Robotics and Autonomous Systems 62 (2014) 556–567

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching

Lorenzo Jamone[a,*], Martim Brandao[a], Lorenzo Natale[b], Kenji Hashimoto[a], Giulio Sandini[c], Atsuo Takanishi[a,d]

[a] *Faculty of Science and Engineering, Waseda University, Tokyo, Japan*
[b] *iCub Facility, Istituto Italiano di Tecnologia, Genoa, Italy*
[c] *Department of Robotics, Brain and Cognitive Sciences, Istituto Italiano di Tecnologia, Genoa, Italy*
[d] *Humanoid Robotics Institute, Waseda University, Tokyo, Japan*

HIGHLIGHTS

- Autonomous online learning of a representation of the robot workspace.
- Locations in space are encoded using gaze-centered motor coordinates.
- The robot is able to estimate the Reachability of visually detected objects.
- The robot can modify its body configuration to improve the quality of arm reaching.
- Overall, we realized a form of intelligent whole-body reaching in a humanoid robot.

ARTICLE INFO

Article history:
Received 3 April 2013
Received in revised form
10 October 2013
Accepted 28 December 2013
Available online 8 January 2014

Keywords:
Kinematic workspace
Online sensorimotor learning
Humanoid robots
Whole-body reaching
Bio-inspired robotics

ABSTRACT

We describe a learning strategy that allows a humanoid robot to autonomously build a representation of its workspace: we call this representation Reachable Space Map. Interestingly, the robot can use this map to: (i) estimate the Reachability of a visually detected object (i.e. judge whether the object can be reached for, and how well, according to some performance metric) and (ii) modify its body posture or its position with respect to the object to achieve better reaching. The robot learns this map incrementally during the execution of goal-directed reaching movements; reaching control employs kinematic models that are updated online as well. Our solution is innovative with respect to previous works in three aspects: the robot workspace is described using a gaze-centered motor representation, the map is built incrementally during the execution of goal-directed actions, learning is autonomous and online. We implement our strategy on the 48-DOFs humanoid robot Kobian and we show how the Reachable Space Map can support intelligent reaching behavior with the whole-body (i.e. head, eyes, arm, waist, legs).

1. Introduction

The ultimate goal of humanoid robots is to become effective helpers for humans, showing some form of autonomy and flexibility that allows them to operate alongside humans during everyday tasks, or even to substitute them in tedious or dangerous works. To achieve this ambitious objective, researchers have been designing more and more complex robots, having an increasing number of degrees of freedom and sensors (see [1,2] for two recent examples, among many others); these robots should be able to cope with the unstructured environment in which humans daily live and act. In particular, they should be able to reach for objects using the arms or even the whole body, and eventually to grasp and to use such objects. A fundamental issue associated with these behaviors is the definition of the space that the robot can reach for, i.e. the reachable space or workspace. In theory, the robot workspace can be computed analytically if a model of the system is available [3]. However, it might be difficult to obtain a closed form solution, especially in the case of complex and highly redundant systems like humanoid robots. In general, computing analytical models of these complex systems is a process that is never error-free; discrepancies between the computed model and the real system arise due to the assumptions that have to be made about the system to keep the model mathematically tractable (e.g. the links are completely

[*] Correspondence to: Instituto Superior Tecnico (Torre Norte), Av. Rovisco Pais 1, 1049-001, Lisbon, Portugal. Tel.: +351 920138558.
E-mail addresses: lorenzojamone@gmail.com, lorejam@liralab.it (L. Jamone), martimbrandao@gmail.com (M. Brandao), lorenzo.natale@iit.it (L. Natale), contact@takanishi.mech.waseda.ac.jp (K. Hashimoto), giulio.sandini@iit.it (G. Sandini), contact@takanishi.mech.waseda.ac.jp (A. Takanishi).

0921-8890/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.12.011

rigid, the sensory measurements are perfect and do not drift over time, there is no backlash in the transmission from the motors to the joints, the axes of rotation of the joints remain constant over time), that in most cases are not realistic. A classical example is the calibration (i.e. finding the zero) of the incremental motor encoders that are typically used to measure joints angular positions. This is a procedure that has to be done every time the robot is switched on, and that typically provides slightly different results from time to time. Therefore, a static analytical model describing, for instance, the robot kinematics is always going to be slightly wrong; online learned models, conversely, can adapt to this kind of changes in the system.

Modern machine learning techniques can be employed to equip these complex robots with the necessary autonomous calibration capabilities (see [4,5] for two recent surveys); moreover, the use of learning techniques seems to be mandatory to eventually provide robots with a human-like level of autonomy and flexibility.

Therefore, in this paper we propose a learning approach to obtain a representation of the workspace of a humanoid robot: we refer to this representation as Reachable Space Map. The map is learned autonomously by the robot during the execution of goal-directed reaching movements. After enough learning has been performed, the robot can use the map to estimate the *Reachability* of a visually detected object before starting the reaching movement. In our design, the Reachability is a continuous value ranging from 0.0 (far from being reachable) to 1.0 (reachable with an optimal arm configuration): therefore, we do not just define the space that the robot can reach for in a binary way (reachable/non-reachable), but instead we provide a Reachability information (i.e. *how well* the robot can reach for) for all the points in space that the robot can visually detect by moving its head and eyes. As a consequence, this map can be exploited to realize body movements (e.g. bending or rotating the torso, bending the knees, walking toward the object) that increase the Reachability of a visually detected object, by placing the robot in a configuration with respect to the object that improves the effectiveness of the subsequent arm reaching action. Our approach focuses on three main concepts, which make our solution innovative with respect to previous works: (i) locations in space are directly encoded using the motor variables involved in the gaze control (i.e. a gaze-centered motor representation), (ii) the sensory data used to build the map are generated by the robot in a goal-directed way and (iii) learning is performed autonomously and online. We have already introduced this solution in two recent works [6,7] in which simulation results are provided. In this paper we apply our system to a real 48-DOFs humanoid robot, and we show how the Reachable Space Map can be used to plan whole-body reaching movements that involve the control of the robot head, eyes, arm, waist and legs. We define this behavior "intelligent" because of two main reasons. First, among all the possible body configurations that allow successful reaching, the robot looks for the one that maximizes a specific performance metric. Second, the knowledge that supports this decision process is acquired by the robot autonomously and incrementally from its own motor experience.

The rest of the paper is organized as follows. In Section 2 we describe the state of the art, discussing the biological inspiration of our approach and reporting the relevant work in robotics. Then in Section 3 we present the robotic platform on which we applied the proposed system. In Sections 4 and 5 we illustrate the gaze and reaching controllers respectively, while in Section 6 we explain how the Reachable Space map can be learned incrementally during the execution of goal-directed reaching movements. In Section 7 we show how the Reachable Space Map can be used to improve the robot reaching skills, involving additional degrees of freedom in the reaching movement. Finally, experimental results are shown in Section 8, while in Section 9 we report our conclusions and we sketch the future work.

2. State of the art

We have chosen to describe the robot kinematic workspace using motor coordinates: in particular, the location of a point in space is defined by the motor positions of the head and eyes after the robot has fixated that point, or in other terms by the *gaze configuration*. This choice is inspired by converging evidence both in monkeys [8–10] and humans [11]: recent studies suggest that areas within the posterior parietal cortex (PPC) maintain updated representations of the reaching targets in different frames of reference (i.e. gaze-centered, visual, body-centered), and perform the reference frame transformations required to switch among the different representations [12]. Head motor signals modulate this neural activity as well [13]. Furthermore, behavioral studies on humans enforce the hypothesis of the presence of a gaze-centered frame of reference for the control of pointing [14–17] and reaching [18], even in the case of whole-body reaching [19].

In fact, the idea of encoding locations in space with the robot gaze configuration has been already investigated in robotics [20–22], as it simplifies both learning and controlling reaching. A motor–motor mapping from arm configuration to gaze configuration can be learned during fixations of the end-effector (i.e. the hand) in different positions, achieved by means of gaze control; then the mapping can be inverted to reach for a fixated object (i.e. retrieving the arm configuration which brings the hand to the fixation point). In [21] the mapping is learned during a training phase separated from the subsequent execution phase, while in [20,22] it is updated during action execution. This mapping itself provides information about the space surrounding the robot, what part of it can be reached and how. A similar approach has been considered also in recent works [23,24] in which the authors mention a motor representation of the space surrounding the robot, either referred as "embodied representation of space" or "visuomotor representation of the peripersonal space". However, none of these works makes this representation explicit, and they only represent the space that the robot can reach (not the space that the robot cannot reach); therefore, it is not clear how this knowledge can be used for planning body movements that facilitate the subsequent reaching action. It would be desirable instead to have a map that provides information about the Reachability of the whole visible space, so that a trajectory from non-reachable to reachable regions can be planned.

Recently, numerical methods have been applied to build an explicit representation of the reachable space of humanoid robots. In [25] an optimization-based method and the Monte Carlo method are compared: locations in space are associated with a binary information (i.e. reachable/non-reachable) which is stored in a database for later utilization. A richer description of the robot workspace is provided in [26], where reachable points in space are ranked by their *reachability*, a measure indicating the number of possible approaching directions for the arm (note that the definition of reachability in [26] is different from the one we propose in this paper); on the basis of that measure a *capability map* is built which has a directional structure and can be used to identify good approach directions for grasping objects, as shown in a subsequent work [27]. In [28] the reachability space is represented by a grid of voxels holding information about the success probability of an IK (inverse kinematics) query, and it is used to speed up a randomized IK solver.

