

Towards a learnt neural body schema for dexterous coordination of action in humanoid and industrial robots

**Ajaz Ahmad Bhat, Sharath Chandra
Akkaladevi, Vishwanathan Mohan,
Christian Eitzinger & Pietro Morasso**

Autonomous Robots

ISSN 0929-5593

Auton Robot

DOI 10.1007/s10514-016-9563-3



Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Towards a learnt neural body schema for dexterous coordination of action in humanoid and industrial robots

Ajaz Ahmad Bhat¹  · Sharath Chandra Akkaladevi² · Vishwanathan Mohan¹ · Christian Eitzinger² · Pietro Morasso¹

Received: 10 March 2015 / Accepted: 17 March 2016
© Springer Science+Business Media New York 2016

Abstract During any goal oriented behavior the dual processes of generation of dexterous actions and anticipation of the consequences of potential actions must seamlessly alternate. This article presents a unified neural framework for generation and forward simulation of goal directed actions and validates the architecture through diverse experiments on humanoid and industrial robots. The basic idea is that actions are consequences of an simulation process that *animates* the internal model of the body (namely the *body schema*), in the context of intended goals/constraints. Specific focus is on (a) *Learning*: how the internal model of the body can be acquired by any robotic embodiment and extended to coordinated tools; (b) *Configurability*: how diverse forward/inverse models of action can be ‘composed’ at runtime by coupling/decoupling different body (body + tool) chains with task relevant goals and constraints represented as multi-referential force fields; and (c) *Computational simplicity*:

how both the synthesis of motor commands to coordinate highly redundant systems and the ensuing forward simulations are realized through *well-posed* computations without kinematic inversions. The performance of the neural architecture is demonstrated through a range of motor tasks on a 53-DoFs robot iCub and two industrial robots performing real world assembly with emphasis on dexterity, accuracy, speed, obstacle avoidance, multiple task-specific constraints, task-based configurability. Putting into context other ideas in motor control like the Equilibrium Point Hypothesis, Optimal Control, Active Inference and emerging studies from neuroscience, the relevance of the proposed framework is also discussed.

Keywords Body schema · Passive motion paradigm · iCub · Motor control · Industrial assembly

Electronic supplementary material The online version of this article (doi:[10.1007/s10514-016-9563-3](https://doi.org/10.1007/s10514-016-9563-3)) contains supplementary material, which is available to authorized users.

✉ Ajaz Ahmad Bhat
ajaz.bhat@iit.it

Sharath Chandra Akkaladevi
sharath.akkaladevi@profactor.at

Vishwanathan Mohan
vishwanathan.mohan@iit.it

Christian Eitzinger
christian.eitzinger@profactor.at

Pietro Morasso
pietro.morasso@iit.it

¹ Robotics, Brain and Cognitive Science Department, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genoa, Italy

² Robotics and Assistive Systems, PROFACTOR GmbH, Im Stadtgut A2, 4407 Steyr-Gleink, Austria

1 Introduction

One of the central challenges in action generation is to control the high-dimensional joint-space of a body (a robot or a human) while acting in a task-space of much lower dimensionality. This dimensionality imbalance termed as motor redundancy implies that an infinite number of solutions exist for reaching the same goal in the task space and the robot has to choose one from many (Bernstein’s Degrees of Freedom Problem ([Bernstein 1967](#))). In addition, a robotic controller operating in real world has to tackle multiple task-specific constraints both internal, such as bounding the range of rotation of a joint; and external, such as avoiding obstacles or reaching the target with a specific end-effector pose to allow further manipulations of grasping, pushing or inserting etc. Redundancy is worthwhile here as it can allow the body to avoid obstacles, joint limits, limb interference and attain

more desirable task-specific postures, but this necessitates the ability to produce many solutions for one goal (Lashley's Principle of Motor Equivalence, [Lashley 1933](#)). Based on the nature of motor task, robotic controllers must also be dynamically configurable i.e. able to recruit in run-time different body segments (arms/torso) and tools on-demand for executing the task. In this view, the desired computational mechanisms for robots must be multi-referential, in the sense of allowing task-modulated bidirectional dynamic interactions among different spaces: end-effector space, joint space, and spaces related to the DoFs of manipulated tools. This is needed because so many tasks particularly in industry for manufacturing, assembling, cutting, welding and painting etc. require robots to use tools. Moreover, in order to exploit the complex body to its maximum potential in unstructured environments, a robotic controller must not only help in shaping the motor output to generate movement but also help provide the robot with information on the feasibility and consequence of actions. In this article, we propose an internal representation of the body (the body schema) as a flexible controller for both simulation and execution of action that addresses the problem of redundancy through well-posed computations, considers various task-specific constraints as well as allows dynamic extension to representations for coordination of different body parts and tools.

1.1 Synergies, equilibrium point hypothesis and optimal control

[Bernstein \(1935\)](#) was among the first to propose that synergies allow the brain to get rid of task-irrelevant degrees of freedom, thus focusing on the simpler problem of mastering a smaller number of task-relevant variables. The concept of synergy was accompanied in early studies by the attempt to assign a regulatory role to the spring-like behavior of muscles ([Bernstein 1935](#)) when such *springness* was indeed suggested by several experimental studies in the 60s and 70s ([Asatryan and Feldman 1965](#); [Bizzi and Polit 1978](#), among others). The central idea was that complex actions can indeed be achieved without a complex, high dimensional optimization process by simply allowing the intrinsic dynamics of the neuromuscular system to seek its equilibrium state when triggered by intended goals. This idea, termed the Equilibrium Point Hypothesis (EPH), exploited two beneficial properties of the neuromuscular apparatus of the body: (1) to induce locally (in a muscle-wise manner) an instantaneous disturbance compensation action, and (2) to induce globally (in a total body-wise manner) a multi-dimensional force field with attractor dynamics. Innovative aspects of the decades old EPH idea were its strong grounding in the biomechanics of the body and the apparent computational simplicity in solving the degrees of freedom problem.

The first attempt for a mathematical formulation of thinning out a particular spatio-temporal pattern for a reaching task among many possible solutions was formulated by [Flash and Hogan \(1985\)](#), in the framework of the classical engineering design technique: optimal control theory (OCT). OCT approaches make use of some form of a Jacobian pseudo-inverse with local null space optimization in order to determine the inverse kinematics ([Baillieul and Martin 1990](#); [Liégeois 1977](#); [Whitney 1969](#)). However, owing to issues like figuring out of the *cost function*, heavy computational costs in calculating pseudo-inverses ([Bryson 1999](#); [Scott 2004](#)), biological implausibility ([Guigon 2011](#); [Todorov 2006](#)) etc., OCT and other prevailing approaches are still being debated ([Friston 2011](#); [Mohan and Morasso 2011](#); [Pickering and Clark 2014](#)). In addition, for cognitive robots interacting with unstructured environments, it is difficult to identify and carefully craft a cost function that may promote the emergence/maturation of purposive, intelligent behavior. This is relevant if we want to go beyond reach/grasp movements to more complex manipulation scenarios in task space, such as recruiting multiple body segments in sequence or parallel, avoiding obstacles, operating in non-deterministic environments with unforeseeable force perturbations ([Khatib 2004](#)) and use of tools which in fact begins once an object of interest is reached and grasped. OCT approaches to task/operational space control discussed in literature can resolve kinematic redundancy at the velocity ([Nakamura and Hanafusa 1987](#)), acceleration ([De Luca and Oriolo 1991](#); [Luca et al. 1992](#); [Hollerbach and Suh 1987](#); [Hsu et al. 1989](#); [Senda 1999](#)) and force levels ([Featherstone and Khatib 1997](#); [Khatib 1987](#)). However, as authors in ([Nakanishi et al. 2008](#)) point out: “While impressive results have been generated with advanced operational space controllers on simulated humanoid robots in recent studies ([Khatib 2004](#); [Sentis and Khatib 2005](#)), there is a lack of understanding of whether simpler control methods could achieve similar results, and in how far the success of idealized simulations extends to actual robot implementations.”

In contrast to Jacobian pseudo-inverse approach which requires methods like damped least squares method (DLS, also called the Levenberg–Marquardt method) to stabilize matrix inversion near singularities ([Buss and Kim 2005](#); [Nakamura and Hanafusa 1986](#); [Wampler 1986](#)), other works propose Jacobian transpose method ([Arimoto 2005](#); [Balestrino et al. 1984](#); [Wolovich and Elliot 1984](#)) to avoid matrix inversions. However application of these Jacobian transpose methods has been found difficult in tasks other than simple reaching to static targets. Another interesting approach well-discussed in literature is the application of neural networks to resolve the forward and inverse robot kinematics problems ([Bekey and Goldberg 2012](#); [Lewis et al. 1998](#)). For example, on a robot named MURPHY, [Mel \(1988\)](#) proposed and implemented a connectionist architecture with

two interconnected populations of neural units, one for retinotopic visual representation and the other value-coded joint representation, which map the relationships between the positions of robot's joints and the state of its visual field. Another (Lee and Kil 1990) of the earlier works proposed a Bidirectional Mapping Neural Network, composed of a multilayer feedforward network with hidden units having sinusoidal activation functions and a feedback network forming a recurrent loop around the feedforward network. The feedforward network is trained to represent the forward kinematics while the feedback network iteratively generates joint angle updates based on a Liapunov function to modify the current joint angles in such a way that the output of the forward network converges to the desired Cartesian position and orientation. Jordan and Rumelhart (1992) proposed a feed-forward network consisting of both a forward and an inverse model where different types of error signals (prediction error, performance error and predicted performance error) are back-propagated into the network to learn the two models. Similarly, methods like counter propagation networks (Nguyen et al. 1990) and recurrent neural networks (Li et al. 2012) have been successfully implemented to encode forward and/or inverse kinematics for redundant robotic manipulators. These studies do address the redundancy resolution problem but only for simple tasks like reaching and do not consider task-specific constraints in most cases. Recent works (Peters and Schaal 2008; Salaün et al. 2009) do address learning in task space control incorporating internal and external constraints to some extent but do neither account for recruitment of different body segments on the fly, nor the learning and extension of body to the use of tools.

1.2 Actions as animations of the body schema

With emerging trends in neuroscience acting as a motivating force to revisit old ideas of synergy formation and equilibrium point hypothesis, in this study we propose that actions (overt as well as covert) are the consequences of a simulation on an internal representation of the body viz., the *body schema*, with the attractor dynamics of force fields induced by the goal and task specific constraints. In this work, we go against the stream of traditional robotics and build a system for a growing, adaptive and configurable body schema that is common to both simulation as well as execution of movement. This neural implementation takes into account growth of the body over time as well as can be coherently extended with other tool-relevant neural networks leading to expansion of the body schema. It brings the system closer to idealizing a developing human body schema where constant changes in kinematics keep occurring in a uniform way. The novelty lies in the fact that a robot's kinematics is not constant as generally assumed in the field of robotics. The proposed

body schema implementation works with well-posed computations, hence is computationally cheap; brings in flexibility and task-specific configurability; and manages task-specific constraints dynamically. The exploitation of a configurable body schema allows the flexibility of assembling appropriate body-chains on the fly based on the nature of the motor task and the body segment (and tool) chosen for its execution. The flexibility of creation and modification of such body/tool-chains in run-time is likely to be a fundamental operation in motor planning and action synthesis.

Both in the context of a human or a humanoid robot, our view is that the body schema must be:

- (a) *somatotopic* i.e. in accordance with the well-known cortical layout;
- (b) *learnt* i.e. acquired by interacting with the world;
- (c) *action-oriented* and not movement-oriented; here action is defined as an animation of the body schema;
- (d) *multi-referential* i.e., should act as a synergy generation machine coordinating multiple motor spaces: joints, end effectors, tools;
- (e) *plastic, task-oriented and expandable* in order to support skill learning and incorporate the internal representation of tools and constraints;
- (f) *mass-less* and not involving precise details of muscle activations etc., so as to operate equally in overt and covert conditions where there is no neuromuscular activity;
- (g) *global but configurable*, in the sense that each action implicitly recruits all the degrees of freedom but configuration is equivalent to a task-dependent pruning.

Our work builds up on these topics, emphasizing the crucial role of an adaptive, configurable body schema. What makes it more interesting is the fact that, such ideas play a fundamental role not just in control, but also in imagination, understanding and reasoning about action through a mechanism of decentralized *local to global* computing and learning.

The rest of the paper is structured as follows. Section 2.1 presents the computational framework for the internal simulation of body schema for goal-directed actions. Section 2.2 discusses the neural implementation of the body schema and how it can be learnt for an embodied robot while discussing learning on humanoid robot, iCub. Section 2.3 discusses how such a plastic body schema can be extended to the coordination of tools. In Sect. 3.1 we show results of training the neural network that models the body schema and how other subtleties like symmetry of the body can be exploited for an efficient body schema representation. In Sect. 3.2 we evaluate the model in a real world industrial assembly task on a humanoid and two industrial robots showing how task-specific constraints and task-based configurability are taken into account. The section also details the use of a perception-

action loop to achieve higher accuracy in an assembly task like ours; ease of porting the neural framework to any other robotic platform with results on two industrial robots; obstacle avoidance using *shaped* trajectories; exploiting a body schema to reason about potential actions and open source release of the developed source code and documentation. We conclude with a discussion on how the proposed model compares with the previous approaches to robot control, how the body-schema is a link between motor control and cognition, and how the proposed model fits into the picture. We end the discussion with a brief outline of future extensions to our work.

