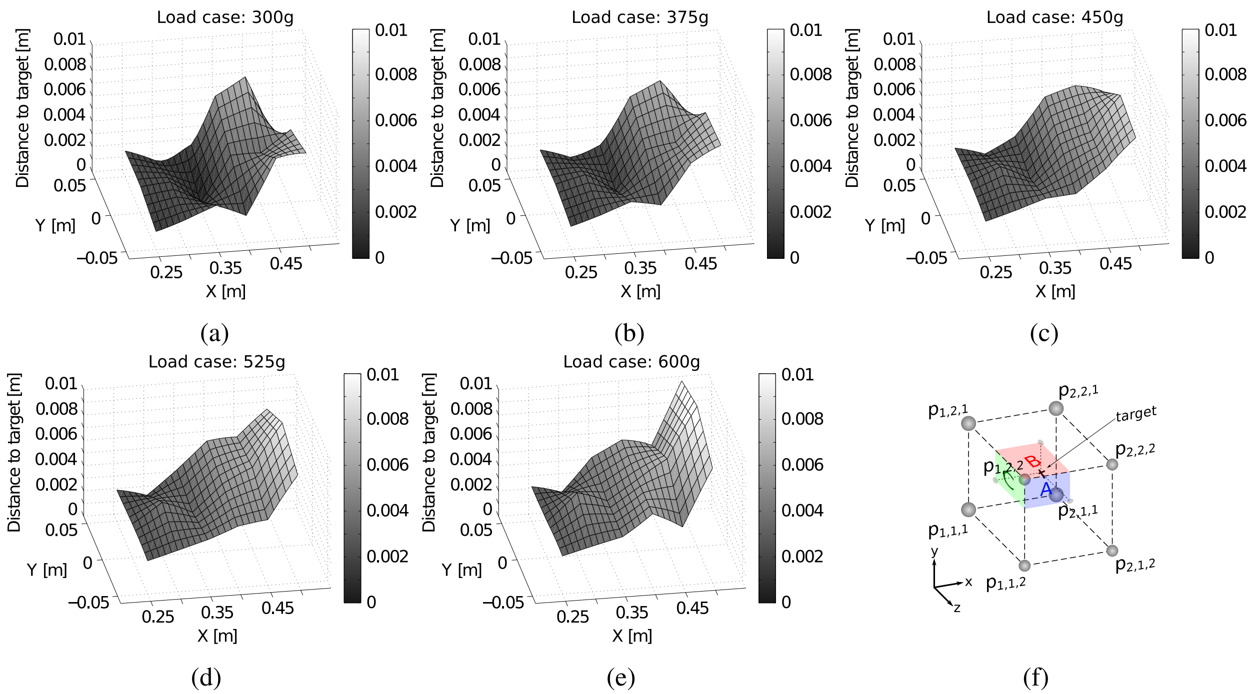


Fig. 3: (a-e) Distance-to-target error during evaluation of goal state and load case mixture. (f) Visual representation of trilinear interpolation between the eight nearest neighbours. The Cartesian goal states are located on the plane spanned by the x- and y-axes. The z-axis is used to indicate the load cases. The volume v_i of the cuboid defined by the surfaces A, B and C represents the mixing coefficient b_i associated with grid node $p_{1,2,2}$ in the form of $b_i = 1 - v_i$ based on a normalized inter-node spacing of the grid.

The approach was evaluated successfully on a simulated robotic limb. It was shown that trajectories could be generated for untrained goal states as well as for untrained load cases. Figure 3(a) and (e) show results for learned load cases and (b)-(d) for interpolated loads. The distance to target error never exceeded 0.01 m.

Acknowledgements

This work was supported by a HIF grant from University of Applied Sciences Bielefeld and by Center of Excellence "Cognitive Interaction Technology" (CITEC, EXC277).

References

Albu-Schaeffer, A., Ott, C. & Hirzinger, G. (2007), 'A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots', *The International Journal of Robotics Research* **26**(1), 23–39.

