Neural Networks 35 (2012) 54–69

Contents lists available at SciVerse ScienceDirect

Neural Networks

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

A bio-inspired kinematic controller for obstacle avoidance during reaching tasks with real robots

Narayan Srinivasa[a,*,1], Rajan Bhattacharyya[a,1], Rashmi Sundareswara[a], Craig Lee[a], Stephen Grossberg[b]

[a] Center for Neural and Emergent Systems, Department of Information and Systems Sciences, HRL Laboratories LLC, 3011, Malibu Canyon Road, Malibu, CA 90265, United States

[b] Center for Adaptive Systems, Graduate Program in Cognitive and Neural Systems, Department of Mathematics, Boston University, 677 Beacon Street, Boston, MA 02215, United States

ARTICLE INFO

Article history:
Received 29 November 2010
Received in revised form 19 June 2012
Accepted 28 July 2012

Keywords:
Learning
Reach
Robotics
Obstacle avoidance
ART
DIRECT
Attentional shroud
Inverse model
Motor control

ABSTRACT

This paper describes a redundant robot arm that is capable of learning to reach for targets in space in a self-organized fashion while avoiding obstacles. Self-generated movement commands that activate correlated visual, spatial and motor information are used to learn forward and inverse kinematic control models while moving in obstacle-free space using the Direction-to-Rotation Transform (DIRECT). Unlike prior DIRECT models, the learning process in this work was realized using an online Fuzzy ARTMAP learning algorithm. The DIRECT-based kinematic controller is fault tolerant and can handle a wide range of perturbations such as joint locking and the use of tools despite not having experienced them during learning. The DIRECT model was extended based on a novel *reactive obstacle avoidance direction* (DIRECT-ROAD) model to enable redundant robots to avoid obstacles in environments with simple obstacle configurations. However, certain configurations of obstacles in the environment prevented the robot from reaching the target with purely reactive obstacle avoidance. To address this complexity, a self-organized process of mental rehearsals of movements was modeled, inspired by human and animal experiments on reaching, to generate plans for movement execution using DIRECT-ROAD in complex environments. These mental rehearsals or plans are self-generated by using the Fuzzy ARTMAP algorithm to retrieve multiple solutions for reaching each target while accounting for all the obstacles in its environment. The key aspects of the proposed novel controller were illustrated first using simple examples. Experiments were then performed on real robot platforms to demonstrate successful obstacle avoidance during reaching tasks in real-world environments.

1. Introduction

The future of industrial robots is progressing towards systems that have a great deal of flexibility in handling various tasks in a self-organized and safe manner. One such task is in the area of intelligent robotic assembly. This task requires the robot to reach and grasp complex objects and manipulate them while satisfying many requirements: the robot must be able to reach for any point in its workspace from any other point with varying speed requirements; the robot must be able to avoid collisions with objects in its environment; it must be able to tolerate unexpected contingencies and perform reliably despite changes in sensor geometry and changes in joint mobility; it must be able to shape its hand around the object to enable stable grasps. In this study the problem of collision avoidance in highly complex environments is developed by drawing inspiration from biological systems.

Previous work has demonstrated obstacle avoidance in complex environments using a path planning approach (Schwarzer & Saha, 2005). In this framework, paths are generated by a path planner, and subsequently evaluated by a separate process to determine collisions with obstacles in the environment (Barraquand et al., 1997; Sanchez & Latombe, 2003). By exhaustively searching through the space of possible paths, a candidate non-collision path is selected for execution. In contrast, path selection and avoidance is an integrated process in human and animal reaching behaviors, and forms the basis of bio-inspired approaches to obstacle avoidance and movement planning.

Another approach to obstacle avoidance is based on modifying path plans created with attractor dynamics (Iossifidis & Schoner, 2007). The controller is developed on the basis of a readily available attractor dynamics model that provides a defined trajectory for the end-effector to reach a target dependent on robot geometry and absolute joint angles. This trajectory serves as the attractor. The influence of the obstacles is superimposed on the attractor

[*] Corresponding author. Tel.: +1 310 317 5870; fax: +1 310 317 5958.
E-mail address: nsrinivasa@hrl.com (N. Srinivasa).

[1] Authors contributed equally.

0893-6080/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2012.07.010

dynamics to modify the trajectory. There are three key differences between this approach and the work described in this paper. First, the controller in Iossifidis and Schoner (2007) requires a specific trajectory to be defined in order to reach a target. The trajectory for the novel controller described in this paper is automatically generated by only using directions to targets and thus is less dependent on system parameters and also much easier to implement. Second, the controller in Iossifidis and Schoner (2007) is dependent on absolute joint angles to define the attractor dynamics while the controller described in this work *learns* a mapping between the changes in the joint angles and changes in direction to the target. Third, the influence of the obstacles modifies the changes in joint angles in this work, while in Iossifidis and Schoner (2007) obstacles modify the absolute joint angles. Thus, the bio-inspired controller described in this paper scales well to robots with more degrees of freedom while being robust to unexpected faults and contingencies.

A prior work that explicitly addresses the obstacle avoidance problem in complex environments is described in Mel (1990). This bio-inspired approach performs a random exploration of arm movements "mentally" using robot models. Each arm configuration that is generated is evaluated for its ability to reach the target while avoiding obstacles. If none of the random movements from the present configuration is successful, then the system "backtracks" to a previous configuration and the search process is continued. This approach creates jerky movements that are arguably unnatural. Furthermore, it is unclear if any subsequent movement smoothing could be applied to avoid collisions.

The approach described in this paper has similarities to Mel (1990) in that "mental rehearsals" (also called lookahead planning) are performed to decide on a plan to avoid obstacles. The mental rehearsal approach employs a perceptual process that is used to generate via-points or intermediate targets during reaching tasks. The via-points are tested for suitability to reach both the target and the initial arm configuration via a reactive obstacle avoidance controller. This approach obviates the need for "backtracking" (Mel, 1990) and is more efficient and practical. Another advantage is that arm movements are smooth due to seamless target switching from via-point to target.

The rest of the paper is organized as follows. Section 2 first describes the basic DIRECT model, followed by a description of the DIRECT-ROAD model for reactive avoidance. Next, the limitation of using the reactive controller in more complex obstacle and target configurations is motivated with an example of a local minima situation. Computer simulations are described to demonstrate obstacle avoidance in complex environments for a planar redundant robot arm. In Section 3, the overall system architecture is described. This is followed in Section 4 by a description of various experiments that were performed on these platforms to demonstrate examples of obstacle avoidance during reaching tasks in complex real-world environments. Section 5 describes related research while concluding remarks are provided in Section 6.

2. Learning to reach to 3D targets

2.1. DIRECT controller

The DIRECT model (Bullock, Grossberg, & Guenther, 1993) uses a mapping of desired movement directions in task space into joint rotations (Fig. 1). The model automatically compensates for externally imposed constraints on effector motion. The use of a directional mapping for movement control is closely related to robotic controllers that utilize a generalized inverse of the Jacobian matrix (Baillieul, Hollerbach, & Brockett, 1984; Hollerbach, 1982; Mussa-Ivaldi & Hogan, 1991). For notational convenience, all vector variables will be in bold while all scalar variables will be italicized.

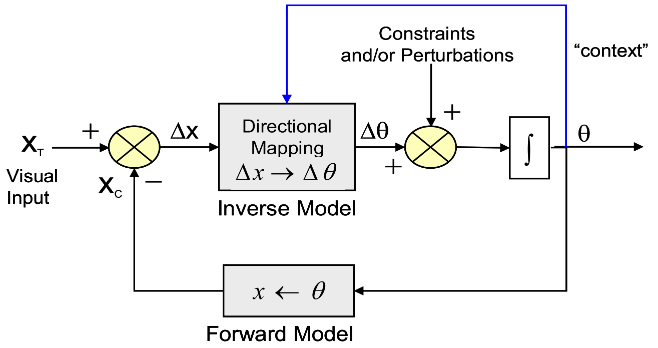

Fig. 1. The control flow depiction of the reactive DIRECT model is shown here. The constraints and/or perturbations are inserted into the control flow based on perceptual influences of obstacles in the environment during hand movements towards targets/goals. The variables $\mathbf{X}_T$ and $\mathbf{X}_c$ represent the visual target and current end-effector position, respectively. $\Delta\boldsymbol{x}$ represents the direction from end-effector to target while $\Delta\boldsymbol{\theta}$ represents the change in joint angles of the robot. $\boldsymbol{\theta}$ represents the current joint angles of the robot.

The relationship between the spatial velocity of the end-effector and the joint velocities of a manipulator such as an arm is given by the following equation:

$$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \tag{1}$$

where $\dot{\mathbf{x}}$ is the spatial velocity vector of the hand, $\dot{\boldsymbol{\theta}}$ is the joint velocity vector and $\mathbf{J}(\boldsymbol{\theta})$ is the manipulator's Jacobian matrix, whose elements depend only on the joint configuration $\boldsymbol{\theta}$. Here $\boldsymbol{\theta}$ serves as a "context" vector (Fig. 1) for selecting the appropriate $\mathbf{J}$ matrix. To obtain a desired joint vector that moves the hand at a desired spatial velocity, Eq. (1) can be rewritten as

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1}(\boldsymbol{\theta})\dot{\mathbf{x}}, \tag{2}$$

where $\mathbf{J}^{-1}(\boldsymbol{\theta})$ is an inverse of the Jacobian matrix. For a redundant manipulator, a unique inverse does not exist. In this case, a generalized inverse, or pseudo-inverse, of the Jacobian matrix must be used. The commonly used generalized inverse is the Moore–Penrose (MP) pseudo-inverse, which has the desirable property of computing the minimum norm joint rotation vector that can produce the desired spatial velocity. The DIRECT model learns approximations of the pseudo-inverse via action-perception cycles (Bullock et al., 1993; Srinivasa, Bhattacharyya, & Grossberg, 2008; Srinivasa & Grossberg, 2008).

Motor babbling in the context of action-perception cycles was used to generate the data to learn forward and inverse models. The procedure consisted of making movements, and recording end-effector positions and changes in end-effector direction. Using directions for movement execution is a well-studied phenomenon in humans and other animals (Georgopoulos, Schwartz, & Kettner, 1986). In this procedure, the end-effector assumed a joint angle designated as a "major posture". The end-effector position served as the reference for subsequent direction computations for the local inverse model. Subsequently, the end-effector was moved by perturbing single joints, designated as "micro-babbles". This procedure was applied at different major postures to estimate directional mappings within the workspace. With such a directional mapping, the spatial trajectories produced by the inverse kinematics transformation are not straight lines in task space, but are slightly curved. These learned mappings are also well behaved at task space singularities (Fiala, 1995; Rosenbaum, Meulenbroek, & Vaughan, 2001).