In all these works the robot workspace is described with respect to a Cartesian frame of reference (either placed in the world or on the robot). With respect to this solution, representing locations in space using the gaze configuration has several advantages. First, we can easily represent all locations the robot can see, even if not reachable, in a compact map, which is limited by the robot joints limits. This is not possible using other representations, and

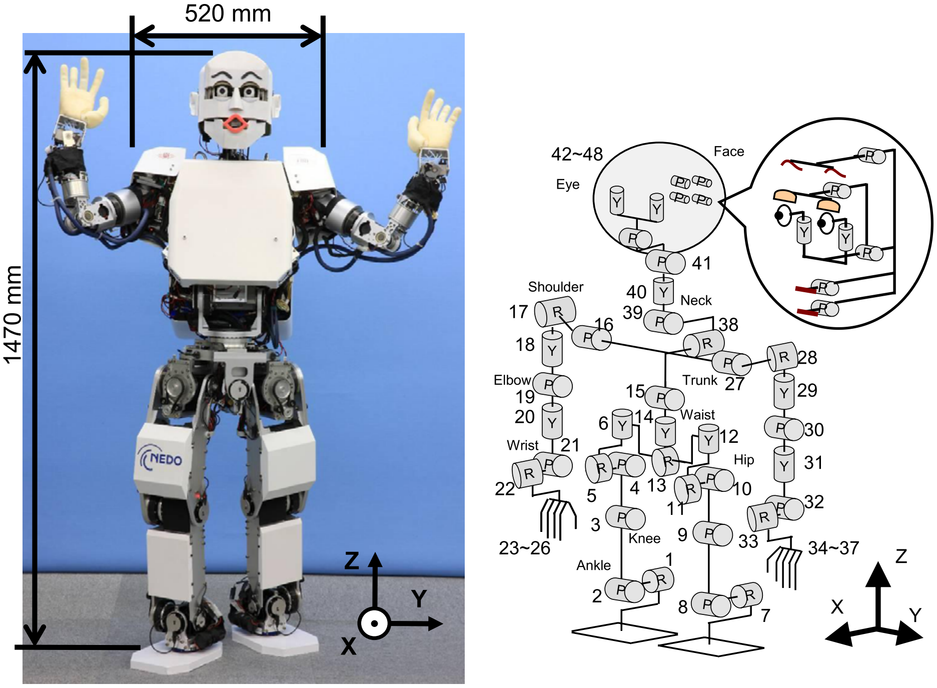


Fig. 1. Kobian humanoid robot. Left image: Kobian expressing surprise. Right image: description of the robot 48 DOFs.

in fact other works either encode only reachable locations or create arbitrary limitations around the robot (e.g. bounding box). Second, this representation is directly linked with the visual search: after a visually detected object is fixated the Reachability information can be retrieved without the need of any additional transformation. Moreover, the gaze configuration can be used to directly trigger the reaching movement, as described in the literature [20–24]. Furthermore, previous works provide a discrete representation of the space (i.e. grid of voxels). Conversely, in our approach the representation of the workspace is approximated using an LWPR (Locally Weighted Projection Regression) neural network [29], which deals with continuous input and output: this is particularly important when using the map for planning body movements. Noticeably, LWPR is an online algorithm for non-linear regression that provides a compact representation, which can be queried fast in real-time.

Another crucial improvement brought by this study with respect to previous works is that the representation of the workspace is learned incrementally (i.e. continuously and online) during the active process of reaching for objects (i.e. goal-directed exploration). Continuous online learning assures that the representation is always up-to-date even in case of changes in the robot kinematics (e.g. link lengths, deformation of materials), actuation (e.g. replacement of a motor) and sensing (e.g. replacement of the cameras). Goal-directed exploration implies that learning is performed in the regions of the space that are more relevant for the task at hand (as opposed for instance to random motor babbling in the motor space): this allows faster convergence inside those regions, as shown for example in [30].

One important feature of the proposed system is the robustness with respect to calibration inaccuracies; this is not a specific, novel, contribution of this work (the same feature is present in previous works as well), but nevertheless it is a noteworthy advantage which is made possible by the adoption of (i) motor representations and (ii) online continuous learning. In most robots the motor positions are measured through incremental encoders, therefore a calibration procedure is needed each time the robot is switched on, to determine the zero position of the encoders. Then, the transformation from motor to Cartesian space requires that the robot cameras are accurately calibrated as well. These calibration procedures lead to slightly different results each time they are executed, and therefore a fixed kinematic model cannot describe the system in an accurate way. Since here the 3D object position is encoded in the motor space (and not in the Cartesian space), the calibration of the cameras is not needed. Moreover, the models that we use (both for controlling reaching and for estimating the robot workspace) are learned online and continuously updated, and therefore they can cope with slight inaccuracies in the calibration of the motor encoders.

Among many possibilities, we have chosen to rely on the LWPR algorithm because of its excellent memory requirements and low computational complexity. Moreover, the algorithm is well known within the robotics community, as proven by many publications, and therefore it could be easier for other researchers in the field to replicate and validate our experiments. However, the system we describe in this paper is very general, and independent of the particular learning algorithm used for the Reachable Space Map estimation; more recent algorithms can be used as well, like for instance Local Gaussian Process Regression (LGPR, [31]), and they may even allow to achieve a more accurate estimation.

3. The humanoid platform

The robotic platform we use in this work is the 48-DOFs full humanoid robot Kobian, that has been designed to integrate the bipedal walking skill of Wabian [32] to the emotion expression capabilities of the human-like head robot WE-4 [33], as described in [34]. Kobian can express different emotions (e.g. happiness, sadness, fear, anger) with face and whole-body movements [1], also during locomotion [35]. The robot size is similar to that of an average Japanese woman (see left image in Fig. 1) and the overall weight is 62 kg. The degrees of freedom of the robot are distributed as follows: 12 in the two legs, 3 in the waist, 14 in the two arms, 8 in the two hands, 4 in the neck and 7 in the head (see right image in Fig. 1).

All the joints are driven by DC motors with encoders activated by electric motor drivers (Tokushu Denso Co., Ltd.); counter readings of the encoders (i.e. measure of joint angular positions) and

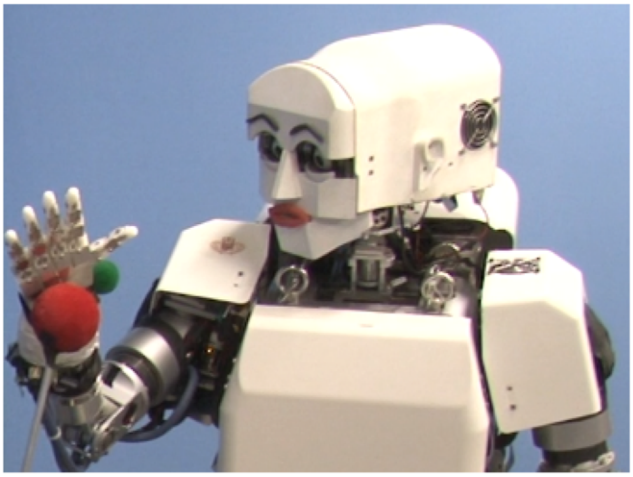


Fig. 2. The humanoid robot Kobian reaching for a red ball. A small green ball is attached to the wrist as a visual marker for the hand (i.e. the robot end-effector). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

output of the velocity references to the motor drivers (i.e. motor commands) are done by a PC (Pentium M 1.8 GHz, QNX Neutrino 6.3.0 operative system) embedded in the robot back through the I/O boards (HRP interface boards of ZUCO, Co., Ltd.). This PC is interfaced through Ethernet (or wireless) connection to an external laptop (SONY VAIO, Intel i7 processor, Windows 7 operative system) on which the higher level programs run (i.e. learning, visual processing, coordinated control). The laptop is a node of a local network of several PCs, that can be exploited to realize distributed computation when multiple behaviors and reasoning processes are executed together. This modular distributed computation is supported by the use of the YARP software framework [36]. Two CMOS color cameras (ARTRAY, ARTCAM-022MINI) are embedded in the robot eyeballs, and directly connected to the laptop through USB connection; the cameras provide images of 640 × 480 pixels at a frame rate of 30 Hz.

As we want to realize visually guided whole-body reaching, we aim at controlling the Kobian head, arm (in this case, the right arm), waist and legs. The robot end-effector is represented by a visual marker (a green ball in this case) that is attached to the wrist (as a marker for the hand); in the experiments, the target of the reaching movements is a red ball. Fig. 2 shows the robot reaching for the red ball; the green ball used as visual marker can be noticed attached to the robot wrist. The position of the centers of the two balls (both the green and the red one) in the camera images is computed through visual processing (i.e. color based segmentation using a simplified version of the technique described in [37]) with a precision of about ±2 pixels (this error is due mainly to slight changes in the illumination, as we perform the experiments in natural environment).

The joints involved in the control are the following:

- $\mathbf{q}_{gaze} = [\theta_{ny}\ \theta_{np}\ \theta_{ev}]^T \in \mathbb{R}^3$
- $\mathbf{q}_{arm} = [\theta_{sp}\ \theta_{sy}\ \theta_{sr}\ \theta_{e}]^T \in \mathbb{R}^4$
- $\mathbf{q}_{waist} = [\theta_{wy}\ \theta_{wp}]^T \in \mathbb{R}^2$
- $\mathbf{q}_{rleg} = [\theta_{rar}\ \theta_{rap}\ \theta_{rk}\ \theta_{rhp}\ \theta_{rhr}\ \theta_{rhy}]^T \in \mathbb{R}^6$
- $\mathbf{q}_{lleg} = [\theta_{lar}\ \theta_{lap}\ \theta_{lk}\ \theta_{lhp}\ \theta_{lhr}\ \theta_{lhy}]^T \in \mathbb{R}^6$

where θ_{ny} and θ_{np} are the neck yaw and pitch rotations (head rotation and elevation/depression, joints 40 and 39 in Fig. 1), θ_{ev} is the eyes vergence angle, which is the coordinated inward/outward yaw rotation of the eyes (joints 43 and 44 in Fig. 1), θ_{sp}, θ_{sy}, θ_{sr} are the shoulder pitch, yaw and roll rotations (elevation/depression, adduction/abduction and rotation of the right arm, joints 16–18 in Fig. 1), θ_e is the right elbow flexion/extension (joint 19 in Fig. 1), θ_{wy} and θ_{wp} are the waist yaw and pitch rotations (rotation and elevation/depression of the torso, joints 14 and 15 in Fig. 1), θ_{rar} and θ_{rap} are the right ankle roll and pitch rotations, θ_{rk} is the right knee rotation, θ_{rhp}, θ_{rhr} and θ_{rhy} are the right hip pitch, roll and yaw rotations (joints from 1 to 6 in Fig. 1), and $\mathbf{q}_{lleg}$ are the same joints for the left leg (joints from 7 to 12 in Fig. 1).