2 Neural acquisition of an internal body model by a humanoid robot

2.1 The PMP model

To explain the neural control of movement, Equilibrium Point Hypothesis (EPH) proposed that the posture of the body is not directly controlled by the brain in a detailed way but is a *biomechanical consequence* of equilibrium among a large set of muscular and environmental forces. In other words, complex actions can indeed be achieved without a complex, high dimensional optimization process by simply allowing the intrinsic dynamics of the neuromuscular system to seek its equilibrium state when triggered by *intended* goals. A plausible extension to the theory of EPH is to consider that what occurs in the brain during both mental simulation and overt execution of action, reflects an endogenous cortical dynamics very similar to the physical dynamics implicit in EPH, but realized through the *animation* of a flexible, plastic and configurable internal representation of the body, with the attractor dynamics of force fields induced by the intended goal. This line of thought emerged during the 80s taking shape as the Passive Motion Paradigm (PMP) and retreaded recently (Mohan and Morasso 2011; Mohan et al. 2009; Mussa-Ivaldi et al. 1988). The basic idea can be formulated in qualitative terms by suggesting that the process by which the brain can determine the distribution of work across a redundant set of joints, when the end effector is assigned the task of reaching a target point in space, can be represented as an internal simulation process on the body schema that calculates how much each joint would move if an externally induced force, i.e. the goal, pulls the end effector by a small amount towards the target. PMP models movements caused by changes in the intrinsic space as movements that cause changes in the extrinsic space; a feature it shares with Active Inference (Friston 2010). This may sound like the *chicken versus egg* problem, but has significance in terms of the computational cost, that inverting the natural direction of causality entails. In addition, we emphasize that it also makes *computational sense* considering that this line of

thinking naturally converts an ill-posed problem into a well-posed problem. The advantage is that the well-posed nature of computation circumvents the need for explicit kinematic inversion and cost function computation (this is evident if we intend to control a 53-DoFs humanoid, like iCub).

2.1.1 Computational model

Let a vector q represent the state of a robot in the intrinsic joint space for a given pose and a vector x identifies the position of the end effector of the robot in the extrinsic workspace for that pose. Then the kinematic transformation $x = f(q)$ can be expressed as: $\dot{x} = J(q) \cdot \dot{q}$ where $J(q)$ is the Jacobian matrix of the transformation. Following steps define the computational structure of PMP (Mohan et al. 2009; Mussa-Ivaldi et al. 1988). Figure 1a gives a graphical representation of the steps involved.

- (1) *Generate a target-centered, virtual force field in the extrinsic space:*

$$F = K_{ext} (x_T - x) \quad (1)$$

where x_T is the target and K_{ext} is the virtual stiffness of the attractive field in the extrinsic space. The intensity of force field F keeps decreasing as the end-effector approaches the target. K_{ext} determines the shape and intensity of the force field. In the simplest case, K_{ext} is proportional to the identity matrix and this corresponds to an isotropic field, converging to the goal target along straight flow lines. More complex curved trajectories like in case of obstacle avoidance or use of tools, can be obtained by either actively modulating or learning the appropriate values of the virtual stiffness (Bhat and Mohan 2015; Mohan et al. 2011).

- (2) *Map the force field from the extrinsic space into virtual torque field in the intrinsic space:*

$$T = J^T F$$

where J^T is the transposed Jacobian matrix which is always well defined. In the next section, we show how these Jacobians can be derived from the learnt body schema model.

- (3) *Relax the arm configuration to the applied field:*

$$\dot{q} = A_{int} T \quad (2)$$

where A_{int} is the virtual admittance matrix in the intrinsic space: the modulation of this matrix affects the relative contributions of the different joints to the overall

reaching movement. In other words, A_{int} is joint compliance matrix that leads to the distribution of the torques among the joint rotations.

- (4) *Map the arm movement into the extrinsic workspace:*

$$\dot{x} = J\dot{q}$$

- (5) *Integrate over time until equilibrium:*

$$x(t) = \int_{t_0}^t J\dot{q}d\tau$$

The last step of integration gives us a trajectory with the equilibrium configuration $x(t)$ defining the final position of the robot in the extrinsic space.

Figure 1a is a graphical representation of the steps involved in case of a single kinematic chain (like an arm). At each time step of this cyclic computational process, target goal x_T in the extrinsic space induces virtual disturbance forces F on the end-effector, which are calculated using the virtual stiffness K_{ext} . These forces are then mapped into equivalent torques T ; this projection is implemented by the transpose Jacobian J^T . These virtual torques T cause incremental joint rotations \dot{q} as allowed by compliances A_{int} of different joints. The incremental change in joint space \dot{q} is mapped to the extrinsic space \dot{x} using the Jacobian matrix J , causing a small displacement of the end-effector towards the intended target. This process progresses cyclically till the time the algorithm converges to an equilibrium state which is reached asymptotically in the following cases:

- When the force field in the extrinsic space is reduced to zero (Eq. 1) which means the end effector reaches the target successfully.
- When the force field in the intrinsic space becomes zero (Eq. 2), although the force field in the extrinsic space is not null and this can happen in the neighborhood of kinematic singularities. In this case, the target cannot be reached, for example because it is outside the workspace; however the final configuration has a functional meaning for the motion planner because it encodes geometric information valuable for re-planning, such as the desired length of a tool to reach the target.

PMP networks are task-specific and hence can be assembled on the fly based on the nature of the motor task and the body chains and tools chosen for execution. As for example, consider upper body coordination in iCub with two target goals (see Fig. 1c, bottom panel). Here iCub has to push the fusebox using the left arm first and then insert the fuse into the fusebox using its right arm. Figure 1b shows composite PMP network for this case with two attractive force fields

applied to the right and left arms of iCub. Note that basic PMP subnetwork shown in Fig. 1a is repeated for the right and the left arm. In this network, there are two additional nodes, (sum) ‘+’ node and (assignment) ‘=’ node. The (sum) ‘+’ node allows the force fields, F_r and F_l , applied to the end-effectors of two or more body segments (e.g. the two arms) to be combined in order to propagate the virtual forces to a common body segment (e.g. the trunk). Therefore, the joint space of this body segment, far away from the end effectors, is recruited by the global force fields and modulated by the local admittance matrix A_w . This motion is then reflected back to the impinging segments, by means of the (assignment) ‘=’ node, thus distributing the movements throughout the overall kinematic structure.

It is to be noted, the above described PMP relaxation process is asymptotically stable, implying that the equilibrium is reached in infinite time. Hence to speed up the operation time of the planner and to control the reaching time of the relaxation process, a time varying gain $\Gamma(t)$ in the form of a time base generator (TBG) is inserted into the model (see Fig. 1a, yellow node). TBG implements the concept of terminal attractor dynamics (Zak 1991). A terminal attractor is an equilibrium point reached in a specified finite time in contrast with the standard attractor systems which converge asymptotically. A TBG can be implemented simply by substituting the relaxation equation (2) with the following one:

$$\dot{q} = \Gamma(t) \cdot A_{int} \cdot T$$

where a possible scalar form of the time-varying gain that implements the terminal attractor dynamics is the following one (it uses a minimum-jerk time base generator $\xi(t)$ with duration τ):

$$\begin{cases} \Gamma(t) = \dot{\xi} / (1 - \xi) \\ \xi(t) = 6 \cdot (t/\tau)^5 - 15 (t/\tau)^4 + 10 (t/\tau)^3 \end{cases}$$

$\Gamma(t)$ grows monotonically as x approaches the equilibrium state and diverges to an infinite value in that state, ensuring the convergence of the relaxation in finite time. Other than implementing the TBG as a scalar function of time like the one given above, interestingly, a neural representation of the time-base generator is also feasible. Appendix presents a neural implementation of the TBG.

2.2 Acquisition of a neural representation of the body schema by iCub humanoid

That humans have an integrated, internal representation of their body (the body image or the body schema¹) is strongly

¹ The difference between body image and body schema is disputed and is somehow fuzzy. For our purpose we assume that they are two sides

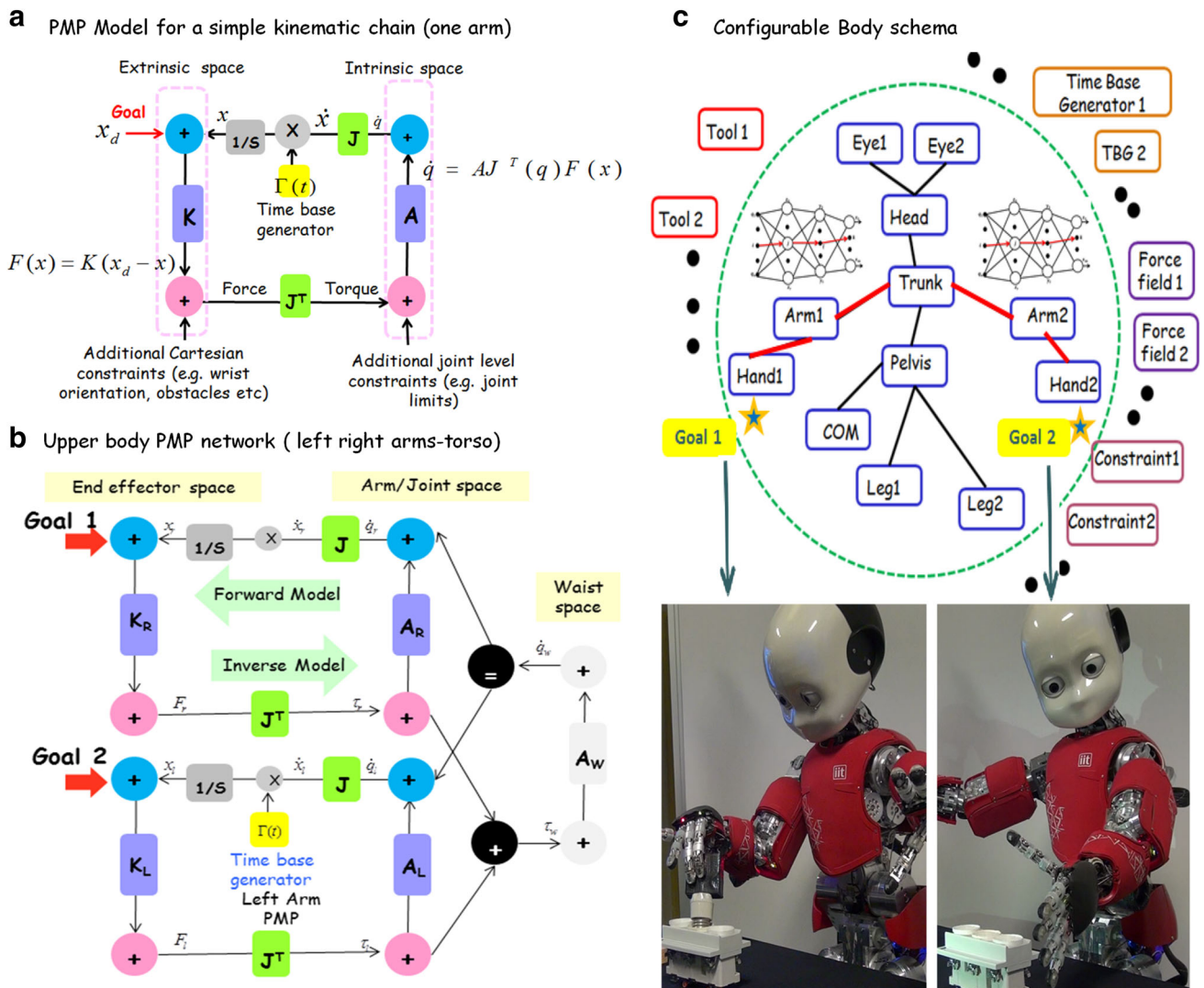


Fig. 1 **a** Shows a basic PMP network for a simple kinematic chain (as the arm). In this simple case, the network is grouped into two motor spaces, extrinsic or the end-effector space and intrinsic or the joint space. Each motor space consists of a generalized displacement node (blue) and a generalized force node (pink). Vertical connections (purple) denote impedances (K stiffness, A admittance) in the respective motor spaces and horizontal connections (green) denote the geometric relation between the two motor spaces, represented by the Jacobian (J). External and internal constraints, represented as other task-dependent force/torque fields, bias the relaxation to equilibrium other penalty functions. The network includes a time base generator (yellow) which enables the system to reach equilibrium in finite time. **b** PMP network for iCub upper body coordination with two target goals, one each to the right and the left arm of the robot. Note that the basic PMP sub-

network shown in **a** is repeated for the right and the left arms. In this network, there are two additional nodes, (sum) '+' node and (assignment) '=' node. The (sum) '+' node allows the two force fields to be combined in determining the motion of the trunk. The (assignment) '=' node propagates the motion of the trunk to the two arms. In this way, the motion of each arm is influenced by both force fields. Observe that the network is fully connected; connectivity articulated in a fashion that all transformations are well-posed. **c** Top panel shows the task-specific configurability of the body schema. Based on the goal and end-effector, only the concerned motor parts are activated (as shown by red lines). As shown in the bottom panel of the figure and the video (Online Resource 1), iCub uses the right arm-torso combination in one case and the left arm-torso one in the other based on the task (Color figure online)

suggested by a variety of pathological conditions which can only be explained by a deficient internal representation of

Footnote 1 continued

of the same coin: the former one stresses the static component, mainly based on proprioceptive information whereas the latter is related to the dynamic synergy formation function.

the body (Head and Holmes 1911). More recent studies (for reviews see: Graziano and Botvinick 2002; Haggard and Wolpert 2005) have identified the different cortical areas that may contribute to this function (area 5 in the superior parietal lobe and possibly premotor and motor areas) and to the multimodal integration of proprioceptive, visual, tactile and

motor feedback signals that is necessary for maintaining a coherent spatiotemporal organization. This body representation is not a static structure like the Penfield's homunculus but keeps on changing as the body keeps growing/ changing over time, for example, changes from an infant to a fully grown man or due to limb amputations. In addition, there can be direct extensions to this schema for example, by the use of tools (Umiltà et al. 2008) which work as new end-effectors of the body. On top of this, a body schema in primates is configurable to a vast number of tasks their bodies have to perform over their lifetime and to the number of end-effectors they use to perform actions. Since the body schema is plastic biologically, the kinematics of the body has to update constantly; this makes it questionable if the current trend in robotics of a *fixed* kinematics is really the right way to look at the problem. Only a flexible system can show a brain-like property, so any system that has to imitate and explain how the brain coordinates multiple degrees of freedom, or perform at a comparable degree of dexterity human body possesses, must have flexibility inbuilt. This is because it is impossible to know beforehand all the tools, the system (body/robot) will use and the tasks it will perform in future. Hence a neural body schema representation of a robot controller, in our view, is the most promising way to achieve task-specific configurability and adaptability in robot action coordination.

