Robotics and Autonomous Systems 60 (2012) 424–440

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

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

Coupled dynamical system based arm–hand grasping model for learning fast adaptation strategies

Ashwini Shukla [*], Aude Billard

Learning Algorithms and Systems Laboratory (LASA), Ecole Polytechnique Federale de Lausanne - EPFL, Switzerland

ARTICLE INFO

Article history:
Available online 27 August 2011

Keywords:
Grasping
Hand–arm coordination
Fast perturbations
Manipulation planning
Programming by demonstration

ABSTRACT

Performing manipulation tasks interactively in real environments requires a high degree of accuracy and stability. At the same time, when one cannot assume a fully deterministic and static environment, one must endow the robot with the ability to react rapidly to sudden changes in the environment. These considerations make the task of reach and grasp difficult to deal with. We follow a Programming by Demonstration (PbD) approach to the problem and take inspiration from the way humans adapt their reach and grasp motions *when perturbed*. This is in sharp contrast to previous work in PbD that uses unperturbed motions for training the system and then applies perturbation solely during the testing phase. In this work, we record the kinematics of arm and fingers of human subjects during unperturbed and perturbed reach and grasp motions. In the perturbed demonstrations, the target's location is changed suddenly after the onset of the motion. Data show a strong coupling between the hand transport and finger motions. We hypothesize that this coupling enables the subject to seamlessly and rapidly adapt the finger motion in coordination with the hand posture. To endow our robot with this competence, we develop a *coupled dynamical system* based controller, whereby two dynamical systems driving the hand and finger motions are coupled. This offers a compact encoding for reach-to-grasp motions that ensures fast adaptation with zero latency for re-planning. We show in simulation and on the real iCub robot that this coupling ensures smooth and "human-like" motions. We demonstrate the performance of our model under spatial, temporal and grasp type perturbations which show that reaching the target with *coordinated hand–arm motion* is necessary for the success of the task.

1. Introduction

Planning and control of constrained grasping motions have often been studied as two separate problems in which one first generates the arm motion [1,2] and then shapes the hand to grasp stably the targeted object [3,4]. The sheer complexity of each of these two problems when controlling high dimensional arm–hand systems has discouraged the use of a single coherent framework for carrying out both tasks simultaneously. In this work, we advocate the use of a single framework to control reach and grasp motion when the task requires very fast adaptation of the motion. We consider the problem of on-the-fly replanning reach and grasp motion for enabling adaption to changes in the position, size or type of object to be grasped. This requires the ability for fast and flexible re-planning.

One widely desired property when designing a robot controller is robustness, i.e. the ability to robustly recover from perturbations. In reach-to-grasp tasks, perturbations may be of the following types: (a) displacement of the robot end-effector and/or the target (spatial perturbations), (b) delays in the task execution due to random factors such as friction in the gears or delays in the underlying controller of the robot (temporal perturbations), (c) change in the target object forcing a change in the type of the grasp required. In the context of controlling for reach and grasp tasks, this problem has been addressed primarily by designing a stable controller (ensured to stop at the target) for both reach and grasp components of motion. Many different ways have been offered to designing task-specific controllers with minimum uncertainties and deviations from the intended trajectory. One drawback of such approaches is that they assume that the trajectory to track is known before hand. It is however not always desirable to return to the original desired trajectory as the path to get there may be unfeasible, especially when a perturbation sent us far from the originally planned trajectory. More recent approaches have advocated the use of dynamical systems as a natural means of embedding sets of feasible trajectories. This offers great robustness in the face of perturbations, as a new desired trajectory can be recomputed on the fly with no need to re-plan [5–7]. We follow this trend and extend our previous works [8,9] on learning a motor control law using time-invariant dynamical systems. Such a control

[*] Corresponding author. Tel.: +41 21 693 6947.
E-mail address: ashwini.shukla@epfl.ch (A. Shukla).

0921-8890/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2011.07.023

law generates trajectories that are asymptotically stable at a single attractor. In this paper, we extend this work to enable coupling across two such dynamical systems for controlling reach and grasp motions in synchrony. Controlling for such coupled dynamical systems entails more complexity than controlling using two independent control laws to ensure satisfaction of convergence constraints and correlations between the two processes [10–13].

We follow a Programming by Demonstration (PbD) approach [14] and investigate how we can take inspiration from the way humans react when perturbed and learn motor control laws from such examples. This departs from the usual approaches in PbD that usually use demonstrations of *unperturbed* motions.

A number of studies of the way humans, and other animals, control reach-and-grasp tasks [10,15,16] have established that the dynamics of arm and finger movements follow a particular pattern of coordination, whereby the fingers start opening (*preshape*) for the final posture at about half of the reaching cycle motion. Humans and other animals adapt both the timing of hand transport and the size of the finger aperture to the object's size and location. When perturbed, humans adapt these two variables seamlessly and in synchrony [17,18]. In [19], we showed a strong coupling between the dynamics of finger aperture and the hand velocity. Finger aperture is composed of a biphasic course, i.e. a short and wide opening after hand peak velocity is followed by a slow closure phase. In this paper, we revisit these observations to derive precise measurement of the correlation between hand transport and finger preshape, which we then use to determine specific parameters of our model of coupled dynamical systems across these two motor programs.

This paper is divided as follows. Section 2 reviews the literature related with the presented work: imitation learning, manipulation planning and biological evidences of reach–grasp coupling. Section 3 starts with a short recap of the background of Dynamical Systems (DSs), their estimation using GMMs and performing regression. We give a formal definition of the Coupled Dynamical System (CDS) model, explain the model construction process and give the algorithm for regression. In Section 4, we present the experimental setup used to learn from perturbed human demonstrations. We validate our approach by presenting a series of experiments on the iCub simulator as well as the real robot. We show that the reach–grasp behavior is reproduced while respecting the correlations and couplings learned during the demonstrations and that it is critical for the success of the overall task. We also show that the post-perturbation re-planning is quick and enables very fast response from the robot.

2. Related work

The presented model relates to different fields of work. It draws inspiration from neuro-physiological studies of human reach-to-grasp motion, exploits the current techniques from imitation learning to add novel contribution in the field of manipulation planning and control. In this section, we review the relevant literature in each of these fields.

2.1. Manipulation planning

The classical approach in robotics for reaching to grasp objects has been to divide the overall problem into two sub-problems, where one first reaches for and then grasps the objects [1,20,21]. Although both the issues of reaching to a pre-grasp pose and formation of grasp around arbitrary objects are intensively studied, very few [22–24] have looked into combining the two so as to have a unified reach–grasp system.

Most manipulation planners typically plan paths in the configuration space of the robot using graph based techniques. Very powerful methods such as those based on probabilistic roadmap and its variants [25,26] use a C-space description of the environment and graph based methods for search. Another approach to the same problem has been adopted by using various control schemes in conjunction with offline grasp planners [21] or visual tracking systems [27]. A synergistic combination of grasp planning, visual tracking and arm trajectory generation is presented in [28]. LaValle and Kuffner [29] proposed RRT as a faster alternative to manipulation planning problems, provided the existence of an efficient inverse kinematic (IK) solver. RRT based methods [1] are currently the fastest online planners due to their efficient searching ability. The reported planning times are of the order of 100 ms for single arm reaching tasks in the absence of any obstacles [20]. While this is certainly very quick, graph based methods lose to take into account the dynamic constraints of the task.

It remains a challenge to design planning algorithms for dynamic tasks under quick perturbations. Moreover, in such cases, re-planning upon perturbation must not take more than a few milliseconds. These are the type of problems we address here. We take inspiration from human studies to understand how humans embed correlations between arm and finger motion to ensure robust response to such fast perturbations. We show that retaining these correlations between the reach and grasp motions is critical to the success of the tasks.

2.2. Biological evidences

The concept of coupling between the reach and grasp motions is inspired by extensive evidence in neuro-physiological studies [10,13,30–35]. The most frequently reported mechanism suggests a parallel, but time-coupled evolution of the reach and grasp motions with synchronized termination. Attempts at quantifying this process in a way that may be usable for robot control are few. Glke et al. [36] and Bae and Armstrong [37] showed that the finger motion during reach to grasp tasks could be described by a simple polynomial function of time, while Ulloa and Bullock [38] modeled the covariation of the arm and the finger motion. Interestingly, these authors also report on an involuntary reopening of the fingers upon perturbation of the target location; an observation which we will revisit in this paper.

A number of models have been developed to simulate the finger–hand coupling, so as to account for the known coordinated pattern of hand–arm motions. The Hoff–Arbib model [39] generates a heuristic estimate of the transport time based on the reaching distance and object size and uses it to compute the opening and closing times of the hand. Since the control scheme presented in their approach was time dependent and the temporal coupling parameters decided prior to the onset of movement, this model did not guarantee handling of temporal or spatial perturbations. Oztop and Arbib [40] argued in their *hand state hypothesis* that during human reach–grasp motion control, the most appropriate feedback is a 7 dimensional vector, including pose of the hand w.r.t the target, hand aperture and thumb adduction/abduction. In the Haggard–Wing model [41], both processes of transport and aperture control have access to each other's spatial state. The variance of hand and finger joint angles is used to set the corresponding control gains, whereas computation of the correlation across the two is used to implement the spatial coupling. The time independency of this model proved to be an elegant way to handle temporal perturbations. A neural network based model presented by Ulloa and Bullock [42] ensured continuous coupling and efficient handling of perturbations. They assumed the *vector integration to end point* (VITE) model [43] as a basis for task dynamics.

The model we propose here specifically exploits the principle of spatial coupling between the palm and finger motion. It ensures that the motion reproduced by the robot exhibits hand–arm coupling and respects termination constraints similar to what are found in natural human motion.

2.3. Imitation learning

Learning how to perform a task by observing demonstrations from an experienced agent has been explored extensively under different frameworks. Classical means of encoding the task information are based on spline or polynomial decomposition and averaging [44,45]. These have been shown to be very fast trajectory generators, useful for tasks like catching moving objects. A different body of work advocates non-linear stochastic regression techniques in order to represent the tasks and regenerate motion in a generalized setting [46]. These methods allow systematical treatment of uncertainty by assuming data noise and hence *estimate* the trajectories as a set of random variables. The regression assumes a model for the underlying process and learns its parameters via machine learning techniques. Subsequently, multiple works under the PbD framework [2,8,47] have shown that this problem can be handled elegantly by using the dynamical systems (DS) approach. Using DS to represent motion removes the explicit time dependency from the model. As a result, transitions between the states during the execution of a task depend solely on the current state of the robot and the environment.[1] However, the removal of time dependency is introduced at the cost of non-trivial stability of the models. The states of a process evolving autonomously under the influence of a DS may diverge away from the goal if initialized outside the basin of attraction of the equilibrium point. Eppner et al. [48] presented a dynamic Bayesian network based approach to learn generalized relations between the world and the robot/demonstrator. While generalizing the task reproduction over different spatial setups, this framework also allows to include constraints not captured from the demonstrations, such as obstacle avoidance. Ijspeert et al. [2] in their *dynamic movement primitive* (DMP) formulation, augment the dynamics learned from motion data with a stable linear dynamics which would take precedence as the state reaches close to the goal. In our previous work [9], it was shown that formulating the problem of fitting data to the Gaussian Mixture Model as a non-linear optimization problem under stability constraints ensures global asymptotic stability of the DS.

Although a large amount of work has been done on learning and improving skills from observing good examples of successful behavior, very few work has looked into the information that can be extracted from the non-canonical demonstrations. In Grollman and Billard [49], we proposed one way to learn from failed demonstrations. Here, we follow a complementary road and investigate how one can learn from observing how humans adapt their motion so as to avoid failure. To the best of our knowledge, this is the first work on PbD that studies human motion recovery under perturbation. We empirically show — (a) the difference in dynamics employed during perturbed and unperturbed demonstrations and (b) the coupling that exists between the reach and grasp components which ensures successful task completion. We present a coupled dynamical system based approach to achieve coordination between hand transport and pre-shape. We show that the DS based formulation enables our model to react under very fast on-the-fly perturbations without any latency for re-planning. We validate the model by implementing our method on the iCub simulator as well as the real robot.

3. Methodology

In this section, we start with a short description of motion encoding using autonomous dynamical systems (DSs) and explain how Gaussian Mixture Models (GMMs) can be used to estimate them. We then present an extension of this GMM estimate to allow coupling across different DSs, which we further refer to as coupled dynamical system (CDS). A formal discussion of the CDS model is presented describing the modeling process and regression algorithm to reproduce the task and a simple 2D example is included to establish intuitive understanding of the working of the CDS model.

3.1. DS control of reaching

We here briefly present our previous work on modeling reaching motion through autonomous dynamical systems with a single attractor at the target. For clarity, we reiterate the encoding presented in [8].