The corresponding limits are defined in Table 1 (note that the limits are the same for both legs, $\mathbf{q}_{rleg}$ and $\mathbf{q}_{lleg}$).

Table 1
Joint limits of the Kobian robot.

	$\mathbf{q}_{gaze}$			$\mathbf{q}_{arm}$				$\mathbf{q}_{waist}$	
min	−40°	−10°	0°	−70°	−30°	−15°	−90°	−90°	−20°
max	40°	20°	20°	−5°	−5°	15°	−5°	−90°	60°

	$\mathbf{q}_{rleg}/\mathbf{q}_{lleg}$					
min	−50°	−45°	−90°	−50°	−10°	−30°
max	50°	45°	0°	90°	25°	50°

4. Gaze control

The gaze controller allows the robot to track and to eventually fixate a 3D point in space by moving the head and eyes. We will refer to this point as the "target" (i.e. target of the gazing action) to explain how this controller works. Then, in the rest of the paper, the target can be either the center of the green ball (i.e. the marker for the hand) or the center of the red ball (i.e. the object, target of the reaching action), and we use the expressions "to gaze", "to fixate" or "to track" to indicate the activation of this controller.

If the target is visible (i.e. inside the image plane) joint velocities are generated as follows:

$$\dot{\mathbf{q}}_{gaze} = -G\mathbf{x} \tag{1}$$

where $G \in \mathbb{R}^{3\times3}$ is a positive definite gain matrix and the position of the target $\mathbf{x} \in \mathbb{R}^3$ is defined as follows:

$$\mathbf{x} = \begin{bmatrix} (u_L + u_R)/2 \\ (v_L + v_R)/2 \\ u_L - u_R \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_L \\ u_R \\ v_L \\ v_R \end{bmatrix}$$

being u_R and v_R the coordinates of the target on the right image plane and u_L and v_L the coordinates of the target on the left image plane. Indeed, the goal of the controller is to reduce to zero $[u_L\ \ u_R\ \ v_L\ \ v_R]^T$, which entails bringing the target to the center of both cameras (i.e. the fixation point). However, since $v_L = v_R$ (perceived targets have the same vertical position on both images) it is sufficient to reduce to zero $[u_L - u_R\ (u_L + u_R)/2\ (v_L + v_R)/2]^T$.

If the target is not visible a stereotyped motion strategy (i.e. random left–right and up–down movements of the neck) is used to detect it; then the gaze controller (Eq. (1)) is activated.

After fixation is achieved we encode the target position in space using the gaze configuration $\mathbf{q}_{gaze}$; since we actuate only 3 DOFs of the head+eyes system the mapping from gaze configuration to target position is unique. If more DOFs are used the redundancy should be solved by the gaze controller, as we did for instance in [38].

5. Reaching control

The approach we use for learning and controlling reaching is very similar to the one described in [39]; minor modifications have been made to assure the repeatability of the redundancy resolution over multiple reaching movements. Hereinafter we provide the details that are important for the understanding of this paper, focusing in particular on the description of the redundancy resolution technique.

The reaching movement starts after the target object has been fixated, therefore the goal is to bring the hand to the fixation point (by actuating the arm joints). Two controllers are used in combination to achieve this goal: open-loop and closed-loop. The open-loop controller is activated first, and presents two interesting features: it can be executed even if the hand is outside the visual field and it does not suffer from velocity limits imposed by visual feedback delays. However, a closed-loop controller which exploits visual feedback (i.e. visual servoing) is necessary to reduce to zero the hand positioning error; this controller is activated when the hand enters the central part of the visual field.

The open-loop controller exploits a kinematic model of the robot arm, head and eyes, namely the arm-gaze forward kinematics, $\mathbf{q}_{gaze} = f_{AG}(\mathbf{q}_{arm})$. The model is inverted to retrieve the arm joints configuration which places the hand in the fixation point. As our system is redundant ($\mathbf{q}_{gaze} \in \mathbb{R}^3$ while $\mathbf{q}_{arm} \in \mathbb{R}^4$) we can have multiple solutions (i.e. the same fixation point defined by $\mathbf{q}_{gaze}$ can be reached with different arm configurations $\mathbf{q}_{arm}$). We solve the IK (inverse kinematics) as an optimization problem by using IpOpt (Interior Point Optimizer [40]), a minimization algorithm which has been proven to be fast and reliable (see for example [41] for a precise evaluation of its performance in solving an inversion problem similar to the one discussed here). The IK computation involves two stages: first we look for a valid solution, then, if at least one exists, we look for an optimal solution according to a given criteria (thus resolving the redundancy).

The first stage is formalized as follows:

$$\mathbf{q}^S_{arm} = \mathbf{argmin}_{\mathbf{q}_{arm} \in \Omega} \left\| \mathbf{q}_{gaze} - f_{AG}(\mathbf{q}_{arm}) \right\| \tag{2}$$

where $\Omega \equiv [\mathbf{q}^{\min}_{arm}, \mathbf{q}^{\max}_{arm}]$ (i.e. arm joint limits) and $\mathbf{q}^S_{arm}$ is the IK solution. We say that $\mathbf{q}^S_{arm}$ is a valid solution if $\left\| \mathbf{q}_{gaze} - f_{AG}(\mathbf{q}^S_{arm}) \right\| \leq \boldsymbol{\epsilon}$, where ϵ is an arbitrary low error threshold (we set $\epsilon = 0.0001$). In that case, the optimal solution is found solving the following problem:

$$\mathbf{q}^S_{arm} = \mathbf{argmin}_{\mathbf{q}_{arm} \in \Omega} M(\mathbf{q}_{arm}) \tag{3}$$

$$s.t. \quad \mathbf{0} \leq \left\| \mathbf{q}_{gaze} - f_{AG}(\mathbf{q}_{arm}) \right\| \leq \boldsymbol{\epsilon} \tag{4}$$

where $M(\mathbf{q}_{arm})$ is the measure we want to minimize. We consider a solution optimal if it maximizes the distance of the arm from joints limits, as proposed in [42], and therefore we define $M(\mathbf{q}_{arm})$ as follows:

$$M(\mathbf{q}_{arm}) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\mathbf{q}_{arm}(i) - \mathbf{a}_i}{\mathbf{a}_i - \mathbf{q}^{\max}_{arm}(i)} \right)^2 \tag{5}$$

where $\mathbf{a}_i = (\mathbf{q}^{\max}_{arm}(i) + \mathbf{q}^{\min}_{arm}(i))/2$.

The closed-loop controller uses a model of the visuo-arm forward kinematics, $\mathbf{x} = f_{AV}(\mathbf{q}_{ah})$, where $\mathbf{q}_{ah} = [\mathbf{q}_{gaze} \;\; \mathbf{q}_{arm}]$. The visuo-arm Jacobian $J_{AV}(\mathbf{q}_{ah}) \in \mathbb{R}^{3\times4}$ is obtained differentiating $f_{AV}(\mathbf{q}_{ah})$ and is used for control. The redundancy is resolved through null-space projection by selecting the solution which maximizes the distance of the joints from the limits (as originally proposed in [42]). Arm motor velocities are generated as follows:

$$\dot{\mathbf{q}}_{arm} = K_m \cdot C_1 + K_s \cdot C_2 \tag{6}$$

where

$$C_1 = J^{\dagger}_{AV}(\mathbf{q}_{ah})\, \dot{\mathbf{x}}^d \tag{7}$$

$$C_2 = (I - J^{\dagger}_{AV}(\mathbf{q}_{ah}) J_{AV}(\mathbf{q}_{ah}))\, \dot{\mathbf{q}}^d_{arm} \tag{8}$$

where $J^{\dagger}_{AV}(\mathbf{q}_{ah})$ is the Jacobian Moore–Penrose generalized inverse, $(I - J^{\dagger}_{AV}(\mathbf{q}_{ah}) J_{AV}(\mathbf{q}_{ah}))$ is a null-space projector, and $\dot{\mathbf{x}}^d$ and $\dot{\mathbf{q}}^d_{arm}$ are defined as follows:

$$\dot{\mathbf{x}}^d = \mathbf{x}^d - \mathbf{x} \tag{9}$$

$$\dot{\mathbf{q}}^d_{arm} = -K_g \, \nabla M(\mathbf{q}_{arm}) \tag{10}$$

where $\mathbf{x}^d$ and $\mathbf{x}$ are the desired and actual positions of the hand in the visual space (in our case, $\mathbf{x}_d = 0$ since we want to bring the hand to the fixation point) and $\nabla M(\mathbf{q}_{arm})$ is the gradient of $M(\mathbf{q}_{arm})$, the function we want to minimize to keep the joints far from their limits (as defined above in Section 5). K_m, K_s and K_g are positive definite diagonal gain matrices.

Both the arm-gaze forward kinematics and the visuo-arm forward kinematics are learned online using LWPR. Whenever the robot is fixating the hand, the arm-gaze forward kinematic is trained with the arm configuration $\mathbf{q}_{arm}$ as input and the gaze configuration $\mathbf{q}_{gaze}$ as output. Fixation of the hand occurs either when the reaching controller brings the hand to fixation or when the robot gazes at the hand (this is done after an unsuccessful reaching action). Then, whenever the hand is moving inside the visual field the visuo-arm forward kinematics is trained with the robot configuration $\mathbf{q}_{ah} = [\mathbf{q}_{arm} \;\; \mathbf{q}_{head}]$ as input and the hand position $\mathbf{x}$ as output.

6. Reachable space map

The Reachable Space Map is a function that defines the Reachability of a fixated object. Because the position of a fixated object is determined by the robot gaze configuration, $\mathbf{q}_{gaze}$, we can express the Reachable Space Map as follows:

$$R = f_{RS}(\mathbf{q}_{gaze}) \tag{11}$$

where R is the Reachability of the fixated object (i.e. the Reachability of the fixated location in space).