This body schema representation has been successfully implemented in experiments on the humanoid robot iCub and two other industrial robotic arms (see Supplementary Videos). iCub is an open-source platform based child humanoid robot (roughly 1m tall), with 53 degrees of freedom: 7 DoFs for each arm, 9 for each hand, 6 for head, 3 for trunk and 6 for each leg. With special focus being given to manipulation and interaction of the robot with the real world, iCub is characterized by highly sophisticated hands, flexible oculomotor system and sizable bimanual workspace. All the low-level code and documentation is provided open source by the RobotCub Consortium (<http://www.robotcub.org>), together with the hardware documentation and CAD drawings. To perform our experiments, we augmented the iCub platform with a Kinect PrimeSense camera for better visual perception. Briefly stating, visual perception system is structured around two major components. The first of these components is concerned with the detection of the robot end effector and objects of interest in RGB-D images that have been captured with the Kinect sensor (Cai et al. 2013). Following the detection process, the location in space of the detected end-effector or the objects is estimated using a 3D pose estimation method from Lourakis and Zabulis (2013).

In this study, PMP body schema model is used to coordinate all the degrees of freedom involved in the left arm-torso-right arm chain of the baby humanoid, i.e. 7+3+7 DoFs in total. The body schema representation is derived

from the kinesthetic learning by the robot. The iCub arms are moved in the peripersonal space of the robot and the end-effector location is tracked using vision and sensory systems. This is very similar to the way motor babbling occurs in infants for body schema representation. To get a more accurate estimate of the correspondence between the extrinsic end-effector space and the intrinsic joint space, we iterate the joint angle values of the robot in a customized volume of its usable workspace, to generate corresponding end-effector locations using forward kinematics of the robot. Both ways are computationally equivalent in terms of giving rise to data: a *training set* of joint rotation readings with the corresponding coordinates of the end-effector. Now using joint angle vectors as input and the corresponding 3D end-effector location vectors as desired output, we train a standard back-propagation neural network with two hidden layers, using MATLAB. Following the training process, we can recover the Jacobian J from the weights of the trained neural network: that represents the body schema, mapping the extrinsic and intrinsic spaces. Here, we mention again that previous studies (Jordan 1990; Jordan and Rumelhart 1992; Lee and Kil 1990; Mel 1988; and others) have also employed back-propagation networks or other kinds of connectionist neural networks to learn the forward kinematics of robots. In our view, many of such supervised learning algorithms can be used for this training process. The choice of a standard back-propagation network for learning of the forward model, in our case, is arbitrary. The core idea to convey is that the proposed body schema model can be learnt and can take a well-plausible neuro-biological representation; which makes it a candidate computational model for action generation in both humans and humanoids. The proposed body schema model goes beyond learning of a single, fixed body-chain kinematics, to a distributed internal representation of overall body schema composed of different body segments and tools. This allows learning and recruitment of different body schema segments on the fly as well as learning and use of tools while incorporating internal and external constraints at the same time for task-specific control. Moreover, in contrast to these works, the proposed body schema model does not require learning the robot's inverse kinematics.

We train a multilayer feed forward neural network with two hidden layers as shown in Fig. 2, to learn the mapping $X = f(Q)$ where $Q = \{q_i\}$ is the input vector (of joint angles), $X = \{x_k\}$ is the output vector (representing 3D position/orientation of the end-effector) and $Z = \{z_j\}$ and $Y = \{y_l\}$ vectors are the output of first and second hidden layer units respectively. Equation 3 expresses the mapping, where $\Omega = \{\omega_{ij}\}$ are connection weights from the input layer to first hidden layer, $O = \{o_{jl}\}$ are the connection weights between two hidden layers, $W = \{w_{lk}\}$ are the connection weights from the second hidden layer to the output layer, $H = \{h_j\}$ are the net inputs to the neurons of the first hidden

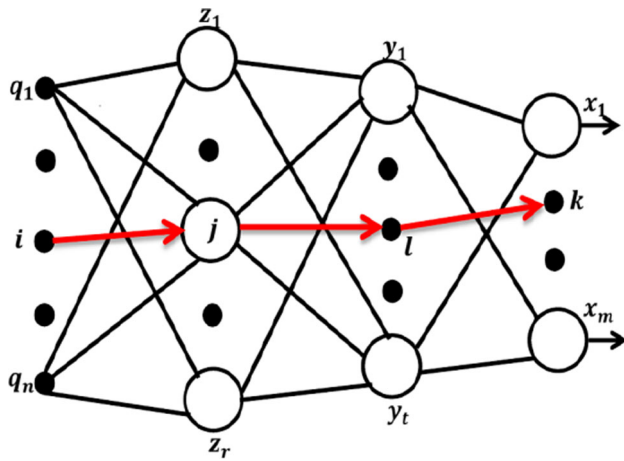


Fig. 2 A feed forward neural network composed of an input layer $\{q_i\}$, two hidden layers $\{z_j\}$ and $\{y_l\}$ and an output layer $\{x_k\}$. See text for details

layer and $P = \{p_l\}$ are net inputs to the second hidden layer. Neurons of the two hidden layers fire using the hyperbolic tangent function; the output layer neurons are linear.

$$X = f(Q) \Rightarrow x_k = \sum_l w_{lk} \cdot g \left(\sum_j o_{jl} \cdot g \left(\sum_i \omega_{ij} q_i \right) \right) \quad (3)$$

After training the neural network using sensorimotor data generated by the robot, the Jacobian J can be extracted from the learnt weight matrices using the following expression:

$$J = \frac{\partial x_k}{\partial q_i} = \sum_l w_{lk} \cdot g'(p_l) \sum_j o_{jl} \cdot g'(h_j) w_{ij}$$

where $h_j = \sum_i \omega_{ij} q_i$ and $p_l = \sum_j o_{jl} g(h_j)$.

2.3 Extension of the body schema to coordinated tools

The capability to use tools in order to achieve goals is a crucial component of motor cognition in humans, primates and some corvids too. The neurophysiological analysis of tool-use in Japanese macaques has shown that the acquisition of these skills is accompanied by long-lasting modifications of their body schema (Iriki et al. 1996; Maravita and Iriki 2004). Extensive tool-use modifies their body schema to incorporate a central representation of the tool. These changes are compatible with the notion of the inclusion of tools in the body schema as if the end effector (e.g. the hand) were elongated to the tip of the tool (Maravita and Iriki 2004; Umiltà et al. 2008).

A full neural implementation of PMP can also be coherently extended with other tool relevant neural networks. The tool becomes an extension of the end effector, leading to the expansion of the body schema. In PMP, the tool space is represented exactly as the body (see Fig. 3b), by means of a generalized force and position node, linked vertically by a virtual admittance matrix A_{Tool} and horizontally by the tool Jacobian matrix J_{Tool} . The admittance matrix A_{Tool} characterizes the incremental transformation from force information to position information in the tool space. The tool Jacobian matrix J_{Tool} forms the interface between the body and the tool and represents the geometrical relationship between the tool and the concerned end effector. While learning to use different tools, it is the tool Jacobians at the interface that are needed to be learnt. Based on the tool being coordinated, it is necessary to load the appropriate device Jacobian associated with it. In order to get the tool Jacobian J_{Tool} , the robot has to practice with the tool to generate sequences of sensorimotor data that includes:

- (1) the instantaneous position of the two hands $Q \in (x_R, y_R, z_R, x_L, y_L, z_L)$ coming from proprioception (and cross-validated by forward model output of PMP i.e. position node in end effector space).
- (2) the resulting consequence i.e. the location of the tool effector $X : (x, y, z)_{Tool}$, perceived through vision and reconstructed to Cartesian coordinates.

As the robot acquires this sensorimotor data by practicing with the tool, a neural network can be trained to learn the mapping $X = f(Q)$. We use a multilayer feedforward network with one hidden layer, where $Q = \{q_i\}$ is the input array (end effector position), $X = \{x_k\}$ is the output array (tool position), and $Z = \{z_j\}$ is the output of the hidden units. The mapping can be expressed as shown in Eq. (4), where $\Omega = \{\omega_{ij}\}$ are the connection weights from the input to the hidden layer, $W = \{w_{lk}\}$ are the connection weights from the hidden to the output layer, $H = \{h_j\}$ are the net inputs to the neurons of the hidden layer. The neurons of the hidden layer are characterized by the logistic transfer function; the output layer is composed of linear neurons.

$$X = f(Q) \Rightarrow \begin{cases} h_j = \sum_i \omega_{ij} q_i \\ z_j = g(h_j) \\ x_k = \sum_j w_{jk} z_j = \sum_j w_{jk} \cdot \left(g \left(\sum_i \omega_{ij} q_i \right) \right) \end{cases} \quad (4)$$

Once the neural network is trained on the sequences of sensorimotor data generated by the robot, the tool Jacobian J_{Tool} can be extracted from the learnt weight matrix by applying chain rule in the following way:

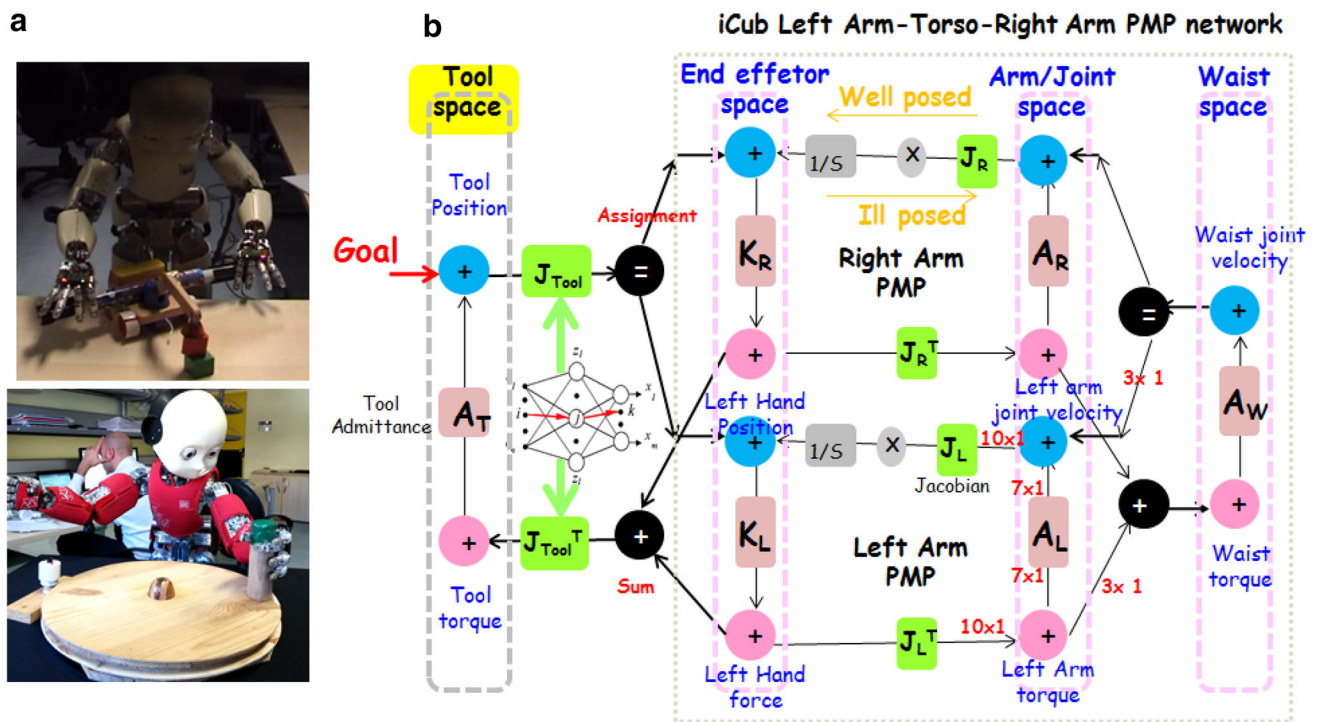


Fig. 3 **a** Shows iCub using two different tools; employing the left arm-torso-right arm combination in one case (*upper*) and the left-arm-torso chain in the other (*lower*). **b** Shows the composite forward/inverse model of a PMP network with a tool held by two arms. The jacobians of the tool are extracted from the neural network as shown. As the target generated force field attracts the tool effector, the end-effectors are con-

sequently pulled to respective positions so as to allow the tool effector to reach the goal. At the same time, the joints of the two arms and the waist are being pulled to a configuration that allows the two hands to reach those desired respective positions. This process of incremental updating of every node in the network continues till the time the tool effector reaches the goal

$$J_{Tool} = \frac{\partial x_k}{\partial q_i} = \sum_j \frac{\partial x_k}{\partial z_j} \frac{\partial z_j}{\partial h_j} \frac{\partial h_j}{\partial q_i} = \sum_j w_{jk} g'(h_j) \omega_{ij}$$

We mention here that the number of hidden layers is chosen heuristically based on the complexity of the mapping problem. In the case of learning of the mapping between arm-torso joints and end-effector space, a two-layer neural network was employed because of a large number of degrees of freedom involved (i.e. 10-DoFs in the arm-torso chain). However, in case of the tools, a single layer network was enough to suffice for learning of the accurate Jacobian as a very small number of DoFs is involved. After the tool Jacobians are learnt by robot, the upper body schema network of Fig. 1b is extended to incorporate the tool as shown in Fig. 3b. The resulting Tool+Body PMP network is complete and fully connected to allow goal directed maneuvering of the tool. During goal directed movements like making the *tool effector* reach a specific pose in the Cartesian space, the targeted pose acts on the tool effector which is now the most distal part of the PMP chain. The target induces a force field on the tool effector, this force field is then incrementally circulated to the proximal spaces i.e., end effectors and joints, according to the information flow in Fig. 3b. In case of tools that require generation of quasi-circular trajectories to operate

(like the ones shown in Fig. 3a), learning of appropriate stiffness K and timing parameters is needed to execute the desired spatio-temporal trajectories using the body + tool chain. The learning of the parameters and related experiments on these tools are discussed in Mohan and Morasso (2011) and Bhat and Mohan (2015) and is out of the scope of this work.