2.2. DIRECT using Fuzzy ARTMAP

The Jacobian-based approach above does not specifically take advantage of the multiple solutions that can be specified for

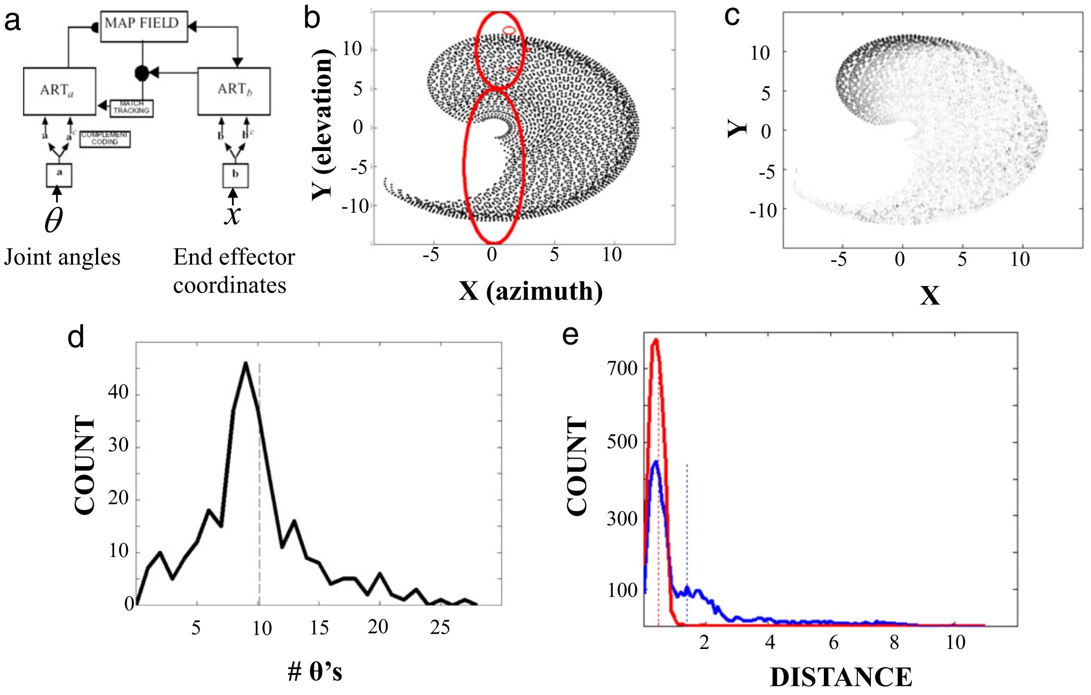

Fig. 2. Neural network for learning forward model, and characterization of predictive performance from simulated motor babbling data. (a) Fuzzy ARTMAP with match tracking for learning the forward model. Here a single end-effector position **x** was mapped to each **θ**. (b) Task space end-effector positions from motor babbling. (c) Error in learned forward model using postures from motor babbling shown as gray scale value at each point from (b), where the maximum error (black) is 10.09 distance units. (d) Histogram showing distribution of associative mapping of different postures (# θ's) to a single end-effector position (COUNT) for motor babbling data shown in (b). This data shows that the network often learned ten different postures for a given task space position. (e) Histogram showing distribution of predictive errors from forward model, with error magnitude on the *x*-axis (DISTANCE), and the number of instances observed for error magnitude using babbling data (COUNT). Prediction errors resulting purely from ARTb (red) in estimating task space positions are compared to overall forward model prediction errors (blue; same data from (c)). Dotted lines show averages of distributions in (d)–(e). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Source: (a) Adapted from (Carpenter et al., 1992), with permission.

each kinematic change. Instead, a learning systems approach was incorporated with learned long term memory (LTM) traces used to generate inverse and forward models (Wolpert & Kawato, 1998). Fuzzy ARTMAP-based learning was used to learn a many-to-one associative mapping. This algorithm is one of a class of learning algorithms called Adaptive Resonance Theory (ART) neural networks that are capable of robust, self-organized incremental learning, and have been used for pattern recognition tasks in wide variety of applications (e.g., Carpenter & Grossberg, 2003; Dagli & Huggahalli, 1995; Gan & Lua, 1992; Srinivasa & Ahuja, 1998; Srinivasa & Jouaneh, 1993; Srinivasa & Sharma, 1997; Tomida, Hanai, Honda, & Kobayashi, 2002; Wienke & Buydens, 1996). While Fuzzy ARTMAP was sufficient for the purposes of demonstrating learning in this study, it should be noted that new versions of the ART family of models (Amis & Carpenter, 2007, 2009; Carpenter, 1997, 2003; Carpenter & Gaddam, 2010; Carpenter, Milenova, & Noeske, 1998) may result in improved learning performance.

Briefly, the collected motor babbling data (Section 4.2) was fed to Fuzzy ARTMAP modules which learned the data as well as associative connections between related data (map field). In the case of the forward model, the associative connections formed from many postures to one end-effector position in task space. For the inverse model, the Fuzzy ARTMAP module learned a mapping from many joint angle changes to one end-effector direction in task space. Fig. 2(a) illustrates the Fuzzy ARTMAP network architecture for learning the forward model, using two ART modules (ARTa and ARTb) that learn to categorize joint angles (Θ) and task space coordinates (x), respectively, along with an associative mapping (MAP FIELD) between the modules. The predictive performance of Fuzzy ARTMAP after learning the forward model for a three-degree-of-freedom (DOF) planar kinematic chain is illustrated in Fig. 2(b)–(e). Task space positions and the corresponding joint angles of the end-effector are recorded during motor babbling and fed to the network. The task space positions for the tip of the end-effector during motor babbling are shown in Fig. 2(b), and the predictive error from the Fuzzy ARTMAP network after learning the forward model is shown in Fig. 2(c), where larger errors are seen near singularities at the edges of the workspace where babbling was limited by the joint angle range. Because the planar arm in this simulation is redundant, multiple joint configurations can place the end-effector in the same task space position. The network learns these many-to-one associations in the MAP FIELD, and Fig. 2(d) shows the distribution is centered near ten postures for one task space position after learning. The distribution of predictive errors was further examined for the entire network (ARTa, ARTb modules and MAP FIELD) as shown in Fig. 2(e) (blue) and for the task space positions alone (ARTb) as shown in Fig. 2(e) (red). Each instance of ARTMAP learning used a baseline vigilance parameter $\rho = 0.95$. A high setting of baseline vigilance helped minimize reaching errors while maximizing code compression. This high vigilance setting also reduced the number mismatch resets during learning, thereby facilitating real-time performance.

Depending on the desired task performance, motor babbling can be performed with varying degrees of granularity in different areas of the major posture space, to tailor the end-effector accuracy. During execution, if the inverse model recalls multiple solutions ($\Delta\boldsymbol{\theta}$) for a desired direction ($\Delta\mathbf{x}$), the forward model is employed to evaluate which solution $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}$ will result in a $\Delta\mathbf{x}$ closest to the desired direction for the end-effector. Examples of generated trajectories are shown in Fig. 3 for a seven-DOF simulated redundant robot.

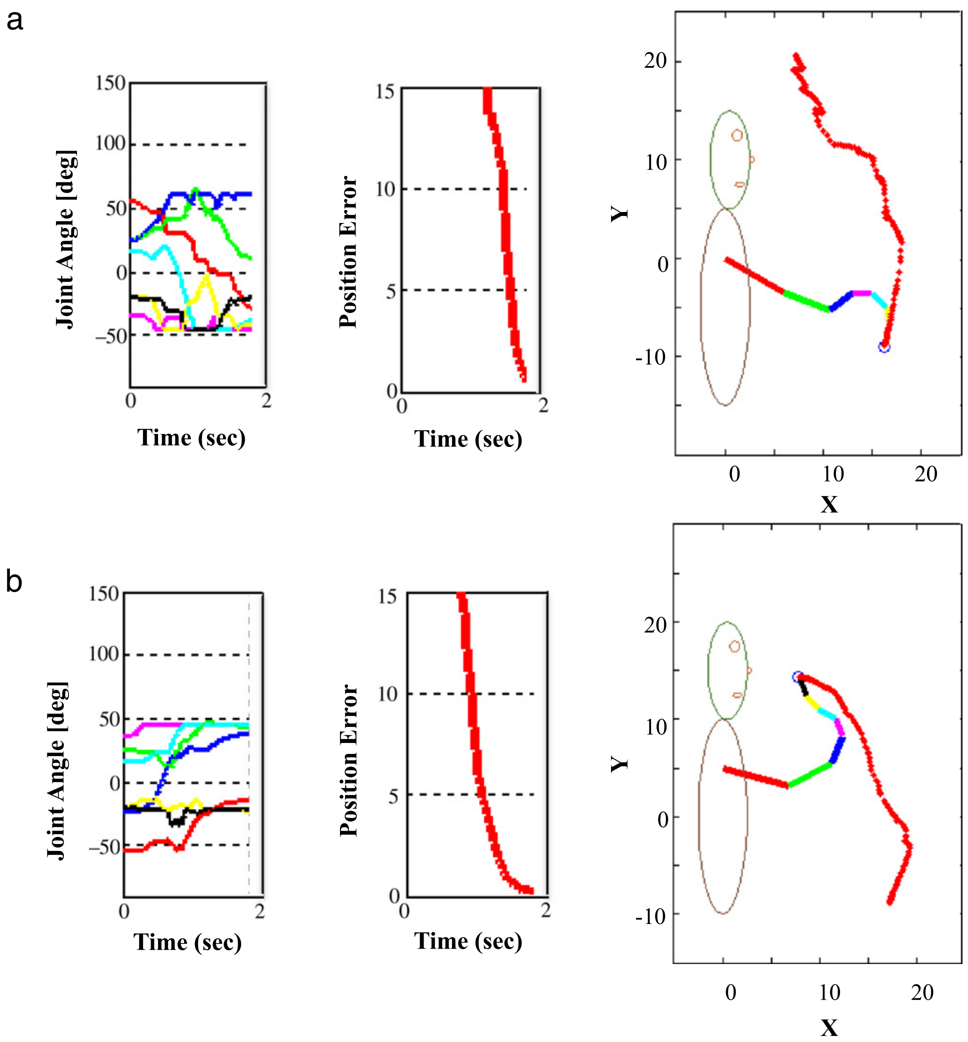

Fig. 3. Seven-DOF planar arm performance using Fuzzy ARTMAP for forward and inverse models for two initial conditions containing different target locations and arm postures. Joint angles, position error for the end-effector in the Cartesian task space, and arm trajectories (red) are shown from left to right. (Colors for joint angles from left panels match limb segments from right panel.) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.3. DIRECT with reactive obstacle avoidance controller (DIRECT-ROAD)

The controller is trained using the learning framework of DIRECT (Bullock et al., 1993; Srinivasa & Grossberg, 2008). This controller depends upon constructing two different models via learning (Fig. 1). First, the inverse model maps the changes in end-effector directions to the corresponding changes in joint angles. Second, the forward model maps the joint angles of the robot into the end-effector coordinates. This controller is fault tolerant and scalable to large numbers of degrees of freedom. However, the feature that is absent is the ability to avoid obstacles while performing actions in space such as reaching to targets. This capability is however essential in applications where the environment is full of obstacles that the robot needs to avoid in order to ensure safety and reliability while performing the task.

To address this, a reactive controller was first designed that was capable of obstacle avoidance in simple obstacle environments. The control model is the same as in Fig. 1. This reactive controller utilizes inputs about obstacles in the operating environment of the system and adapts the framework of potential field methods, which frame kinematic control in terms of force (Khatib, 1986). The target serves as an attractive force, whereas obstacles serve as repulsive forces in the potential field, and typically kinematic controllers follow a gradient in the field to realize a path to the target around the obstacles.

In the DIRECT-ROAD controller the directions of obstacles to limbs serve as constraints instead of repulsive forces. These constraints are described by $\mathbf{v}_{\text{on}}$ vectors, which represent the optimal direction for limb n to avoid an obstacle o based on the point on the obstacle closest to the limb and the corresponding point on the limb closest to the obstacle (Fig. 4). These points were computed dynamically in both simulations and experiments (Section 4); vision-based sensing was employed in experiments; however, proximity sensors at fixed locations on the limbs may also be used.