The Reachable Space Map is estimated using an LWPR neural network, which is trained online during goal-directed reaching movements. Training data is collected after each reaching action in the form $\langle \mathbf{q}_{gaze}, R \rangle$: $\mathbf{q}_{gaze}$ is the gaze configuration (which identifies the position of the target object) and R is an evaluation of the Reachability following the reaching action. The value R is in inverse proportion to the final error of the closed-loop controller if the target object is not reached (case A), while it is proportional to a measure of "optimality" of the arm configuration if the target object is reached (case B):

$$(\mathrm{A})\, R = \frac{1 - e_{cl}}{2} \tag{12}$$

$$(\mathrm{B})\, R = \frac{1 + opt_{arm}}{2} \tag{13}$$

where $e_{cl} = \|\mathbf{x}_d - \mathbf{x}\| / e^{\max}_{cl}$ is the normalized final visual error of the closed-loop controller (i.e. the 3D distance from the end effector to the fixation point, in camera coordinates, measured after the reaching movement is finished), $e^{\max}_{cl}$ is the maximum possible visual error (i.e. image boundaries) and $opt_{arm} = (1 - M(\mathbf{q}_{arm}))$ is the optimality measure, which in our implementation is the distance from joint limits. R ranges from 0.0 to 1.0, where 0.0 means far from being reachable (high error of the closed-loop controller) and 1.0 means reachable with an optimal arm configuration (as far as possible from the arm joint limits).

7. Whole-body object tracking and reaching

The Reachable Space Map defined in (11) can be exploited to plan and execute preparatory whole-body movements that place the robot in an optimal posture with respect to the object before performing the arm reaching movement; for optimal posture we mean a posture from which the object can be reached with an optimal arm configuration (i.e. as far as possible from joint limits, as defined in Section 6).

Assuming that the robot is maintaining fixation on an object, any motion of the robot body with respect to the target object causes the gaze configuration to change in order to keep the object in fixation; as a consequence, the Reachability of the object (i.e. the output of the Reachable Space Map) also changes. This means that, at least implicitly, it exists a motor–motor relation (i.e. a kinematic model) between the movement of the robot body and, through $\mathbf{q}_{gaze}$, a change in the Reachability. Assuming that this kinematic model is known (either computed analytically or learned from motor experience), given a desired change in the Reachability dR^d (e.g. from low Reachability to high Reachability) an appropriate body motion can be retrieved and executed: we call this motion a *preparatory whole-body movement*. In previous work [6] we proposed to learn such a model from motor data: the model was estimated online using LWPR, and initialized with data gathered during a motor babbling phase in which the robot was moving the waist randomly while keeping fixation of the object.

The main limitations of this approach are that: (i) body movements are executed only after fixation of the target object has been achieved, and (ii) accurate fixation of the object must be necessarily kept during the execution of the preparatory whole-body movement (otherwise the learned model is no longer valid). Due to the first limitation the overall robot behavior follows a static sequence (control the neck and the eyes to track and fixate the target object, then control the body, then control the arm reaching) that does not look natural and does not allow to exploit the body motion also during the visual tracking of the object; the second limitation may impose limits on the velocity of the body motion.

Here we propose a different approach, in which the body motion is used not only to place the robot in an optimal configuration with respect to the object, but also to extend the robot tracking capabilities. Indeed, it can happen that target objects that are detected in the periphery of the visual field cannot be fixated using only the head joints, whereas the use of additional joints may allow to achieve fixation. In particular, we employ the joints of the waist and the legs. Another important constraint is that we need to add a postural task to ensure that the robot maintains its balance (i.e. it does not fall).

While the waist joints $\mathbf{q}_{waist}$ are used to tilt and rotate the torso, the legs joints $\mathbf{q}_{legs} = [\mathbf{q}_{lleg}\ \ \mathbf{q}_{rleg}]$ are employed to control the Cartesian position of the waist with respect to a reference frame placed on the ground, between the feet, namely $\mathbf{p}_{waist} = [x_{waist}\ y_{waist}\ z_{waist}]$, as follows:

$$\dot{\mathbf{q}}_{legs} = J_L^{\dagger}(\mathbf{q}_{legs})\dot{\mathbf{p}}_{waist}^{d} \tag{14}$$

where $\dot{\mathbf{p}}_{waist}^{d}$ is the desired velocity of the waist and $J_L^{\dagger}(\mathbf{q}_{legs})$ is the pseudo-inverse of the Jacobian obtained differentiating the legs forward kinematics, $\mathbf{p}_{waist} = f_L(\mathbf{q}_{legs})$, which is computed analytically.

The three components of $\dot{\mathbf{p}}_{waist}^{d}$ are obtained in different ways, and serve different purposes: while $\dot{z}_{waist}$ (i.e. the vertical component) contributes, together with $\dot{\mathbf{q}}_{waist}$, both to realize tracking with the whole-body and to place the robot in an optimal configuration for reaching, $\dot{x}_{waist}$ and $\dot{y}_{waist}$ (i.e. the horizontal and lateral components) keep the robot balance. For the sake of simplicity, we describe these control contributions separately, even if they are realized simultaneously.

Balance control aims at canceling the ZMP (Zero Moment Point) error, keeping the ZMP in the center of the support polygon, by setting $\dot{x}_{waist}$ and $\dot{y}_{waist}$ as follows:

$$\begin{bmatrix} \dot{x}_{waist} \\ \dot{y}_{waist} \end{bmatrix} = \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} \begin{bmatrix} ZMP_x^d - ZMP_x \\ ZMP_y^d - ZMP_y \end{bmatrix} \tag{15}$$

where α_x and α_y are control gains, ZMP_x and ZMP_y are the x and y components of the measured ZMP, and ZMP_x^d and ZMP_y^d are the correspondent reference values. For each component, we call $ZMP^d - ZMP$ the ZMP error.

To describe the other controllers we define a vector of body velocities as follows:

$$\dot{\mathbf{q}}_b = [\dot{\mathbf{q}}_{waist}\ \dot{z}_{waist}] \in \mathbb{R}^3. \tag{16}$$

During object tracking the head and eyes joints, $\mathbf{q}_{gaze}$, are actuated using the gaze controller described in Section 4 (Eq. (1)).

If $\mathbf{q}_{gaze}$ reaches the limits, the waist and legs joints are used to extend the robot range of action, as follows:

$$\dot{\mathbf{q}}_b = -G_w\mathbf{x} \tag{17}$$

where $G_w \in \mathbb{R}^{3x3}$ is a positive definite gain matrix and $\mathbf{x} \in \mathbb{R}^3$ is the visual position of the target as defined in Section 4. The intuitive idea behind the design of this controller is that the rotation of the head, θ_{ny}, can be replaced by the rotation of the waist, θ_{wy}. The same holds for θ_{ev} and θ_{wp} (i.e. eyes vergence replaced by bending of the torso), and for θ_{np} and z_{waist} (i.e. head elevation/depression replaced by vertical motion of the waist).

If $\mathbf{q}_{gaze}$ is within the limits, the tracking (i.e. fixating) task can be accomplished actuating only the $\mathbf{q}_{gaze}$ joints (using the controller of Eq. (1)), and therefore the waist and legs joints can be used to position the robot in an optimal configuration with respect to the object. To realize this task, first we compute the head and eyes velocity that would realize a desired change in the Reachability. To do this, we exploit the pseudo-inverse of the Jacobian of the Reachable Space Map, which is obtained differentiating Eq. (11), as follows:

$$J_{RS}(\mathbf{q}_{gaze}) : \dot{R} = J_{RS}(\mathbf{q}_{gaze})\dot{\mathbf{q}}_{gaze} \tag{18}$$

$$\dot{\mathbf{q}}_{gaze}^{d} = J_{RS}^{\dagger}(\mathbf{q}_{gaze})dR^d \tag{19}$$

where dR^d is the desired change in the Reachability, which is typically chosen as the difference between the maximum Reachability ($R_{\max} = 1.0$) and the actual one R, leading to $dR^d = R_{\max} - R$. Then, from the desired head and eyes motion $\dot{\mathbf{q}}_{gaze}^{d}$ we derive a motion of the body $\dot{\mathbf{q}}_b$ that, if executed, would cause the head and eyes to follow the desired motion $\dot{\mathbf{q}}_{gaze}^{d}$ because of the gaze controller of Eq. (1). This transformation from head and eyes motion to body motion is obtained exploiting an approximated kinematic model, as follows:

$$\dot{\mathbf{q}}_b = K_{bh}\dot{\mathbf{q}}_{gaze}^{d} \tag{20}$$

where K_{bh} is the constant matrix defining the model. Again, the idea behind is that the robot redundancy can be exploited replacing the motion of the head and eyes joints with the motion of other body parts.

Summarizing, during this whole-body object tracking and reaching behavior the head and eyes joints ($\mathbf{q}_{gaze}$) are controlled using Eq. (1), the waist horizontal ($\dot{x}_{waist}$) and lateral ($\dot{y}_{waist}$) motion is controlled using Eq. (15), while the waist vertical motion ($\dot{z}_{waist}$) and the waist joints ($\mathbf{q}_{waist}$) are controlled using Eq. (17) when the head and eyes reach the physical limits, and using Eq. (20) otherwise. The horizontal, lateral and vertical motion of the waist is realized actuating the legs joints ($\mathbf{q}_{legs}$) through Eq. (14). This allows the robot to realize two tasks simultaneously (while keeping the balance): to track and eventually fixate the target object moving the whole body (i.e. head, eyes, waist, legs), therefore increasing the range of fixation, and to find an optimal body configuration to perform reaching. We show an example of this behavior in Section 8.2.