Let $\boldsymbol{\xi}$ denote the end-effector position and $\dot{\boldsymbol{\xi}}$ its velocity. We further assume that the state of the system evolves in time according to a first order autonomous Ordinary Differential Equation (ODE):

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi}). \tag{1}$$

$\mathbf{f}: \mathbb{R}^d \mapsto \mathbb{R}^d$ is a continuous and continuously differentiable function with a single equilibrium point at the attractor, denoted by $\boldsymbol{\xi}^*$ and we have:

$$\lim_{t\to\infty} \boldsymbol{\xi}(t) = \mathbf{0}. \tag{2}$$

We do not know $\mathbf{f}$, but we are provided with a set of N demonstrations of the task where the state vector and its velocities are recorded at particular time intervals, yielding the data set $\{\boldsymbol{\xi}_n^t, \dot{\boldsymbol{\xi}}_n^t\} \forall t \in [0, T_n]; n \in [1, N]$. T_n denotes the number of data points in demonstration n. We assume that this data was generated by our function $\mathbf{f}$ subjected to a white Gaussian noise ϵ and hence we have:

$$\dot{\boldsymbol{\xi}} = \mathbf{f}(\boldsymbol{\xi}; \boldsymbol{\theta}) + \epsilon. \tag{3}$$

Notice that $\mathbf{f}$ is now parameterized by the vector $\boldsymbol{\theta}$, that represents the parameters of the model we will use to estimate $\mathbf{f}$.

We build a model-free estimate of the function $\hat{\mathbf{f}}$ in two steps. We first build a probability density model of the data by modeling it through a mixture of K Gaussian functions. The core assumption when representing a task as a Gaussian Mixture Model (GMM) is that each recorded point $\boldsymbol{\xi}(t)$ from the demonstrations is a sample drawn from the joint distribution:

$$\boldsymbol{\xi} \sim \mathcal{P}\left(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}} | \boldsymbol{\theta}\right) = \sum_{k=1}^{K} \pi^k \mathcal{N}(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}; \boldsymbol{\theta}^k) \tag{4}$$

with $\mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\theta}^k) = \frac{1}{\sqrt{(2\pi)^{2d}|\Sigma^k|}} e^{\frac{1}{2}(\boldsymbol{\xi}-\boldsymbol{\mu}^k)^{\mathrm{T}}(\Sigma^k)^{-1}(\boldsymbol{\xi}-\boldsymbol{\mu}^k)^{\mathrm{T}}}$ and where π^k, $\boldsymbol{\mu}^k$ and Σ^k, are the component weights, means and covariances of the k-th Gaussian.

Taking then the posterior mean estimate of $\mathcal{P}\left(\dot{\boldsymbol{\xi}} | \boldsymbol{\xi}\right)$ yields a noise-free estimate of our underlying function:

$$\dot{\boldsymbol{\xi}} = \sum_{k=1}^{K} h^k(\boldsymbol{\xi})(\mathrm{A}^k + \mathbf{b}^k) \tag{5}$$

where,

$$\left.\begin{aligned} A^k &= \underset{\dot{\xi}\xi}{\overset{k}{\Sigma}} (\underset{\xi}{\overset{k}{\Sigma}})^{-1} \\ \mathbf{b}^k &= \mu_{\dot{\xi}}^k - A^k \mu_{\xi}^k \\ h^k(\boldsymbol{\xi}) &= \frac{\pi^k \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\theta}^k)}{\sum_{i=1}^{K} \pi^i \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\theta}^i)} \end{aligned}\right\}. \tag{6}$$

To ensure that the resulting function is asymptotically stable at the target, we use the *stable estimator of dynamical system* (SEDS) approach, see [9] for a complete description. In short, SEDS

[1] Even in a DS formulation, time dependency is present but only implicitly in the form of time derivatives of the state variables.

determines the set of parameters $\boldsymbol{\theta}$ that maximizes the likelihood of the demonstrations being generated by the model, under strict constraints of global asymptotic stability. Next, we explain how this basic model is exploited to ensure that the hand and fingers reach the target even when perturbed. We further show how it is extended to build an explicit coupling between hand and finger motion dynamics to ensure robust and coordinated reach and grasp.

3.2. Why coupling reach and grasp?

We are now facing the problem of extending our reaching model presented in the previous section to allow successful hand–arm coordination when performing reach and grasp. The scheme presented in our previous works, as explained in previous sub-section, assumes that the motion is point-to-point in high dimensional space. As a result, the state vector $\boldsymbol{\xi}$ converges uniformly and asymptotically to the target. To perform reach–grasp tasks, such a scheme could be exploited in two ways. One could either:

1. Learn two separate and independent DSs with end-effector pose and finger configurations as state vectors.
2. Or learn one DS with an extended state vector consisting of degrees of freedom of the end-effector pose as well as finger configurations.

Learning two DSs would not be desirable at all since then the two sub-systems (hand transport and pre-shape) would evolve independently using their respective learned dynamics. Hence, any perturbation in hand transport would leave the two sub-systems temporally out of synchronization. This may lead to the failure of the overall reach–grasp task even when each of the individual DS will have converged to their respective goal states.

At first glance, the second option is more appealing as one could hope to be able to learn the correlation between hand and finger dynamics, which would then ensure that the temporal constraints between the convergence of transport and hand pre-shape motions will be retained during reproduction. In practice, good modeling of such an *implicit* coupling in high-dimensional system is hard to ensure. The model is as good as the demonstrations are. If one is provided with relatively few demonstrations (in PbD one targets less than ten demonstrations for the training to be bearable to the trainer), chances are that the correlations will be poorly rendered, especially when querying the system far away from the demonstrations. Hence, if the state of the robot is perturbed away from the region of the state space which was demonstrated, one may not ensure that the two systems will be properly synchronized. We will establish this by the means of a simulation experiment in Section 4.

We here take an intermediary approach in which two separate DSs are first learned and then coupled explicitly. In the context of reach-and-grasp tasks, the two separate DSs correspond to the hand transport (dynamics of the end-effector motion) and the hand pre-shape (dynamics of the finger joint motion). We will assume that the transport process evolves independently of the fingers' motions while the instantaneous dynamics followed by the fingers depends on the state of the hand. This will result in the desired behavior, namely that the fingers will reopen when the object is moved away from the target. Note that the finger–hand coupling will be parameterized. We will show in the experiments that this coupling can be tuned by changing the model parameters to favor either "human-like" motion or fast adaptive motion to recover from quick perturbations.

3.3. Coupled dynamical system

In the following subsections, we present the formalism behind the coupled dynamical system (CDS), describing how we learn the model and how we then query the model during task execution. To facilitate understanding of the task reproduction using the CDS model, we illustrate the latter in a 2D example that offers a simplistic representation of the high-dimensional implementation presented in the result section.

3.3.1. CDS model

Let $\boldsymbol{\xi}_x \in \mathbb{R}^3$ denote the cartesian position of the hand and $\boldsymbol{\xi}_f \in \mathbb{R}^{d_f}$ the joint angles of the fingers. d_f denotes the total number of degrees of freedom of the fingers. The hand and the fingers follow separate autonomous DS with associated attractors. For convenience, we place the attractors at the origin of the frames of reference of both the hand motion and the finger motion and hence we have: $\boldsymbol{\xi}_x^* = 0$ and $\boldsymbol{\xi}_f^* = 0$. In other words, the hand motion is expressed in a coordinate frame attached to the object to be grasped, while the zero of the finger joint angles is placed at the joint configuration adopted by the fingers when the object is in the grasp. We assume that there is a single grasp configuration for a given object. Since the reach and grasp dynamics may vary depending on the object to be grasped, we will build a separate CDS model for each object considered here. We denote the set $\mathcal{G}$ of all objects for which grasping behaviors are demonstrated.

The following three joint distributions, learned as separate GMMs, combine to form the CDS model:

1. $\mathcal{P}\left(\boldsymbol{\xi}_x, \dot{\boldsymbol{\xi}}_x | \boldsymbol{\theta}_x^g\right)$: encoding the dynamics of the hand transport
2. $\mathcal{P}\left(\Psi(\boldsymbol{\xi}_x), \boldsymbol{\xi}_f | \boldsymbol{\theta}_{inf}^g\right)$: encoding the joint probability distribution of the inferred state of the fingers and the current hand position
3. $\mathcal{P}\left(\boldsymbol{\xi}_f, \dot{\boldsymbol{\xi}}_f | \boldsymbol{\theta}_f^g\right)$: encoding the dynamics of the finger motion

$\forall g \in \mathcal{G}$. Here, $\Psi : \mathbb{R}^3 \mapsto \mathbb{R}$ denotes the *coupling function* which is a monotonic function of $\boldsymbol{\xi}_x$ satisfying:

$$\lim_{\boldsymbol{\xi}_x \to \mathbf{0}} \Psi(\boldsymbol{\xi}_x) = 0. \tag{7}$$

$\boldsymbol{\theta}_x^g$, $\boldsymbol{\theta}_f^g$ and $\boldsymbol{\theta}_{\text{inf}}^g$ denote the parameter vectors of the GMMs encoding the hand-transport dynamics, finger motion dynamics and the inference model, respectively.

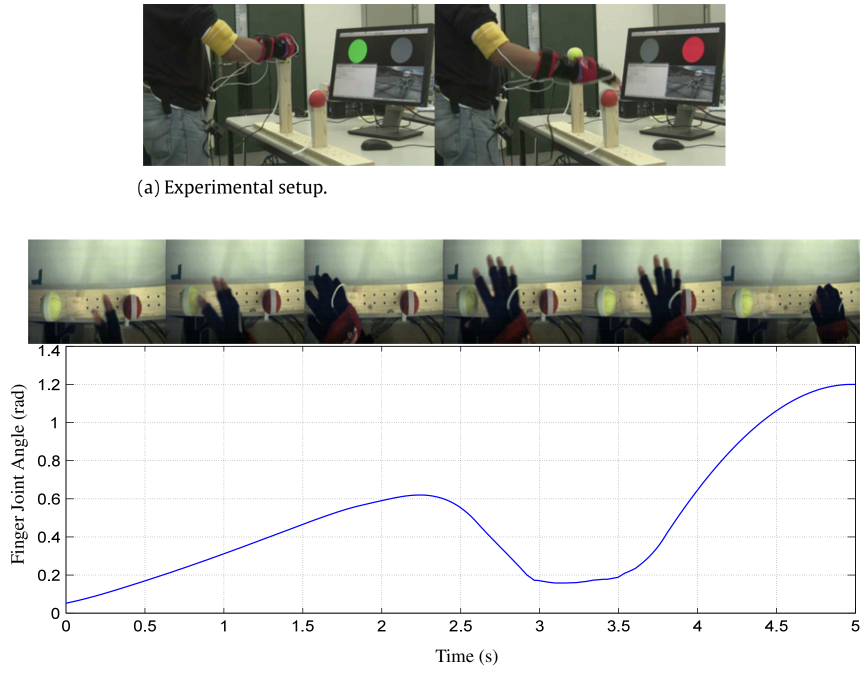
The distributions $\mathcal{P}\left(\boldsymbol{\xi}_x, \dot{\boldsymbol{\xi}}_x | \boldsymbol{\theta}_x^g\right)$ and $\mathcal{P}\left(\boldsymbol{\xi}_f, \dot{\boldsymbol{\xi}}_f | \boldsymbol{\theta}_f^g\right)$ that represent an estimate of the dynamics of the hand and finger motion, respectively, are learned using the same procedure as described in Section 3.1. To recall, each density is modeled through a mixture of Gaussian functions. As explained in Section 3.1, in order to ensure that the resulting mixture is asymptotically stable at the attractor (here the origin of each system), we use the SEDS learning algorithm [9]. Note that SEDS allows to only learn models where the input and output variables have the same dimensions. Since the variables of the distribution $\mathcal{P}\left(\Psi(\boldsymbol{\xi}_x), \boldsymbol{\xi}_f | \boldsymbol{\theta}_{inf}^g\right)$ have not the same dimension, we learned this distribution through a variant of SEDS where we maximize the likelihood of the model under the constraint:

$$\lim_{x \to 0} \mathbb{E}\left[\boldsymbol{\xi}_f | x\right] = \mathbf{0}. \tag{8}$$

3.3.2. Reproduction

While reproducing the task, the model essentially works in three phases: *Update hand position* → *Infer finger joints* → *Increment finger joints*. The palm position is updated independently at every time step and its current value is used to modulate the dynamics of the finger motion through the coupling mechanism. Fig. 2 shows this flow of information across the sub-systems and the robot. Such a scheme is desired since it ensures that any perturbation is reflected appropriately in both sub-systems.

The process starts by generating a velocity command for the hand transport sub-system and increments its state by one time step. $\Psi(\boldsymbol{\xi}_x)$ transforms its current state which is fed to the



(a) Experimental setup.

(b) Hand closeup.

Fig. 1. (a) Experimental setup to record human behavior under perturbations. On screen target selector is used to create a sudden change in the target location for reaching. (b) Motion of the fingers as seen from a high speed camera @ 100 fps. Note the decrease in the joint angle values (re-opening of fingers) starting at the onset of perturbation.