It should be noted that the tool admittance A_{Tool} is a property of the tool itself and can be approximately estimated as the ratio of the total force exerted by robot with its hand and the corresponding displacement of the tool. Since the displacement of the tools connected to iCub, in our case, is proportional to the displacement of the iCub's hand, in such normal conditions, the tool admittance is approximated as an identity matrix.

3 Experiments and results

3.1 Training the neural network and exploiting symmetry

To model the body schema on iCub, we use the forward kinematics of the iCub to generate about 200,000 data points for training the neural network. As another way to record

Table 1 Statistical evaluation of the trained neural network

Statistical parameter	Training dataset	Test dataset	Validation dataset
Root mean square error	0.810	1.039	1.101
Standard deviation	1.911	2.784	2.974
Values in mm			

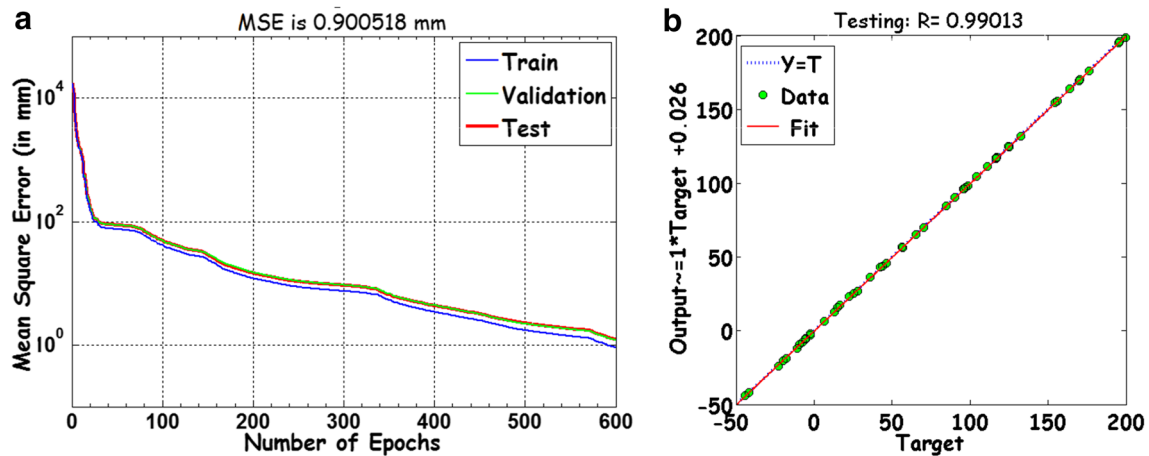


Fig. 4 **a** Shows the decreasing mean square errors in the neural network during the training process for the training data set, validation data set and the test data set. The *error values* decreased to less than 0.1 cm in this session; **b** shows the regression analysis plot for the results on the

y-component of the test data i.e., the graph between the target y-values and the corresponding network output values. The regression value is very close to 1, showing goodness of the fit (Color figure online)

the data, we kinesthetically move the iCub's right arm and torso to babble in its peripersonal space and record the 3D coordinates, returned by the Kinect sensor, of the end effector locations reached by the robot in its workspace and the corresponding joint angles using which these positions were achieved. Using the joint angle values as input vectors and 3D end effector locations as target output, we trained a feed forward neural network using a standard error back-propagation algorithm implemented in MATLAB. Taking a point on the axis of rotation of the torso pitch-joint of the humanoid as frame of reference, our experiments involved working with iCub's two arms and the torso. In order to achieve the neural representation for such a body schema, we exploit the fact that human/humanoid bodies exhibit bilateral mirror symmetry along the sagittal plane. Since each joint angle lies within the same range of values for the two arms, and the torso is shared; the only mirroring would be along one axis with respect to the robot's frame of reference. In case of iCub, it is the y-axis through origin. So for the same set of joint angles of two arms, forward kinematics results in same coordinate values for the two axis (x-axis and z-axis in case of iCub) but the y-axis value is mirrored (meaning positive in sign for one arm and the negative for the other). Hence a neural representation of one half of the body can also be used to employ the other half, thus effectively reducing the burden of training, execution speed and storage.

Since each iCub arm has 7 joints and the torso has 3 DoFs, the input vector consisted of 10 values and the target vector consists of the corresponding 3D location. Number of neurons in first and second hidden layers of the neural network was determined heuristically as 48 and 55 respectively. Five random samples of training datasets, test datasets and validation datasets of sizes 100,000, 20,000 and 20,000 data points respectively were selected from the generated 200,000 data points. The neural network was trained using Levenberg–Marquardt algorithm for all the five samples. Table 1 lists the averages of the root mean square error and the standard deviation over the different learning runs. Values are expressed in millimetres [mm] scale.

The results showed that the averaged root mean square error of the approximator was around 0.1 cm as indicated in Table 1. Figure 4a is a snapshot during one of the training sessions showing graphically the decreasing error curves of the different datasets. Figure 4b shows the regression analysis plot of the corresponding training session where a nice overlap is shown achieved between the y-component of the test data and the resulting output data of the trained network. We also experimented with the size of the training dataset necessary to achieve a well approximating neural network. We observed that decreasing the size of the training dataset even up to around 60,000 data points did not impair the network performance drastically. However networks when trained for

Table 2 Evaluation of neural network performance using variable sizes of training dataset (values in mm)

Dataset size	100,000	60,000	30,000	10,000	5000
RMSE (training)	0.810	3.179	27.22	58.66	310.7

smaller datasets, failed validation tests earlier during the training process and resulted in bad approximators. Table 2 summarizes the results of using different sizes of training datasets, averaged over 5 random samples for each given size. RMSE values are expressed in millimetres [mm] scale.

Figure 5a–c show results of the neural PMP architecture in case of a reaching task, after incorporating the trained neural

network into the model. The right and the left hand are given two target points which are just reflections of each other along the sagittal plane of the humanoid body. The initial positions of the two end-effectors are also mirror reflections of each other. The trajectories of the end-effectors to reach the target also mirror each other, as shown in Fig. 5a. Figure 5b shows how for the two given targets, trajectories evolve along x, y and z directions. Since the targets to reach are placed symmetrically in the y-direction, the x and z trajectories of the two arms overlap each other whereas their y trajectories are symmetric along origin as shown in Fig. 5b. Figure 5c shows the evolving trajectories of ten joint angles of iCub's torso-arm while reaching the corresponding given targets.

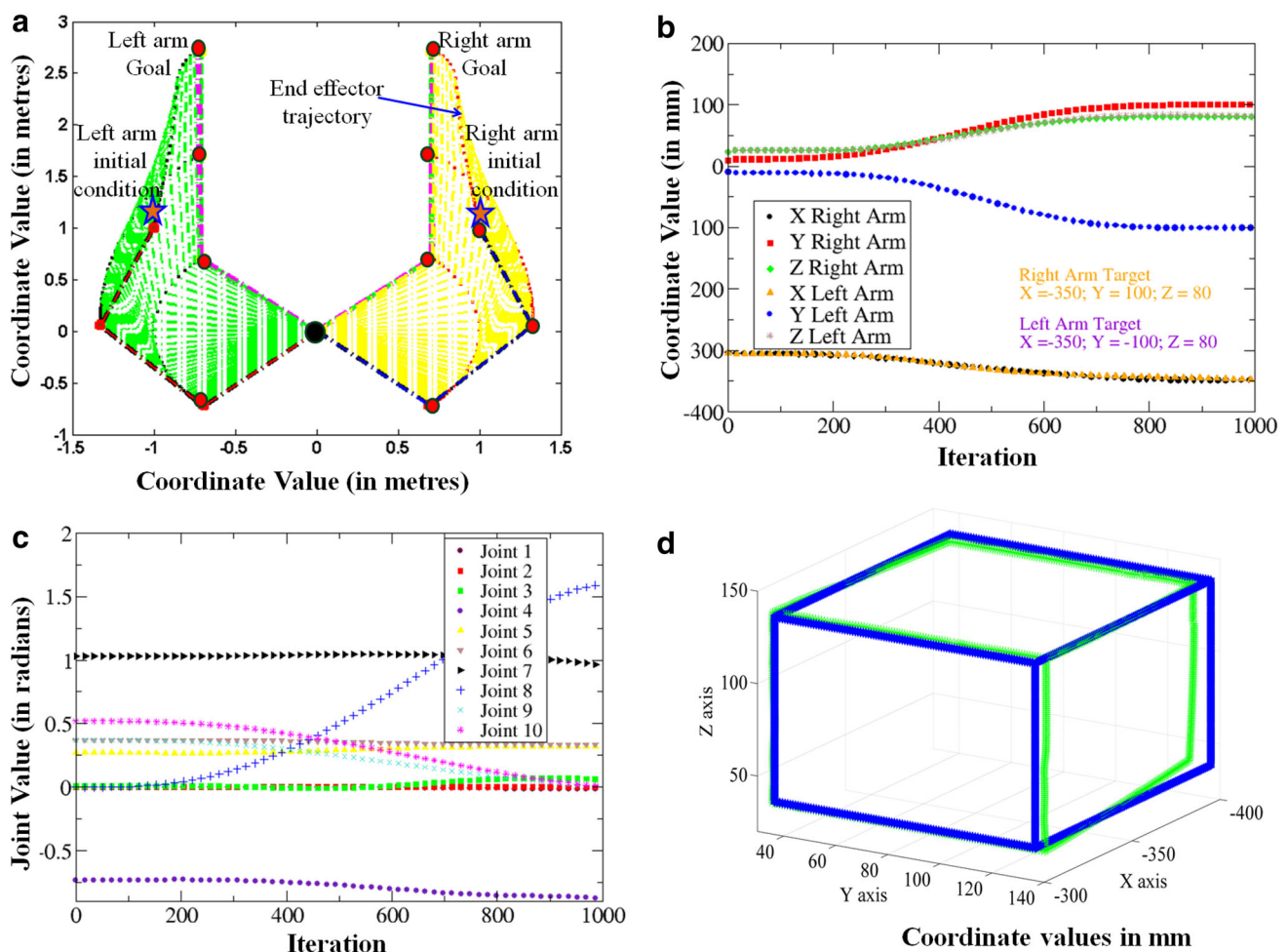


Fig. 5 **a** Shows the end-effector trajectories for the two arms of a body with bilateral symmetry, when they start from same initial joint angle values. If the target points to two end effectors are such that they are mirror reflections of each other along the sagittal plane, the trajectories generated by two arms will also be mirror images if there are same internal and external constraints. **b** Shows the evolution of trajectories along x, y and z direction for two targets placed symmetrically along y

direction reached by left and right arm. The graph shows the x and z trajectories of the two arms overlap each other whereas their y trajectories are symmetric along origin. **c** Depicts the evolution of 10 joint angles of iCub's torso-arm for reaching the targets of **b**. **d** Shows desired targets to the end effector (in blue) in and outside the 3D workspace of the robot and the corresponding solutions reached (in green) (Color figure online)

3.2 Experiments on iCub and industrial robots

3.2.1 Neural architecture for action generation in a real world industrial assembly task

We apply the above described neurally schemed PMP framework in an assembly task setting inspired by practical day-to-day challenges in industrial arena. All the experiments are carried out in the context of a real world assembly scenario where the robot is presented with a fuse and a fuse box, and the task of the robot is to pick the fuse up and finally insert it into a hole of the fuse box. Such a task is very challenging from a motor action/cognition perspective as the sub-tasks involved to realize this goal include reaching, picking, pushing and inserting dynamically during the execution of the plan, which in of themselves are non-trivial. The robot has to reach the object (fuse) while maintaining a desirable posture of the body where joints are in their mid-range values in order to minimize the strain and avoid self-damage. Also the robot has to reach with a high level of accuracy on top of the fuse to make the next step of, picking the fuse up, successful. After reaching, the robot has to grasp the fuse with a particular wrist orientation so that the object is picked up properly making the insertion feasible later on. After picking up the fuse, the robot goes out to reach the fusebox and if the fusebox is not reachable to the hand that has the fuse, the fusebox has to be pushed by the other hand into the workspace area of the hand that has the fuse, to make the insertion possible. Once the fuse box is reachable, the system has to align the fuse on top of the fusebox to reduce any chances of failure while inserting the fuse into fusebox. Failure is quite possible given the fact that the fusebox hole is small with a diameter of 2.5cm.