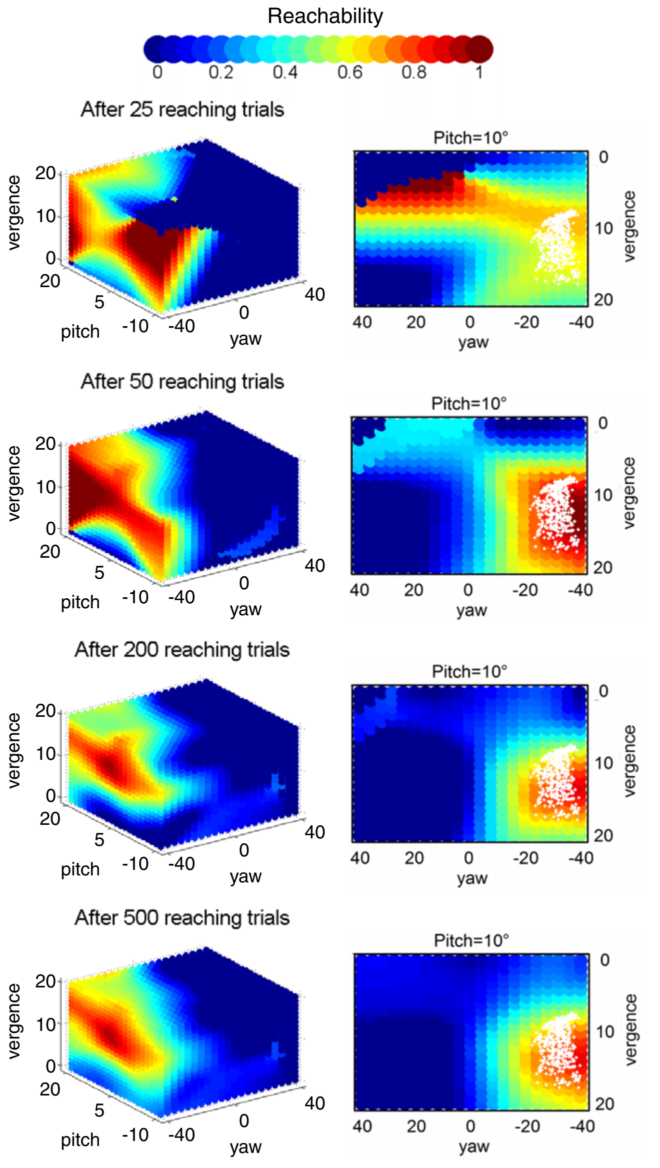


Fig. 3. From top to bottom, Reachable Space Map of the right arm after 25, 50, 200, 500 reaching trials respectively. Left image: 3D visualization. Right image: 2D visualization on the vergence/yaw plane, with $pitch = 10°$. Green/blue colors identify non-reachable regions ($R < 0.5$), yellow/red identify reachable regions ($R > 0.5$), as shown in the color bar on the top. The white dots in the right images are representative reachable points (i.e. ground truth).

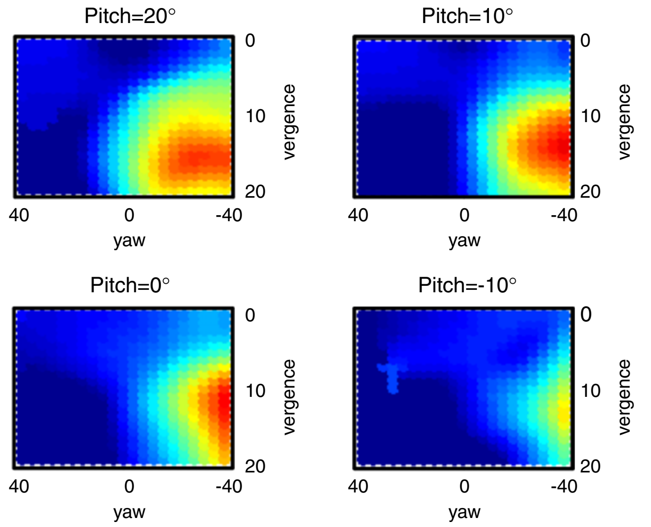


Fig. 4. Reachable Space Map of the right arm after 500 reaching trials. Four 2D visualizations on the vergence/yaw plane are depicted, with different values of pitch, namely [−10° 0° 10° 20°].

8. Experimental results

We have performed experiments with the humanoid robot Kobian to show that the robot is able to learn autonomously a motor representation of the workspace (i.e. the Reachable Space Map) and use it to perform complex whole-body reaching movements. During a first phase we showed a red ball to the robot, placed in 500 different positions: the robot used the gaze controller described in Section 4 to fixate the ball, and then the reaching controller described in Section 5 to reach for it with the right arm. The learned models used for reaching control (namely, $\mathbf{q}_{gaze} = f_{AG}(\mathbf{q}_{arm})$ and $\mathbf{x} = f_{AV}(\mathbf{q}_{ah})$) had been trained already during previous goal-directed movements (this part of learning is not considered here), and therefore they allowed to successfully reach for reachable targets; nevertheless, those models were also updated online during the movements, as described in Section 5. For more details about the autonomous online learning of the reaching controller the reader should refer to [39]. Object positions were chosen randomly, and they were either reachable (about 60%) or non-reachable. During the execution of these goal-directed movements the Reachable Space Maps was learned incrementally as described in Section 6: the data collected during this experiment is reported in Section 8.1. Then, in Section 8.2 we show how the robot was able to exploit the learned Reachable Space Map to realize optimal whole-body reaching, as introduced in Section 7.

8.1. Estimation of the robot workspace

Fig. 3 shows the progressive acquisition of the Reachable Space Map of the right arm, during the sequence of 500 reaching movements. A 3D and a 2D visualization of the map (left images and right images respectively) are depicted at four stages of the learning, namely after 25, 50, 200 and 500 reaching trials. Referring to the description in Section 3, yaw, pitch and vergence are θ_{ny}, θ_{np} and θ_{ev} respectively. The color bar on the top of Fig. 3 relates the output of the map (i.e. the Reachability $R = f(\mathbf{q}_{gaze})$) to the colors used in the images: green/blue colors identify non-reachable regions ($R < 0.5$), yellow/red identify reachable regions ($R > 0.5$). The normalized mean squared error (NMSE) of the estimation has been computed with respect to a given test set of 250 $\langle \mathbf{q}_{gaze}, R \rangle$ samples, not used for training: the NMSE is 0.35 after the first 25 reaching trials, and it drops to 0.17 (after 50 trials), to 0.10 (after 200 trials), and eventually to 0.08 (after 500 trials). In the 2D visualization of the map (a slice on the vergence/yaw plane, with $pitch = 10°$) some reachable points are displayed (i.e. the white dots) to provide a "visual" understanding of the quality of the estimation: the envelope of these points coarsely indicates the boundary between the reachable and the non-reachable part of the workspace. Noticeably, the yellow/red area highly matches the envelope of the reachable points already after 200 reaching trials, and even better after 500.

Fig. 4 provides a more detailed 2D visualization of the Reachable Space Map after 500 reaching trials, showing four slices on the vergence/yaw plane, with $pitch = [-10° \ 0° \ 10° \ 20°]$.

The vergence variable is proportional to the distance of the fixated point from center of the eyes: the more the eyes are rotated inward (i.e. high value of vergence), the closer the fixated point is. Therefore, points with very low vergence are not reachable; as we are showing the map of the right arm, this is especially true for points that lie on the robot left side ($yaw > 0°$, i.e. head looking to the left). Also some points with high vergence cannot be

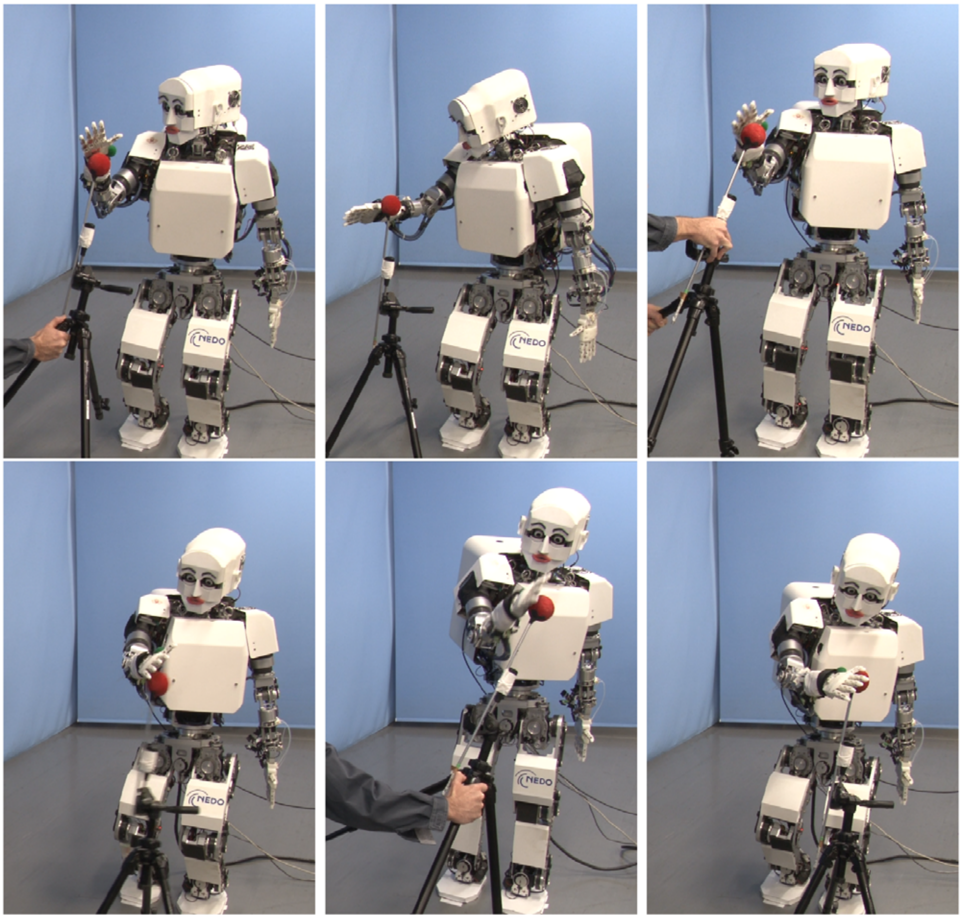


Fig. 5. Different optimal whole-body configurations for different positions of the object with respect to the robot.

reached (or can be reached but with a non-optimal configuration of the arm); on the robot right side ($yaw < 0°$), this is mainly due to the limit of the elbow joint θ_e (i.e. the elbow cannot bend more than 90°). As expected, when the yaw variable is close to the higher limit ($yaw = 40°$, robot looking to its left side) most targets are not reachable with the right arm, mainly due to the lower limit of the shoulder yaw joint θ_{sy} (i.e. arm adduction/abduction); as the yaw variable gets closer to the lower limit ($yaw = -40°$, robot looking to its right side) the Reachability progressively increases. In general, when the robot looks down ($pitch > 0°$) the reachable part of the workspace (i.e. yellow/red area) is bigger; this is due to the upper limit of the shoulder pitch joint θ_{sp} (i.e. arm elevation/depression), which does not allow the robot to reach for most of the target placed over the head.