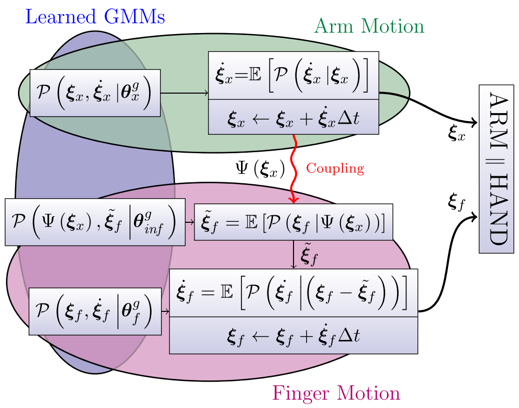

Learned GMMs
Arm Motion
$\mathcal{P}\left(\xi_x, \dot{\xi}_x \middle| \theta_x^g\right)$
$\dot{\xi}_x = \mathbb{E}\left[\mathcal{P}\left(\dot{\xi}_x \middle| \xi_x\right)\right]$
$\xi_x \leftarrow \xi_x + \dot{\xi}_x \Delta t$
ξ_x
$\Psi(\xi_x)$
Coupling
ARM || HAND
ξ_f
$\mathcal{P}\left(\Psi(\xi_x), \tilde{\xi}_f \middle| \theta_{inf}^g\right)$
$\tilde{\xi}_f = \mathbb{E}\left[\mathcal{P}\left(\xi_f \middle| \Psi(\xi_x)\right)\right]$
$\tilde{\xi}_f$
$\mathcal{P}\left(\xi_f, \dot{\xi}_f \middle| \theta_f^g\right)$
$\dot{\xi}_f = \mathbb{E}\left[\mathcal{P}\left(\dot{\xi}_f \middle| \left(\xi_f - \tilde{\xi}_f\right)\right)\right]$
$\xi_f \leftarrow \xi_f + \dot{\xi}_f \Delta t$
Finger Motion

Fig. 2. Task execution using CDS model. Blue region shows the three Gaussian Mixture Models which form the full CDS model. Green region shows the sub-system which controls the dynamics of the hand transport. Magenta region shows the sub-system controlling finger motion, while being influenced by the state of the hand transport sub-system. Coupling is ensured by passing selective state information in the form of $\Psi(\boldsymbol{\xi}_x)$ as shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

inference model that calculates the desired state of the finger joint angles by conditioning the learned joint distribution. The velocity to drive the finger joints from their current state to the inferred (desired) state is generated by *Gaussian Mixture Regression* (GMR) conditioned on the error between the two. The fingers reach a new state and the cycle is repeated until convergence. Algorithm 1 explains the complete reproduction process in pseudo-code. Note that the coupling function $\Psi(\boldsymbol{\xi}_x)$ also acts as a phase variable which updates itself at each time step and, in the event of a perturbation, will command the fingers to re-adjust so as to maintain the same correlations between the sub-system states as learned from the demonstrations. Two other parameters governing the coupled behavior are scalars $\alpha, \beta > 0$. Qualitatively speaking, they respectively control the speed and amplitude of the robot's reaction under perturbations.

Algorithm 1 Task Execution using CDS

```
Input: $\boldsymbol{\xi}_x(0)$; $\boldsymbol{\xi}_f(0)$; $\boldsymbol{\theta}_x^g$; $\boldsymbol{\theta}_{\text{inf}}^g$; $\boldsymbol{\theta}_f^g$; $\alpha$; $\beta$; $\Delta t$; $\epsilon$
  Set $t = 0$
  repeat:
    if perturbation then
      update $g \in \mathcal{G}$
    end if
    Update Hand Position:   $\dot{\boldsymbol{\xi}}_x(t) \sim \mathcal{P}\left(\dot{\boldsymbol{\xi}}_x | \boldsymbol{\xi}_x; \boldsymbol{\theta}_x^g\right)$
                            $\boldsymbol{\xi}_x(t+1) = \boldsymbol{\xi}_x(t) + \dot{\boldsymbol{\xi}}_x(t)\Delta t$
    Infer Finger Joints:    $\tilde{\boldsymbol{\xi}}_f(t) \sim \mathcal{P}\left(\boldsymbol{\xi}_f | \Psi(\boldsymbol{\xi}_x); \boldsymbol{\theta}_{inf}^g\right)$
    Update Finger Joints:   $\dot{\boldsymbol{\xi}}_f(t) \sim \mathcal{P}\left(\dot{\boldsymbol{\xi}}_f | \beta\left(\boldsymbol{\xi}_f - \tilde{\boldsymbol{\xi}}_f\right); \boldsymbol{\theta}_f^g\right)$
                            $\boldsymbol{\xi}_f(t+1) = \boldsymbol{\xi}_f(t) + \alpha\dot{\boldsymbol{\xi}}_f(t)\Delta t$
    $t \leftarrow t + 1$
  until: Convergence   $\left(\|\dot{\boldsymbol{\xi}}_f(t)\| < \epsilon \text{ and } \|\dot{\boldsymbol{\xi}}_x(t)\| < \epsilon\right)$
```

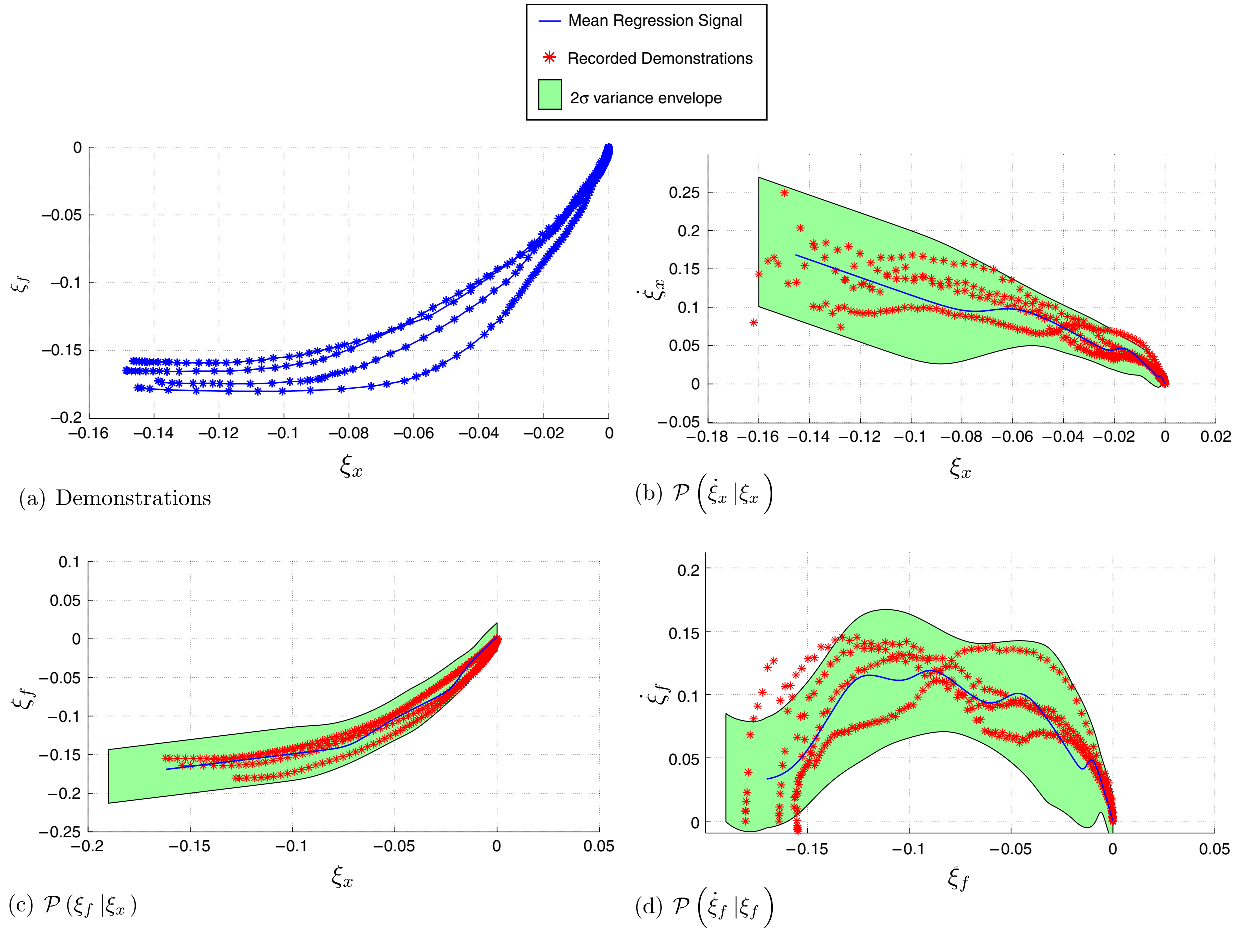
As described in Section 3.1, learning using SEDS ensures that the model for reaching and the model for grasping are both stable at their respective attractors. We however need now to verify that when one combines the two models using CDS, the resulting model is stable at the same attractors.

Definition. A CDS model is globally asymptotically stable at the attractors $\boldsymbol{\xi}_x^*$, $\boldsymbol{\xi}_f^*$ if by starting from any given initial conditions $\boldsymbol{\xi}_x(0)$, $\boldsymbol{\xi}_f(0)$ and coupling parameters $\alpha, \beta \in \mathbb{R}$ the following conditions hold:

$$\lim_{t\to\infty} \boldsymbol{\xi}_x(t) = \boldsymbol{\xi}_x^* \tag{9a}$$

$$\lim_{t\to\infty} \boldsymbol{\xi}_f(t) = \boldsymbol{\xi}_f^*. \tag{9b}$$

Such a property is fundamental to ensure that the CDS model will result in a reach and grasp motion terminating at the desired target. Most importantly, showing that the attractors for hand and fingers are also globally asymptotically stable will ensure that this model benefits from the same robustness to perturbation as described for



(a) Demonstrations

(b) $\mathcal{P}\left(\dot{\xi}_x \left|\xi_x\right.\right)$

(c) $\mathcal{P}\left(\xi_f \left|\xi_x\right.\right)$

(d) $\mathcal{P}\left(\dot{\xi}_f \left|\xi_f\right.\right)$

Fig. 3. GMMs which combine to form the CDS model for 2D example. (a) shows the human demonstrations. Large number of datapoints around the end of trajectories depict very small velocities. (b) shows the GMM encoding the velocity distribution conditioned on the position of reaching motion (ξ_x). (c) shows the GMM encoding the desired value of ξ_f (i.e. $\tilde{\xi}_f$) given the current value of ξ_x as seen during the demonstrations. (d) shows the GMM encoding the dynamic model for the finger pre-shape.

the simple reaching model in Section 3.1. See Appendix A for the proof of stability.

3.3.3. Minimal example

To establish an intuitive understanding, we instantiate the CDS model as a 2D representative example of actual high-dimensional reach–grasp tasks. We consider 1-D cartesian position ξ_x of the end-effector and 1 finger joint angle ξ_f, both expressed with respect to their respective goal states so that they converge to the origin. In this way, the full fledged grasping task is just a higher dimensional version of this case by considering 3-dimensional cartesian position instead of ξ_x and all joint angles (or eigen-grasps) of the fingers instead of ξ_f.

Under the given setting, typical demonstrations of reach–grasp task are as shown in Fig. 3(a), where the reaching motion converges slightly faster than the finger curl. We extract the velocity information at each recorded point by finite differencing and build the following models from the resulting data: $\mathcal{P}\left(\xi_x, \dot{\xi}_x \left|\boldsymbol{\theta}_x\right.\right)$, $\mathcal{P}\left(\xi_x, \xi_f \left|\boldsymbol{\theta}_{inf}\right.\right)$ and $\mathcal{P}\left(\xi_f, \dot{\xi}_f \left|\boldsymbol{\theta}_f\right.\right)$. The resulting mixtures for each of the models are shown in Fig. 3. For reproducing the task, instead of using the earlier approach of [8] where the system evolves under the velocities computed as $\mathbb{E}\left[(\dot{\xi}_x; \dot{\xi}_f) \left|(\xi_x; \xi_f)\right.\right]$, we proceed as in Algorithm 1. Fig. 4 shows reproduction of the task in the (ξ_x, ξ_f) space overlaid on the demonstrations. It clearly shows that a perturbation in ξ_x creates an effect in ξ_f, i.e., generating a negative velocity, the magnitude of which is tunable using the α parameter. This change is brought due to the need of tracking the inferred ξ_f values i.e. $\tilde{\xi}_f$, at all ξ_x. $\tilde{\xi}_f$ represents the expected value of ξ_f given ξ_x as seen during the demonstrations. The variation of the trajectories of ξ_f with α and β is shown in Fig. 5. α modulates the speed with which the reaction to perturbation occurs. On the other hand, a high value of β increases the amplitude of reopening. Fig. 6 shows the streamlines of this system for two different α values in order to visualize the global behavior of trajectories evolving under the CDS model.