3.2.2 Performance evaluation of the neural architecture on iCub platform

To evaluate the neural PMP controller, we first performed experiments on iCub Simulator (http://wiki.icub.org/brain/group_icub_Simulation.html) to reach different target locations in the reachable workspace. We took an exhaustive set of 24,000 reachable locations distributed uniformly in the workspace. The results showed that the target locations can be reached accurately with a root mean square error of

Table 3 Results of the neural controller on iCub Simulator in reaching a given target (values in mm)

Statistical parameter	Value
Root mean square error	3.073
Standard deviation	2.984
Root mean square error $[x , y , z]$	[4.12, 3.11, 1.30]
Standard deviation $[x , y , z]$	[3.67, 3.10, 1.91]

3.073 mm. Table 3 gives the statistical details of experimental results. All the values are expressed in millimetres [mm].

As obvious from the tabulated statistics, the RMSE error as well as the standard deviation in z-axis was much lower than the corresponding values along x-axis and y-axis. This is due to that fact the volume of chosen workspace (about $275 \times 200 \times 120 \text{ mm}^3$) for generating the training data was smaller in dimension along z-axis than in the other two axes. The resulting greater density of data points along the z-direction lead to better training and performance of the learnt controller along the z-axis than in the other two axes. A further exploration of about 900 target points in and outside the workspace (see Fig. 5d) showed that even if the target is outside the reachable workspace, the controller nevertheless tries to approach the target as well as possible by fully extending the arm to a position that is at a minimum distance from the target. Hence, what we see in such cases is a gentle degradation of performance that characterizes humans in the same situations. Although there is no exact solution to the problem, the network *does its best*.

As this research work was carried out under European Union FP7 project named DARWIN (<http://www.darwin-project.eu>) which resulted in the development of a dexterous cognitive system for assembly tasks, the experimental results were carried out to validate the overall DARWIN cognitive architecture wherein the proposed PMP body schema controller was implemented for action generation in iCub and two industrial robots. A broad set of experiments was carried out to document the performance of the overall cognitive architecture in different variations of the assembly task. A complete evaluation and detailed specifications of the DARWIN cognitive system can be found in a deliverable (D9.4) of the EU FP7 project DARWIN (DARWIN D9.4 2014). To summarize the results here, we define;

$$\text{Pick Success Rate} = \frac{\text{Number of times a fuse is picked up successfully by iCub}}{\text{Number of attempts iCub makes to pick up the fuse}}$$

$$\text{Insertion Success Rate} = \frac{\text{Number of times a fuse is successfully inserted by iCub}}{\text{Number of attempts iCub makes to insert the fuse}}.$$

$$\text{Overall Task Success Rate} = \frac{\text{Number of successful fusebox assemblies}}{\text{Number of fusebox assembly attempts}}$$

Table 4 Results of iCub performance in an industrial assembly task

Pick success rate	Insertion success rate	Overall task success rate
70 %	70 %	60 %

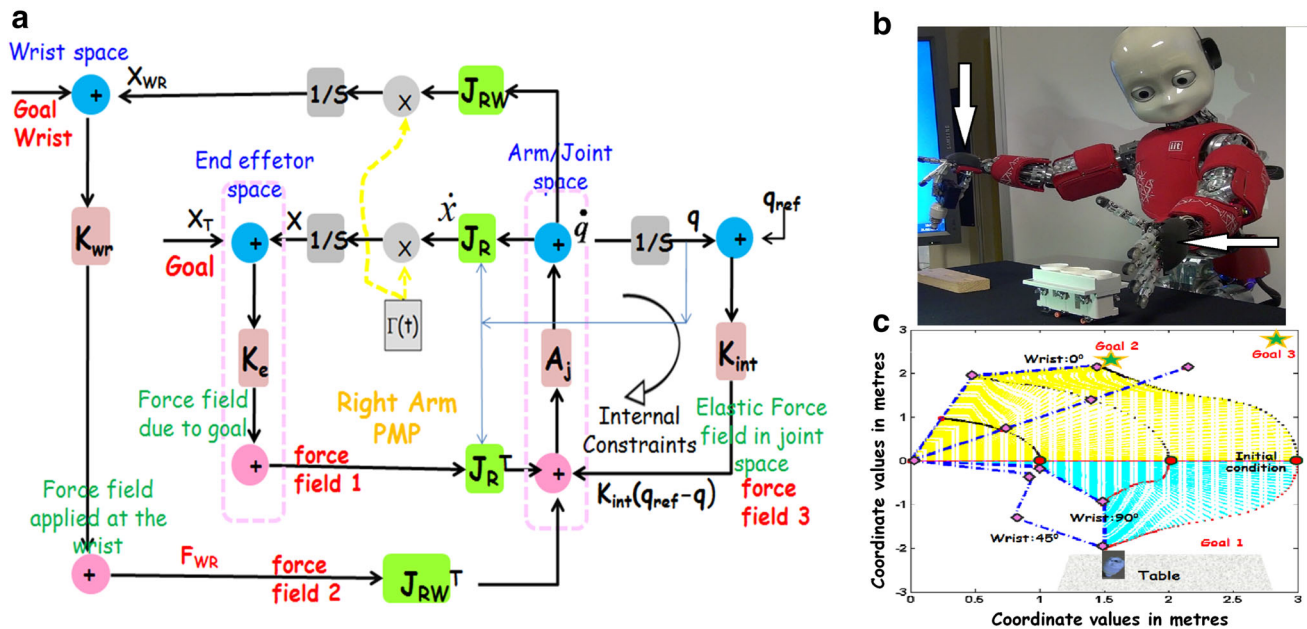


Fig. 6 **a** Composite forward/inverse model with two attractive force fields applied to the arm, a field F_1 that identifies the desired position of the hand/fingertip and a field F_2 that helps achieving a desired pose of the hand via an attractor applied to the wrist. This network assumes that the shoulder of the robot is *grounded*: this means that the two force fields do not propagate beyond the shoulder joint. Force fields representing other constraints like joint limits and net effort to be applied (scaled appropriately based on their relevance to the task) are also superimposed on the earlier fields F_1 and F_2 . The time base generator takes

care of the temporal aspects of the relaxation of the system to equilibrium. In this way, superimposed force fields representing the goals and task relevant mixtures of constraints can pull a network of task relevant parts of an internal model of the body to equilibrium. **b** iCub picking up the fuse with the right hand with wrist orientation different than that it uses for pushing with the left hand; refer to the video (Online Resource 1). **c** Shows a representation of the PMP relaxation process dynamics of a body with three joints when reaching to targets in a 2D space with different wrist orientations

Table 4 shows the evaluation results over 40 different trials in a basic assembly task carried out by iCub. It has to be noted that these performance rates are equally affected by other components of the DARWIN cognitive architecture. Problems like non-detection, mis-localization, improper grasping etc. cumulatively affect the performance of the overall system. In fact, a major upgrade of the other sub-components of DARWIN cognitive system improved the overall performance to 95% on the industrial platform (see details in a subsection on industrial robots later in this article).

3.2.3 Dealing with multiple constraints including wrist orientation

Visual goals (i.e. fuse and fusebox) identified by the visual perception systems are translated into corresponding spatial goals (3D locations) and directed towards the appropriate end-effector (hand) for subsequent realization by the action. Based on the goal, and the dynamically presented set of con-

straints related to the nature of the task, for example, just grasping a fuse with a particular wrist orientation or using an end effector to push the fusebox, a custom relaxation network is built in real time as described in below.

The distributed nature of the forward inverse model allows us to easily incorporate the process of relaxation at a target position with a desired orientation of the wrist. Orienting the wrist appropriately is critical when the robot has to pick up an object (fuse) from above or push it (fusebox) sideways. It is also important when iCub tries to pick up tools placed at arbitrary orientations on the table, for further use to attain goals. Further, for many tools, the effective extension of reaching space that the robot could get using a tool is a function of the orientation with which it is grasped. The resulting computational network is shown in Fig. 6. The new addition to the scheme is the outer loop that generates a new force field F_2 that defines an attractor applied to the wrist. In addition, in many cases, joints need to be constrained not to exceed some pre-defined limits, for example in case of pushing, the upper

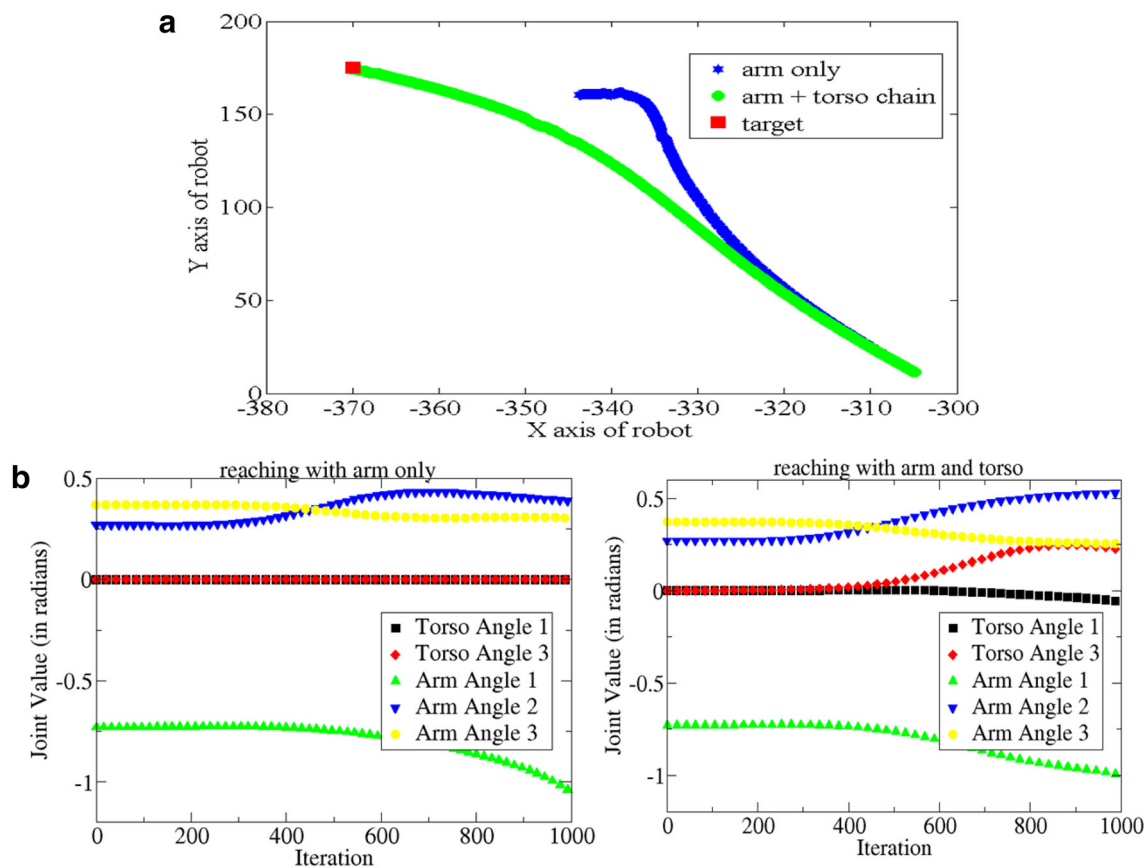


Fig. 7 **a** Trajectories of the robot in x-y plane while attempting to reach a target ($x = -370$ mm; $y = 175$ mm; $z = 23$ mm) using different kinematic chains after starting from same initial position ($x = -303$ mm; $y = 9$ mm; $z = 23$ mm); **b** shows some of the joint trajectories involved

arm should not touch torso (to avoid self-damage). Hence, in this case there are three weighted, superimposed force fields that shape the spatio-temporal behavior of the system: (1) To the end-effector (to reach the target); (2) To the wrist (for orientation); and (3) A force field in joint space as internal constraints of Joint limits.

3.2.4 Dynamic recruitment of additional motor spaces

PMP allows the flexibility to engage different parts of the body schema run-time in performing a task. Based on the requirements of the task, different degrees of freedom can be recruited to accomplish goals. For example, when a goal is not reachable by stretching the arm alone towards the goal, the DoFs of the waist can be mobilized to help attain the desired posture for reaching the goal. Figure 7a shows the system being asked to reach a target using only the arm in one case (blue trajectory) and the torso-arm chain in the other case (green trajectory). The target location (red square in the figure) is such that a fully stretched arm cannot reach if

in two cases. The contribution of torso joints makes the goal reachable; trajectories of torso joint angles can be seen changing in the right panel (in the left panel they are zero throughout)

there is no torso movement. As the figure shows, the system relaxes to the desired solution only when the joints of the waist are coupled in the kinematic chain. Figure 7b shows some of the joint trajectories involved in two cases. The left panel shows joint trajectories when the waist is not recruited in the relaxation process; the waist joint values are static at zero radians and the target cannot be achieved by extending the arm alone. Only after the torso joints contribute to the relaxation process, the goal is reached; trajectories of torso joint angles can be seen changing in the right panel.