8.2. *Intelligent whole-body reaching*

As we described in Section 7, the Reachable Space Map can be used to support the execution of complex reaching behaviors. In particular, we aim at realizing an integrated behavior in which the robot tracks and gazes at an object by controlling the head, eyes, waist and legs joints, finding an optimal posture with respect to the object, and then reaches for the object with an arm movement. The optimality is defined with respect to the reaching movement: optimal is a configuration that allows to perform the arm reaching movement optimally. In this specific implementation of our strategy, the optimality of reaching is defined as a kinematic metric, measured as the distance of the final arm configuration from the joint limits.

To test this behavior we conducted an experiment in which the robot makes reaching attempts toward targets placed in different positions. Each reaching attempt is organized as follows:

- the robot starts from a given whole-body configuration, $\mathbf{q}^s$ (the same for all the reaching attempts);
- the target (i.e. red ball) is placed within the visible space of the robot, and perceived by the robot with position $\mathbf{x}_i$ (in camera coordinates);
- the robot gazes at the target actuating the head, eyes, waist and legs, using the control strategy described in Section 7;
- as soon as the target is in fixation, and the Reachability is maximized, the robot reaches for the target with the right arm, using the controller described in Section 5;
- when reaching is completed, we measure the final arm configuration $\mathbf{q}^f_{arm}$.

During the experiment the robot performed 25 reaching attempts toward target objects placed in different positions. These targets were distributed over the whole visual field, as it can be seen in Fig. 6, where the initial visual positions of the targets are plotted: for visualization purposes only the first two dimensions of the $\mathbf{x}$ vector are displayed. All these target locations require whole-body movements to be reached for: most of them cannot be directly fixated using only the head, due to the head joint limits, and the ones that can be fixated have a low value of Reachability (i.e. lower than 0.5) if whole-body movements are not exploited. Fig. 7 shows the final values of the Reachability for all the 25 reaching attempts: all the values are higher than 0.5, meaning that the target objects were always made reachable by the whole-body movement, and

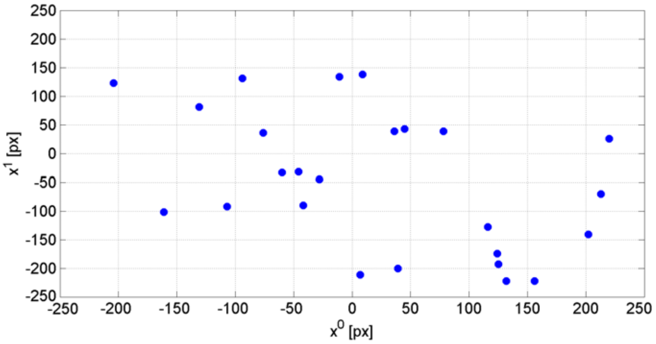


Fig. 6. Initial position of the target in the visual field, for the 25 reaching attempts. The first two components of the **x** vector are displayed.

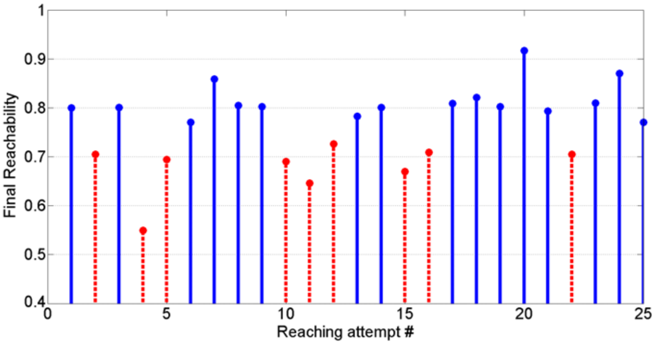


Fig. 7. Final value of the Reachability for the 25 reaching attempts. In red (dashed line), the attempts in which waist or legs joint limits are reached, making impossible to achieve a higher value of Reachability. In blue (solid line), all the other attempts. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

in fact all the 25 reaching attempts were successful. Some of the values in the plot are marked in red: in those cases the robot reached some of the joint limits of the waist or legs, making impossible to achieve a higher value of Reachability. Moreover, we evaluated the error of the Reachability estimation for each reaching attempt, $R_{err} = \|R_a - R\|$, where R_a is the actual Reachability value obtained from the final arm configurations $\mathbf{q}^f_{arm}$, using Eq. (13), and R is the Reachability value estimated with the Reachable Space Map: we verified that the Map provides a good estimation of the Reachability, as the average of R_{err} over the 25 reaching attempts was $\text{avg}(R_{err}) = 0.0240$, with standard deviation $stdev(R_{err}) = 0.0238$. Fig. 5 shows some of the final body configurations obtained by the robot to reach for the targets.

To describe the robot behavior more in detail, we show a series of figures related to one of the 25 whole-body reaching movements. In Fig. 8 a sequence of robot images taken during the execution of the movement is presented: from the initial body configuration, $\mathbf{q}^s$ (first image on the left), to the final one, $\mathbf{q}^f$. The plots in Figs. 9 and 10 depict the robot motor trajectories during the movement, as well as the position of the target in the visual field and the estimated Reachability. Fig. 11 shows the ZMP error, that is almost always equal to zero; for reference, the limits of the support polygon in meters are $[-0.10, 0.14]$ and $[-0.18, 0.18]$, on the X and Y directions respectively. Fig. 12 displays the trajectory of the robot gaze projected on a slice of the Reachable Space Map for which the head pitch angle is equal to 20°, showing how the Reachability changes during the movement as a function of the robot gaze configuration; this 2D visualization is consistent, as for most of the trajectory the head pitch angle (i.e. $\mathbf{q}^1_{gaze}$) is actually equal to 20°, as it can be seen in the gaze trajectory depicted in the top right image of Fig. 9. The estimated Reachability increases during the motion to reach a final value that is higher than 0.5. However, it should be noted that what is estimated is the Reachability of the fixated location (following the definition in Section 6), and therefore it corresponds to the Reachability of the target only when the target is in fixation; when the target is not in fixation it is not reachable by definition, given the framework that we use to control arm reaching (see Section 5). The gaze motion is aimed at both (i) bringing the target in fixation and (ii) maximizing the Reachability. Looking at the gaze trajectory in Fig. 12 it can be noticed that in the very beginning the estimated Reachability is decreased (this drop in the Reachability can be seen also in the corresponding plot in Fig. 9): this is because initially the fixation error is high (i.e. the target lies in the periphery of the visual field), and therefore the gaze motion is mainly driven by this error. Then, after this initial transient, the Reachability of the fixated location starts to increase (due to the combination of waist and legs motion), and when the fixation error is reduced it corresponds to the Reachability of the target object: finally, as soon as the Reachability is maximized and the fixation error brought to zero, the arm reaching motion can be executed.

9. Conclusions and future work

We presented a learning strategy to build a map of the workspace of a humanoid robot: a Reachable Space Map. The workspace is represented with gaze-centered motor variables, and it is learned autonomously by the robot during the execution of goal-directed reaching movements. In this regard, our approach is innovative with respect to previous works in the literature, in which the robot workspace is described with Cartesian coordinates

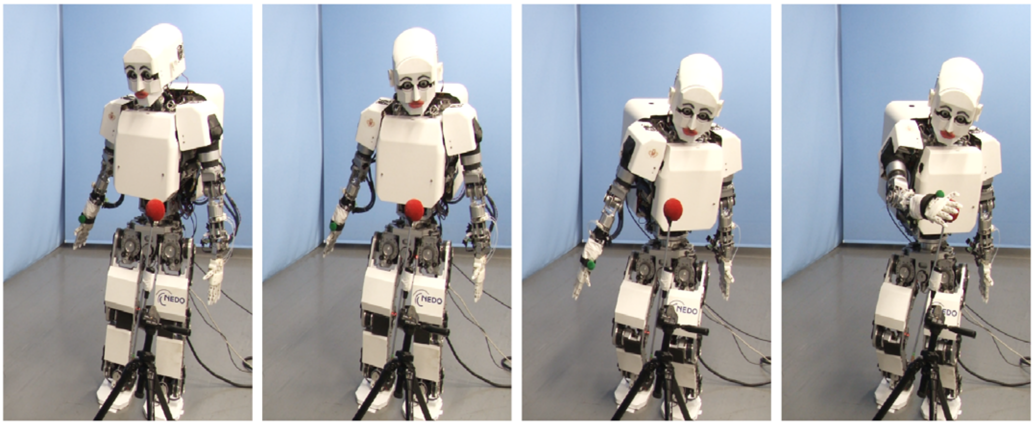


Fig. 8. Sequence of robot images during the execution of one of the optimal whole-body reaching movements.