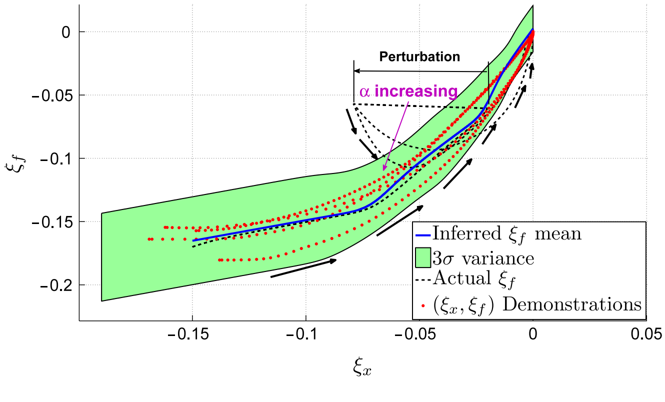


Fig. 4. Reproducing the task under the CDS model. The reproduction (dashed) is overlaid on the demonstrations for reference. The model is run for different α values and the flow of the state values in time is depicted by the arrows. It is evident that the model tries to track the desired ξ_f (blue) values at the current ξ_x by reversing the velocity in ξ_f direction. The tracking is more stringent for larger α. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

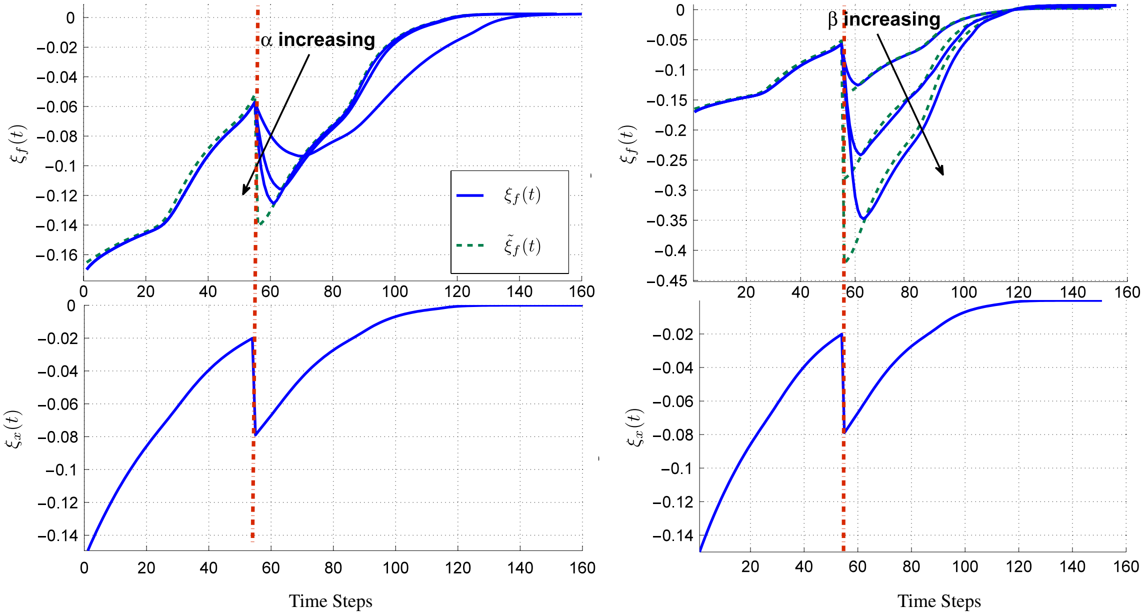


Fig. 5. Variation of obtained trajectories with α and β. Vertical red line shows the instant of perturbation when the target is suddenly pushed away along positive ξ_x direction. Negative velocities are generated in ξ_f in order to track $\tilde{\xi}_f$. Speed of retracting is proportional to α (*left*) and amplitude is proportional to β (*right*). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

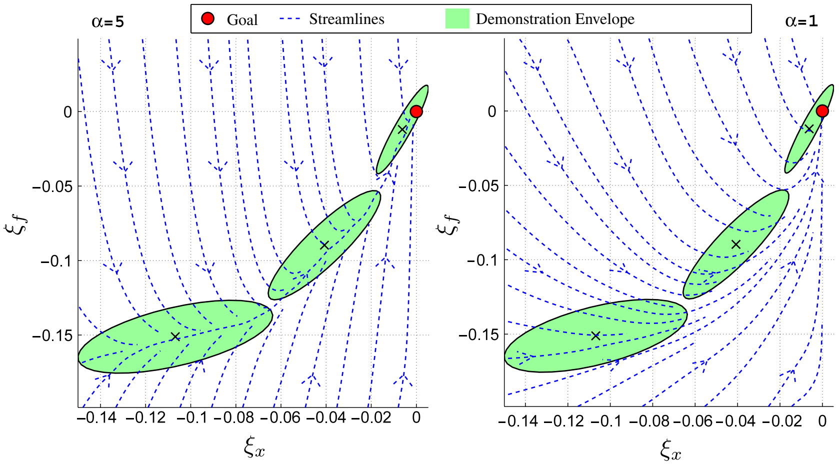


Fig. 6. Change in α affecting the nature of streamlines. Larger α will tend to bring the system more quickly toward the (ξ_x, ξ_f) locations seen during demonstrations.

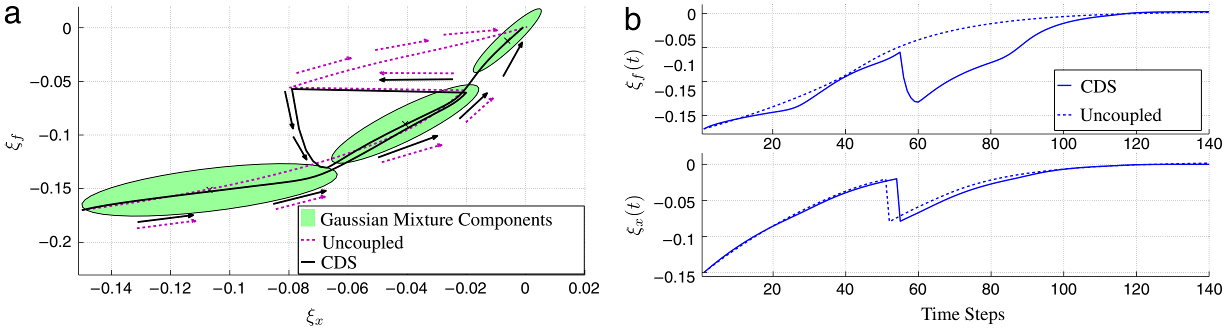


Fig. 7. Task reproduction with explicit and implicit coupling shown in (a) state space, (b) time variation. Dotted lines show the implicitly coupled task execution. Note the difference in the directions from which the convergence occurs in the two cases. In the explicitly coupled execution, convergence is faster in ξ_x than in ξ_f.

At this point, it is important to distinguish our approach from the single GMM approach of [8] mentioned in Section 3.2. Fig. 7 shows a comparison of the CDS trajectories with those obtained using the single GMM approach, where the coupling is only implicit. It shows the behavior when a perturbation is introduced only on the abscissa. Clearly, in the implicitly coupled case, the perturbation is not appropriately transferred to the unperturbed dimension ξ_f and the motion in that space remains unchanged.

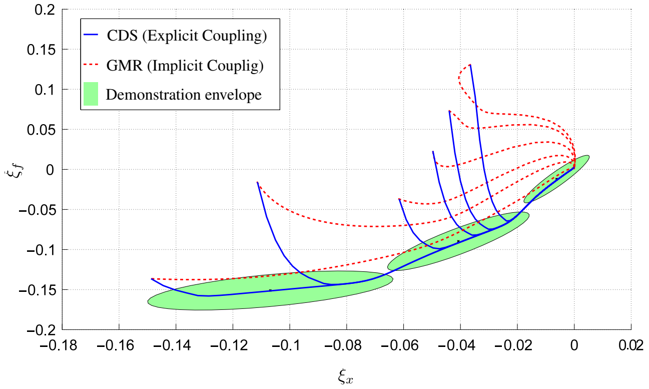


Fig. 8. Comparison between the motions obtained by single GMM and CDS approaches. Note how the order of convergence can be remarkably different when starting at different positions in state space.

This behavior can be significantly different depending on the state of the two sub-systems just after the perturbation.

To investigate this, we initialize the single GMM model as well as the CDS model at different points in state space and follow the two trajectories. Fig. 8 shows this experiment. Notice the sharp difference in the trajectories as the CDS trajectories try to maintain correlation between the state space variables and always converge from within the demonstration envelope. On the other hand, the trajectories of the single GMM approach have no definite convergence constraint.[2] This difference is significantly important in the context of reach–grasp tasks. If the trajectories converge from the top of the envelope, it means that the variable ξ_f (fingers) is converging faster than the ξ_x (hand position). This translates to premature finger closure as compared to what was seen during the demonstrations. If they converge from below, it means that the fingers are closing later than what was seen during the demonstrations. While the former is undesirable in any reach–grasp task, the latter is undesirable only in the case of moving/falling objects.

4. Experiments and results

A core assumption of our approach lies in the fact that human motor control exploits an explicit coupling between hand and finger motions. In this section, we first validate this hypothesis by reporting on a simple motion studies conducted with five subjects performing a reach and grasp task under perturbations. Data from human motion are used in three capacities: (a) to confirm that the CDS model captures well the coupling across hand and fingers found in human data; (b) as demonstration data to build the probability density functions of the CDS model; (c) to identify relationships across the variables of the system and to use these to instantiate the two free parameters (α and β) of the CDS model.

In the second part of this section, we perform various experiments in simulation and with the real iCub robot to validate the performance of the CDS model as a good model to ensure robust control of reach and grasp in robots. In particular, we test that the CDS model is indeed well suited to handle fast perturbations which typically need re-planning and are difficult to handle online. Videos for all robot experiments and simulations are cross-linked to the corresponding figures.

4.1. Instantiation of CDS variables

In all the experiments presented here, including human data, the state of our system is composed of the cartesian position and orientation of the end-effector (human/robot wrist) and of the following 6 finger joint angles:

- 1 for curl of the thumb.
- 2 for index finger proximal and distal joints.
- 2 for middle finger proximal and distal joints.
- 1 for combined curl of ring and little finger.

We use the norm-2 for the coupling function, i.e. $\Psi(.) = \|.\|$ in the CDS implementation for modeling both human data and for robot control. As a result, fingers reopen and close as a function of the distance of the hand to the target.

Since CDS controls for the hand displacement, we use the Moore–Penrose inverse kinematic function to convert the end-effector pose to joint angles of the arm. In simulation and on the real iCub robot, we control the seven degrees of freedom (DOFs) of the arm and 6 finger joints at an update rate of 20 ms.

4.2. Validation against human data

As discussed in Section 2.2, many physiological studies reported a natural coordination between arm and fingers when humans reach for objects. In order to assess quantitatively these observations and provide data in support of our model of a coupling between the two processes of hand transport and finger motion, we performed experiments with human subjects performing reach-to-grasp tasks under *fast and random perturbations*.

4.2.1. Experimental procedure

The experimental setup is shown in Fig. 1(a). The subject stands in front of two stationary targets, a green and a red ball. An on-screen target selector prompts the subject to reach and grasp one of the two balls depending on the color shown on the screen. To start the experiment, one of the ball is switched on and the subject starts to reach toward the corresponding object. As the subject is moving his hand and preshaping his fingers to reach for the target ball, a perturbation is created by abruptly switching off the target ball and lighting up the second ball. The switch across targets occurs only once during each trial about 1–1.5 s after the onset of the motion. The subject's hand has usually by then traveled more than half the distance separating it from the target. The trial stops once the subject has successfully grasped the second target.

To ensure that we are observing natural response to such perturbations, subjects were instructed to proceed at their own pace and no timing for the overall motion was enforced. As a result, the time it took for each subject to complete the motion varied across subjects and across trials. Since encoding in the CDS model is time-invariant, modeling is not affected by these changes in the duration of experiment.

We recorded the kinematic of the hand, fingers and arm motions of 5 subjects across 20 trials. 10 of the trials were unperturbed, i.e. the target was not switched during the motion. Subjects did the 20 trials in one swipe. Unperturbed and perturbed trials were presented in random order for each subject. The arm and hand motion was recorded using three XSens^{TM} IMU motion sensors attached to the upper arm, forearm and wrist of the subject at a frame rate of 20 ms.[3] The fingers' motion was recorded using a 5DT^{TM}data glove. Angular displacements of the arm joint and finger joints were re-constructed and mapped to the iCub's arm joint angles and finger joint angles. To assess visually that the correspondence between human motion and robot motion is well done, the iCub simulator runs simultaneously while the human is performing the trials.

[2] It is also worth mentioning that the two trajectories are fairly similar when initialized close to the demonstration envelope.

[3] To compensate for drifts from the IMU and data glove measurements, subjects were instructed to proceed to a brief calibration procedure after each trial. This procedure lasted not more than 5 s.

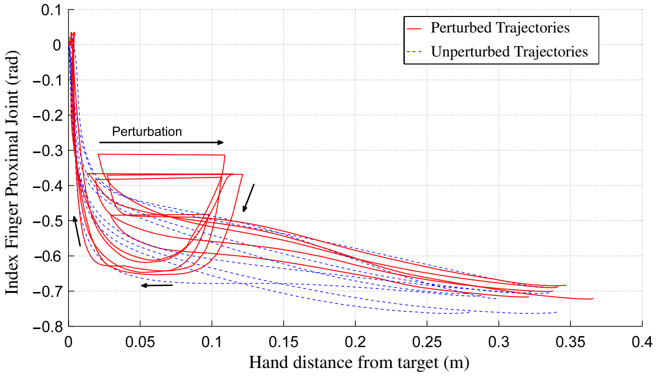


Fig. 9. Hand–finger coordination during perturbed and unperturbed demonstrations.

Data from the 10 unperturbed trials and from the 5 subjects are used to train the 3 GMMs which serve as the basis for the CDS model. The next section discusses how well the CDS model renders human behavior under perturbations.

4.2.2. Qualitative analysis of human motion

Visual inspection of the human data confirms a steady coupling between hand transport and fingers closing in the unperturbed situation, whereby fingers close faster as the hand approaches faster the target, and conversely. This coupling persists across trials and for all subjects. Most interesting is the observation that, during *perturbed* trials, just after the target is switched, the fingers first reopen and then close again synchronously with the hand, as the hand moves toward the new target; see Fig. 1(b). Note that, in all trials, the fingers re-opened irrespective of the fact that the aperture of the fingers at the time of perturbation was large enough to accommodate the object. This suggests that this reaction to perturbation is not driven by the need of accommodating the object within the grasp, but may be the result of some inherent property of finger–hand motor control. It appears as if the fingers would first "reset" to a location that corresponds to the expected location for the fingers given the new hand–target distance. Once reset, fingers and hand would resume their usual coupled hand–finger dynamics. Fig. 9 shows this typical two-phase motions after perturbation, plotting the displacement of the proximal joint of the index finger against the distance of the hand to the target ball.