The obstacle-to-limb vector is scaled by the inverse of the distance from the obstacle o to each limb (based on the $\mathbf{v}_{\text{on}}$ vectors), summed across obstacles, normalized, and scaled by the inverse of the minimum distance from each limb to the obstacles:

$$\mathbf{v}_{\text{on}} = k_0 \frac{\left\|\mathbf{v}_{\text{target}}\right\|}{\min \|v_{\text{on}}\|} \left(\frac{\sum_{i}^{\#\ obstacles} \left(\frac{1}{\|\mathbf{v}_{\text{in}}\|}\right) \mathbf{v}_{\text{in}}}{\left\| \sum_{i}^{\#\ obstacles} \left(\frac{1}{\|\mathbf{v}_{\text{in}}\|}\right) \mathbf{v}_{\text{in}} \right\|} \right), \tag{3}$$

where $\|\cdot\|$ is the L_2 norm operator and $\mathbf{v}_{\text{target}}$ is the direction to the target (Fig. 4). The fraction $\frac{1}{\min\|\mathbf{v}_{\text{on}}\|}$ is the inverse of the minimum

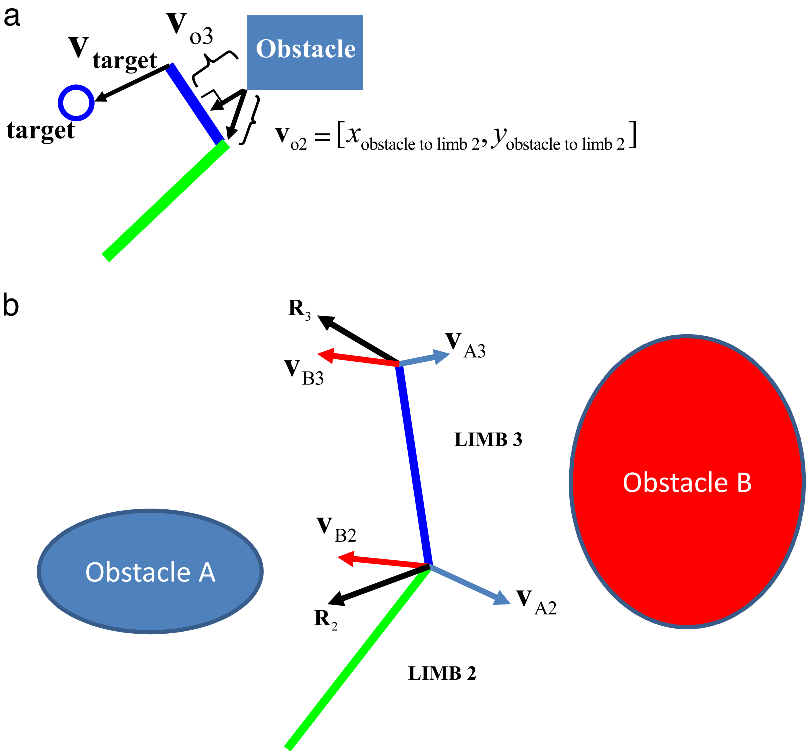
a
V target
V o3
Obstacle
target
$\mathbf{v}_{o2} = [x_{\text{obstacle to limb 2}}, y_{\text{obstacle to limb 2}}]$
b
R 3
V A3
V B3
LIMB 3
Obstacle B
V B2
Obstacle A
R 2
V A2
LIMB 2

Fig. 4. (a) A 2D Cartesian space is used as the workspace with an arm that contains three joints. The vector from the endpoint of the arm (endpoint of limb 3) to the target is $\boldsymbol{v}_{\text{target}}$, which is a 2D vector. The diagram illustrates the obstacle-to-limb vectors $\boldsymbol{v}_{\text{in}}$ which represent the minimum distance between the obstacle and the limb. In the case of Limb 3, $\boldsymbol{v}_{o3}$ is the normal vector from the point on the obstacle closest to Limb 3. In the case of Limb 2, the obstacle is farther than the endpoint of Limb 2; thus $\boldsymbol{v}_{o2}$ is the closest point from the obstacle to the endpoint of Limb 2. (b) Diagram illustrating the scaled obstacle avoidance vectors from Obstacle A, $\boldsymbol{v}_{A2}$ and $\boldsymbol{v}_{A3}$, and Obstacle B, $\boldsymbol{v}_{B2}$ and $\boldsymbol{v}_{B3}$, for Limb 2 and Limb 3 are shown here (see Eq. (3)). The resultant obstacle avoidance vectors $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$ for each limb are shown in black (see Eqs. (4) and (5)).

distance from each limb n to the obstacle o. This factor makes $\mathbf{v}_{\text{on}}$ have unequal lengths, as shown in Fig. 4(b).

The inverse of the minimum distance was chosen as the principal method of scaling the $\mathbf{v}_{\text{on}}$ vectors for simplicity. Other functions of distance such as an exponential can also be employed. In this study, $0 < k_0 < 1$ is a user-defined constant, with increasing values causing greater avoidance of obstacles. k_0 is multiplied by the distance to the target to balance the influence of obstacle avoidance with that of the target for the change in joint angle.

Obstacle avoidance for a given limb depends on the position of the limb in the kinematic chain. The endpoint of last limb m in the chain $\mathbf{R}_m$ is computed as

$$\mathbf{R}_m = \sum_{o=1}^{l} v_{om}, \tag{4}$$

where l is the total number of obstacles. The endpoints of all the other limbs n are iteratively calculated moving back from the last limb m inwards as follows:

$$\mathbf{R}_n = \mathbf{R}_{n+1} + \sum_{o=1}^{l} v_{\text{on}}, \tag{5}$$

where $n = m - 1, \ldots, 1$. Fig. 4(b) provides an example using two obstacles that shows scaled obstacle-to-limb vectors and the total obstacle avoidance vectors for the limbs. Note that each limb has different obstacle avoidance vectors based on the limb and obstacle configurations.

The DIRECT controller joint angle update for each joint n is

$$\theta_n^t = \theta_n^{t-1} + \mu \Im_n (\mathbf{v}_{\text{target}} - \mathbf{R}_n)\delta t, \tag{6}$$

where θ_n^t is the current joint angle, t is time elapsed from movement initiation, μ is a scalar which represents movement gate opening from the basal ganglia, a bio-inspired parameter, and δt is a time integration factor. The basal ganglia is a complex subcortical structure in the brain that provides the means for graded voluntary choice of alternate actions (Tsuji, Morasso, Shigehashi, & Kaneko, 1995). The choice of μ in this model is a simple approximation to the actual neural dynamics of the basal ganglia (Brown, Bullock, & Grossberg, 2004; Dranias, Grossbegr, & Bullock, 2008). In this study, the change in joint angles is based on the direction to the target *in addition to* the change in directions to avoid obstacles where $\Im_n$ is a row vector from the inverse of the learned Jacobian matrix $\mathbf{J}$ within the context of the current joint configuration $\boldsymbol{\theta}$. The number of elements in $\Im_n$ corresponds to the dimensions of the task space. In this study, $\mathbf{J}$ was estimated in an online fashion as a map between $\boldsymbol{\theta}$ and $\mathbf{v}_{\text{target}} - \mathbf{R}_n$ using the Fuzzy ARTMAP learning algorithm during action-perception cycles.

DIRECT is able to successfully execute control of a redundant robot under four altered conditions: joint lock, longer limbs, larger body size and a rigid tool added to the last joint at a fixed angle. The Fuzzy ARTMAP approach to the inverse model solution augments the original DIRECT model (Bullock et al., 1993) to successfully control the robot under these altered conditions. The performance of the DIRECT-ROAD controller for a given obstacle and target configuration is shown in Fig. 5. Computer simulations show that the DIRECT-ROAD for articulated redundant robots preserves these properties of the original DIRECT model.

A novel aspect of the approach described in this paper is that it modulates the elements of the controller trained in *obstacle-free space* via perceptual influences of obstacles by utilizing the directions of the obstacles to limbs to enable the robot arm to avoid obstacles. The DIRECT-ROAD controller retains the robustness and fault-tolerant properties of the original DIRECT controller. A problem with reactive controllers however is that they cannot avoid collisions with obstacles in more complex

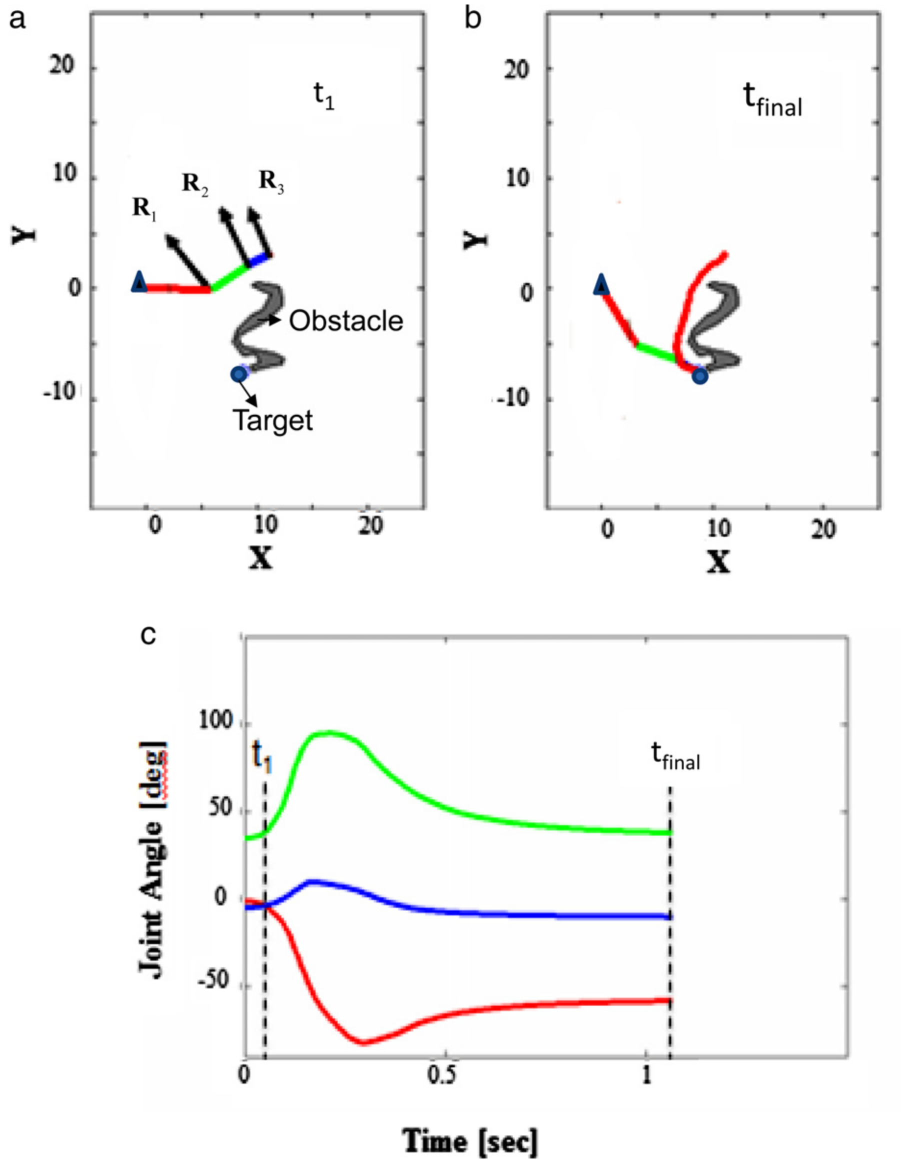

Fig. 5. (a) An early stage in the DIRECT-ROAD controller trajectory is illustrated here. For clarity, the triangle represents the fixed shoulder joint location at origin (0, 0). The vectors $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$ represent the obstacle avoidance vectors for the three limbs of the robot. (b) Trajectory for successful completion of movement to target. (c) Joint angles for entire movement trajectory. Colors correspond to limbs (e.g., red indicates the joint angle at the shoulder, or first limb.) The wiggly gray shape is the obstacle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

obstacle configurations such as the local minima configuration described in the next section.

2.4. Local minima with reactive obstacle avoidance

In the DIRECT-ROAD model, obstacle avoidance vectors are formed from the configuration of obstacles in the environment and the positions of the limbs. These vectors are applied to the limbs of the robot to influence the changes in joint angle directly and independently for each limb. Consequently, certain configurations of obstacles in the environment create obstacle avoidance vectors which prevent the robot from reaching the target. This situation can occur even with a clear path to the target around an obstacle, as illustrated in Fig. 6(a). The path to the target from the initial configuration of the robot arm requires a circuitous route around the obstacle. The arm moves initially from the lower left toward the target in the upper right. Since the obstacle is symmetrical, the obstacle avoidance vectors for the limbs trap it in a local minimum because it is driven to the target precisely in the direction it must avoid.

