Coupled dynamical system based hand-arm grasp planning under real-time perturbations

Ashwini Shukla and Aude Billard

Learning Algorithms and Systems Laboratory (LASA)
Ecole Polytechnique Federale de Lausanne (EPFL)
Lausanne, Switzerland - 1015
Email: {ashwini.shukla;aude.billard}@epfl.ch

Abstract—Robustness to perturbation has been advocated as a key element to robot control and efforts in that direction are numerous. While in essence these approaches aim at "endowing robots with a flexibility similar to that displayed by humans", few have actually looked at how humans react in the face of fast perturbations. We recorded the kinematic data from human subjects during grasping motions under very fast perturbations. Results show a strong coupling between the reach and grasp components of the task that enables rapid adaptation of the fingers in coordination with the hand posture when the target object is perturbed. We develop a robot controller based on *Coupled Dynamical Systems* that exploits coupling between two dynamical systems driving the hand and finger motions. This offers a compact encoding for a variety of reach and grasp motions that adapts on-the-fly to perturbations without the need for any re-planning. To validate the model we control the motion of the iCub robot when reaching for different objects.

I. INTRODUCTION

Performing manipulation tasks interactively in real environments requires a high degree of accuracy and stability. At the same time, except in completely determined and static environments, machine perception of the environment may suffer from real-time perturbation. To handle these, it requires flexibility on the part of the robot. These considerations make the task of reaching to grasp under fast perturbations difficult to deal with. Planning of constrained grasping motions has often been studied as two separate problems of grasping [15, 3] and generating arm-motion [2, 11] i.e., decoupling the reach and grasp components. The high dimensionality and complexity of these problems has discouraged the use of a single coherent framework for carrying out both the tasks. Constrained motion planning for only the reaching motion in high dimensional arm-hand systems is a challenging problem itself and requires the use of predefined heuristics. In Programming by Demonstration (PbD), these heuristics are embedded in the demonstrations provided by a human agent which can be used to generate a generalized task description [4].

In previous work Gribovskaya and Billard [9] have used PbD to encode the dynamics of a task as a Dynamical System (DS) where the end effector of the robot moves under the influence of an attractor positioned at the target. It has been shown that such an approach ensures reproduction of the learned behavior in a generalized manner while efficiently
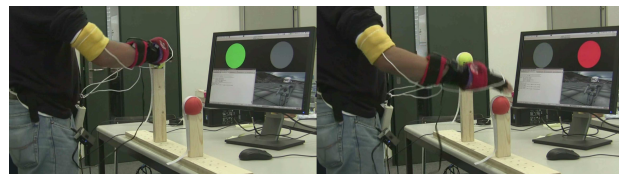

Fig. 1. Experimental setup to record human behavior under perturbations. Two fixed targets are accompanied by the on-screen target selector which activates different targets by changing colors and hence create a spatial perturbation. The subject wears motion sensors and data glove which record the whole arm-hand kinematics at fixed time intervals. The kinematic values are simultaneously transferred to the iCub simulator for visualization of the task.

handling spatial and temporal perturbations. In this work, we address the problem of having a combined encoding scheme for both reaching and grasping behaviors capable of adapting against realtime perturbations during task reproduction. It is worth mentioning here that this entails more complexity than two independent attractor based tasks. Human neurophysiological studies [5, 7] have shown that apart from the parallel evolution of hand transport and preshape, there exist convergence constraints and correlations between the two processes. In addition to the advantage of "human-like" motion, it is critical to maintain the coupling between hand and arm motion while performing a task in order to ensure successful grasp formation at the target. In addition to studying the correlations in the unperturbed human behaviors of performing a grasping task, we also study perturbed demonstrations of human subjects in order to biologically inspire our approach of handling perturbations. We present the Coupled Dynamical System (CDS) model which ensures that the motion constraints mentioned above are respected and at the same time ensures very fast adaptation under perturbations.

II. RELATED WORK

Classically, the behavior of reaching to grasp objects has been treated as two different problems. Especially in motion planning works, reaching to a pre-grasp pose and motion of the hand and fingers are considered as independent processes and are triggered sequentially [2, 10, 18]. Although both the issues of reaching to a pre-grasp pose and formation of grasp around arbitrary objects are intensively studied, very few [1, 8]

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. Classical approaches use probabilistic roadmap and its variants [6, 17] which assume a predefined and static environment and are unable to handle any perturbations at the run-time. Moreover, they require huge preprocessing steps and are not suitable for replanning of grasping tasks online. More recently, LaValle and Kuffner [14] proposed *rapidly exploring random trees* RRTs as a faster alternative to manipulation planning problems, provided the existence of an efficient inverse kinematic (IK) solver. RRT based methods [2, 18] are currently the fastest online planners which exploit the efficient searching ability of RRTs in order to find suitable paths between the start and the goal configurations. Note that a major drawback of all graph based approaches is that they lose to retain any explicit correlations that exist between the two processes of hand transport and preshape as naturally performed by humans.