The CDS model, using the distance of the hand to the target for the coupling function $\Psi(\xi_x)$, gives a very good account of this two phase response and is shown overlaid on the data; see Fig. 10(a). Observe further that the trajectories followed by the fingers after perturbations remain within the covariance envelope of the model. This envelope represents the variability of finger motion observed during the *unperturbed* trials. This hence confirms the hypothesis that the fingers resume their unperturbed motion model shortly after responding to the perturbation. This is particularly visible when looking at Fig. 10(b), zoomed in on the part of the trajectories during and just after the perturbation. Three different demonstrations are shown. It can be seen that, irrespective of the state of the fingers ξ_f at the time of perturbation, the finger trajectories tend to follow the mean of the regressive model (which is representative of the mean of the trajectories followed by the human finger during the unperturbed trials) before the perturbation occurs. Just after the perturbations, the fingers then re-open (trajectory goes down) and then close again (trajectory goes up).

4.2.3. Modeling human motion

The previous discussion assessed the fact that the CDS model gives a good account of the qualitative behavior of the fingers's motion after perturbation. We here discuss how, by tuning the two open parameters of the CDS models, namely α and β (see Table 1), we can better reproduce individual trajectories of the fingers for a particular trial and subject.

Table 1
Variation of time taken to recover from perturbation with the instant of perturbation. Note that the total task duration is of the order of 4 s. The values were taken at constant α and hence do not change with the instant of perturbation. This shows the robustness of the proposed method in adapting against perturbations.

T_p (s)	ΔT_{adapt} (s)
1.2	2.82
1.5	2.77
1.8	2.76
2.1	2.80

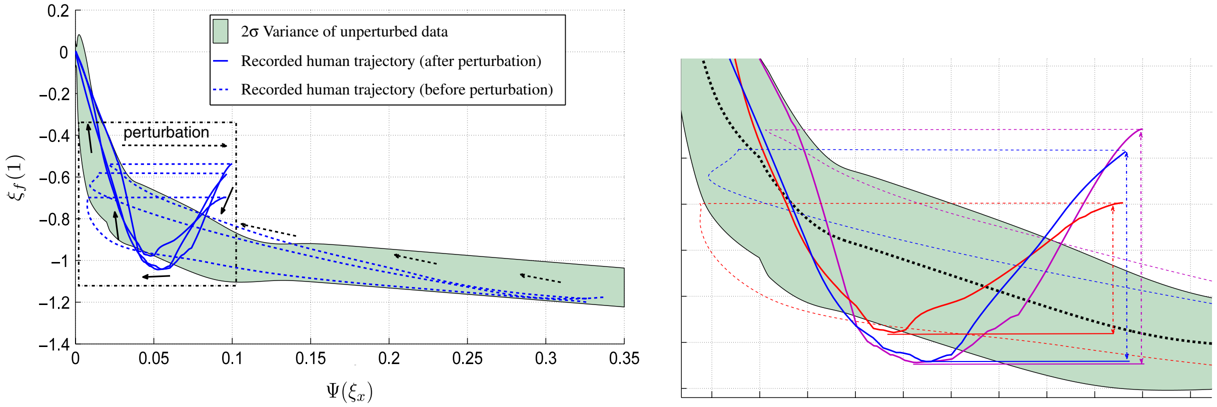
As illustrated in Section 3.3, these two parameters control, respectively, for the speed and amplitude of the motion of the reopening of the fingers after perturbation. Although these parameters can be set arbitrarily in our model, a closer analysis of human data during the perturbed trials shows that one can estimate these parameters by observing the evolution of hand motion *prior* to perturbation. When plotting the average velocity of the hand prior to perturbation and the amplitude of finger reopening, we see that the two parameters are linearly correlated; see Fig. 11(a). Similarly, when plotting the velocity at which fingers reopen against the amplitude of reopening, we see that these two parameters are also linearly correlated. In other words, the faster the hand moves toward the target, the less the fingers reopen upon perturbation. Further, the faster the fingers reopen the larger the amplitude of the reopening of the fingers. Note that while there is a correlation, this correlation is subject dependent. To reproduce human data for a particular trial with CDS, we can hence use the above two observations combined with the fact that α and β control the speed and amplitude of the fingers' motion.

Fig. 11(c) shows that this results in a good qualitative fit of the motion after perturbation. 3 perturbed trials chosen from subject 1 are shown. Similar plots for other subjects can be found in Appendix B. We analyze the quality of the fit by comparing it to the motion obtained from the model with *optimal* parameter values. We find the optimal values of α and β for a particular demonstration by performing a grid search and optimizing the fit between the model generated and demonstrated motion.[4] The fitting is evaluated using the absolute error between the joint angle values (radians) summed over a time window from the instant of perturbation till the end of demonstration. Fig. 12 shows the variation of this error term with α and β. It can be seen that the error first decreases and then increases with progressively increasing α and β. The contours of the error function on variation with α and β are shown in Fig. 12(c).

Note that it is not the aim of this analysis to find the optimal parameters, but to give the reader an idea of how good is the fit obtained from a brute force grid search as compared to what we can infer *prior* to the perturbation. Further, the discrepancy between the model run with inferred parameters and the actual data is only due to the noise in the linear correlation.

It is important to emphasize that the CDS model is built using data from the unperturbed trials and the parameters α and β are inferred from the perturbed trials. It is a representative of a generic pattern of finger–hand coupled dynamics that is present across subjects and trials but that is not subject specific. Further, the estimation of the parameters α and β is done based on an observation of a coupling across variables in a single subject, and is

[4] This estimate of the *optimal* α and β is only accurate up to the width of the grid chosen.



(a) Recorded trajectory.

(b) Zoomed around perturbation.

Fig. 10. (a) shows the data recorded from perturbed human demonstrations. The adaptation behavior under perturbation follows the same correlations between hand position and fingers as in the unperturbed behavior. The region where perturbation is handled is indicated in red and zoomed in (b) where 3 different demonstrations (red, blue and magenta) from the same subject are shown. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

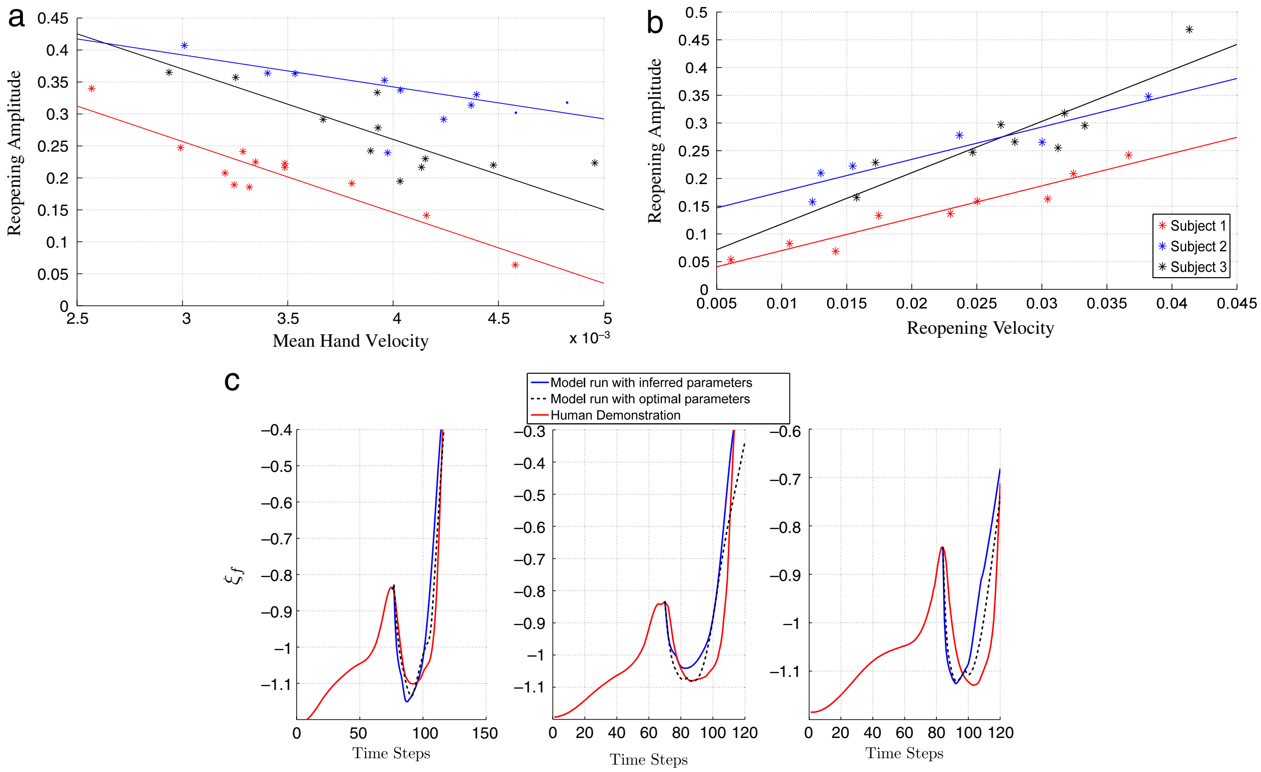


Fig. 11. Correlations deduced from the experiments. (a) Shows the linear correlation found between mean hand velocity prior to perturbation and the amplitude of finger reopening. (b) Shows the same between speed and amplitude of finger reopening. (c) Compares the trajectories obtained using the inferred parameters with the actual human demonstration. Finger motion obtained using optimal parameter values is also shown in black (dashed).

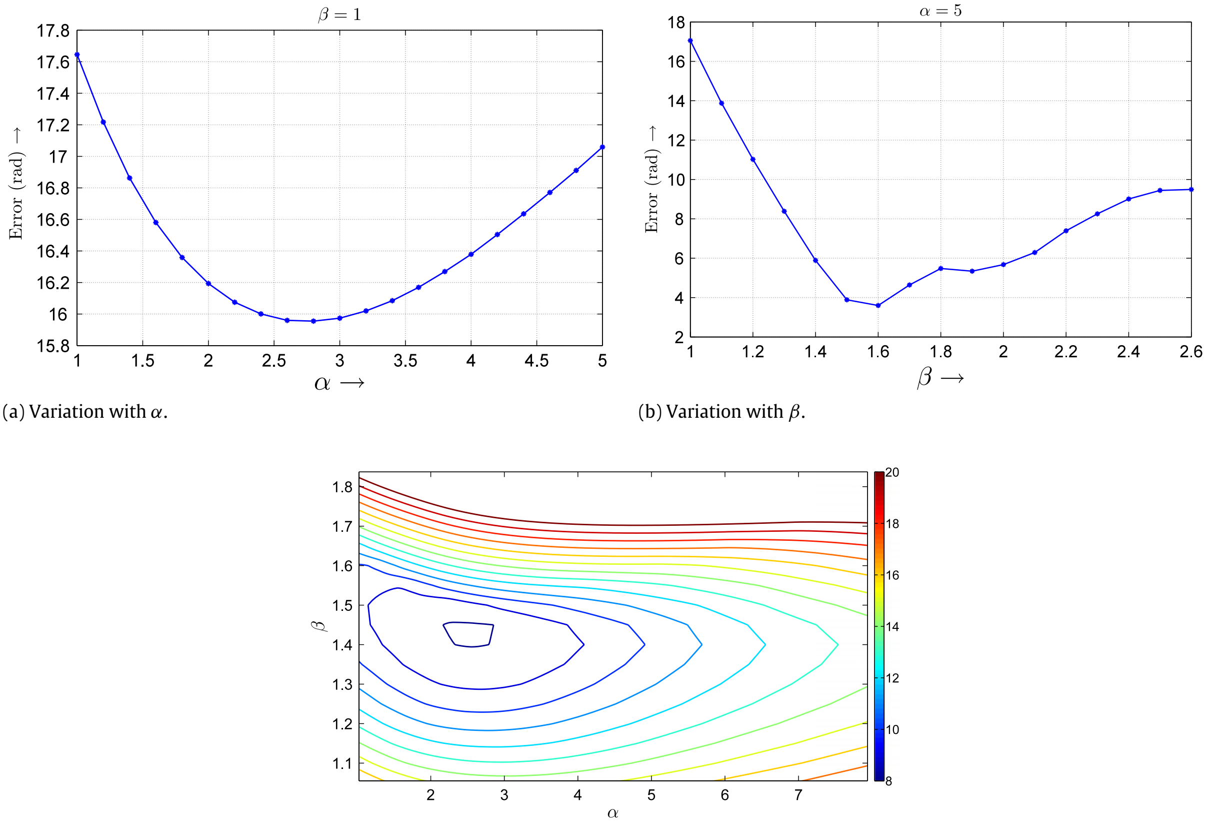
not fitted for a particular trial. The CDS model, hence, encapsulates general patterns of finger–hand motions inherent to human motor control. We discuss next how such human-like dynamics of motion can be used for robust control of hand–finger motion for successful grasp during perturbations.