3.2.5 Perception–action–proprioception loop for higher accuracy in an assembly task

Challenges in performing such an industrial assembly task also arise when the visual perception is not very accurate, e.g., because of minor errors in camera calibration; and the objects to deal with are small in size. Given that both the objects (fuse and a hole in fusebox) are of very small size, an insertion task like ours requires a reaching accuracy to about 0.5 cm.

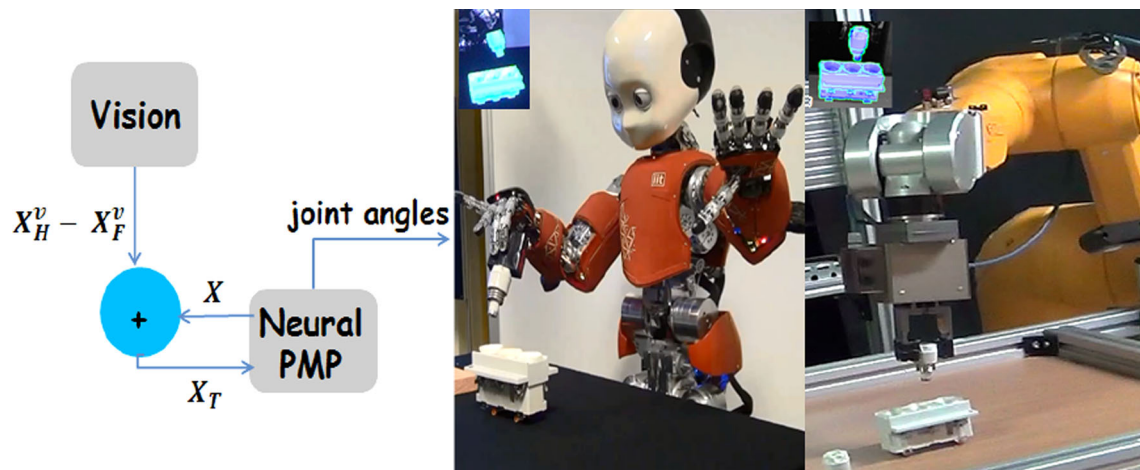


Fig. 8 The loop of integration between vision, motion and proprioception. Using information from vision the distance between the location of hole X_H^v and the location of fuse X_F^v is added to the current location of the end effector (from the forward model of neural PMP) to pro-

vide a new goal to the end effector. The alignment continues till vision estimated distance between the fuse and the hole is greater than 5 mm. Refer to Supplementary Videos (Online Resources 1 and 2) to see the robots performing alignment

Reaching directly to the hole, using visual information only, is not successful as there is some inaccuracy in estimating the location of the object. In order to overcome this limitation, we exploit the fact that the error by the visual perception in the localization of both the fuse and the hole of the fusebox is relatively same. Hence from inaccurate object locations, we can still estimate the distance between the fuse and the hole with fair accuracy. Using this distance as a measure of a virtual force field that pulls the end effector (hand of the iCub) such that the distance between the two objects (fuse and hole) is minimized, the goal of aligning the fuse on top of the hole is realized. Each time the current location of the end effector is provided by the forward model of the neural PMP. If X_F^v , X_H^v denote the perceptually estimated locations of the fuse and the hole respectively, then Eq. (1) calculating the attractive force field for PMP dynamics can be rewritten as:

$$F = K_{ext} (X_H^v - X_F^v)$$

Figure 8 shows how the information from vision is used in a loop to drive the motion of the robots using the proprioception from neural PMP. The loop continues until $X_H^v - X_F^v < 5$ mm.

3.2.6 Porting the neural PMP framework to coordinate for other embodiments

The neural PMP framework can easily be ported to any other embodiment with the only change required being the substitution of the neural network trained for mapping between the intrinsic and extrinsic spaces. For a new robot, a new neural

network can be trained, which learns the mapping between its joint space and the end effector space, and then imported into the PMP framework. We implemented the same framework on two industrial robots (Stäubli RX130B and TX90L each with 6 degrees of freedom) by training a new feedforward neural network for each. The trainings converged much faster than that of iCub given the fewer DoFs. The average mean square error of the TX controller in the reaching task was also much lower (<0.05 cm) than in the case of iCub. The same assembly task scenario was carried out using the two robots (see Fig. 9); the only difference here was two robots were used as two arms that work together to perform the assembly.

Here also, the proposed PMP architecture was implemented for action generation and control of the two robots as part of the DARWIN cognitive system. In this case, the DARWIN cognitive system was extensively evaluated/benchmarked against a state of the art industrial system performing different versions of the afore-mentioned assembly task. The overall performance of the presented architecture was comparable in all aspects against the industrial benchmarking system. A complete evaluation and detailed specifications of both systems can be found in a deliverable (D9.5) of the EU FP7 project DARWIN (DARWIN D9.5 2015). Table 5 shows (a) the evaluation results of an industrial robot (TX) in a reaching task and (b) success rates in a basic assembly task.

3.2.7 Obstacle avoidance using continuous reaching

For a robot operating in unstructured environments, obstacles do arise as the robot moves its body to reach out to target

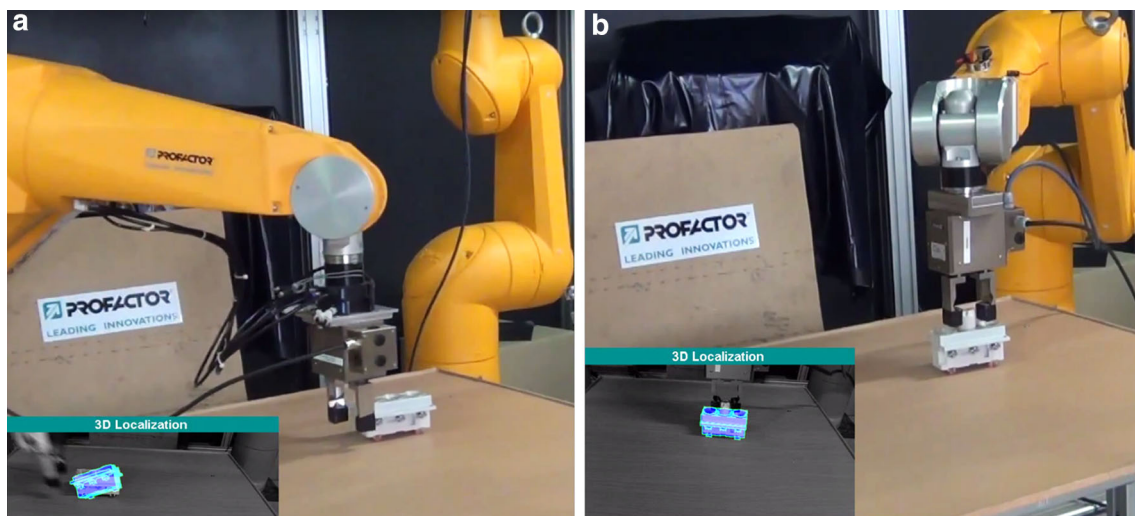


Fig. 9 The two robots performing the assembly task together. The robot on the left (a) pushes the fusebox and the robot on the right (b) completes the insertion. Refer to Supplementary Videos (Online Resources 2 and 3)

Table 5 Statistical results (a) of industrial robot TX in simple reaching task in a simulated environment; (b) performance of Industrial robots in an industrial assembly task

Statistical parameter	Value	
a		
Root mean square error	0.0377	
Standard deviation	0.0369	
Mean square error[x , y , z]	[0.0512, 0.0381, 0.0142]	
Standard deviation [x , y , z]	[0.0494, 0.0381, 0.0140]	
Pick success rate	Insertion success rate	Overall task success rate
b		
81 %	98 %	95 %

goals. Hence an obstacle avoidance mechanism is needed to be incorporated in the motion planning framework so as to complete a task in a safe way. In the assembly task, since objects are scattered in the scene at random locations, some of them can obstruct the path of the robotic arm while it tries to reach the others. In order to complete the assembly task successfully in presence of multiple objects (fuses and fuseboxes), the neural PMP coordinating the robot is augmented with a *virtual trajectory generation system* that dynamically generates a virtual path for the motion of the end effector based on the notion of *shape* formation (see Mohan et al. 2011, for details). By taking into account the geometric information related to possible obstacles to be avoided and the target goal (provided by vision), a spatio-temporal virtual trajectory of a desired *shape*, that avoids all obstacles, is synthesized for the robot to follow. The synthesized virtual trajectory acts as a *moving point attractor* that the end effector of the robot has to track continuously. This in turn, results in generation of necessary motor commands through PMP simulation. In this sense, while the PMP deploys a parallel composition of force fields, the virtual trajectory gen-

eration system deploys a serial composition of equilibrium points that *animates* the internal body model. The framework operates real time, takes into account multiple obstacles and does not get stuck in local minima. Figure 10 shows cases of goal directed reaching in an industrial robot where the robot successfully performs the assembly task by moving in a *bump* shaped trajectory in order to avoid obstacles in the environment. The concept of different kinds of *shapes*, their composition, generation and application using the *virtual trajectory generation system* is discussed in Mohan et al. (2011).

3.2.8 Exploiting the power of imagined actions

Considering that real and imagined actions turn out to be similar indeed, the proposition that even overt actions are a product of an *internal simulation* is a defining feature of PMP architecture. Since PMP body schema model relies on simulating the action on the body schema before execution in real, it can act as a forward model to predict the consequence of an action. Such mental simulations can be exploited by

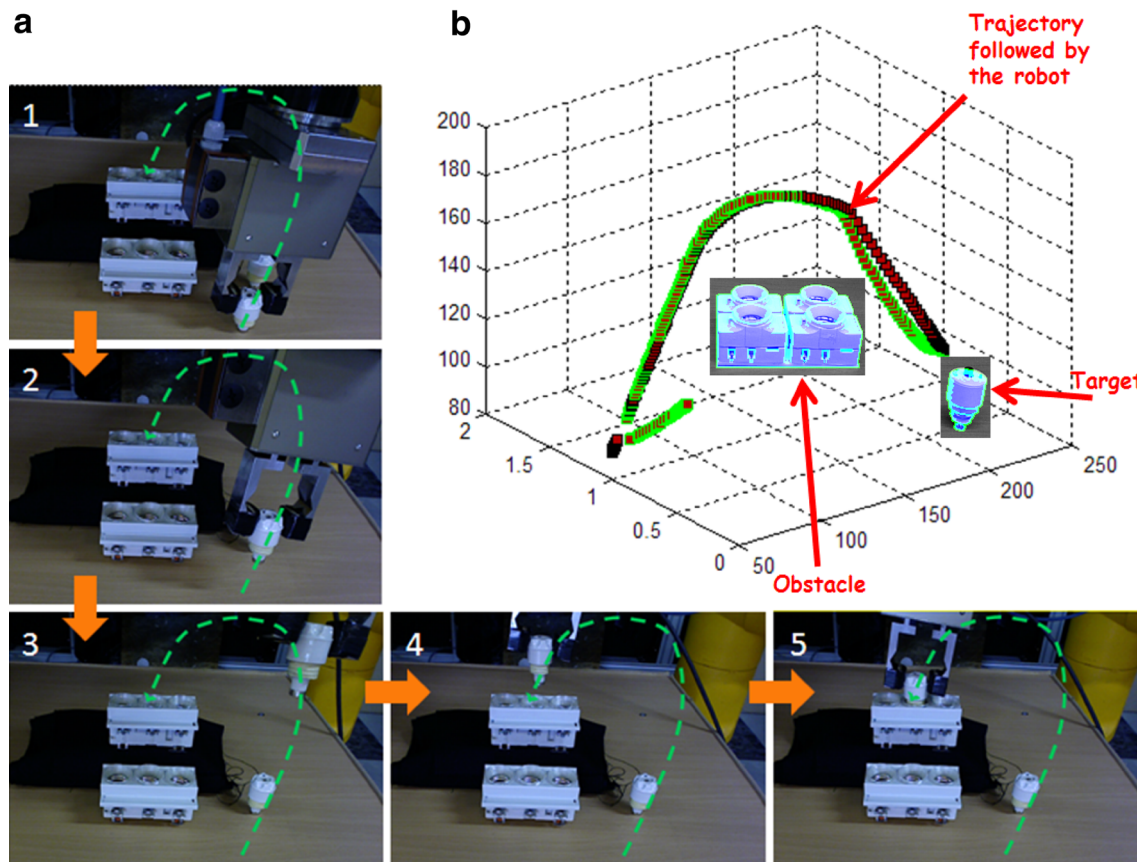


Fig. 10 **a** (1–5) The robot end effector/gripper following a *bump* shaped trajectory to avoid obstacles when assembling the fusebox. **b** In a different setup, the virtual trajectory (in red) that the robot end

effector has to follow when reaching a target and avoiding collision with the other and the actual trajectory of the robot end effector (in green) (Color figure online)

a higher level reasoning system to validate the consistency of the motor plan before the actual delivery of the generated motor commands to the controller and to re-plan the sequence of necessary actions if needed. We exploited this feature of PMP in two cases:

- (a) *Dealing with unreachability*: When targeting to reach for an object (fusebox), the PMP dynamics on the body schema structure can infer using the forward model, if the object is in an arm's peripersonal space and hence reachable or not. If it happens to be not-reachable to that arm, the higher level reasoning system can use this inference from PMP to plan a different course of actions to complete the task. In our case, the reasoning system planned to employ another body schema chain i.e., the other arm to push the object (fusebox) into the desired reachable workspace (see Figs. 6b, 9), following which goal of insertion was realized. The higher level reasoning could also use the residual *error* or measure of inconsistency in reaching, as an inference to trigger the possible conjunction of an appropriate tool into the body schema network, to reach the goal.