The concept of coupling between the reach and grasp motions is inspired by evidence in physiological studies [5, 16]. The most frequently reported mechanism suggests a parallel, but time-coupled evolution of the reach and grasp motions. However, directly mimicking this behavior makes the system time-dependent and hence unlikely to handle temporal perturbations.

In this work, we use PbD as a means to encode the dynamics of reach-grasp motion and the coupling information in a compact way. We present the Coupled Dynamical System (CDS) task modeling approach to coordinate the motions of hand transport and preshape without being explicitly dependent on the flow of time. We also show that this method efficiently handles on-the-fly perturbations by incorporating information from perturbed human demonstrations. We conduct experiments on the iCub robot which show that this coupling is necessary in order to successfully complete the tasks under perturbations of different types.

III. METHODOLOGY

In this section, we start with a short formalism of autonomous dynamical system (DS) in the context of robotic manipulation tasks and its estimation with a Gaussian Mixture Model (GMM). For more details on this the reader is referred to Gribovskaya and Billard [9]. We present an extension to this formulation and introduce the notion of coupling between different DS by the means of a coupling function. A formal discussion of the CDS model is presented describing the modeling process and regression algorithm to reproduce the task. A simple 2D example is presented to establish intuitive understanding of the working of the CDS model.

A. Preliminaries

Consider a state vector $\boldsymbol{\xi}(t) \in \mathbb{R}^d$ which can be used to uniquely define the state of the robot while performing a task (e.g. joint angles, position and orientation of the end-effector etc.). Let there be N demonstrations of the task where the state

TABLE I
GAUSSIAN MIXTURE REGRESSION

Let us assume that a process is characterized by some output $\boldsymbol{\xi}_O$ produced by input $\boldsymbol{\xi}_I$ at each time step and our aim is to have a regression model for the same inputs and outputs. To this end, we model the joint probability of input and output variables using Gaussian Mixtures. The probability that a full data point $\boldsymbol{\xi} = [\boldsymbol{\xi}_I; \boldsymbol{\xi}_O]$ is generated by this process is defined by

$$
\begin{aligned}
\mathcal{P}(\boldsymbol{\xi}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} \pi^k \mathcal{N}(\boldsymbol{\xi}_I, \boldsymbol{\xi}_O; \boldsymbol{\theta}^k) \\
&= \sum_{k=1}^{K} \pi^k \frac{1}{\sqrt{(2\pi)^{2d}|\Sigma^k|}} e^{-\frac{1}{2}(\boldsymbol{\xi}-\boldsymbol{\mu}^k)^T (\Sigma^k)^{-1} (\boldsymbol{\xi}-\boldsymbol{\mu}^k)^T}
\end{aligned}
$$

where $\boldsymbol{\theta}^k$ contains the parameters viz. priors π^k, centers $\boldsymbol{\mu}^k$ and covariances Σ^k to define the gaussian distribution $\mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\theta}^k)$ for the kth gaussian. Separating the input and output components, the parameters can be further represented as

$$
\boldsymbol{\mu}^k = \begin{pmatrix} \boldsymbol{\mu}_I^k \\ \boldsymbol{\mu}_O^k \end{pmatrix} \quad , \quad \Sigma^k = \begin{pmatrix} \Sigma_I^k & \Sigma_{IO}^k \\ \Sigma_{OI}^k & \Sigma_O^k \end{pmatrix}.
$$

Gaussian Mixture Regression allows to compute for a given input variable $\boldsymbol{\xi}_I$ and a given component k, the expected distribution of $\boldsymbol{\xi}_O$ as

$$
\mathcal{P}\left(\boldsymbol{\xi}_O | \boldsymbol{\xi}_I ; \boldsymbol{\theta}^k\right) \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}^k, \hat{\Sigma}^k\right)
$$

where,

$$
\hat{\boldsymbol{\mu}}^k = \boldsymbol{\mu}_O^k + \Sigma_{OI}^k \left(\Sigma_I^k\right)^{-1} \left(\boldsymbol{\xi}_I - \boldsymbol{\mu}_I^k\right) \quad , \quad \hat{\Sigma}^k = \Sigma_O^k - \Sigma_{OI}^k \left(\Sigma_I^k\right)^{-1} \Sigma_{IO}^k
$$