4.3. *Validation of the model for robot control*

We here test the performance of CDS for robust control of reach and grasp motion in the iCub robot. We first show using the iCub simulator that the approach presented in this work is decidedly better than our previous approach of learning task dynamics using only one dynamical system. It ensures successful task completion under spatial perturbation of the target where the previous approach fails. We also investigate the adaptability of the CDS model in reacting quickly to counter fast perturbations (even when not demonstrated a priori). Finally, we conduct experiments on the iCub robot to validate the ability of the model to adapt on-the-fly reach and grasp motion under various forms of perturbations.

4.3.1. *Comparison with single DS approach*

In Section 3.3, we discussed the fact that the "naive" approach in which one would learn the hand–finger coupling by using a single GMM (the state vector in this case comprises the hand position and finger joints) would likely fail at encapsulating explicitly the

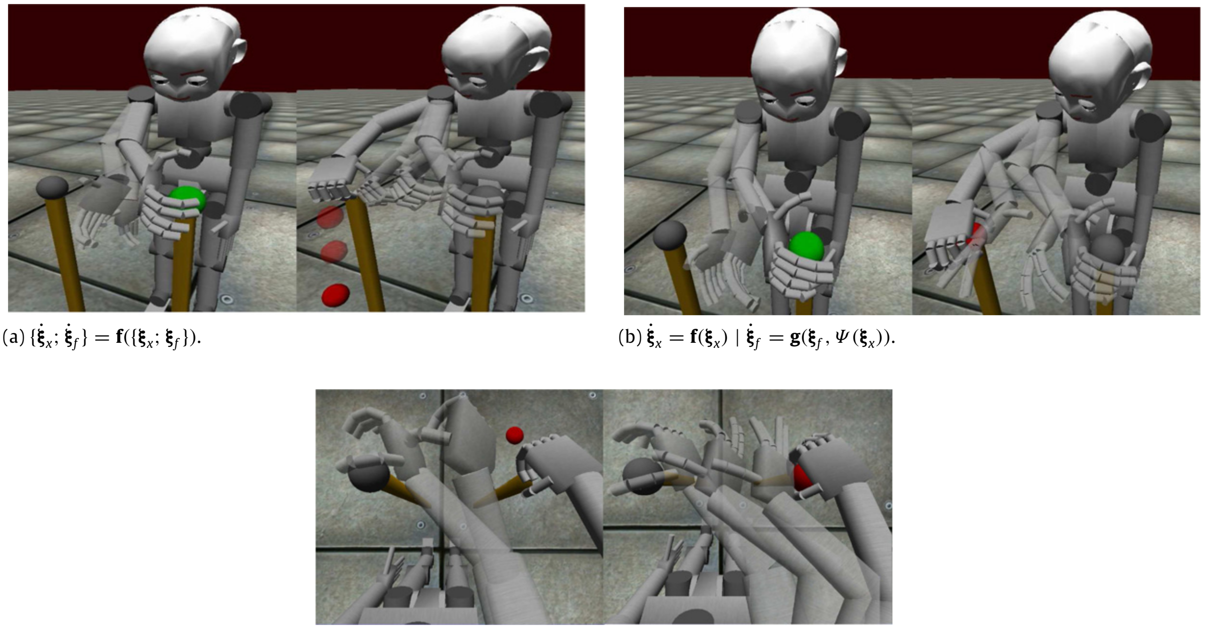


(a) Variation with α.

(b) Variation with β.

(c) Contour plot.

Fig. 12. Effect of changing parameters α and β on the error between model run and mean human demonstrations.



(a) $\{\dot{\boldsymbol{\xi}}_x; \dot{\boldsymbol{\xi}}_f\} = \mathbf{f}(\{\boldsymbol{\xi}_x; \boldsymbol{\xi}_f\})$.

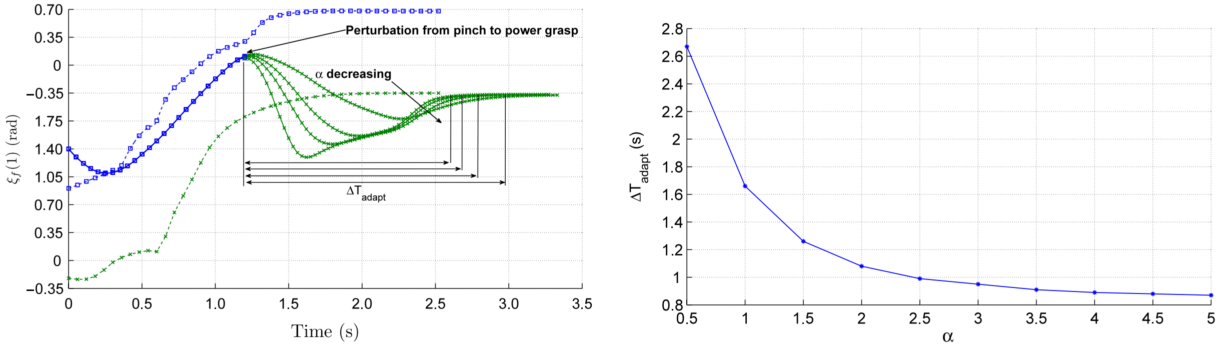
(b) $\dot{\boldsymbol{\xi}}_x = \mathbf{f}(\boldsymbol{\xi}_x) \mid \dot{\boldsymbol{\xi}}_f = \mathbf{g}(\boldsymbol{\xi}_f, \Psi(\boldsymbol{\xi}_x))$.

(c) Hand closeup.

Fig. 13. Reach–grasp task executions with and without explicit coupling. The explicitly coupled execution (b) prevents premature finger closure, ensuring that given any amount of perturbation, formation of the grasp is prevented until it is safe to do so. In the implicitly coupled execution (a), fingers close early and the grasp fails. (c) Shows the closeup of hand motion post perturbation with implicit (*left*) and explicit (*right*) coupling.

correlation between the two dynamics. We had then advocated the use of an explicit coupling function to couple the dynamics of finger and hand motion, each of which are learned through separate GMMs, leading to the CDS model. We here illustrate this in simulation when reproducing the human experiment. The iCub robot first reaches out for the green ball. Midway through the motion, the target is switched and the robot must go and reach for the red ball.

Fig. 13(a) shows that using a single GMM for the hand and finger dynamics fails at embedding properly the correlations between the



(a) Adaptation with varying α.

(b) α vs ΔT_{adapt}.

Fig. 14. (a) Speed of reaction varies with varying α. Adaptation is qualitatively the same, but faster as α increases. (b) shows the variation of recovery time ΔT_{adapt} with α.

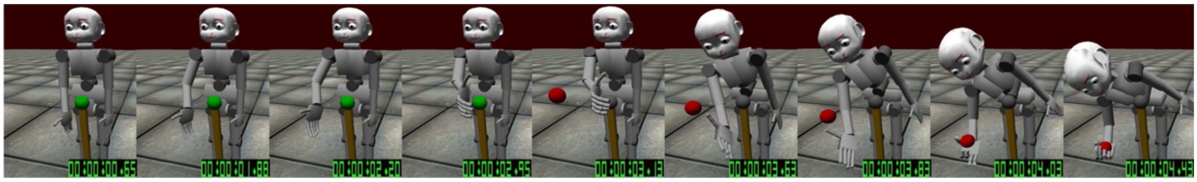


Fig. 15. Fast adaptation under perturbation from palm-down to palm-up power grasp.

reach and grasp sub-systems and does not adapt well the fingers' motion to grasp for the new ball target. Lack of an explicit coupling leads to a poor coordination between fingers and hand motion. As a result, the fingers close too early, leading the ball to fall. Fig. 13(b) shows the same task when performed using the CDS model. The fingers first reopen following the perturbation, hence delaying the grasp formation, and then close according to the correlations learned during the demonstrations, leading to a successful grasp. Fig. 13(c) shows the hand from top view where the re-opening of fingers can be seen clearly in the explicitly coupled task.

4.3.2. *Adaptability to fast perturbations*

An important aspect of encoding motion using autonomous dynamical systems is that it offers a great resilience to perturbations. We here show that this offers robust control in the face of very rapid perturbations.

In Section 4.2, we showed that the two free parameters α and β of the CDS model could be inferred from human data. We then already emphasized the role of these two parameters to control for speed and amplitude of finger reopening. To recall, the larger the α the faster the motion. Hence, executing the task with values for α that differ from that set from human data may be interesting for robot control for two reasons: (a) as robots can move much faster than humans, using larger values for α could exploit the robot's faster reaction times while retaining the coupling between finger and hand motion found in human data, (b) also, using values of α that depart from these inferred from human demonstration may allow to generate better responses to perturbations that send the system to the area of the state space not seen during demonstrations.

Fig. 14(a) illustrates the role that α plays in controlling for the reaction time. As expected, the time it takes for the finger to adapt to perturbation decreases when increasing α (i.e. ΔT_{adapt}) corresponds to the time elapsed between the onset of the perturbation and the time when the finger position re-join the original desired position, i.e. the position the finger should have been in the unperturbed case. Fig. 14(b) plots α against ΔT_{adapt}. Recovery times can be significantly reduced by increasing α. This is the time it takes for the robot to completely recover from the perturbation and reach the target position successfully. It is important to emphasize once more that this trajectory "re-planning" is performed at run-time, i.e within the 20 ms close-loop control of the robot. Again, there is no replanning, adaptation to perturbation results from providing the CDS model with the current position of the finger, hand and target. Importantly, this provides a smooth response that enables the robot to change its trajectory *without stopping to re-plan*.

We illustrate this capacity to adapt to rapid perturbation in an experiment with the iCub robot when the robot must not only adapt the trajectory of its hand but also switch across grasp types; see Fig. 15. Due to hardware constraints on the real platform, we perform this particularly high speed perturbation experiment in the iCub simulator. As the robot moves toward the target located on its left, the target ball suddenly disappears and reappears on the right of the robot. In contrast to our previous experiment, the ball is no longer supported against gravity and hence, starts falling. For the robot to reach and grasp the object before it reaches the floor, the robot has to act very quickly. This requires a fast adaptation from palm-up to palm-down grasp as well as for fingers, while the target keeps on moving. To perform this task, we first trained two separate CDS models to learn two different dynamics for power grasp in palm-up and palm-down configurations, respectively. Learning was done by using five (unperturbed) human demonstrations of this task. During reproduction, the robot initially starts moving toward the target using the CDS model for palm-down configuration grasp. After perturbation, the robot switches to the CDS model for palm-up power grasp.

To ensure that the robot intercepts the falling object in its workspace, we use the approach presented in our previous work on catching flying objects by Kim et al. [50].[5] This allows us to determine the catching point as well as the time it will take the object to reach this point (assuming here a simple free fall for the dynamics of the object). This determines the maximal value for ΔT_{adapt}, which we then use to set the required α to intercept the object in time.

[5] In Kim et al. [50] we had used a single GMM to control for both arm and hand motion. This would not allow to quickly adapt to different grasp on the fly as shown in Section 4.3.1.

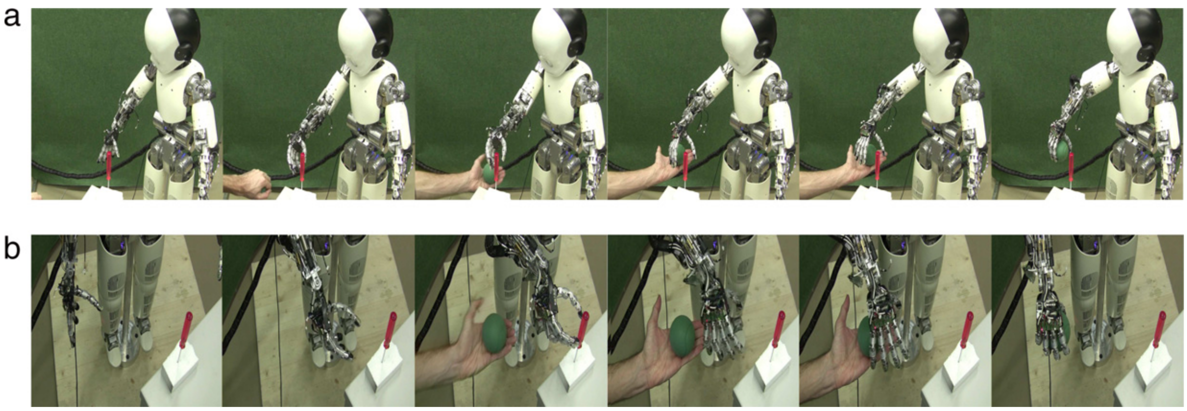


Fig. 16. Validating the model on the real iCub platform. The robot adapts between pinch and power grasps at different spatial positions in real time without any delays for re-planning. (a) and (b) show the same task from front and top view to better visualize the motion of the fingers.

As shown in Fig. 15, switching to the second CDS model ensures that replanning of the finger motion is done in coordination with the hand motion (now redirected to the falling object). Precisely, the orientation of the hand and the finger curl are changed synchronously yielding the hand to close its grasp on the falling object at the right time. Notice that as the distance-to-target suddenly increases, the CDS model forces the fingers to reopen. Subsequently, the fingers close proportionally as the distance between the falling ball and the robot hand decreases, hence maintaining the correlations seen during the demonstrations. Note that to generate this task, we control also for the torso (adding two more variables to the inverse kinematics so as to increase the workspace of the robot).

4.3.3. Switching between different grasp types

Here, we perform another experiment showing the ability of our system to adapt the *fingers' configuration* (in addition to adapting the fingers' dynamics of motion) so as to switch between pinch and power grasps. We learn two separate CDS models for pinch–grasp of a thin object (screw-driver) and power grasp of a spherical object, respectively, from five demonstrations of each task during unperturbed trials.