2.5. Novel controller for reaching amidst obstacles with planning

In order to get around the obstacle, a process which actively plans trajectories must be implemented. A potential approach to solve this problem is inspired by human movement planning process (Cohen & Rosenbaum, 2004). This study adapts these ideas in planning by introducing the concept of mental rehearsal, or lookahead planning, of movements, as described below. These rehearsal movements in combination with the DIRECT-ROAD controller offer a solution to escape local minima situations. The initial training process is identical to the process in Bullock et al. (1993) where the robot arm babbles in space to learn forward and inverse models of kinematic control. The perceptual influence of local obstacles introduces constraints for limb movements in a manner that can help avoid obstacles in simple configurations. The planning process is introduced within the same control framework, as shown in Fig. 6(b). Here the controller is provided with new perceptual information in the form of via-points or intermediate targets around obstacles in its environment. The Fuzzy ARTMAP

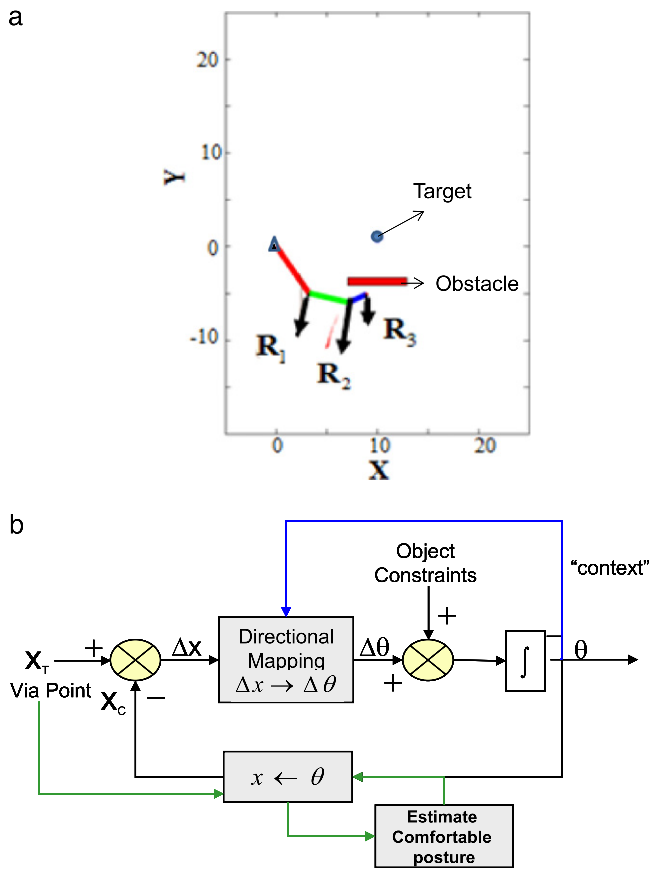

Fig. 6. (a) The obstacle avoidance paradigm applies constraints to the joint angles so that the arm moves toward the target but away from the direction of obstacles. The blue triangle represents the fixed shoulder joint location at (0, 0). The example shows the arm with the obstacle avoidance vectors $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$ overlaid on the endpoint of each limb. The direction of the vector from the endpoint to the target is nearly opposite of that of $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$ vectors on the limbs. Though a clear path around the obstacle exists, the application of such obstacle avoidance is purely reactive. (b) The novel control architecture that incorporates planning based on perceptually estimated via-points is shown with the green arrows representing the modifications to the DIRECT controller. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

forward model evaluates potential via-points, and the inverse model is leveraged to mentally rehearse trajectories using the DIRECT-ROAD controller. If the robot reaches the target while maintaining a certain minimum distance from obstacles along the entire path, the mental rehearsal is deemed successful, and is selected for movement execution. If these conditions are not met, a more deliberate planning process is initiated.

2.6. *Planning process*

Movement plans to escape local minima require paths around the obstacles. To generate these paths, the DIRECT-ROAD controller was used to rehearse movements to intermediate targets, or via-points, and paths from the initial point to via-points. The generation of via-points relies on visual perception of the environment and objects within it. In this study obstacles are salient elements in addition to the target in the visual scene. It has been hypothesized that such elements receive a large share of the cognitive processing during perception, and in spatial terms these elements form an attentional shroud (Cao, Grossberg, & Markowitz, 2011; Fazl, Grossberg, & Mingolla, 2009; Foley, Grossberg, & Mingolla, 2012; Grossberg, 2007, 2009). An attentional shroud is an object-fitting distribution of spatial attention that enhances object-based processing. This concept is employed in the model to define attentional shrouds encompassing obstacle objects, as illustrated in Fig. 7(a). The increased visual attention produces an enhanced spatial perception of the localized area, and may cause the robot to actively model the obstacle (and its occluded volume), or recall the obstacle volume from memory.

The obstacle volume model, or recalled model, allows the robot to then generate points in the space around the obstacle model in Fig. 7(b). These points serve as via-points for the robot. They represent locations near the obstacle that may provide a path around it. Some via-points are directly accessible by visual perception; however, others may implicitly rely on the model, or recalled model of the obstacle, and place points in areas that are occluded from the robot's view. A bounding box is placed around the obstacle with the assumption that the obstacle is symmetric and occluded dimensions are equal to visible dimensions, and points are sampled on the bounding box to generate first approximation via-points. Points which are known to be inaccessible *a priori*, for example because they are beyond the span of robot arm, are not included (Fig. 7(a)).

The via-points generated from the attentional shrouds are used to seed the mental rehearsal process. For redundant robot arms, a via-point has many associated postures, or joint configurations, that are learned using the forward model and that potentially place the arm near the via-point. Via-point and end-effector locations corresponding to postures that are associated with a three-DOF planar arm are shown in Fig. 7(c). The postures that are recalled must be checked for erroneous placing of the arm close to or within the volume of an obstacle. The remaining postures serve as candidates for each via-point. In this study, a single posture is selected from all the candidates to test for a via-point in the planning process. However, it is possible to test more than one candidate posture during planning. The selection criterion is based on scores for each candidate posture based on two metrics.

(1) Comfort of the arm in joint space:

$$S_{\text{comfort}} = \sum_{i=1}^{m} \left(\theta_i - \overline{\theta}_i\right)^2, \tag{7}$$

where θ_i is the joint angle for joint i and $\overline{\theta_i}$ is the mean joint angle calculated using the joint angle range for joint i.

(2) Distance of the robot's endpoint when realizing the candidate posture from the via-point:

$$d_1 = \left\lfloor x_{viapoint} - x_{posture},\ y_{viapoint} - y_{posture}, \ldots, \right\rfloor. \tag{8}$$

These two metrics are combined to form a score for each candidate posture:

$$S = \frac{1}{S_{\text{comfort}} \left\| d_1 \right\|}. \tag{9}$$

The candidate posture with the highest score is selected for a given via-point in subsequent mental rehearsals during planning. The winning candidate postures are shown with the via-points in Fig. 7(d).

The next step in the planning process is to mentally rehearse paths using the candidate postures from the via-points. It is possible to test them in a random order. However, a heuristic can reduce the number of tests necessary to realize a successful path on average. The candidate postures that place the robot arm's end-effector "in between" the initial point and target are selected for earlier testing. Via-points and candidate postures are numbered according to this heuristic in Fig. 7(d). Note that these distances are Euclidean and do not consider obstacles. A scheme that considers the distance to circumvent the obstacle, or other factors, may provide a better heuristic for ordering and further reduce the number of mental rehearsals. The end-effector position at each selected posture serves as the via-point for subsequent mental rehearsals. Henceforth these postures are referred to as *posture via-points*. Examples of posture via-points are illustrated in Fig. 7(d) as blue asterisks.

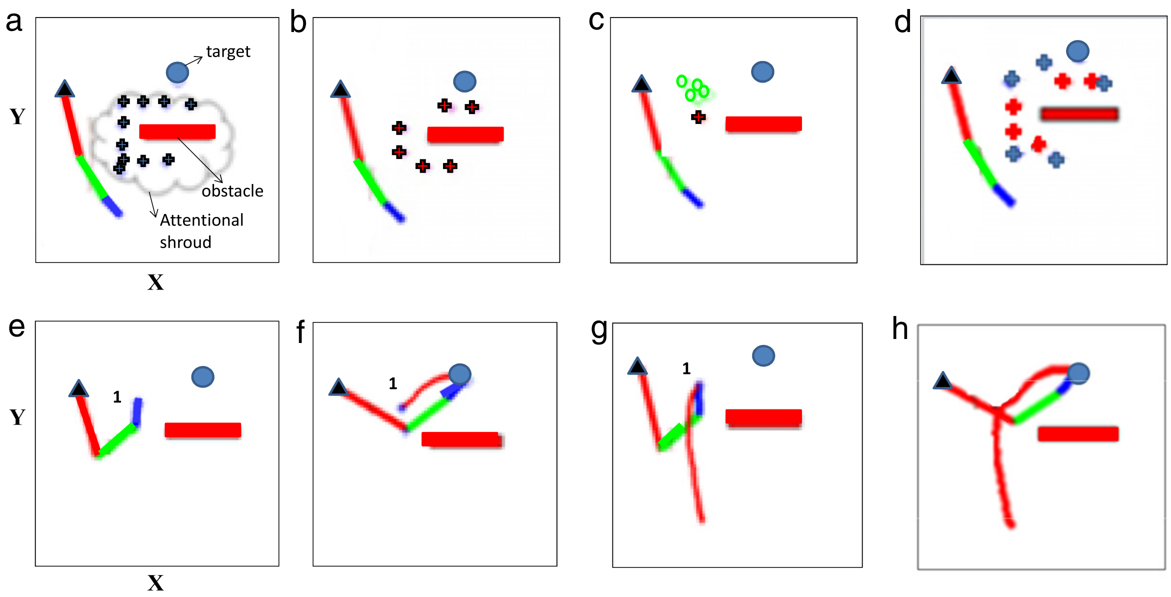

Fig. 7. (a) Visual perception of the environment identifies and segments obstacles in the path to the target (blue circle). The blue triangle represents the fixed shoulder joint location while the small cross marks represent the via-points. Attention is directed toward such objects, and produces a "shroud" that envelopes the object. (b) The potential via-points (shown as red crosses) generated from the attentional shroud undergo an associative procedure. The resulting associations recall the unique areas of space (green circles) that were learned, and serve as a subset of the potential via-points. (c) A selected via-point is examined for candidate postures. The end-effector locations of these postures are displayed as green circles. (d) The via-points, and the associated recalled postures, are ordered for testing the possible paths from them to the target, shown by the numbers. (e) Robot arm configuration at the first selected via-point posture during mental rehearsal. (f) Trajectory in mental rehearsal from posture in (e) to target. (g) Trajectory in mental rehearsal from initial point to first selected via-point posture. (h) Entire movement execution trajectory from initial point to target using selected via-point posture. The partial outline of the dashed circle around the via-point indicates the neighborhood for target switching, where the radius is α_0 from Eq. (11). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.7. Mental rehearsals and movement execution

The selected posture via-point is used to mentally rehearse paths between the target and initial point using the planning process. Fig. 7(e) shows the configuration of the arm at the first selected posture via-point being tested for a path to the target. Fig. 7(f) shows that a successful path exists from a posture via-point to the target, and Fig. 7(g) shows that a successful path exists from the initial point to the same posture via-point. The mental rehearsals during planning continue until there is a successful path from initial point to target, or until all via-points are exhausted (note that new via-points could be generated for testing). A final movement plan is formed based on the successful paths in the various stages of planning.

Once a final movement plan is selected, the robot is ready to execute the movement. It is possible to execute the planned movements from initial point to via-point(s) and from via-point(s) to the target as separate discrete movements. The DIRECT model employs a form of velocity control and includes a factor based on the distance to a target for a given movement:

$$\mathrm{GO} = f\left(\mathbf{v}_{\mathrm{target}}, t, G_0\right) = G_0 \left\|\mathbf{v}_{\mathrm{target}}\right\| \frac{t^2}{1+t^2}, \qquad (10)$$

where $\mathbf{v}_{\mathrm{target}}$ is the vector from the robot arm's endpoint to the target, t is elapsed time from movement onset, and G_0 is a user-defined constant (G_0 values between 1 to 200 was used to simulate various speeds of movement). This method produces Gaussian velocity curves for joint angles that are analogous to those observed from humans during reaching (Bullock et al., 1993). In addition, a single "user"-defined parameter can control the peak velocity of the movement without affecting the trajectory path.

In the present model, a single, continuous movement from the initial point to the target is produced without the need for stopping at via-points. This requires a modification of the velocity control term GO, and switching from the via-point to the target by the controller during the movement:

$$\mathrm{Go}_{\mathrm{switch}} = d_1\alpha(d_1) + (\alpha_0 - d_2\alpha(d_1))G_0\frac{t^2}{1+t^2}, \qquad (11)$$

where d_1 is the distance to the via-point (as in Eq. (8)) and d_2 is the distance to the target. The function α used in this work was linear compression ($\alpha = x$ for $x < x_0$; $\alpha = \alpha_0$ for $x = x_0$). This can however be changed to realize different behaviors for the adaptive control of velocity when switching targets (e.g., faster than linear). The parameter $\alpha_0 = 3$ in all simulations. The complete movement execution path trajectory is shown in Fig. 7(h).

The operation of the planning process and the DIRECT-ROAD dynamics in another obstacle and target configuration is illustrated in Fig. 8. The local minimum can be seen in the first mental rehearsal from the initial point to the target in Fig. 8(a). The planning process is shown in Fig. 8(b). The movement execution of the final plan along with target switching to produce a continuous movement is illustrated in Fig. 8(c).

Unlike simulation studies described thus far, where targets were chosen to be points in space, experiments with real robots and environments require visual processing of scenes to acquire information about the target objects. This process is now described in detail in the next section.