Using the linear transformation of gaussian distributions, the conditional expectation of $\boldsymbol{\xi}_O$ given $\boldsymbol{\xi}_I$ can be re-written as a single normal distribution with the parameters

$$
\hat{\boldsymbol{\mu}} = \sum_{k=1}^{K} h_k \hat{\boldsymbol{\mu}}^k \quad , \quad \hat{\Sigma} = \sum_{k=1}^{K} h_k^2 \hat{\Sigma}^k
$$

where,

$$
h_k = \pi^k \frac{\mathcal{N}\left(\boldsymbol{\xi}_I; \boldsymbol{\theta}^k\right)}{\sum_{i=1}^{K} \pi^i \mathcal{N}\left(\boldsymbol{\xi}_I; \boldsymbol{\theta}^i\right)}
$$

is the probability that $\boldsymbol{\xi}_I$ was generated by component k.

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 recordings in the n-th demonstration. Without loss of generality, we assume that the dynamics of the task can be represented by a first order autonomous Ordinary Differential Equation (ODE):

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

where $\mathbf{f} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a continuous and continuously differentiable function with a single equilibrium point $\dot{\boldsymbol{\xi}}^* = \mathbf{f}(\boldsymbol{\xi}^*) = 0$ and $\boldsymbol{\epsilon}$ represents white gaussian noise. It is evident from the formulation of DS that it does not explicitly depend on time and hence is robust towards temporal perturbations. To handle spatial perturbations, i.e. sudden displacement of the target or the manipulator, we consider $\boldsymbol{\xi}$ in the reference frame of the target.

By estimating the function $\mathbf{f}$, we can have a compact mathematical description of the task as a Dynamical System. For this purpose, we use GMM to encode the demonstrated trajectories in a probabilistic framework. The core assumption when representing a task as a Gaussian Mixture Model is that each recorded point $\boldsymbol{\xi}(t)$ from the demonstrations is a sample
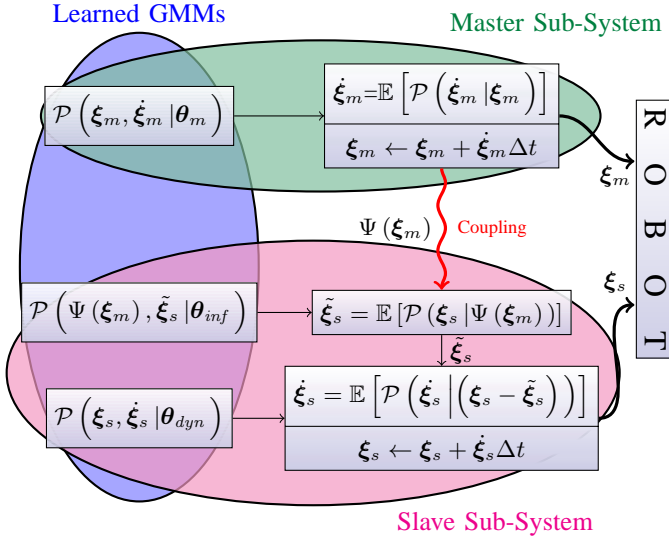
Fig. 2. Task execution using CDS model. Blue region shows the three Gaussian Mixture Models which form the full CDS model. Green region shows the master sub-system where the cartesian position of the robot end-effector evolves in time as a DS and is continuously fed to the robot. Magenta region shows the slave sub-system where the finger joint angles evolve in time as a DS, but also influenced by the state of the master system and fed to the robot. Coupling is ensured by passing selective state information in the form of $\Psi(\boldsymbol{\xi}_m)$ as shown in red.

drawn from the joint distribution:

$$\mathcal{P}\left(\boldsymbol{\xi}|\boldsymbol{\theta}\right) = \sum_{k=1}^{K} \pi^k \mathcal{N}\left(\boldsymbol{\xi}; \boldsymbol{\theta}^k\right).$$

As detailed in Table I, this model makes it possible to perform probabilistic regression for the value of output variable $\boldsymbol{\xi}_o$ given the value of input variable $\boldsymbol{\xi}_I$ at each time instant. E.g., as applied in previous works [9, 12] in the special case of learning dynamics, the desired state velocities can be queried conditioned on the current state. Note that in this special case which models only the dynamics of the task, the partitions $\boldsymbol{\xi}_I$ and $\boldsymbol{\xi}_o$ correspond respectively to the spatial positions and velocities of the robot's end-effector. Since this is not always the case in the CDS model, we will keep this generalized for now and define the partitions for the different components of the coupled model in later subsections. In the next sections, we show that the CDS model harnesses much more from the GMM than just states and velocities by learning the coupling information in addition to the dynamics.