Fig. 16 illustrates the experiment. While the robot reaches for the thin object, pre-shaping its fingers to the learned pinch–grasp, we suddenly present the spherical object in the robot's field of view. The robot then redirects its hand to reach for the spherical object in place of the thin one. Since this experiment does not require very rapid reacting time, the experiment could be conducted on the real iCub robot. In this experiment, the two objects are color-tracked using the iCub's on-board cameras. Change of target is hardcoded. As soon as the green object is detected in the cameras, the target location is switched from the red object to the green one and the robot's CDS model is switched accordingly.

Fig. 17 shows the motion of robot's index finger proximal joint as it adapts to the induced perturbation. During the first phase of the motion, the finger closes rapidly so as to yield a pinch–grasp. After perturbation, the fingers reopen to yield the power grasp that would better accommodate the spherical object. The robot smoothly switches from following the pinch–grasp model requiring smaller hand aperture (i.e. larger joint value) to the power grasp model which requires a larger aperture (smaller joint value) by reopening the fingers and subsequently closing them on the target, thereby, completing the task successfully.

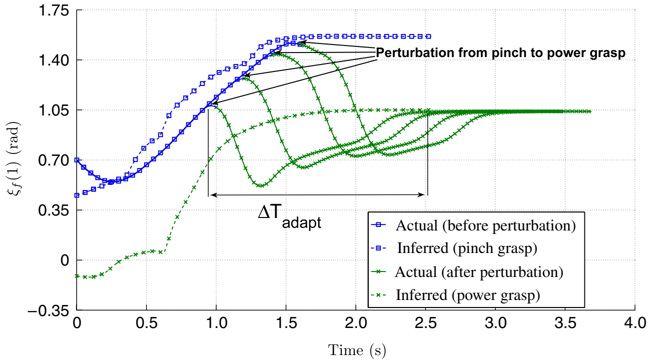


Fig. 17. Motion of one finger joint angle under grasp-type perturbation as recorded from the simulation. The inferred joint positions predicted by the models, both the grasp models (power and pinch), are shown in dotted. The adaptation is smooth and robust w.r.t the instant when perturbation was applied. Time taken to recover from perturbation (ΔT_{adapt}) remains constant.

While we discussed in the previous section the advantage to adapt the parameter α when one needs to perform tasks that require very high reaction times (reaction times that are higher than what humans could achieve), we here show that, using the α parameter inferred from human data is sufficient when required reaction times are sufficiently slow. We emphasize once more the benefit of the model to achieve very robust behavior in the face of various perturbations. To this end, we perform two variants on the task described above where we introduce perturbations.

First, we run the same switching tasks but present the spherical objects at different instants after the onset of motion. The perturbation instants vary from middle of the task duration to almost completion of the task. Table 1 gives the recovery time and time instant of perturbation. T_p denotes the instant at which the perturbation was introduced. Since we do not change the value of the parameter α, at each run, the time taken by the robot to recover from the perturbation remains the same. Fig. 17 shows the resulting trajectories for the index finger. Even when the perturbation occurs shortly before the completion of pinch–grasp, the model readapts the grasp smoothly, yielding a correct grasp at the second object.

Second, to highlight the performance of CDS to adapt continuously and on the fly control for coordinated motion of hand and fingers on the real iCub robot, we introduce perturbation during the first part of the previous task in which the iCub robot reaches with a pinch–grasp (here the robot reaches for a glass of wine[6]);

[6] The location of the grasping point on the glass of wine is indicated by a red patch that is detected through two external cameras running at 100 Hz.

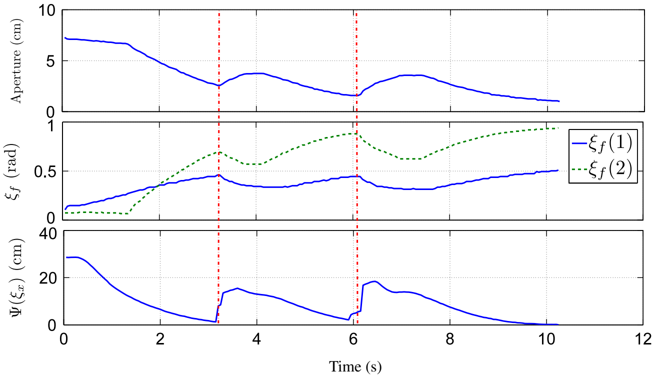


Fig. 18. Coordinated hand–arm motion while the robot was perturbed multiple times in different directions. Finger joint angles, and hence, the hand aperture, change according to the learned correlations. Vertical red line marks the instants at which the robot was perturbed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

see Fig. 18. To introduce perturbations on-the-fly during execution of the pinch–grasp, we implement a reflex behavior using the iCub's skin touch sensors on the forearm, such that, when the robot detects a touch on its forearm, it immediately moves its arm away from the point where it was touched. This reflex overlays the CDS controller. When the CDS controller takes over again, it uses the new position of the arm to predict the new finger and hand motion.

Fig. 18 shows the displacement along time of the proximal and distal finger joints of the index finger, hand aperture and the distance-to-target, when the robot is solely reaching with a pinch–grasp and is being perturbed once on its way toward the object. The hand aperture is computed as the distance between the tips of the thumb and index fingers. As expected, as the hand is moved away from the target, the fingers reopen in agreement with the correlations learned in the CDS model and then close into pinch–grasp on the object. The finger motion robustly adapts to the perturbation, changing the hand aperture in coordination with the perturbed hand position and finally reaches the target state for the pinch–grasp with ≈ 1 cm hand aperture. Fig. 19 shows one cycle of tactile perturbation on the real robot.

5. Conclusion and discussion

In this paper, we presented a model for encoding and reproducing different reach-to-grasp motions that allows to handle fast perturbations in real-time. We showed that this capacity to adapt without re-planning could be used to allow to switch across grasp types smoothly. The model was strongly inspired from the way humans adapt reach and grasp motion under perturbation. Human data was used to determine a generic coupling between control of hand transport and finger aperture. It was also used to determine quantitative values for this coupling.

We first showed that the model gives a good qualitative account of human reach and grasp motions during perturbation. We then showed through both simulation and real robot experiments that the CDS model provides a robust controller for a variety of reach and grasp motions in robots.

Importantly, we showed that, while human behavior is a good source of inspiration, one can depart from this model, by tuning the parameters of the model, to induce better robot performance if needed and for tasks which humans struggle to perform. This follows a trend in Programming by Demonstration that emphasizes the fact that what is good for the human is not necessarily good for the robot (biological inspiration should be taken with a grain of salt).

In this work, biological inspiration pertains to our use of coupled dynamical systems to control for hand motion and finger motion. We showed that, introducing such an explicit coupling was advantageous over a more implicit coupling that could be learned with other density based methods for estimating the correlations across all the variables. The two free parameters of the CDS model offer a variety of ways in which the model can be adapted to realize motion that are optimal for the particular robot platform or the particular task.

5.1. Shortcomings and future work

In this work, we have shown that the model parameters can be tuned in order to either generate "human-like" motion or to induce "un-human-like" very fast reactions. Determining which parameter to use is task and platform dependent. One could, like we did here, learn how reaction times vary as a function of the open parameter α and choose the optimal α taking into consideration task and platform-related constraints, such as minimizing jerk, choosing a time window large enough to ensure successful task completion. The advantage of having such simple and explicit parametrization of the model is that it allows to reuse the same model and adapt the speed to different tasks, e.g. a power grasp may be realized in a very rapid manner when one must hurriedly grasp a support to prevent a fall. It may on the other hand be performed very slowly when grasping a raw egg.

In this work, we have assumed that there exists only one way in which the object can be grasped and that this corresponds to the way demonstrated by the human. This assumption makes it difficult to complete the grasp if the object can be grasped in multiple ways and is presented in a pose different from the one seen during the demonstrations, or when the robot does not have the same dexterity as the human. By construction, the CDS model allows to have a single attractor point and hence it is constrained to yield a single grasp pose. We showed that one could switch across different CDS models and one could hence consider learning different CDS models for each possible grasp pose. A secondary mechanism, e.g. based on measuring the closest distance between the current posture and the variety of grasp pose could determine which CDS model to activate and when. Our current work is investigating how the CDS model could be extended to learn not just a single attractor point but an attractor surface. Indeed, most objects can be grasped at different points along their graspable surface.

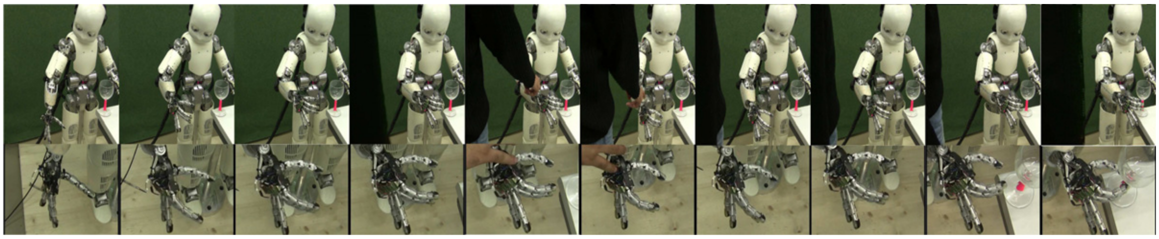
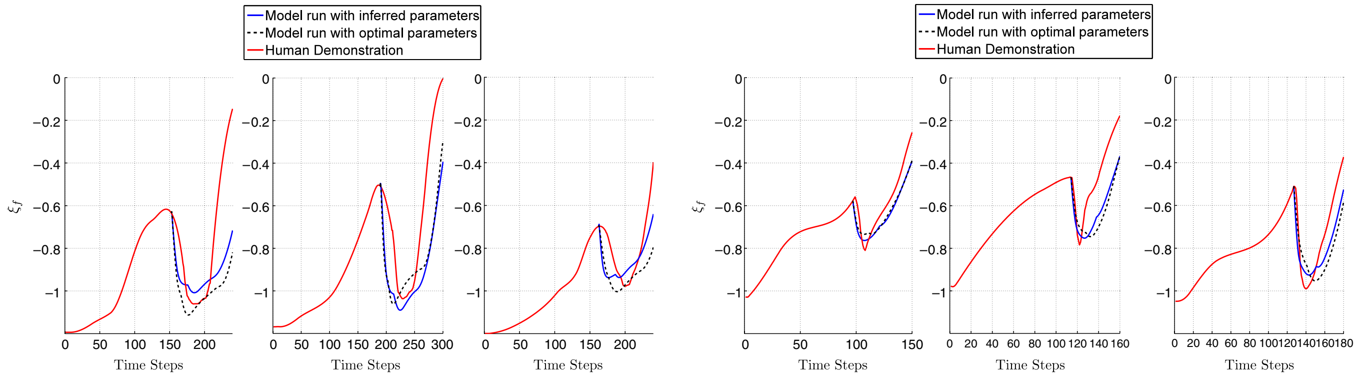


Fig. 19. Spatio-temporal perturbations created using the tactile interface of the iCub. Top row shows the motion of the robot. Bottom row shows closeup of the hand.



(a) Subject 2.

(b) Subject 3.

Fig. B.20. Comparison of model run with inferred parameter values, optimal parameter values and the actual human demonstrations under perturbation.

This work ignored the notion of obstacle. In this paper, perturbations related to either a displacement of the target or to a displacement of the end-effector during task completion. Obstacle avoidance is certainly a major source of disturbances during reach-and-grasp tasks and current planning techniques offer advanced and robust solutions to trajectory planning that can deal with very cluttered environments. Embedding obstacle avoidance while retaining the time-invariance property of autonomous dynamical systems which we exploit here would provide an interesting extension to the present model and would offer an interesting alternative to planning techniques. Even the quickest planning techniques still require planning times of the order of 0.5 s [20] when considering obstacles. This is too slow when the required reaction times are of the order of only a few milliseconds (such as when reaching for a falling object as shown in this paper). Current work in our group is investigating the possibility to embed obstacle avoidance in the scheme proposed by the CDS model.

Throughout the work presented here we have assumed that we had access to an efficient inverse kinematic (IK) solver. Thanks to which, we were able to learn the models of the task and control the robot in the operational space. This is however too restrictive and creates a major drawback in practice. While we advocated that the CDS model needs no replanning, we were still running at a 20 ms close-loop. Most of these 20 ms were devoted to computing the IK. Indeed, computing the output of the regressive models of CDS requires less than a couple of ms. Another obvious drawback of controlling in task space is that the motion in task space inferred by the model may not always yield a feasible solution in joint space, especially when transferring human data to robots, and when a perturbation sends the motion very far from the motion demonstrated. In the experiments presented here, we did not encounter this problem as the task was always chosen so as to remain within the center of the robot's workspace.

Finally, the nature of the coupling assumed in our model is only one directional, i.e., a perturbation in the reaching motion is reflected in the finger motion and not the other way round. A typical case where such a coupling is useful is when the dynamic controller for the fingers is malfunctioning and is too slow to follow the desired trajectory. In such a scenario, the arm must also slow down and synchronize with the finger motion. Such a behavior, even if not biologically inspired, is desirable in the context of robotics where controller noise is ubiquitous. Although we did not explicitly include this in our experiments, the CDS model is actually capable in its current formulation, to handle bidirectional coupling. This is because the learned GMM is a joint distribution of distance-to-target and finger configuration. One could take the conditional on either of the variables (in the presented scheme, we take it on distance-to-target). However, it needs to be studied what effect (in terms of stability) this bi-directional dependency will have on the overall system.