3. Overall system architecture

The system for implementing the DIRECT-ROAD model on real robotic platforms consists of three main modules: the visual perception module, the DIRECT-ROAD controller and the robot movement execution module, as shown in Fig. 9. The visual perception module takes in stereo images from the stereo camera and processes them using an object recognition model (Section 4.1). The object recognition model is built using templates from an object database. In addition to the object recognition model, there is an end-effector marker detector based on the color and shape of the marker. There are three servers (the visual perception server, the

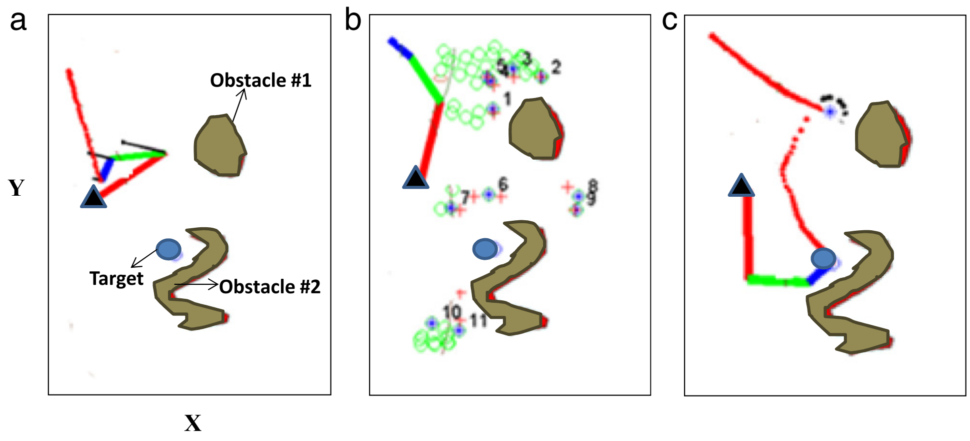

Fig. 8. (a) Final arm posture and trajectory during mental rehearsal from initial point to target shows that the DIRECT-ROAD controller trajectory falls into a local minimum for these obstacles (shown in gray) and target (shown as blue circle). Note that the initial arm configuration can be seen in Fig. 8(b). (b) Selected via-points (crosses), all associated postures (green circles), winning candidate postures (blue asterisks) and order of via-point posture testing for mental rehearsals. (c) Execution of final movement plan using target switching with a via-point posture (the trajectory of the end-effector is shown as the red curve). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

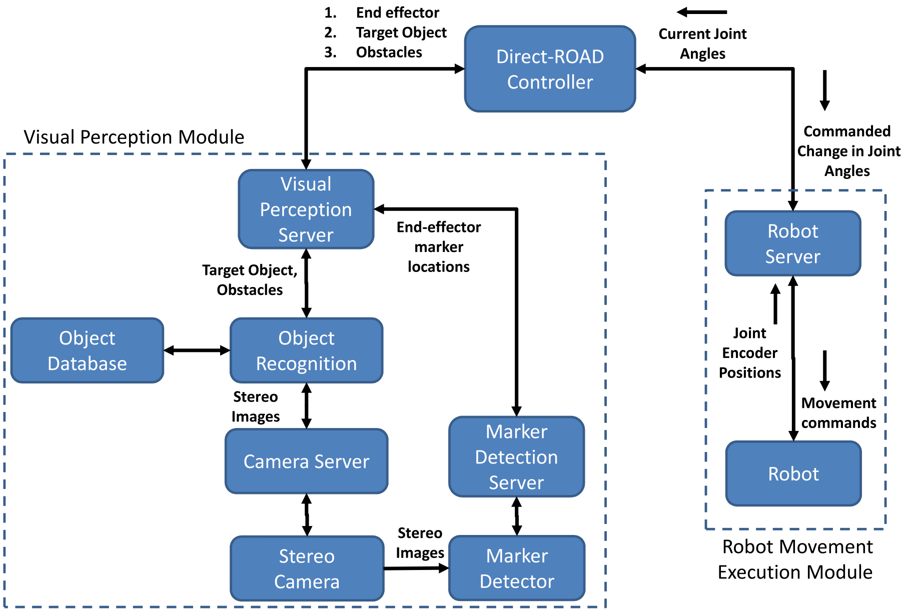

Fig. 9. The overall system diagram is shown here with the various interactions between the visual perception module, DIRECT-ROAD controller and the robot movement execution module.

marker detection server and the camera server) that constantly send in requests for new inputs from their respective modules. For example, the visual perception server requests for information about the end-effector location, the target object location and the location of obstacles to the object recognition model and the marker detection server. These three inputs are fed from the visual perception server to the DIRECT-ROAD controller. The DIRECT-ROAD controller also receives the current posture of the robot arm in the form of joint angles for all its joints. Using these inputs, the DIRECT-ROAD controller produces the change in joint angles using the model described in the previous section. The commanded change in joint angles is then combined with the current joint angles by the robot server. The robot server then sends movement commands to the joint encoders of the robot arm. The encoder readings are also sent back to the robot server to provide proprioceptive feedback about the current robot arm posture to the DIRECT-ROAD controller.

4. Experiments

Experiments in learning and control using DIRECT were first performed on the ST Robotics R17 platform. Fuzzy ARTMAPs were used to learn the forward and inverse models for four DOFs on the robot (waist rotate, shoulder swing, elbow swing, and hand swing) in 3D task space. Additionally, experiments in learning and control using DIRECT and DIRECT-ROAD were also performed on a custom robot platform developed by the Shadow Robot Company, consisting of a Shadow Hand mounted on a custom pneumatically driven arm containing tendon-driven antagonistic pairs of air muscles. Forward and inverse models were learned for six DOFs (waist rotate, shoulder swing, elbow swing, elbow rotate, and two wrist DOFs for flex/extension and pro/supination) and six DOFs in task space corresponding to the end-effector position and orientation of the outer palm. Section 4.2 provides the details of the joint angle workspace range for each robot.

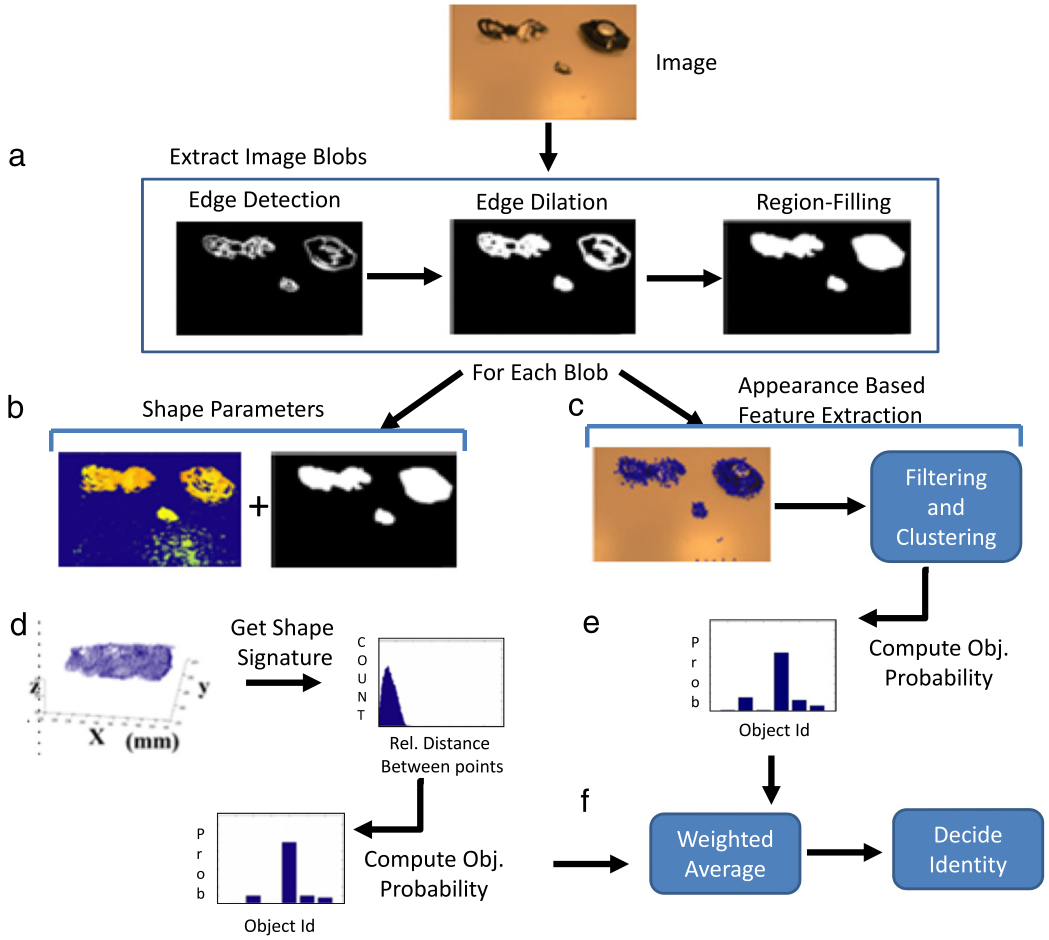

Fig. 10. A flowchart demonstrating the testing process in the object detection system is shown here. An image taken by the Tyzx stereo camera system. In (a), this image is processed to obtain image blobs on which further object detection techniques will be applied. (b) The depth map obtain is processed to remove outliers and to fill in 'holes' using interpolation. (c) Interest points or features are extracted from 2D images and then analyzed to remove outliers (filtering) and then clustered to represent objects. (d) A probability distribution based on shape is computed. (e) A probability distribution is computed to determine the identity of the cluster based on the features. (f) The final decision on the identity of the object is computed based on both appearance and shape.

The visual workspace was viewed by a Tyzx stereo camera, which provided depth maps along with 312 × 500 RGB images of scenes monitored in real time. An example image is shown in Fig. 10 with the corresponding depth map in Fig. 10(b). Images were subsequently processed online to locate the end-effector for control purposes and also for object detection of the target objects within the scene. An object library was constructed using the procedure in Section 4.1. Once objects were identified, the experimenter selected an object from a list of identified objects as the target for a given reaching task.

4.1. Visual perception module

In order to reach for 3D targets, the first task is to obtain the position of the 3D targets with respect to a camera reference frame, followed by identification of the target object from all other objects in the visual scene. The 3D location of targets were computed from stereo-pair of images and the associated depth maps obtained in real time using on-board processing on the Tyzx stereo camera system. The 2D image on the left is subjected further to blob extraction techniques (Fig. 10(a)) and the depth map is refined using interpolation (Fig. 10(b)).

In order to identify an object based on its appearance in the image, the object recognition system makes use of a supervised learning method. The training consists of using an interest point detector known as Speed-Up Robust Features or SURF (Hebert, Andreas, Tinne, & Van Gool, 2008). SURF is a Hessian-matrix-based detector which finds points of interest on the image that is generally scale, rotation and illumination invariant. This descriptor contains the distribution of Haar-wavelet responses around the neighborhood of each point of interest. These points of interest typically are edges and texture points on the image of an object. The SURF features are extracted from color images by treating each channel of the color image as a monochrome image, applying one of the interest point detectors, and then taking a weighted average of all three descriptors for a given pixel location. An example of this is shown in Fig. 10(c).

In this study, a shape statistic was then used to extract shape signatures out of the 3D visual data. This statistic based on Osada, Funkhouser, Chazelle, and Dobkin (2001) is computed based on the following steps.

(1) Randomly select points on the surface of the object.
(2) Compute the Euclidean distance between all point pairs on the object surface.
(3) Histogram these distances.

This histogram represents a probability distribution of the frequency of 3D distances on the surface of the object (Fig. 10(e)). During the experiment or test time, the 2D features are extracted and then analyzed to remove false positives. Following this step, clusters of objects are extracted (based on *k*-shared nearest neighbors Jarvis & Patrick, 1973). For each cluster, the closest matching shape is extracted from a database of objects and the object in the current image is identified based on a weighted combination (Fig. 10(d)). The entire test process is summarized as shown in Fig. 10.