B. Coupled Dynamical System

In the classical case of learning position and orientation dynamics, one GMM each for hand and finger motions would suffice to model the dynamics. However, to model a reach-grasp with hand-arm coordination, it is required to have a more complex approach. Note that Gribovskaya and Billard [9] have used a coupling between position and orientation in the inverse kinematics. However, the main purpose served by this approach was to avoid unfavorable joint postures

The CDS model derives inspiration from the biological evidences of reach-grasp coupling [7, 16]. These studies advocate the fact that there is a parallel, but time-coupled evolution of these sub-tasks combined with synchronized termination constraints. E.g., if the fingers close before the hand reaches the object, the task fails. Moreover, this order needs to be maintained under spatial, temporal and grasp-type perturbations. Another example of a situation where coupling is needed is that of change in grasp type. When the required grasp type is changed on the fly, if the change occurs from a low-aperture grasp to a high aperture grasp, a full reopening may be needed which is only possible if the coupling is active.

We first formally discuss the CDS model in subsections describing the model creation and the procedure for task execution. To establish intuitive understanding, we present a 2D example as a representative of higher dimensional grasping tasks. Subsequently, we show that the CDS model retains the global stability endowed in the individual GMMs by *stable estimator of dynamical systems* (SEDS) Khansari-Zadeh and Billard [12] and the fact that different GMMs are coupled using a coupling function does not affect the overall stability of the model.

1) Model Building: Let $\boldsymbol{\xi}_m$ denote the state of the master sub-system and $\boldsymbol{\xi}_s$ that of the slave sub-system in their respective goal reference frames. Consider the set $\mathcal{G}$ of all objects for which grasping behaviors are demonstrated. The following three joint distributions are learned as explained in Table. I -

1) $\mathcal{P}\left(\boldsymbol{\xi}_m, \dot{\boldsymbol{\xi}}_m | \boldsymbol{\theta}_m^g\right)$: encoding the dynamics of the master
2) $\mathcal{P}\left(\Psi(\boldsymbol{\xi}_m), \boldsymbol{\xi}_s | \boldsymbol{\theta}_{\inf}^g\right)$: encoding the inferred state of the slave conditioned on the master (we will refer to this quantity as $\tilde{\boldsymbol{\xi}}$)
3) $\mathcal{P}\left(\boldsymbol{\xi}_s, \dot{\boldsymbol{\xi}}_s | \boldsymbol{\theta}_{\mathrm{dyn}}^g\right)$: encoding the dynamics of the slave

$\forall g \in \mathcal{G}$. Here $\Psi : \mathbb{R}^{d_m} \mapsto \mathbb{R}$ denotes the *coupling function* which is a monotonic function of $\boldsymbol{\xi}_m$ with the constraint

$$\lim_{\boldsymbol{\xi}_m \to \mathbf{0}} \Psi(\boldsymbol{\xi}_m) = 0 \qquad (2)$$

and d_m denotes the dimension of the master sub-system. The purpose of the coupling function is to transfer relevant information between the sub-systems so as to ensure coupling between them. The distributions for learning dynamics (i.e. $\mathcal{P}\left(\boldsymbol{\xi}_m, \dot{\boldsymbol{\xi}}_m | \boldsymbol{\theta}_m^g\right)$ and $\mathcal{P}\left(\boldsymbol{\xi}_s, \dot{\boldsymbol{\xi}}_s | \boldsymbol{\theta}_{\mathrm{dyn}}^g\right)$) is learned using SEDS given by which produces globally stable model but can only handle models with $|\boldsymbol{\xi}_I| = |\boldsymbol{\xi}_o|$. On the other hand, the distribution $\mathcal{P}\left(\Psi(\boldsymbol{\xi}_m), \boldsymbol{\xi}_s | \boldsymbol{\theta}_{\inf}^g\right)$ is learned using non-linear programming to fit gaussians to the data under the constraint

$$\lim_{\boldsymbol{x} \to \mathbf{0}} \mathbb{E}\left[\boldsymbol{\xi}_s | \boldsymbol{x}\right] = \mathbf{0}. \qquad (3)$$

In the context of reach to grasp tasks studied in this work, the master sub-system corresponds to reaching motion with state vector as the cartesian position of the end-effector. The slave sub-system corresponds to the motion of the fingers with state vector as the finger joint angles. Note that the same model can be used with orientation variables (Euler angles or Axis-angle