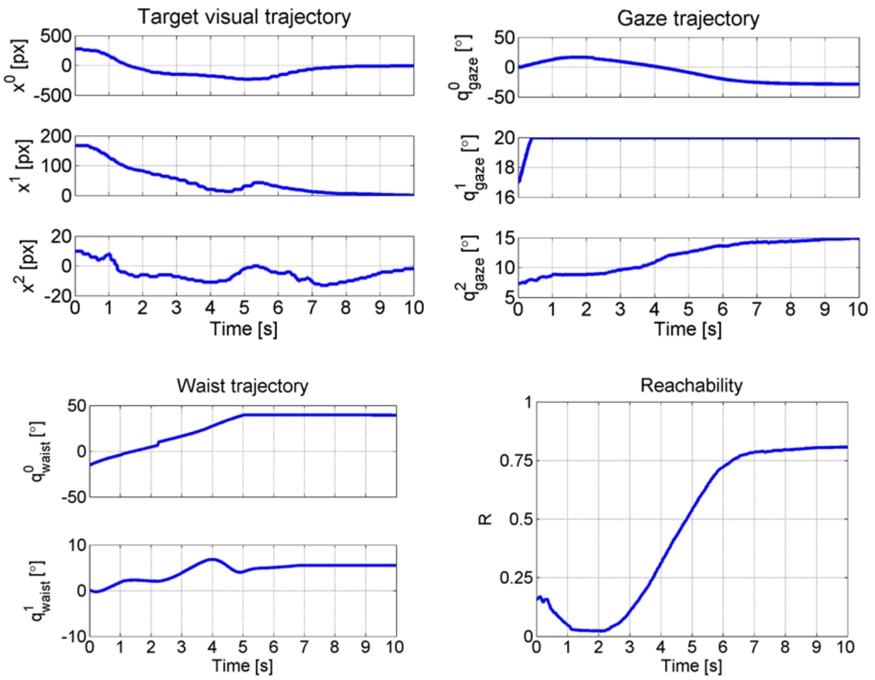


Fig. 9. Motor and sensory trajectories during the execution of the optimal whole-body reaching movement. Top left: position of the target in the image plane. Top right: gaze motor trajectory. Bottom left: waist motor trajectory. Bottom right: estimated Reachability.

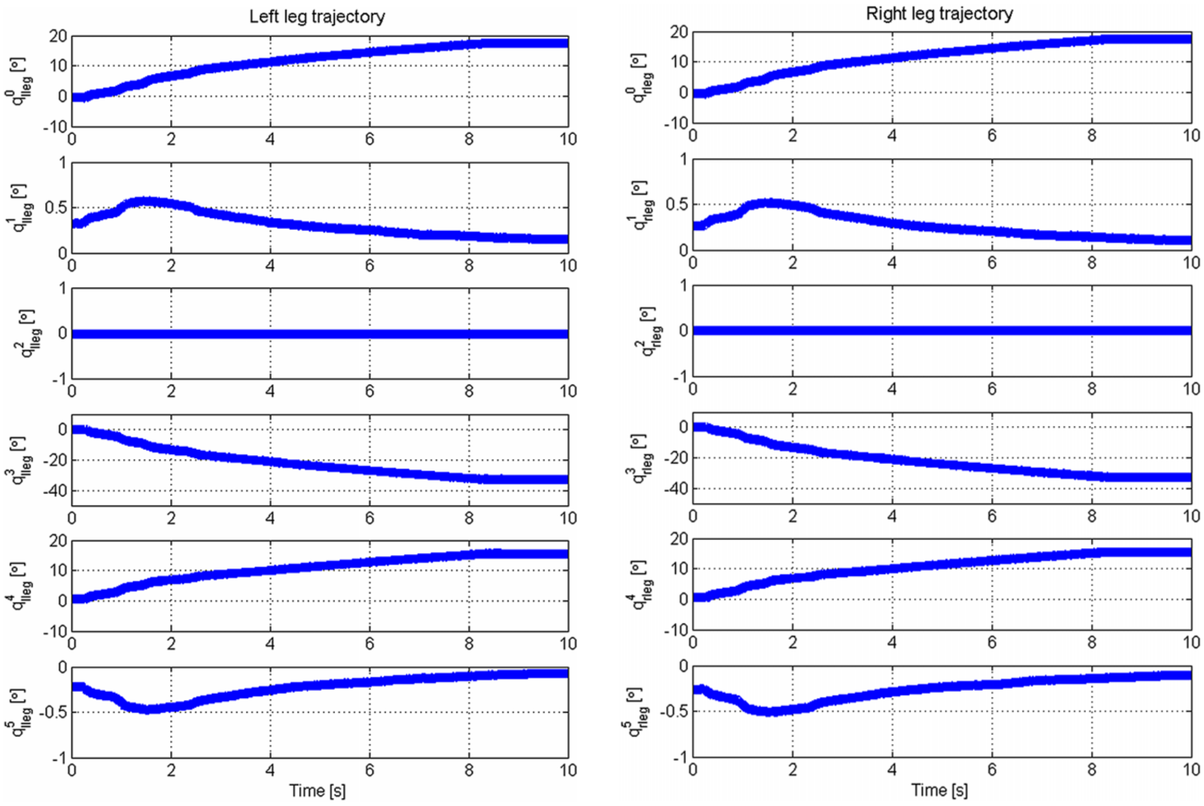


Fig. 10. Legs motor trajectories during the execution of the optimal whole-body reaching movement.

and it is learned offline. To support the effectiveness of our approach, we provide experimental results using the 48-DOF humanoid robot Kobian. We first show how the robot learns incrementally the Reachable Space Map for the right arm; then, we present a possible application in which the robot uses the learned map to realize a form of intelligent whole-body reaching. Overall, we propose an autonomous behavior in which the robot performs visual tracking of a target object using the whole body (i.e. head, eyes, waist, legs) and then it is able to find a final body configuration which is optimal for reaching.

The approach we propose to obtain the robot workspace is very general and can be implemented on any robot equipped with arm and binocular head. In the experiments presented in this paper we exploit the map to include movements of the waist (bending

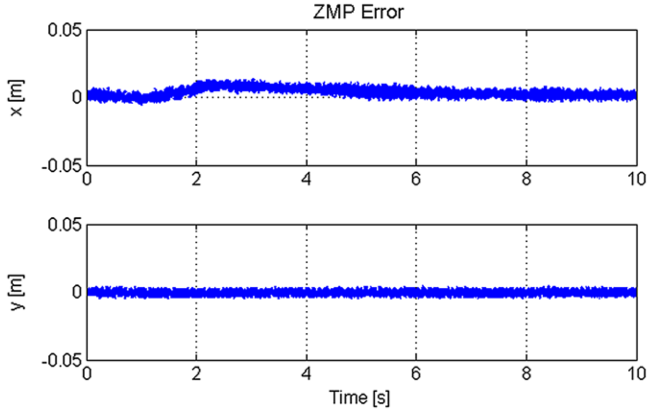


Fig. 11. ZMP error during the execution of the optimal whole-body reaching movement; *X* and *Y* components. The limits of the support polygon in meters are [−0.10, 0.14] and [−0.18, 0.18], on the *X* and *Y* directions respectively.

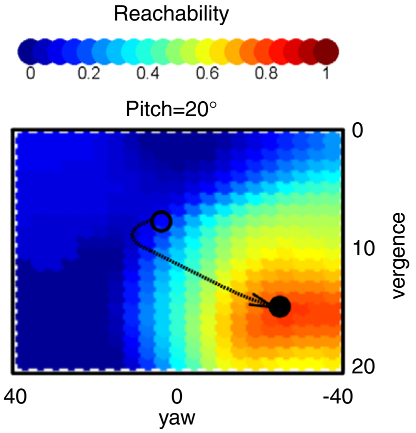


Fig. 12. Trajectory of the robot gaze configuration visualized on a slice of the Reachable Space Map (with *pitch* = 20°, upper joint limit), during the execution of the optimal whole-body reaching movement. The colors in the Map indicate the Reachability value, as described on the color bar in the top. The empty circle is the initial gaze configuration, the filled circle is the final one.

and rotation of the torso) and of the legs (flexion/extension of the knees, hips and ankles). However, this strategy can be extended to include locomotion as well: a low value of Reachability, combined with a low value of vergence (i.e. indicating that the fixated target object lies far from the robot), triggers a walking behavior in which the robot follows the direction given by the gaze and walks forward until it gets close enough to the object (as measured from both the vergence and the Reachability values) to start the whole-body reaching. This would eventually constitute an integrated system for complex goal-directed reaching behavior, able to deal with the high level of redundancy that is typical of humanoid robots. Preliminary results in this direction are presented in [43].

Finally, because reaching is typically realized in order to manipulate objects, our ultimate goal is to extend the proposed approach to the case in which it is required not only to position the hand, but also to orient it with respect to the object to be grasped. This would require to extend the reaching map taking into account the hand orientation: this can be done by augmenting the gaze configuration (that encodes objects 3D position) with the 3D object orientation expressed in the cameras reference frame. Based on this controller, the Reachable Space Map can be extended accordingly: the output of the map could estimate how well the fixated object can be reached given a certain orientation and position. This map could be learned incrementally following a similar approach to the one presented in this paper. This extension represents another interesting research direction we are currently pursuing.

References

[1] N. Endo, A. Takanishi, Development of whole-body emotional expression humanoid robot for adl-assistive rt services, J. Robot. Mechatronics 23 (6) (2011) 969–977.

[2] A. Parmiggiani, M. Maggiali, L. Natale, F. Nori, A. Schmitz, N. Tsagarakis, J. Santos-Victor, F. Becchi, G. Sandini, G. Metta, The design of the icub humanoid robot, Int. J. Human. Robot. 9 (4) (2012) 1250027.

[3] K.C. Gupta, On the nature of robot workspace, Int. J. Robot. Res. 5 (1986).

[4] O. Sigaud, C. Salan, V. Padois, On-line regression algorithms for learning mechanical models of robots: a survey, Robot. Auton. Syst. 59 (12) (2011) 1115–1129.

[5] D. Nguyen-Tuong, J. Peters, Model learning in robotics: a survey, Cogn. Process. 12 (4) (2011) 319–340.

[6] L. Jamone, L. Natale, G. Sandini, A. Takanishi, Interactive online learning of the kinematic workspace of a humanoid robot, in: International Conference on Intelligent Robots and Systems, IEEE-RSJ, 2012.

[7] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, A. Takanishi, Learning the reachable space of a humanoid robot: a bio-inspired approach, in: International Conference on Biomedical Robotics and Biomechatronics, IEEE, 2012.

[8] R.A. Andersen, G.K. Essick, R.M. Siegel, Encoding of spatial location by posterior parietal neurons, Science 230 (1985) 456–458.

[9] J.R. Duhamel, C.L. Colby, M.E. Goldberg, The updating of the representation of visual space in parietal cortex by intended eye movements, Science 255 (1992) 90–92.