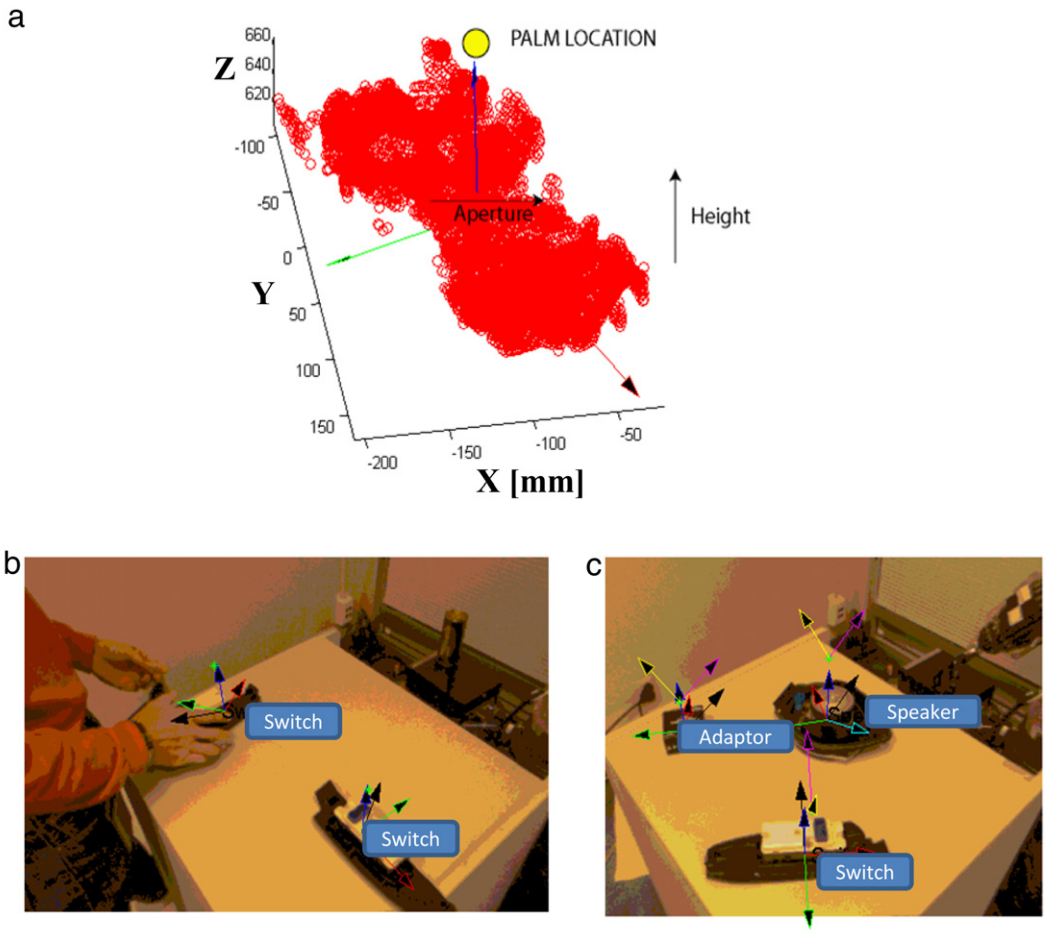

Fig. 11. An example of a "switch" object and (a) its 3D point cloud data from the Tyzx camera. The point cloud is also used to roughly estimate the aperture and height of the object on the table. (b) The estimation of the object's major and minor axes necessary for robot palm orientation before grasping. (c) Examples of object axes estimated for other objects and the same "switch" object in other orientations are shown here.

This vision algorithm offers the benefit of being robust to partial occlusions, discarding unknown objects, and demonstrating object pose invariance.

Along with the identity and the location of the target object, the vision module also estimates the target location and pose for the palm to grasp the object. To accomplish this, the visual module estimates the normal vector of the object (with respect to the surface it rests on) and the centroid of the palm is translated along that normal by 10 cm. This step is necessary to ensure that the fingers of the hand do not collide with the surface of the table. To calculate the object normal, it first uses 3D point cloud data of the object that is obtained from the Tyzx camera. This 3D data is then projected onto the table axes and a Principal Component Analysis (PCA; see Bell & Sejnowski, 1997) is performed on the data. The PCA provides the major and minor axes of the object so that the robot hand can orient itself accordingly. Examples of the point cloud data for a "switch" object and the estimated object axes are shown in Fig. 11(a) and (b). The extracted axes for a few other example objects are shown in Fig. 11(c). For power grasping of the object, the 3D point cloud data can also be used to estimate an aperture and height for each object (Fig. 11(a)).

4.2. R17 robot babbling and reaching experiments without obstacles

Motor babbling was performed on the R17 robot, with three major postures and nine micro-babbling positions corresponding to 0°, ±3°, ±6°, ±9°, and ±12° for each joint ($n = 4$). This results in $9^4 = 6561$ micro-babbling movements at each major posture, and separate inverse models were learned for each major posture. The workspace was viewed by the Tyzx stereo camera, and end-effector locations were extracted from each image to provide data to the Fuzzy ARTMAP learning of forward and inverse models. An example of joint angle data for two DOFs during motor babbling for three major postures are shown in Fig. 12(a). The resulting positions in Cartesian space (visual coordinates) and end-effector directions from micro-babbling at a single major posture are shown with their distributions in Fig. 12(b) and (c). This process of motor babbling produced fine-grained inverse models containing many redundant solutions (Fig. 12(e)).

During execution, the inverse model with the closest match to the current posture (computed based on the Euclidian distance between joint angles) was recalled and used for Fuzzy ARTMAP matching to generate solutions ($\Delta\boldsymbol{\theta}$). The chosen solution was one with the direction ($\Delta\mathbf{x}$) closest to the desired direction and which did not place a joint closer to singularity. The robot was able to reliably converge within 5 cm of target locations placed on table in a 60 cm × 120 cm area. The initial postures were varied and did not need to be any of the trained postures from babbling. However, they generally needed to be within 10°–15° of a trained posture for success. Subsequently, the motor babbling data was subsampled by a factor of ten at each major posture and found to perform with the same accuracy. An example of reaching using the DIRECT with Fuzzy ARTMAPs is shown in Fig. 12(f).

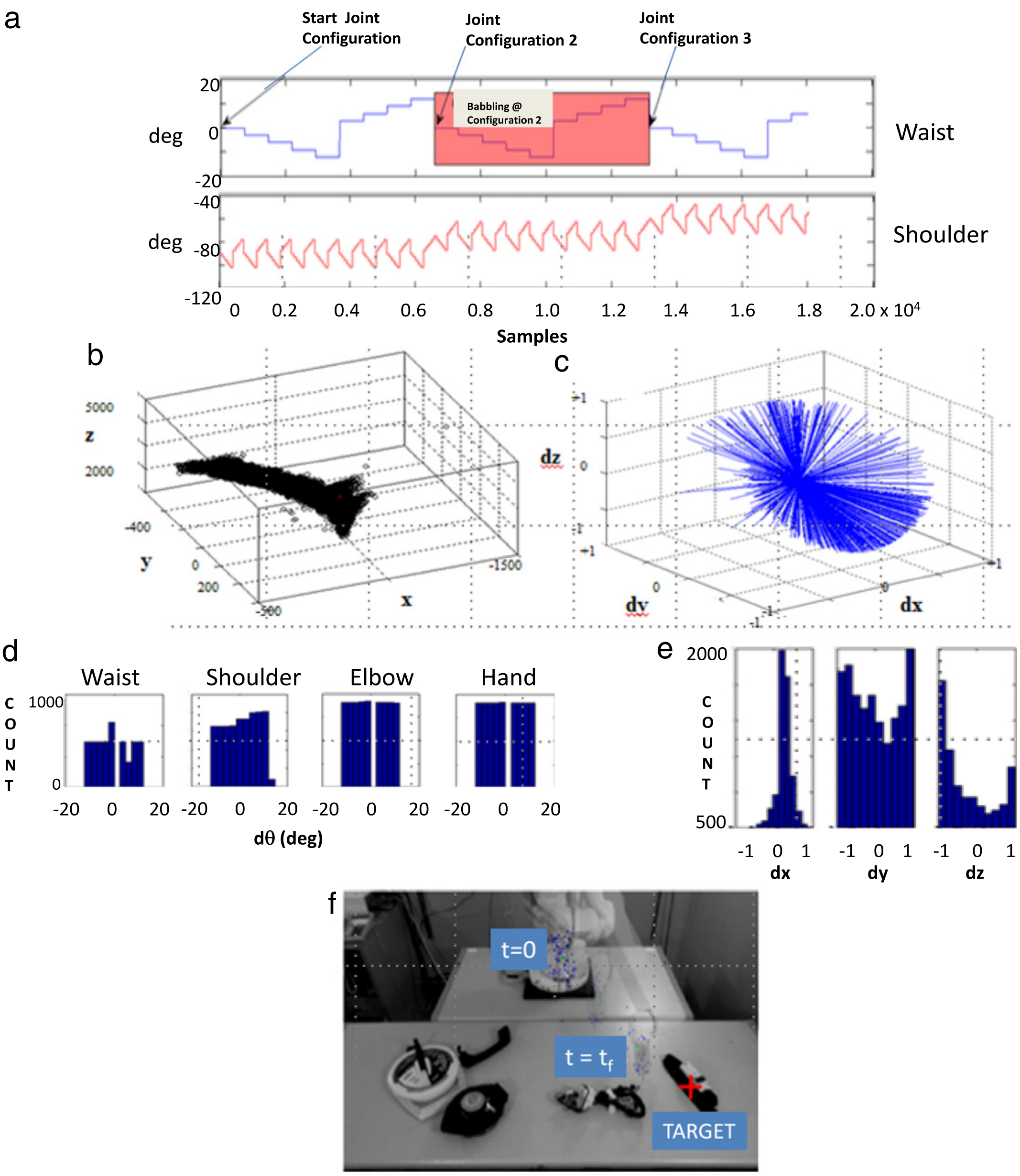

Fig. 12. Motor babbling data and reach experiment on R17 robot platform. (a) Joint angle data for structured motor babbling for two DOFs (elbow and hand not shown for simplicity). Nine micro-babbling positions for each joint (0°, ±3°, ±6°, ±9°, and ±12° for four DOFs) were recorded for each major posture to learn an inverse model for it. The red box highlights micro-babbling at the second major posture, which is further displayed in (b)–(e). (b) End-effector locations in visual coordinates using the visual perception algorithm to recognize the end-effector for the micro-babbles. (c) End-effector directions were generated for the micro-babbling to learn the inverse model at the major posture. (d) Histogram of joint angle changes ($\Delta\boldsymbol{\theta}$) from micro-babbling. (e) Histogram of end-effector directions ($\Delta\mathbf{x}$) from micro-babbling. (f) Trajectory of a movement to a target (red cross) shown at three time points. Blue dots indicate features identified for the end-effector to compute its location. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.3. Reaching to 3D targets with obstacles using a Shadow robot

In addition to the R17 robot experiments for reaching, similar experiments were performed on a Shadow robot platform with redundant degrees of freedom. The platform was used to reach, grasp, and manipulate a target to place it in a defined pose. In this experiment, the amount of motor babbling was reduced to a single major posture and micro-babbles at 0° and ±5° for six DOFs (216 micro-babbling movements). DIRECT and Fuzzy ARTMAPs were used to control both the end-effector location and palm orientation in order to perform power grasps for target objects.