Acknowledgments

This work was supported by EU Project *First-MM(FP7/2007–2013)* under grant agreement number 248258.

The authors thank Florent D'Halluin and Jean-Baptiste Keller for the help in setting up the experiments on the iCub robot.

Appendix A. Stability of CDS model

To prove that the CDS model indeed follows the conditions (9), we use the properties of its individual components. For simplicity, we shift all the data into the goal reference frame so that $\boldsymbol{\xi}_x^* = \boldsymbol{\xi}_f^* = 0$. The condition (9a) holds true due to the global stability of SEDS. To investigate the stability of the coupling, we consider

$$\begin{aligned}\lim_{t\to\infty} \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_f \,|\Psi\left(\boldsymbol{\xi}_x\right)\right] &= \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_f \middle|\Psi\left(\lim_{t\to\infty}\boldsymbol{\xi}_x\right)\right] \\ &= \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_f \middle|\lim_{\boldsymbol{\xi}_x\to\mathbf{0}}\Psi\left(\boldsymbol{\xi}_x\right)\right] \text{(By (9a))} \\ &= \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_f \,|\mathbf{0}\right] \quad \text{(By (7))} \\ &= \mathbf{0} \quad \text{(By (8))}. \end{aligned} \tag{A.1}$$

The model which governs the evolution of the coupled variable $\boldsymbol{\xi}_f$ is given by

$$\dot{\boldsymbol{\xi}}_f = \mathbb{E}\left[\dot{\boldsymbol{\xi}}_f \middle|\left(\boldsymbol{\xi}_f - \tilde{\boldsymbol{\xi}}_f\right)\beta\right].$$

Taking the limiting values and using (A.1), we get

$$\lim_{t\to\infty} \dot{\boldsymbol{\xi}}_f = \mathbb{E}\left[\dot{\boldsymbol{\xi}}_f \,|\beta\boldsymbol{\xi}_f\right]$$

which is again globally asymptotically stable due to SEDS. However, as seen from Algorithm 1, the multiplier α boosts the velocity before incrementing the state. It is trivial to see that this does not affect the global asymptotic behavior of the model since negative definite $\frac{A+A^T}{2} \Rightarrow \alpha\frac{A+A^T}{2}$ is also negative definite for $\alpha > 0$. For details on why such a condition is required for global stability, the reader is referred to [9].

Appendix B. Model performance with inferred parameter values

See Fig. B.20.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at doi:10.1016/j.robot.2011.07.023.

References

[1] D. Berenson, S. Srinivasa, D. Ferguson, A. Collet, J. Kuffner, Manipulation planning with workspace goal regions, in: IEEE International Conference on Robotics and Automation. ICRA, IEEE, 2009, pp. 618–624.

[2] A. Ijspeert, J. Nakanishi, S. Schaal, Movement imitation with nonlinear dynamical systems in humanoid robots, in: IEEE International Conference on Robotics and Automation. Proceedings. ICRA, vol. 2, IEEE, 2002, pp. 1398–1403.

[3] A. Miller, S. Knoop, H. Christensen, P. Allen, Automatic grasp planning using shape primitives, in: IEEE International Conference on Robotics and Automation. Proceedings. ICRA, vol. 2, IEEE, 2003, pp. 1824–1829.

[4] G. Bone, A. Lambert, M. Edwards, Automated modeling and robotic grasping of unknown three-dimensional objects, in: IEEE International Conference on Robotics and Automation. ICRA, IEEE, 2008, pp. 292–298.

[5] P. Pastor, H. Hoffmann, S. Schaal, Movement generation by learning from demonstration and generalization to new targets, in: Adaptive Motion of Animals and Machines AMAM.

[6] L. Dan, E. Todorov, Hierarchical optimal control of a 7-dof arm model, in: IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL '09, IEEE, 2009, pp. 50–57.

[7] D. Grimes, D. Rashid, R. Rao, Learning nonparametric models for probabilistic imitation, Advances in Neural Information Processing Systems 19 (2007) 521.

[8] E. Gribovskaya, A. Billard, Learning nonlinear multi-variate motion dynamics for real-time position and orientation control of robotic manipulators, in: Proceedings of IEEE-RAS International Conference on Humanoid Robots, IEEE, 2009, pp. 472–477.

[9] S. Khansari-Zadeh, A. Billard, Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming, in: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, IEEE, 2010, pp. 2676–2683.

[10] U. Castiello, K. Bennett, H. Chambers, Reach to grasp: the response to a simultaneous perturbation of object position and size, Experimental Brain Research 120 (1998) 31–40.

[11] M. Gentilucci, S. Chieffi, M. Scarpa, U. Castiello, Temporal coupling between transport and grasp components during prehension movements: effects of visual perturbation, Behavioural Brain Research 47 (1992) 71–82.

[12] L. Jakobson, M. Goodale, Factors affecting higher-order movement planning: a kinematic analysis of human prehension, Experimental Brain Research 86 (1991) 199–208.

[13] M. Jeannerod, The timing of natural prehension movements, Journal of Motor Behavior 16 (1984) 235.

[14] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Survey: robot programming by demonstration, in: Handbook of Robotics, 2008 (Chapter 59).

[15] Y. Paulignan, M. Jeannerod, C. MacKenzie, R. Marteniuk, Selective perturbation of visual input during prehension movements, Experimental Brain Research 87 (1991) 407–420.

[16] U. Castiello, K. Bennett, G. Stelmach, Reach to grasp: the natural response to perturbation of object size, Experimental Brain Research 94 (1993) 163–178.

[17] M. Kawato, Internal models for motor control and trajectory planning, Current Opinion in Neurobiology 9 (1999) 718–727.

[18] D. Engstrom, J. Kelso, Coordination dynamics of the complementary nature, Gestalt Theory 30 (2008) 121.

[19] M. Christel, A. Billard, How monkeys morphology constrain natural prehension kinematics. Unconstrained conditions and simulation, Behavioural Brain Research 131 (2001) 169–184.

[20] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, R. Dillmann, Humanoid motion planning for dual-arm manipulation and re-grasping tasks, in: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, 2009, pp. 2464–2470.

[21] K. Harada, K. Kaneko, F. Kanehiro, Fast grasp planning for hand/arm systems based on convex model, in: IEEE International Conference on Robotics and Automation. ICRA, 2008.

[22] J.-H. Bae, S. Arimoto, Y. Yamamoto, H. Hashiguchi, M. Sekimoto, Reaching to grasp and preshaping of multi-dofs robotic hand-arm systems using approximate configuration of objects, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 14, IEEE, 2006, pp. 1605–1610.

[23] M. Gienger, M. Toussaint, C. Goerick, Task maps in humanoid robot manipulation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, 2008, pp. 2758–2764.

[24] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, A.Y. Ng, Reactive grasping using optical proximity sensors, in: IEEE International Conference on Robotics and Automation, 2009. ICRA '09, Institute of Electrical and Electronics Engineers, 2009, pp. 2098–2105.

[25] J.-P. Saut, A. Sahbani, S. El-Khoury, V. Perdereau, Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2007, pp. 2907–2912.

[26] A. Gasparri, G. Oliva, S. Panzieri, Path planning using a lazy spatial network PRM, in: 17th Mediterranean Conference on Control and Automation. MED, IEEE, 2009, pp. 940–945.

[27] A. Morales, T. Asfour, P. Azad, S. Knoop, R. Dillmann, Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2007, pp. 5663–5668.

[28] D. Kragic, A. Miller, P. Allen, Real-time tracking meets online grasp planning, in: IEEE International Conference on Robotics and Automation. Proceedings. ICRA, vol. 3, IEEE, 2001, pp. 2460–2465.

[29] S. LaValle, J. Kuffner, Rapidly-exploring random trees: Progress and prospects, in: Algorithmic and Computational Robotics: New Directions: The Fourth Workshop on the Algorithmic Foundations of Robotics, AK Peters, Ltd., 2001, p. 293.

[30] J. Vilaplana, J. Batlle, J. Coronado, A neural model of hand grip formation during reach to grasp, in: IEEE International Conference on Systems, Man and Cybernetics, vol. 1, IEEE, 2005, pp. 542–546.

[31] M. Gentilucci, U. Castiello, M. Corradini, M. Scarpa, C. Umilta, G. Rizzolatti, Influence of different types of grasping on the transport component of prehension movements, Neuropsychologia 29 (1991) 361–378.

[32] M. Saling, S. Mescheriakov, E. Molokanova, G. Stelmach, M. Berger, Grip reorganization during wrist transport: the influence of an altered aperture, Experimental Brain Research 108 (1996) 493–500.

[33] A. Mitz, M. Godschalk, S. Wise, Learning-dependent neuronal activity in the premotor cortex: activity during the acquisition of conditional motor associations, Journal of Neuroscience 11 (1991) 1855.

[34] R. Meulenbroek, D. Rosenbaum, C. Jansen, J. Vaughan, S. Vogt, Multijoint grasping movements, Experimental Brain Research 138 (2001) 219–234.

[35] Y. Paulignan, C. MacKenzie, R. Marteniuk, M. Jeannerod, The coupling of arm and finger movements during prehension, Experimental Brain Research 79 (1994) 431–435.

[36] J. Glke, N.J. Wachter, T. Geyer, H. Schll, G. Apic, M. Mentzel, Motion coordination patterns during cylinder grip analyzed with a sensor glove, The Journal of Hand Surgery 35 (2010) 797–806.

[37] S. Bae, T.J. Armstrong, A finger motion model for reach and grasp, International Journal of Industrial Ergonomics 41 (2011) 79–89.

[38] A. Ulloa, D. Bullock, A neural network simulating human reach-grasp coordination by continuous updating of vector positioning commands, Neural Networks 16 (2003) 1141–1160.

[39] B. Hoff, M. Arbib, Models of trajectory formation and temporal interaction of reach and grasp, Journal of Motor Behavior 25 (1993) 175–192.

[40] E. Oztop, M. Arbib, Schema design and implementation of the grasp-related mirror neuron system, Biological Cybernetics 87 (2002) 116–140.

[41] P. Haggard, A. Wing, Coordinated responses following mechanical perturbation of the arm during prehension, Experimental Brain Research 102 (1995) 483–494.

[42] A. Ulloa, D. Bullock, A neural circuit for coordinating reaching with grasping: autocompensating variable initial apertures, perturbations to target size, and perturbations to target orientation, in: International Joint Conference on Neural Networks. Proceedings. IJCNN, vol. 2, IEEE, 2002, pp. 1047–1052.

[43] D. Bullock, S. Grossberg, The VITE model: a neural command circuit for generating arm and articulator trajectories, Dynamic Patterns in Complex Systems (1988) 305–326.

[44] J. Hwang, R. Arkin, D. Kwon, Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control, in: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS. Proceedings, vol. 2, IEEE, 2003, pp. 1444–1449.

[45] M. Keshmiri, M. Keshmiri, A. Mohebbi, Augmented online point to point trajectory planning, a new approach in catching a moving object by a manipulator, in: 8th IEEE International Conference on Control and Automation ICCA, IEEE, 2010, pp. 1349–1354.

[46] S. Schaal, C. Atkeson, Constructive incremental learning from only local information, Neural Computation 10 (1998) 2047–2084.

[47] S. Schaal, S. Kotosaka, D. Sternad, Nonlinear dynamical systems as movement primitives, in: Humanoids. First IEEE-RAS International Conference on Humanoid Robots, CD-Proceedings, 2000.

[48] C. Eppner, J. Sturm, M. Bennewitz, C. Stachniss, W. Burgard, Imitation learning with generalized task descriptions, in: ICRA'09, IEEE (2009) 3968–3974.

[49] D.H. Grollman, A. Billard, Donut as I do: learning from failed demonstrations, in: International Conference on Robotics and Automation, IEEE, Shanghai, 2011.

[50] S. Kim, E. Gribovskaya, A. Billard, Learning motion dynamics to catch a moving object, in: 10th IEEE-RAS International Conference on Humanoid Robots, IEEE, 2010.

Ashwini Shukla is a Ph.D. student in the Learning Algorithms and Systems Laboratory (LASA) at the Swiss Federal Institute of Technology in Lausanne (EPFL). He received his Bachelors and Masters degree in Mechanical Engineering (2009) from the Indian Institute of Technology Kanpur (IITK), India. His research interests focus upon Programming by Demonstration and machine learning techniques for robot control and fast grasp planning techniques.

Aude G. Billard is Associate Professor and head of the LASA Laboratory at the School of Engineering at the Swiss Federal Institute of Technology in Lausanne (EPFL). Prior to this, she was Research Assistant Professor at the Department of Computer Sciences at the University of Southern California, where she retains an adjunct faculty position to this day. Aude Billard received a B.Sc. (1994) and M.Sc. (1995) in Physics from EPFL, with specialization in Particle Physics at the European Center for Nuclear Research (CERN), an M.Sc. in Knowledge based Systems (1996) and a Ph.D. in Artificial Intelligence (1998) from the Department of Artificial Intelligence at the University of Edinburgh. Her research interests focus on machine learning tools to support robot learning through human guidance. This extends also to research on complementary topics, including machine vision and its use in human–machine interaction and computational neuroscience to develop models of learning in humans.