[10] A.P. Batista, C.A. Buneo, L.H. Snyder, R.A. Andersen, Reach plans in eyecentered coordinates, Science 285 (1999) 257–260.

[11] W.P. Medendorp, H.C. Goltz, T. Vilis, J.D. Crawford, Gaze-centered updating of visual space in human parietal cortex, J. Neurosci. 23 (15) (2003) 6209–6214.

[12] A. Bosco, R. Breveglieri, E. Chinellato, C. Galletti, P. Fattori, Reaching activity in the medial posterior parietal cortex of monkeys is modulated by visual feedback, J. Neurosci. 30 (44) (2010) 14773–14785.

[13] P.R. Brotchie, R.A. Andersen, L.H. Snyder, S.J. Goodman, Head position signals used by parietal neurons to encode locations of visual stimuli, Nature 375 (1995) 232–235.

[14] J.F. Soechting, M. Flanders, Sensorimotor representations for pointing to targets in three-dimensional space, J. Neurophysiol. 62 (1989) 582–594.

[15] J. McIntyre, F. Stratta, F. Lacquaniti, Viewer-centered frame of reference for pointing to memorized targets in three-dimensional space, J. Neurophysiol. 62 (1997) 582–594.

[16] D.Y.P. Henriques, E.M. Klier, M.A. Smith, D. Lowy, J.D. Crawford, Gaze-centered remapping of remembered visual space in an open-loop pointing task, J. Neurosci. 18 (4) (1998) 1583–1594.

[17] A. Blangero, Y. Rossetti, J. Honor, L. Pisella, Influence of gaze direction on pointing to unseen proprioceptive targets, Adv. Cogn. Psychol. 1 (1) (2005) 9–16.

[18] K. Fiehler, I. Schtz, D.Y.P. Henriques, Gaze-centered spatial updating of reach targets across different memory delays, Vis. Res. 51 (2011) 890–897.

[19] M. Flanders, L. Daghestani, A. Berthoz, Reaching beyond reach, Exp. Brain Res. 126 (1) (1999) 19–30.

[20] M. Marjanovic, B. Scassellati, M. Williamson, Self-taught visually guided pointing for a humanoid robot, in: International Conference on Simulation of Adaptive Behavior, MA, 1996.

[21] S. Rougeaux, Y. Kuniyoshi, Robust tracking by a humanoid vision system, in: International Workshop on Humanoid and Human Friendly Robotics, Tsukuba, Japan, 1998.

[22] G. Metta, G. Sandini, J. Konczak, A developmental approach to visually-guided reaching in artificial systems, Neural Netw. 12 (10) (1999) 1413–1427.

[23] M. Hulse, S. McBride, J. Law, M. Lee, Integration of active vision and reaching from a developmental robotics perspective, Trans. Auton. Mental Dev. 2 (4) (2010) 355–367.

[24] E. Chinellato, M. Antonelli, B.J. Grzyb, A.P. del Pobil, Implicit sensorimotor mapping of the peripersonal space by gazing and reaching, Trans. Auton. Mental Dev. 4 (2010) 43–53.

[25] Y. Guan, K. Yokoi, X. Zhang, Numerical methods for reachable space generation of humanoid robots, Int. J. Robot. Res. 27 (2008) 935–950.

[26] F. Zacharias, C. Borst, G. Hirzinger, Capturing robot workspace structure: representing robot capabilities, in: International Conference on Intelligent Robots and Systems, IROS, 2007.

[27] F. Zacharias, C. Borst, G. Hirzinger, Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace, in: International Conference on Advanced Robotics, 2009.

[28] N. Vahrenkamp, T. Asfour, R. Dillmann, Efficient inverse kinematics computation based on reachability analysis, Int. J. Human. Robot. 9 (4) (2012) 1250035.

[29] S. Vijayakumar, S. Schaal, Locally weighted projection regression: an $o(n)$ algorithm for incremental real time learning in high dimensional space, in: International Conference on Machine Learning, ICML, 2000.

[30] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, A. Takanishi, Learning task space control through goal directed exploration, in: International Conference on Robotics and Biomimetics, IEEE-RAS, Phuket, Thailand, 2011.

[31] D. Nguyen-Tuong, M. Seeger, J. Peters, Model learning with local Gaussian process regression, Adv. Robot. 23 (15) (2009) 2015–2034.

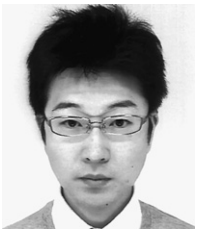
[32] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. Lim, A. Takanishi, Development of a new humanoid robot wabian-2, in: International Conference on Robotics and Automation, IEEE-RAS, 2006.

[33] H. Miwa, T. Okuchi, H. Takanobu, A. Takanishi, Development of a new human-like head robot we-4, in: International Conference on Intelligent Robots and Systems, IEEE-RSJ, 2002.
[34] M. Zecca, N. Endo, S. Momoki, K. Itoh, A. Takanishi, Development of a new human-like head robot we-4, in: International Conference on Intelligent Robots and Systems, IEEE-RSJ, 2002.
[35] N. Endo, K. Endo, K. Hashimoto, T. Kojima, F. Iida, A. Takanishi, Integration of emotion expression and visual tracking locomotion based on vestibulo-ocular reflex, in: International Symposium on Robot and Human Interactive Communication, 2010.
[36] G. Metta, P. Fitzpatrick, L. Natale, Yarp: yet another robot platform, Int. J. Adv. Robot. Syst. 3 (1) (2006) 43–48. Special issue on software development and integration in robotics.
[37] M. Taiana, J. Santos, J. Gaspar, J. Nascimento, A. Bernardino, P. Lima, Tracking objects with generic calibrated sensors: an algorithm based on color and 3D shape features, Robot. Auton. Syst. 58 (6) (2010) 784–795.
[38] L. Natale, F. Nori, G. Metta, G. Sandini, Learning precise 3d reaching in a humanoid robot, in: International Conference of Development and Learning, IEEE, London, UK, 2007.
[39] L. Jamone, L. Natale, G. Metta, F. Nori, G. Sandini, Autonomous on-line learning of reaching behavior in a humanoid robot, Int. J. Human. Robot. 9 (3) (2012) 1250017.
[40] A. Wachter, L.T. Biegler, On the implementation of a primal–dual interior point filter line search algorithm for large-scale nonlinear programming, Math. Program. 106 (1) (2006) 25–57.
[41] U. Pattacini, F. Nori, L. Natale, G. Metta, G. Sandini, An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots, in: International Conference on Intelligent Robots and Systems, IROS, Taipei, Taiwan, 2010.
[42] A. Ligeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, Trans. Syst. Man Cyber. 7 (1977) 868–871.
[43] M. Brandao, L. Jamone, P. Kryczka, N. Endo, K. Hashimoto, A. Takanishi, Reaching for the unreachable: integration of locomotion and whole-body movements for extended visually guided reaching, in: International Conference on Humanoid Robots, IEEE-RAS, 2013.

Lorenzo Jamone received his M.S. degree in Computer Engineering from the University of Genoa, and his Ph.D. degree in Humanoid Technologies from the University of Genoa and the IIT (Italian Institute of Technology), in 2006 and 2010, respectively. He was a Research Fellow at the RBCS department of the IIT, in 2010, and Postdoctoral Researcher at the Takanishi Laboratory in Waseda University, Tokyo, from 2010 to 2012. He is currently Postdoctoral Researcher at the Instituto Superior Tecnico, Lisbon, Portugal. His research interests include humanoid robotics, online sensorimotor learning, bio-inspired motor control and developmental robotics.

Martim Brandao received his M.S. degree in Electrical and Computer Engineering from the Instituto Superior Tecnico (Lisbon, Portugal) in 2010. He was a Research Student at the Computer Vision Lab of the Instituto Superior Tecnico from 2010 to 2011, and he is currently Research Student at the Takanishi Laboratory in Waseda University, Tokyo, since 2011. His research interests include image processing, 3D scene representation and humanoid robotics.

Lorenzo Natale received his degree in Electronic Engineering (with honors) in 2000 and Ph.D. in Robotics in 2004 from the University of Genoa. During his M.Sc. and Ph.D. he worked in LIRA-Lab (Laboratory for Integrated Advanced Robotics), at the University of Genoa. Between 2005 and 2006 he was Postdoctoral researcher at the MIT Computer Science and Artificial Intelligence Laboratory, in the Humanoid Robotics Group. He is now team leader at the IIT. His research focuses on developmental robotics, sensorimotor learning and perception in artificial and biological systems.



Kenji Hashimotois an Assistant Professor of the Research Institute for Science and Engineering, Waseda University, Japan. He received the B.E. and M.E. degrees in Mechanical Engineering from Waseda University, Japan, in 2004 and 2006, respectively. He received the Ph.D. degree in Integrative Bioscience and Biomedical Engineering from Waseda University, Japan, in 2009. While a Ph.D. candidate, he was funded by the Japan Society for the Promotion Science as a Research Fellow. He was a Postdoctoral Researcher at the Laboratoire de Physiologie de la Perception et de l'Action in UMR 7152 College de France-CNRS, France from 2012 to 2013. His research interests include walking systems, biped robots, and humanoid robots.

Giulio Sandini is Director of Research at the IIT (department of Robotics, Brain and Cognitive Sciences) and full professor of bioengineering at the University of Genoa. He graduated in Electronic Engineering at the University of Genoa in 1976 and he was research fellow and assistant professor at the Scuola Normale Superiore in Pisa until 1984. After his return to Genoa in 1984 as associate professor, in 1990 he founded the LIRA-Lab. He is with IIT since 2006. He is author of more than 300 publications and five international patents. His research activities are in the fields of Biological and Artificial Vision, Computational and Cognitive Neuroscience and Robotics.

Atsuo Takanishi is a Professor of the Department of Modern Mechanical Engineering, Waseda University and a concurrent Professor and one of the core members of the HRI (Humanoid Robotics Institute), Waseda University. He received the B.S.E. degree in 1980, the M.S.E. degree in 1982 and the Ph.D. degree in 1988, all in Mechanical Engineering from Waseda University. He is author of more than 300 publications in the areas of Humanoid Robotics and Medical Robotics.