The workspace was viewed by a 3D camera, and the location and orientation for the end-effector were estimated frame by frame to control the robot. The identity and pose of target objects, including the major and minor axes of the object for palm orientation, were estimated using the algorithms described in Section 4.1. The location of the object to be grasped was estimated using the 3D point cloud from the Tyzx camera. In order to estimate the palm orientation of the robot, three visual markers were recognized from the captured images. An example of such markers is shown as green, red, and blue concentric circles in Fig. 13. Two vectors $\mathbf{p}$ and $\mathbf{q}$ were computed from the marker locations, as shown in Fig. 13(a), where $\mathbf{p}X\mathbf{q}$ was used to compute the palm normal vector. The vector $\mathbf{p}$ was also used to control the palm orientation. In all, a 6D vector was controlled using three DOFs for palm orientation. Two separate controllers were used for palm position (3D) and palm orientation (6D). Grasping was performed open loop once the hand was in the desired location and pose

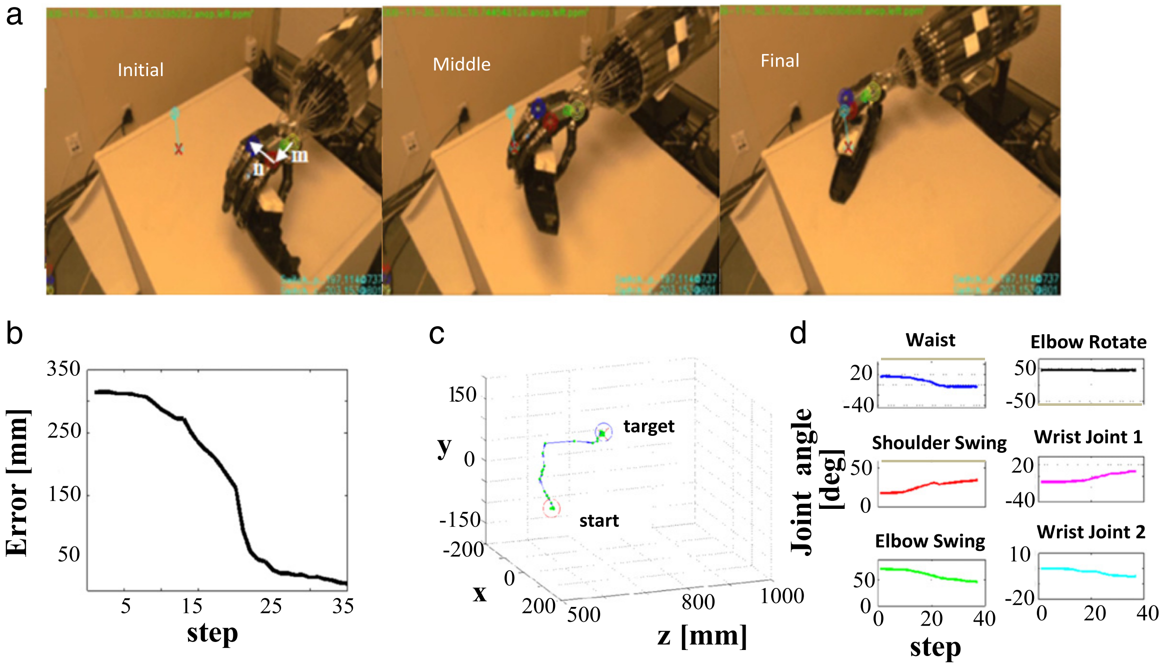

Fig. 13. Manipulation of a target object using DIRECT with Fuzzy ARTMAPs. (a) Snapshots from the camera showing the trajectory of the Shadow robot starting from when the object was grasped (left; initial reach and grasp not shown), during manipulation (middle), and final placement (right). (b) End-effector location error shown for the reach trajectory. (c) End-effector location (green dots on blue line) in visual coordinates, where the starting location is shown with a red circle, and the target location is shown with a blue circle. (d) Joint angle trajectories shown for six DOFs to control end-effector location and palm orientation (elbow, wrist joints 1 and 2). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

($\leq$ 10 mm for location, ± 0.2 normalized direction error for palm orientation).

The workspace for the robot can be seen in Fig. 13, and the single major posture used for training placed the end-effector at the center of it. The amount of motor babbling performed was adequate for the robot to reach from any posture within the workspace/camera view to a target anywhere on the table. The trajectory for placing a grasped object in a new target location and pose is shown in Fig. 13(c). The target location and pose were specified by first placing an identical object in the scene, estimating its location and pose using the same visual perception algorithms, and then removing it before any reach movements were executed.

DIRECT-ROAD is capable of controlling the amount of avoidance performed for obstacles using a single parameter, k_o. Manipulating k_o controls obstacle avoidance both in simulation and for experiments conducted on the Shadow robot platform, as illustrated in Fig. 14. Obstacles and objects were detected at the beginning of an experiment using visual perception algorithms. This informed the controller of the target location and pose for the initial reach and grasp, final target location and pose, and location of the obstacle. DIRECT-ROAD computes avoidance at each step based on the current pose of the robot with respect to the obstacles. In this experiment, obstacle avoidance was performed by using the points on the static obstacle that were closest to the end-effector at each step, though it could readily be applied to a moving obstacle with this approach as well. Example trajectories are shown for different values of k_o in Fig. 14 for the robot performing an initial reach and grasp movement while avoiding an obstacle placed on the table. The robot was able to reach within ± 1 cm of the object and grasp it while avoiding the obstacle using DIRECT-ROAD with Fuzzy ARTMAPs from a variety of initial postures spanning the workspace. Manipulation was achieved by controlling the palm orientation through three DOFs (elbow, wrist joints 1 and 2) continuously during the movement execution using the same learned models to place the object in the target pose.

5. Discussion

The robustness and fault tolerance exhibited by the real robot systems trained using the self-organizing neural model is akin to behaviors by biological systems wherein reliability and fault tolerance are cornerstones of performance. These key features ensure the survivability of biological systems when faced with previously unforeseen environments and disturbances. It seems that by learning the appropriate transform between various sensory modalities, robotic systems can also exhibit fault tolerance and robustness to unexpected changes. A similar transform was learned for reaching (Bullock et al., 1993) and saccading to 3D targets (Srinivasa & Grossberg, 2008). In Bullock et al. (1993), motor-equivalent reaching and tool use of a three-DOF planar arm were demonstrated. There the arm was able to adapt to new disturbances such as restrictions to joint rotations, tool use, and prism rotations of images. They also point out that their model's properties match monkey neurophysiological data and human psychophysical data. Similar results were obtained for the control of a highly redundant head–neck–eye simulated robot system (Srinivasa & Grossberg, 2008). The results from this study further point to the utility of using a DIRECT-style learned transform from desired movement directions into joint rotations for the control of multimodal interacting systems with redundant degrees of freedom in three dimensions.

The robot and camera system presented in this paper utilizes a basic mechanism for learning that is derived from action-perception cycles. These cycles provides self-consistent movement commands that activate correlated visual, spatial and motor information that are used to learn an internal coordinate transformation between vision and motor systems. A neural correlate for this coordinate transformation can be found in the parietal cortex (Snyder, Batista, & Andersen, 2000), where the 3D position of a reach goal is represented using visual coordinates during movement planning (Bhattacharyya, Musallam, & Andersen, 2009). While this mechanism is established in primates, no claim is made that the underlying neural architecture in this work replicates brain function.

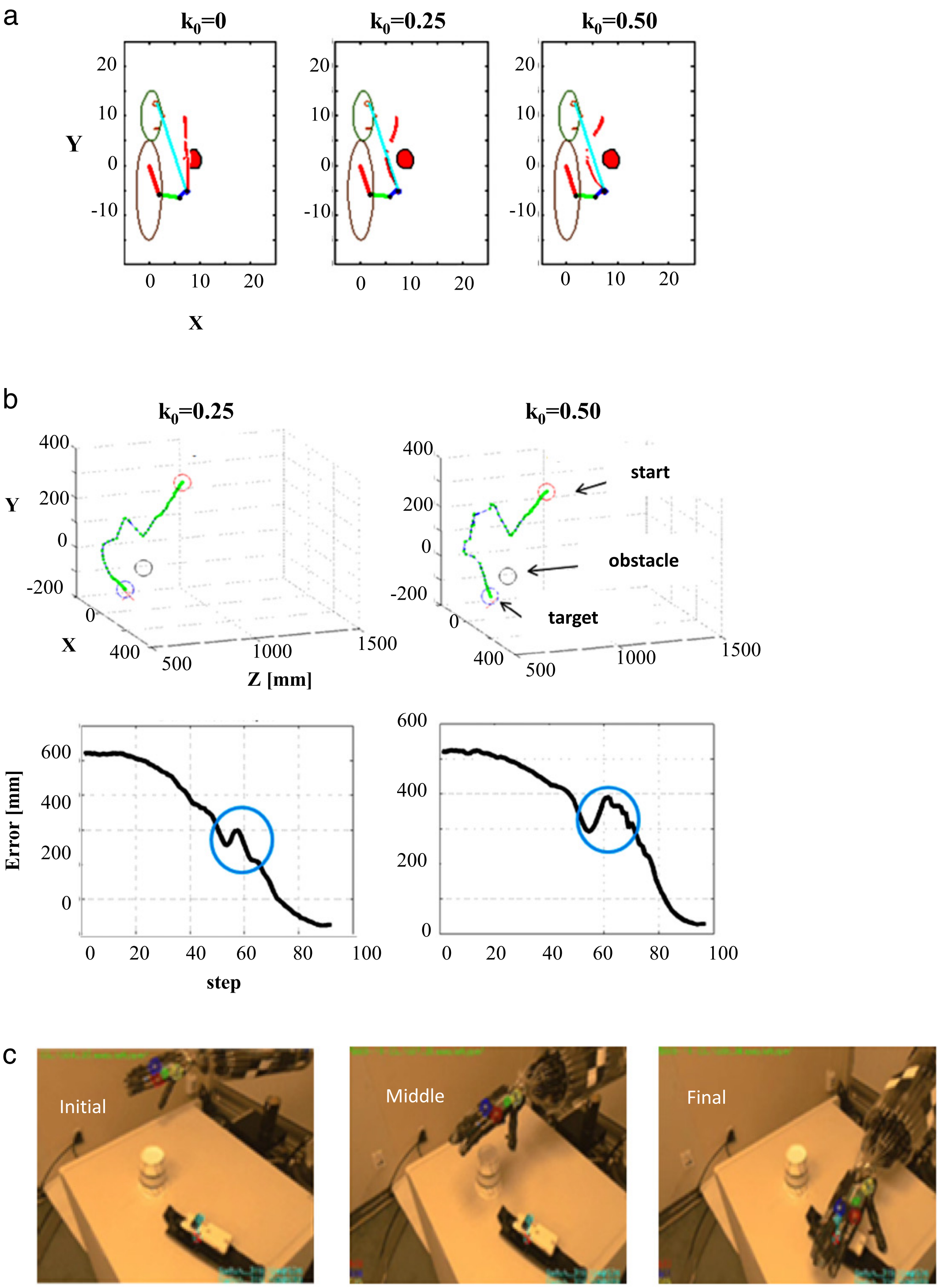

Fig. 14. DIRECT-ROAD with Fuzzy ARTMAP for reaching with obstacle avoidance. (a) The amount of avoidance can be controlled using a single parameter, k_o, in DIRECT-ROAD; see Eq. (4). For $k_o = 0$, no avoidance is used, and the robot trajectory is through the obstacle. Avoidance increases with larger values of k_o, and is shown for $k_o = 0.25$ and $k_o = 0.5$. Obstacles and trajectories are shown in red, with the minimum distance to the obstacle observed over the whole trajectory shown underneath the simulations. (b) Results from two experiments showing Shadow robot reach movement end-effector trajectories and errors in visual coordinates for $k_o = 0.25$ and $k_o = 0.5$. The red circle shows the initial position, the blue circle shows the target position, and the black circle shows the cup obstacle position, with green dots showing the end-effector location in each iteration. The table was also considered an obstacle in this experiment; however is not shown in the diagram. Varying the level of avoidance changes trajectory and error profile during the reach, G_o parameters were the same for both experiments. (c) Snapshots from the camera as the Shadow robot reaches for the target object while avoiding the obstacles (cup and table) with $k_o = 0.25$. Early in the trajectory, the end-effector is drawn to the target on the table, and is unaffected by the closest obstacle points (all on the table) since the distance to obstacle points remains large. In the middle, the closest obstacle points are on the cup instead of the table, and they cause DIRECT-ROAD to control the arm to avoid the cup smoothly, and reach the target. Circles on the hand indicate localization of features on the end-effector to calculate location and palm orientation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The stereo camera processing steps leverage on-board computer algorithms while the visual perception of shape and appearance are also motivated by computer vision and image processing algorithms. The plan is to gradually replace these algorithms with more biologically motivated models (Fazl et al., 2009; Grossberg & Huang, 2009; Huang & Grossberg, 2010).

The approach in this study draws inspiration from biology based on various studies of motor systems in monkeys and humans. First, the DIRECT-ROAD controller is designed to map directions of targets to rotations of joint angles. The locations of targets in visual coordinates are known to be coded in the dorsal stream of visual processing (Goodale & Milner, 1992; Mishkin & Ungerleider, 1982). In later stages of the dorsal stream, the directions to targets are represented in the posterior parietal cortex in visual coordinates and are transformed into a variety of other coordinate systems such as body centered and hand centered during reach planning (Snyder et al., 2000). The final motor command is specified in hand coordinates to encode the direction of the hand to reach for targets, consistent with direction coding in the primary motor cortex (Cisek, Crammond, & Kalaska, 2003; Georgopoulos et al., 1986).