- (b) *Avoiding collisions*: In robotic platforms with robots/arms operating in parallel while working in a shared workspace, collision prediction/avoidance mechanisms are compulsory for safe and robust functionality. In a similar setup (see Fig. 11) when two robots have to reach two targets/objects in parallel, the Neural PMP forward/Inverse model when given a spatial configuration of objects in the scene, simulates reachability and anticipated motion trajectories of the robots during the task. The anticipated trajectories are then exploited by a higher level reasoning to detect possible collisions and re-plan the course of action if a possible collision using the two motion trajectories is anticipated.

3.2.9 Open source release of source code and documentation

The latest version of the software implementation (i.e. source code in C++) of the neural PMP architecture is released under GPL license and is available in iCub's open-source repository (robotcub) which is hosted on www.sourceforge.net. Both source code and user manual for installation

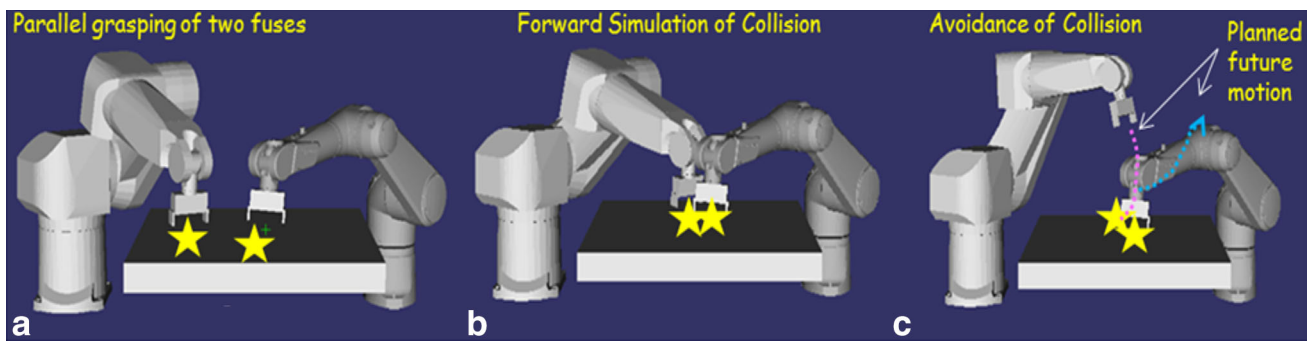


Fig. 11 Some simulated results of parallel operation of two industrial robots in a shared workspace **a** without collision, **b** simulated collision (i.e. actions that are anticipated but not executed) and **c** planned future

motion. Refer to Supplementary Video (Online Resource 3) to see the robots working in parallel without any collisions

and customized use of the architecture on different robotic platforms is available at <https://svn.code.sf.net/p/robotcub/code/trunk/iCub/contrib/src/morphoGen/>. The source code of modules for iCub (named *PMP*) and industrial robots (named *PMPRX_Neural*) are inside the subdirectories of *morphoGen/src/control* folder, while *morphoGen/app* contains all the configuration files for loading the neural networks and codes for data-generation and training of the neural networks (see folders *morphoGen/app/perception ActionCycleApp/* and *morphoGen/app/RXneural* for iCub and industrial robots respectively). The implementations use YARP platform (<http://wiki.icub.org/yarpdoc/>) for communication of messages. The User manual can be found inside *morphoGen/documentation* folder. It gives a detailed description on how to run the modules and how to use the available code for training new neural networks. In addition, videos of the robots performing assembly tasks using neural *PMP* are provided as supplementary material.

4 Discussion

This paper proposed a neural framework for action generation and control of humanoid and industrial robots employing the internal representation of the body, i.e., the *body schema*, as the controller. The body schema consists of distributed representations of the different kinematic segments of the body and the tools all networked together in serial, parallel or hybrid geometries. This body schema is learnt in the form of different neural networks as exemplified in the article. It is plastic and can be extended as well as configured as per the requirements of a given task. For action generation, the body schema is subjected to multiple virtual force fields to reach different goals and fulfil multiple constraints; these forces induce a passive motion in the body schema to relax it to an equilibrium configuration in finite time reaching the goal.

The overall framework is robust and operates by means of incremental, well-posed direct computations while avoiding

any model inversions. There are no predefined cost functions/ optimization constraints like minimum torque, minimum jerk etc. common to most control models in literature; hence there is flexibility to operate online facilitating run-time co-evolution of plans and the corresponding control processes needed to realize them. The framework allows task-specific imposition of force fields onto the model dynamics to maneuver the body movement in a task-dependent way. Different body segments and tools can be linked together into custom networks systematically as per the task at hand. Moreover, different degrees of freedom of each kinematic chain can be recruited as per their variable compliances.

In comparison to other force field based methods in control, our approach has two main advantages:

- (a) Typical force field based methods (Featherstone and Khatib 1997; Khatib 1987) to motor control are designed to apply to systems which are characterized by a single geometric relationship/Jacobian matrix and hence are not flexible to run-time recruitment of different kinematic chains as per task requirements. In contrast, the networked distributed representation of different kinematic chains as a body schema facilitates their use on demand.
- (b) The *PMP* body schema model not only allows generating unique solutions, but a class of solutions by the use of impedances, i.e. stiffness and admittances, as tunable parameters. This is different from the approach of using an inertia-weighted pseudo-inverse (Khatib 1987) because generally the controller cannot tune directly the inertial properties of the manipulator. Moreover, after several experimental tests on previously proposed force field operational controllers, Nakanishi et al. (2008) reported that these approaches degrade in the face of modeling errors and did not score well in their experimental evaluations. They attribute this degradation to the inaccuracies of the estimated inertia matrix, as this matrix and its inverse are used at many places of the control law. As an aside, calculation of the inertia-weighted pseudo-inverse

in these control models is computationally expensive (Featherstone 1987) whereas PMP body schema computations are local, well-posed and hence, cheap.

Another interesting feature of the proposed body schema controller is that it is inherently a forward-inverse model that can generate the feasible motor commands to transform the current position of the body into the desired one (inverse model) as well as predict the future state (forward model) of the body from the motor commands generated by inverse model. The body schema plays the role of a shared representation on which both forward and inverse models are realized during the cyclic exchange of energy between different motor spaces inside the PMP networks. Architecture of forward-inverse models and their role in the cognitive architectures is under recent debate (Pickering and Clark 2014).

4.1 Real and imagined actions: the butterfly and pupa

The article presented a neural framework to the PMP mechanism through the incorporation of an extendable body schema discussing the results from motor control perspective. However, the neural controller described here can not only generate but also simulate action. This is very relevant in the context of studies on motor cognition where questions regarding the representations of actions in brain, how these are formed and handled, are still open. How does the brain perceive and understand not only the intended actions that will eventually be executed, but also imagined actions; how does it learn by observation, or even understand the behavior and intentions of other people's actions? How is the brain able not only to shape the motor system in anticipation to execution, but also provide the self with information on the feasibility and the meaning of potential actions? In this quest, mounting research using advanced techniques like brain imaging for studying motor imagery (Frey and Gerry 2006; Grafton 2009; Kranczioch et al. 2009), and embodied cognition (Gallese and Lakoff 2005; Gallese and Sinigaglia 2011; Pickering and Clark 2014; Sevdalis and Keller 2011) confirm that distributed, multi-centered neural activity is consistently detected during different conditions like imagination of movement, observation/imitation of other's actions, and comprehension of language. These studies aimed at direct access to the mental states in the absence of overt movements, make clear that actions involve a covert stage. Covert and overt stages thus represent a continuum like two life stages of a butterfly, the pupa and the adult; such that every overtly executed action (the adult fly) implies the existence of a covert stage (the pupa), whereas a covert action may not necessarily turn out into an overt action. In our view, the *link or the middleware* between the two forms/stages is the body schema mechanism. Running internal simulations on an interconnected set of neuronal net-

works must be the main function of the body schema. Hence we believe our model provides a computational basis to validate the results from neuroscience in this direction. In this context, we are currently investigating the incorporation of the model in cognitive architectures for action simulation in different social contexts like observation, imitation and inference about the actions of others. Other future developments to the work include experiments with learning and use of a range of tools to address the current needs in social and industrial robotics. Problems of task-specific tool-use by social robots for household activities like cooking, cleaning etc., and industrial robots in manufacturing and assembly like, cutting, drilling, painting are planned to be addressed as future extensions to this work.

Acknowledgments This work presented in this article is supported by Robotics, Brain and Cognitive Sciences Department IIT, the EU FP7 Project DARWIN (www.darwin-project.eu, Grant No. FP7-270138) and US Dept. of Defense Grant (W911QY-12-C0078).

Appendix: A neural implementation of time base generator

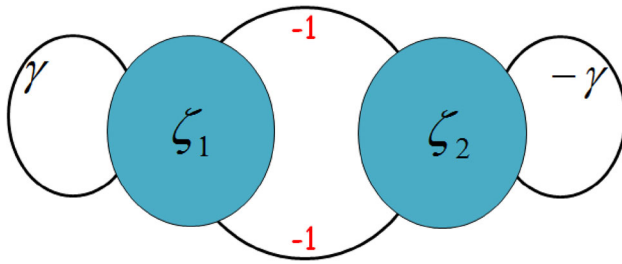
A time base generator (TBG) is a scalar dynamical system in the normalized variable ξ given by:

$$\begin{aligned}\dot{\xi} &= \gamma(\xi(1-\xi))^\beta \\ \beta &\in (0, 1),\end{aligned}\tag{5}$$

where $\xi(t)$ is a smooth sigmoid from $\xi(0) = 0$ to $\xi(t_f) = 1$, with a bell shaped velocity profile and desired finite movement duration t_f . The system has two equilibrium points, an unstable one at $\xi = 0$ and a stable one at $\xi = 1$, consequently the system always approaches stably to $\xi = 1$. The time history of the TBG can be regulated using β . The γ parameter has a dual function: controlling the convergence time and to reset the TBG and make it excitable for subsequent activation cycles. As regards to the exponent β , it can be shown that the condition,² $\beta > 2/3$ is essential in order for the third derivative of $\xi(t)$ (Jerk) to be defined at $t = 0$ and $t = t_f$. Under these conditions, it can be seen that the dynamics of the system are Non Lipschitzian,³ i.e. equilibrium configurations do not satisfy Lipschitz condition for ODE since $|\partial\dot{\xi}/\partial\xi| \rightarrow \infty$.

² Condition to have a bounded acceleration, $\partial^2\xi/\partial t^2 = -\beta\gamma^2(\xi(1-\xi))^{2\beta-1}(1-2\xi)$, at equilibrium point is, $0.5 < \beta < 1$. The Jerk of $\xi(t)$, $\partial^3\xi/\partial t^3 = \beta\gamma^3(\xi(1-\xi))^{3\beta-2}\{(2\beta-1)(1-2\xi)^2 - 2\xi(1-\xi)\}$ imposes an additional restriction of having $0.66 < \beta < 1$ for bounded jerk.

³ Non Lipschitzian systems have point attractors of infinite stability in the sense that the gradient of their Lyapunov function diverges at equilibrium point, a consequence is that they reach equilibrium in finite time (it is a terminal attractor). $\partial\dot{\xi}/\partial\xi = \beta\gamma(\xi(1-\xi))^{\beta-1}(1-2\xi)$, as $\beta < 1$, the expression tends to ∞ , at equilibrium points.



Neural Network of Time Base Generator

Fig. 12 Reciprocal inhibition neural network for TBG

This implies that equilibrium point is a terminal attractor, and systems with terminal attractor dynamics always converge in finite time (Zak 1991).

To derive the convergence time, let us consider a simpler dynamical system:

$$\dot{\xi} = \gamma \xi^\beta.$$

$$t_f = \int_0^{t_f} dt = \int_0^1 \partial \xi / \gamma \xi^\beta = 1/\gamma(1 - \beta)$$

Once again we can see that equilibrium point is a terminal attractor as convergence time is always finite and can be precisely specified through the constant $\gamma = 1/t_f(1 - \beta)$.

Remarkably, the above dynamical system can be approximated using a reciprocal inhibition network consisting of two neurons. A single neural element is an integrate-and-fire neuron comprised of a multiplier, an integrator and a power function. In the integrate-and-fire model, input spikes are multiplied by their respective synaptic weights, summed and integrated over time. If the integral exceeds a threshold, the neuron fires and the integration restarts. The functionality in this case can be expressed as:

$$\dot{\xi}_i = \prod w_i \xi_i$$

where

$$\zeta_i = \xi_i^\beta$$

The reciprocal inhibition network of two neurons modeling the TBG is shown in Fig. 12.