Atkeson, C. G., Moore, A. W. & Schaal, S. (1997), 'Locally weighted learning', *Artificial Intelligence Review* **11**(1-5), 11–73.

Boblan, I. & Schulz, A. (2010), A humanoid muscle robot torso with biologically inspired construction, *in* '41th International Symposium on Robotics - ISR 2010 and 6th German Conference on Robotics - ROBOTIK 2010', pp. 934–939.

Da Silva, B. C., Konidaris, G. & Barto, A. G. (2012), Learning parameterized skills, *in* 'Proceedings of International Conference of Machine Learning'.

Hogan, N. (1985), 'Impedance control - An Approach to Manipulation 1. Theory', *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME* **107**(1), 1–7.

Ijspeert, A. J., Billard, A., Nakanishi, J. & Schaal, S. (2003), Learning attractor landscapes for learning motor primitives, *in* 'in Advances in Neural Information Processing Systems', MIT Press, pp. 1523–1530.

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P. & Schaal, S. (2013), 'Dynamical movement primitives: Learning attractor models for motor behaviors', *Neural Computation* **25**(2), 328–373.

Ijspeert, A. J., Nakanishi, J. & Schaal, S. (2002), Movement imitation with nonlinear dynamical systems in humanoid robots, *in* 'In IEEE International Conference on Robotics and Automation (ICRA2002)', Vol. 2, Washington, DC, pp. 1398–1403.

Kober, J., Muelling, K., Kroemer, O., Lampert, C. H., Schoelkopf, B. & Peters, J. (2010), Movement templates for learning of hitting and batting, *in* M. Rakotondrabe & I. Ivan, eds, 'IEEE International Conference on Robotics and Automation - ICRA 2010', IEEE International Conference on Robotics and Automation ICRA, IEEE, IEEE, 345 E 47TH ST, New York, NY 10017 USA, pp. 853–858.

Kupcsik, A. G., Deisenroth, M. P., Peters, J. & Neumann, G. (2013), Data-efficient generalization of robot skills with contextual policy search, *in* 'Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI)'.

Mülling, K., Kober, J., Kroemer, O. & Peters, J. (2013), 'Learning to select and generalize striking movements in robot table tennis', *International Journal of Robotics Research* **32**(3), 263–279.

Paskarbeit, J., Annunziata, S., Basa, D. & Schneider, A. (2013), 'A self-contained, elastic joint drive for robotics applications based on a sensorized elastomer coupling design and identification', *Sensors and Actuators A: Physical* **A**(199), 56–66.

Pratt, G. A. & Williamson, M. M. (1995), 'Series elastic actuators', *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots'* **1**, 399–406.

Schaal, S., Mohajerian, P. & Ijspeert, A. J. (2007), Dynamics systems vs. optimal control - a unifying view, *in* P. Cisek, T. Drew & J. Kalaska, eds, 'Computational Neuroscience: Theoretical Insights into Brain Function', Vol. 165 of *Progress in Brain Research*, Elsevier Science BV, Sara Burgerhartstraat 25, PO BOX 211, 1000 AE Amsterdam, NETHERLANDS, pp. 425–445.

Tsagarakis, N. G., Laffranchi, M., Vanderborght, B. & Caldwell, D. G. (2009), A compact soft actuator unit for small scale human friendly robots, *in* 'IEEE International Conference on Robotics and Automation, 2009', pp. 4356–4362.

Vijayakumar, S. & Schaal, S. (2000), Locally weighted projection regression : An O(n) algorithm for incremental real time learning in high dimensional space, *in* P. Langley, ed., 'Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)', Stanford University, Morgan Kaufmann Publishers, pp. 1079–1086.

Zinn, M., Khatib, O., Roth, B. & Salisbury, J. K. (2004), 'Playing it safe', *IEEE Robotics & Automation Magazine* **11**(2), 12–21.