Second, the approach described in this paper combines direction-based control strategy with a posture-based control strategy (Desmurget & Prablanc, 1997; Graziano, 2006; Rosenbaum et al., 2001) during mental rehearsals. This is consistent with studies where the motor system is found to follow an optimal control hypothesis (Todorov & Jordan, 2002). This hypothesis postulates that there is no single, preferred parameter for motor control, such as direction or end posture. Instead, the parameters being controlled depend on the task being performed. For example, in hitting a nail with a hammer, the final position of the hammer head is of critical importance and is highly conserved, whereas other variables such as the exact trajectory of hand or speed of rotation of the arm joints are less important. For tasks of obstacle avoidance with simple obstacle configurations, the modulation of directions to targets seems to be adequate. However, for tasks with more complex obstacle configurations, the model switches to posture-based internal search, or mental rehearsals, for possible via-points that can be used to avoid obstacles. Elsinger and Rosenbaum (2003) have confirmed that humans use feedforward modeling of prospective movements to select end postures prior to overt movement production. In fact, neural coding in the posterior parietal cortex predicts future positions of the end-effector during movement control, consistent with the operation of a feedforward model (Mulliken, Musallam, & Andersen, 2008). This evidence has guided DIRECT-ROAD to multiple stages of reaching from movement and posture planning to movement production. Although the control model does not incorporate detailed neural mechanisms such as alpha–gamma compensation when contacting obstacles (Bullock, Cisek, & Grossberg, 1998) or spiking dynamics (Rieke, Warland, van Steveninck, & Bialek, 1997), its design principles are inspired by biological studies.

6. Conclusions

In this study, a novel kinematic controller was developed, inspired by human and animal behaviors that are capable of controlling redundant robot arms to avoid obstacles while reaching for targets. The core of the kinematic controller for trajectory generation is based on a self-organizing neural model that is capable of learning to reach 3D targets by self-generated vision and motor signals during action-perception cycles. The approach in this study integrates the process of trajectory generation and planning during obstacle avoidance, thereby making the process of reaching amidst obstacles very efficient. The controller was tested with real robotic platforms along with cameras to train the kinematic controller. This learning strategy seems to offer a trade-off between optimality in functioning within a limited scope or rigid environments versus flexibility and robust performance in a wide variety of environments. The proposed approach offers a possible solution to enable robots to reach for targets in complex environments in a fast, safe and reliable manner.

Acknowledgments

We would like acknowledge the support of a Collaborative Research grant CR07C202 at HRL for this research work. S.G. was supported in part by CELEST, an NSF Science of Learning Center (SBE-0354378), and by the SyNAPSE program of DARPA (HR0011-09-C-0001).

References

Amis, G., & Carpenter, G. (2007). Default ARTMAP 2. In *Proceedings of the international joint conference on neural networks*, IJCNN07. (pp. 777–782). Orlando, Florida: IEEE press.

Amis, G., & Carpenter, G. (2009). Self-supervised ARTMAP. *Neural Networks*, *23*, 265–282.

Baillieul, J., Hollerbach, J., & Brockett, R.W. (1984). Programming and control of kinematically redundant manipulators. In *Proc. of 23rd IEEE conference on decision and control* (pp. 768–774).

Barraquand, J., Kavraki, L., Latombe, J. C., Motwani, R., Li, T. Y., & Raghavan, P. (1997). A random sampling scheme for path planning. *International Journal of Robotics Research*, *16*(6), 754–774.

Bell, A., & Sejnowski, T. (1997). The independent components of natural scenes are edge filters. *Vision Research*, *37*(23), 3327–3338.

Bhattacharyya, R., Musallam, S., & Andersen, R. A. (2009). Parietal reach region encodes reach depth using retinal disparity and vergence angle signals. *Journal of Neurophysiology*, *102*, 805–816.

Brown, J. W., Bullock, D., & Grossberg, S. (2004). How laminar frontal cortex and basal ganglia circuits interact to control planned and reactive saccades. *Neural Networks*, *17*, 471–510.

Bullock, D., Cisek, P., & Grossberg, S. (1998). Cortical networks for control of voluntary arm movements under variable force conditions. *Cerebral Cortex*, *8*, 48–62.

Bullock, D., Grossberg, S., & Guenther, F. H. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience*, *5*, 408–435.

Cao, Y., Grossberg, S., & Markowitz, J. (2011). How does the brain rapidly learn and reorganize view- and positionally-invariant object representations in inferior temporal cortex? *Neural Networks*, *24*, 1050–1061.

Carpenter, G. A. (1997). Distributed learning, recognition, and prediction by ART and ARTMAP neural networks. *Neural Networks*, *10*, 1473–1494.

Carpenter, G.A. (2003). Default ARTMAP. In *Proceedings of the international joint conference on neural networks*, IJCNN'03 (pp. 1396–1401).

Carpenter, G. A., & Gaddam, C. S. (2010). Biased ART: a neural architecture that shifts attention toward previously disregarded features following an incorrect prediction. *Neural Networks*, *23*(3), 35–451.

Carpenter, G. A., & Grossberg, S. (2003). Adaptive resonance theory. In *The handbook of brain theory and neural networks, vol. 2* (pp. 87–90).

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, *3*, 698–713.

Carpenter, G. A., Milenova, B. L., & Noeske, B. W. (1998). Distributed ARTMAP: a neural network for fast distributed supervised learning. *Neural Networks*, *11*, 793–813.

Cisek, P., Crammond, D. J., & Kalaska, J. F. (2003). Neural activity in primary motor and dorsal pre-motor cortex in reaching tasks with the contralateral versus ipsilateral arm. *Journal of Neurophysiology*, *89*, 922–942.

Cohen, R. G., & Rosenbaum, D. A. (2004). Where objects are grasped reveals how grasps are planned: generation and recall of motor plans. *Experimental Brain Research*, *157*, 486–495.

Dagli, C., & Huggahalli, R. (1995). Machine-part family formation with the adaptive resonance theory paradigm. *International Journal of Production Research*, *33*, 893–914.

Desmurget, M., & Prablanc, C. (1997). Postural control of three-dimensional prehension movements. *Journal of Neurophysiology*, *77*, 452–464.

Dranias, M. R., Grossberg, S., & Bullock, D. (2008). Dopaminergic and non-dopaminergic value systems in conditioning and outcome-specific re-evaluation. *Brain Research*, *1238*, 239–287.

Elsinger, C. L., & Rosenbaum, D. A. (2003). End posture selection in manual positioning: evidence for feedforward modeling based on a movement choice method. *Experimental Brain Research*, *152*, 499–509.

Fazl, A., Grossberg, S., & Mingolla, E. (2009). View-invariant object category learning, recognition, and search: how spatial and object attention are coordinated using surface-based attentional shrouds. *Cognitive Psychology*, *58*, 1–48.

Fiala, J.C. (1995). Neural network models for motor timing and coordination. Ph. D. dissertation. Boston University.

Foley, N. C., Grossberg, S., & Mingolla, E. (2012). Neural dynamics of object-based multifocal visual spatial attention and priming: object cueing, useful-field-of-view, and crowding. *Cognitive Psychology*, *65*, 77–117.

Gan, K. W., & Lua, K. T. (1992). Chinese character classification using an adaptive resonance network. *Pattern Recognition*, *25*, 877–882.

Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, *233*, 1416–1419.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, *15*, 20–25.

Graziano, M. (2006). The organization of behavioral repertoire in motor cortex. *Annual Review of Neuroscience*, *29*, 105–134.

Grossberg, S. (2007). Towards a unified theory of neocortex: laminar cortical circuits for vision and cognition. *Progress in Brain Research*, *165*, 79–104.

Grossberg, S. (2009). Cortical and subcortical predictive dynamics and learning during perception, cognition, emotion and action. *Philosophical Transactions of the Royal Society of London—Series B: Biological Sciences*, *364*, 1223–1234.

Grossberg, S., & Huang, T. R. (2009). ARTSCENE: a neural system for natural scene classification. *Journal of Vision*, *9*, 1–19.

Hebert, B., Andreas, E., Tinne, T., & Van Gool, L. (2008). SURF: speeded up robust features. *Computer Vision and Image Understanding, CVIU*, *110*(3), 346–359.

Hollerbach, J. M. (1982). Computers, brains, and the control of movement. *Trends in Neurosciences*, *5*, 189–192.

Huang, T.-R., & Grossberg, S. (2010). Cortical dynamics of contextually cued attentive visual learning and search: spatial and object evidence accumulation. *Psychological Review*, *117*, 1080–1112.

Iossifidis, I., & Schoner, G. (2007). Dynamical systems approach for the autonomous avoidance of obstacles and joint-limits for an redundant robot arm. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 580–585).

Jarvis, R. A., & Patrick, E. A. (1973). Clustering using a similarity measure based on shared nearest neighbors. *IEEE Transactions on Computers*, *22*(11), 1025–1034.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, *5*(1), 90–98.

Mel, B. W. (1990). *Connectionist robot motion planning: a neurally-inspired approach to visually-guided reaching*. San Diego, CA, USA: Academic Press Professional.

Mishkin, M., & Ungerleider, L. G. (1982). Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural Brain Research*, *6*(1), 57–77.

Mulliken, G. H., Musallam, S., & Andersen, R. A. (2008). Forward estimation of movement state in posterior parietal cortex. *Proceedings of the National Academy of Sciences*, *105*, 8170–8177.

Mussa-Ivaldi, F. A., & Hogan, N. (1991). Integrable solutions of kinematic redundancy via impedance control. *The International Journal of Robotics Research*, *10*, 481–491.

Osada, R., Funkhouser, T., Chazelle, B., & Dobkin, D. (2001). Matching 3D models with shape distributions. In *SMI 2001 international conference on shape modeling and applications* (pp. 154–166).

Rieke, F., Warland, D., van Steveninck, R. R., & Bialek, W. (1997). *Spikes: exploring the neural code*. MIT Press.

Rosenbaum, D. A., Meulenbroek, R. G. J., & Vaughan, J. (2001). Planning reaching and grasping movements: theoretical outlines and practical implications. *Motor Control*, *5*, 99–115.

Sanchez, G., & Latombe, J. C. (2003). A single-query bi-directional probabilistic roadmap planner with lazy collision checking. *Robotics Research*, 403–417.

Schwarzer, F., & Saha, M. (2005). Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. *IEEE Transactions on Robotics*, *21*, 338–353.

Snyder, L. H., Batista, A. P., & Andersen, R. A. (2000). Intention-related activity I the posterior-parietal cortex: a review. *Vision Research*, *40*, 1433–1441.

Srinivasa, N., & Ahuja, N. (1998). A learning approach to fixating on 3D targets with active cameras. In *Proceedings of Asian Conference on Computer Vision*, 1 (pp. 623–631).

Srinivasa, N., Bhattacharyya, R., & Grossberg, S. (2008). A bio-inspired kinematic controller for obstacle avoidance during reaching tasks with redundant robots. In *Proceedings of the IEEE RAS/EMBS international conference on biomedical robotics and biomechatronics* (pp. 7–14).

Srinivasa, N., & Grossberg, S. (2008). A head–neck–eye system that learns fault-tolerant saccades to 3-D targets using a self-organizing neural model. *Neural Networks*, *21*, 1380–1391.

Srinivasa, N., & Jouaneh, M. (1993). An invariant pattern recognition machine using a modified ART architecture. *IEEE Transactions on Systems, Man and Cybernetics*, 335–341.

Srinivasa, N., & Sharma, R. (1997). Execution of saccades for active vision using a neuro-controller. *IEEE Control Systems*, *17*(2), 18–29.

Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, *5*(11), 1226–1235.

Tomida, S., Hanai, T., Honda, H., & Kobayashi, T. (2002). Analysis of expression profile using fuzzy adaptive resonance theory. *Bioinformatics*, *18*(8), 1073–1083.

Tsuji, T., Morasso, P., Shigehashi, K., & Kaneko, M. (1995). Motion planning for manipulators using artificial potential field approach that can adjust convergence time of generated arm trajectory. *Journal of the Robotics Society of Japan*, *13*(3), 285–290.

Wienke, D., & Buydens, L. (1996). Adaptive resonance theory based neural network for supervised chemical pattern recognition (Fuzzy ARTMAP) part 1: theory and network properties. *Chemometrics and Intelligent Laboratory Systems*, *32*, 151–164.

Wolpert, D., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.