Dynamic behavior of the neuron can be written as

$$\dot{\xi}_1 = -\gamma \xi_1^\beta \xi_2^\beta = -\dot{\xi}_2$$

$$\xi_1(t) + \xi_2(t) = 1$$

$$\therefore \dot{\xi}_2 = \gamma \xi_1^\beta \xi_2^\beta = \gamma \xi_2^\beta (1 - \xi_2)^\beta = \gamma (\xi_2(1 - \xi_2))^\beta$$

This is same as Eq. 5.

To perform any reaching movement, several joints—shoulder, elbow, wrist, fingers move cooperatively forming

a synergy in a flexible and dynamic fashion. While groups of fingers may operate synergistically while playing a guitar chord, individual fingers are controlled while playing a lead. One of the basic problems of motor control is to understand how neural control structures quickly and flexibly organize and engage different parts of the body schema to cooperate synergistically in a movement sequence. The above TBG can be used to dynamically couple and decouple synergies in different ways based on task specification. In sum, by selecting two parameters of the TBG (t_f and β), a family of time-varying signals can be generated. From the point of view of real-time implementation, it is possible to use any scalar function of time satisfying the properties of described above or a look-up table etc.

References

- Arimoto, S., et al. (2005). Natural resolution of ill-posedness of inverse kinematics for redundant robots: A challenge to Bernstein's degrees-of-freedom problem. *Advanced Robotics*, 19(4), 401–434.
- Asatryan, D. G., & Feldman, A. G. (1965). Functional tuning of the nervous system with control of movements or maintenance of a steady posture. *Biophysics*, 10, 925–935.
- Baillieul, J., & Martin, D. P. (1990). Resolution of kinematic redundancy. *Proceedings of Symposia in Applied Mathematics*, 41, 49–89.
- Balestrino, A., De Maria, G., & Sciavicco, L. (1984). Robust control of robotic manipulators. In *Proceedings of the 9th IFAC world congress* (Vol. 5, pp. 2435–2440).
- Bekey, G., & Goldberg, K. Y. (Eds.). (2012). *Neural networks in robotics* (Vol. 202). Berlin: Springer.
- Bernstein, N. (1935). The problem of the interrelationships between coordination and localization. Retrieved November 13th, 2015 from <http://www.cns.nyu.edu/~bijan/courses/sm10/Readings/Glimcher/Problem%20of%20the%20Interrelation%20of%20Coord%20and%20Local%20-%20PGArt.pdf>.
- Bernstein, N. (1967). *The coordination and regulation of movements*. Oxford: Pergamon Press.
- Bhat, A. A., & Mohan, V. (2015). How iCub learns to imitate use of a tool quickly by recycling the past knowledge learnt during drawing. In *Biomimetic and biohybrid systems* (pp. 339–347). Berlin: Springer.
- Bizzi, E., & Polit, A. (1978). Processes controlling arm movements in monkeys. *Science*, 201, 1235–1237.
- Bryson, E. (1999). *Dynamic optimization*. Menlo Park, CA: Addison Wesley Longman.
- Buss, S. R., & Kim, J.-S. (2005). Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3), 37–49.
- Cai, H., Werner, T., & Matas, J. (2013). Fast detection of multiple textureless 3-D objects. In *Computer vision systems* (pp. 103–112). Berlin: Springer.
- DARWIN D9.4. (2014). Deliverable D9.4: Third year demonstrators and evaluation report. EC FP7 project DARWIN Grant No. 270138. Retrieved November 10th, 2015 from http://darwin-project.eu/wp-content/uploads/2010/07/D94_Y3_Demonstrators_Evaluation_v3.0.pdf.
- DARWIN D9.5. (2015). Deliverable D9.5: Industrial assembly demonstrator and final evaluation. EC FP7 project DARWIN Grant No. 270138. Retrieved November 10th, 2015 from http://darwin-project.eu/wp-content/uploads/2010/07/D95_Y4_Demonstrators_Evaluation.pdf.

- De Luca, A., & Oriolo, G. (1991). Issues in acceleration resolution of robot redundancy. In *Third IFAC symposium on robot control* (pp. 93–98).
- De Luca, A., Oriolo, G., & Siciliano, B. (1992). Robot redundancy resolution at the acceleration level. *Laboratory Robotics and Automation*, 4, 97–106.
- Featherstone, R. (1987). *Robot Dynamics Algorithms*. Dordrecht: Kluwer.
- Featherstone, R., & Khatib, O. (1997). Load independence of the dynamically consistent inverse of the Jacobian matrix. *International Journal of Robotics Research*, 16(2), 168–170.
- Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5, 1688–1703.
- Frey, S. H., & Gerry, V. E. (2006). Modulation of neural activity during observational learning of actions and their sequential orders. *Journal of Neuroscience*, 26, 13194–13201.
- Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11, 127–138.
- Friston, K. (2011). What is optimal about motor control? *Neuron*, 72(3), 488–498.
- Gallese, V., & Lakoff, G. (2005). The brain's concepts: The role of the sensory-motor system in reason and language. *Cognitive Neuropsychology*, 22(3), 455–479.
- Gallese, V., & Sinigaglia, C. (2011). What is so special about Embodied Simulation. *Trends in Cognitive Sciences*, 15(11), 512–519.
- Grafton, S. T. (2009). Embodied cognition and the simulation of action to understand others. *Annals of the New York Academy of Sciences*, 1156, 97–117.
- Graziano, M. S. A., & Botvinick, M. M. (2002). How the brain represents the body: Insights from neurophysiology and psychology. In W. Prinz & B. Hommel (Eds.), *Common mechanisms in perception and action: Attention and performance* (pp. 136–157). Oxford: Oxford University Press.
- Guigon, E. (2011). Models and architectures for motor control: Simple or complex? In F. Danion & M. L. Latash (Eds.), *Motor control* (pp. 478–502). Oxford: Oxford University Press.
- Haggard, P., & Wolpert, D. M. (2005). Disorders of body schema. In H. J. Freund, M. Jeannerod, M. Hallett, & R. Leiguarda (Eds.), *Higher-order motor disorders: From neuroanatomy and neurobiology to clinical neurology* (pp. 261–271). Oxford: Oxford University Press.
- Head, H., & Holmes, G. (1911). Sensory disturbances in cerebral lesions. *Brain*, 34, 102–254.
- Hollerbach, J. M., & Suh, K. C. (1987). Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation*, 3(4), 308–316.
- Hsu, P., Hauser, J., & Sastry, S. (1989). Dynamic control of redundant manipulators. *Journal of Robotic Systems*, 6(2), 133–148.
- Iriki, A., Tanaka, M., & Iwamura, Y. (1996). Coding of modified body schema during tool use by macaque postcentral neurones. *Neuroreport*, 7, 2325–2330.
- Jordan, M. I. (1990). Motor learning and the degrees of freedom problem. In M. Jeannerod (Ed.), *Attention and performance XIII*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc.
- Jordan, M. I., & Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3), 307–354.
- Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1), 43–53.
- Khatib, O., et al. (2004). Human-centered robotics and interactive haptic simulation. *International Journal of Robotics Research*, 23(2), 167–478.
- Kranczioch, C., Mathews, S., Dean, J. A., & Sterr, A. (2009). On the equivalence of executed and imagined movements. *Human Brain Mapping*, 30, 3275–3286.
- Lashley, K. S. (1933). Integrative function of the cerebral cortex. *Physiological Reviews*, 13(1), 1–42.
- Lee, S., & Kil, R. M. (1990, June). Robot kinematic control based on bidirectional mapping neural network. In *1990 IJCNN international joint conference on neural networks, 1990* (pp. 327–335). New York: IEEE.
- Lewis, F. W., Jagannathan, S., & Yesildirak, A. (1998). *Neural network control of robot manipulators and non-linear systems*. Boca Raton: CRC Press.
- Li, S., Chen, S., Liu, B., Li, Y., & Liang, Y. (2012). Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks. *Neurocomputing*, 91, 1–10.
- Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, 7(12), 868–871.
- Lourakis, M., & Zabulis, X. (2013). Model-based pose estimation for rigid objects. In *Computer vision systems* (pp. 83–92). Berlin: Springer.
- Maravita, A., & Iriki, A. (2004). Tools for the body (schema). *Trends in Cognitive Science*, 8, 79–86.
- Mel, B. W. (1988). MURPHY: A robot that learns by doing. In *Neural information processing systems* (pp. 544–553).
- Mohan, V., & Morasso, P. (2011). Passive motion paradigm: An alternative to optimal control. *Frontiers in Neurobotics*, 5, 4.
- Mohan, V., Morasso, P., Metta, G., & Sandini, G. (2009). A biomimetic, force-field based computational model for motion planning and bimanual coordination in humanoid robots. *Autonomous Robots*, 27, 291–301.
- Mohan, V., Morasso, P., Zenzeri, J., Metta, G., Chakravarthy, V. S., & Sandini, G. (2011). Teaching a humanoid robot to draw ‘Shapes’. *Autonomous Robots*, 31(1), 21–53.
- Mussa-Ivaldi, F. A., Morasso, P., & Zaccaria, R. (1988). Kinematic networks. A distributed model for representing and regularizing motor redundancy. *Biological Cybernetics*, 60, 1–16.
- Nakamura, Y., & Hanafusa, H. (1986). Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measurement, and Control*, 108, 163–171.
- Nakamura, Y., & Hanafusa, H. (1987). Optimal redundancy control of robot manipulators. *International Journal of Robotics Research*, 6(1), 32–42.
- Nakanishi, J., Cory, R., Mistry, M., Peters, J., & Schaal, S. (2008). Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6), 737–757.
- Nguyen, L., Patel, R. V., & Khorasani, K. (1990, June). Neural network architectures for the forward kinematics problem in robotics. In *1990 IJCNN international joint conference on neural networks* (pp. 393–399). New York: IEEE.
- Peters, J., & Schaal, S. (2008). Learning to control in operational space. *The International Journal of Robotics Research*, 27(2), 197–212.
- Pickering, M. J., & Clark, A. (2014). Getting ahead: Forward models and their role in cognitive architecture. *Trends in Cognitive Sciences*, 18(9), 451–456.
- Salaün, C., Padois, V., & Sigaud, O. (2009, October). Control of redundant robots using learned models: An operational space control approach. In *IROS 2009 IEEE/RSJ international conference on intelligent robots and systems, 2009* (pp. 878–885). New York: IEEE.
- Scott, S. (2004). Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5, 534–546.
- Senda, K. (1999). Quasioptimal control of space redundant manipulators. *AIAA Guidance, Navigation, and Control Conference*, 3, 1877–1885.
- Sentis, L., & Khatib, O. (2005). Synthesis of wholebody behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4), 505–518.

- Sevdalis, V., & Keller, P. E. (2011). Captured by motion: Dance, action understanding, and social cognition. *Brain & Cognition*, 77, 231–236.
- Todorov, E. (2006). Optimal control theory. In K. Doya, et al. (Eds.), *Bayesian brain: Probabilistic approaches to neural coding* (pp. 269–298). Cambridge, MA: MIT Press.
- Umiltà, M. A., Escola, L., Intskirveli, I., Grammont, F., Rochat, M., Caruana, F., et al. (2008). When pliers become fingers in the monkey motor system. *Proceedings of the National Academy of Sciences of the United States of America*, 105(6), 2209–13.
- Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least squares methods. *IEEE Transaction on Systems, Man, and Cybernetics*, 16, 93–101.
- Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man Machine Systems*, 10(2), 47–53.
- Wolovich, W. A., & Elliot, H. (1984). A computational technique for inverse kinematics. In *Proceedings of the 23rd IEEE conference on decision and control* (pp. 1359–1363).
- Zak, M. (1991). Terminal chaos for information processing in neurodynamics. *Biological Cybernetics*, 64, 343–351.



Ajaz Ahmad Bhat is a Fellow-Ph.D. at the Robotics, Brain and Cognitive Sciences Department of the Italian Institute of Technology (IIT), Italy. He holds a Masters in Computer Applications and is currently working for his doctoral thesis on endowing robots with human-like memories. He has worked extensively in EU project DARWIN endeavoring for development of robots with embodied intelligence.



Sharath Chandra Akkaladevi is a researcher in the Robotics and Assistive Systems department at Profactor GmbH. He completed his Master studies in Information Technology with distinction at Alpen-Adria University, Austria. He has been actively involved and responsible for research activities related to EU funded project DARWIN. He is currently pursuing his Ph.D with focus on Human Robot Cooperation at Alpen-Adria University.



Vishwanathan Mohan is a senior post-doctoral researcher at the Robotics, Brain and Cognitive Sciences Department of the Italian Institute of Technology (IIT), Italy. He was originally trained in Microelectronics and VLSI design at the IIT Madras followed by research training in the field of cognitive robotics at IIT Italy, where he has been a learning/playing partner to the humanoid iCub since its inception, helping it to gradually expand its cognitive horizons and at the same time trying to understand the computational principles that endow us with similar capabilities. Most of his present research is funded through the EU's projects iTalk, DARWIN, EFRA, US Dept. of Defence.



Christian Eitzinger is a senior researcher and Head of Machine Vision department at Robotics and Assistive Systems, PRO-FACTOR GmbH, Austria. He is coordinating the EU funded project DARWIN.



Pietro Morasso is currently Senior Researcher at the Italian Institute of Technology (Department of Robotics, Brain and Cognitive Sciences), Genoa, Italy, director of the Motor Learning and Robotic Rehabilitation lab (RBCS dept), since 2010. His scientific interests include neural control of movement, motor learning, neuromotor rehabilitation, robot-therapy, computational neuroscience, biomimetic humanoid robot control. He is an author of 6 books and over 400 publications of which more than 80 are published in peer-reviewed journals.