(a) Demonstrations   (b) $\mathcal{P}\left(\dot{\xi}_x \,|\, \xi_x\right)$ (c) $\mathcal{P}\left(\xi_f \,|\, \xi_x\right)$ (d) $\mathcal{P}\left(\dot{\xi}_f \,|\, \xi_f\right)$
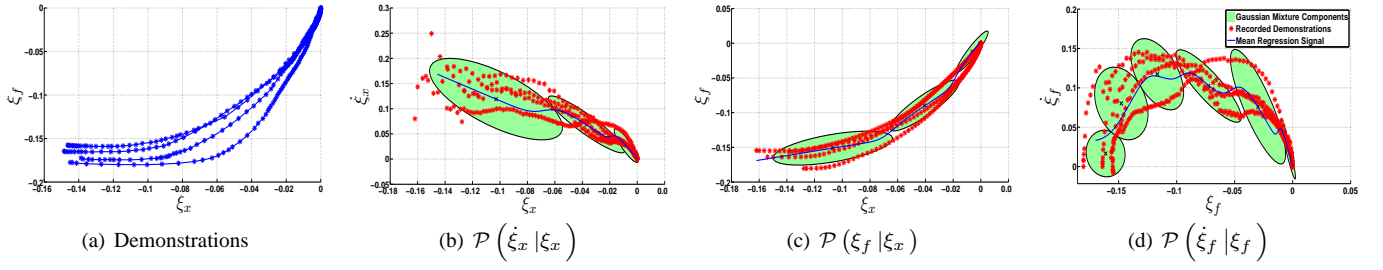
Fig. 3. GMMs which combine to form the CDS model for the 2D example. (a) shows the human demonstrations. Large number of data points around the end of trajectories depict very small velocities. (b) shows the GMM encoding the velocity distribution conditioned on the position of master sub-system (ξ_x). (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 slave-subsystem (ξ_f). Legend for (d) holds for all.

representation) and hence can be used to couple orientation control with position control by having another slave sub-system.

2) Reproduction: While reproducing the task, the model essentially works in three phases: *Increment master → Infer slave → Increment slave*. The master sub-system evolves in time independently and the corresponding end-effector commands are issued to the robot. Moreover, it modulates the commands for the slave system which also evolves in a similar way but while sharing information with the master subsystem due to the coupling mechanism. Fig. 3 shows this flow of information among the sub-systems and the robot. Such a scheme is desired since it ensures that any spatial, temporal or grasp-type perturbations are reflected appropriately in all the sub-systems. The process starts by generating a velocity command for the master sub-system and thus increments the state by one time step. $\Psi(\boldsymbol{\xi}_m)$ transforms the current state of the master sub-system which is fed to the inference model that infers the desired state of the slave sub-sytem by conditioning the learned joint distribution on the appropriate variable. The velocity command to drive the slave sub-system from the current state to the inferred (desired) state is generated by GMR conditioned on the error between the two. The slave sub-system achieves a new state and the cycle is repeated until convergence. Note that if at any instant, the robot is presented with a different object for which the grasping model is stored in the set $\mathcal{G}$, we select the corresponding CDS model and continue the same operations using master and slave components. Algorithm 1 explains the complete reproduction process in a pseudocode.

Note that the coupling function $\Psi(\boldsymbol{\xi}_m)$ also acts as a phase variable which updates itself every time step and in the event of a perturbation, will command the slave system to re-adjust so as to maintain the same correlations 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. E.g., in the experiments presented in Section IV, they control the speed and the extent of re-opening of fingers when the target is changed to a farther location.

Example. Suppose we represent a reach-grasp task as two

Algorithm 1 Coupled task execution

Input: $\boldsymbol{\xi}_m(0)$; $\boldsymbol{\xi}_s(0)$; $\boldsymbol{\theta}_m^g$; $\boldsymbol{\theta}_{\inf}^g$; $\boldsymbol{\theta}_{dyn}^g$; α; β; Δt; ϵ
    Set $t = 0$
    repeat:
       if *perturbation* **then**
          update $g \in \mathcal{G}$
       end if
       Increment Master: $\dot{\boldsymbol{\xi}}_m(t) \sim \mathcal{P}\left(\dot{\boldsymbol{\xi}}_m | \boldsymbol{\xi}_m; \boldsymbol{\theta}_m^g\right)$
                            $\boldsymbol{\xi}_m(t+1) = \boldsymbol{\xi}_m(t) + \dot{\boldsymbol{\xi}}_m(t)\Delta t$
       Infer Slave: $\tilde{\boldsymbol{\xi}}_s(t) \sim \mathcal{P}\left(\boldsymbol{\xi}_s | \Psi\left(\boldsymbol{\xi}_m\right); \boldsymbol{\theta}_{inf}^g\right)$
       Increment Slave: $\dot{\boldsymbol{\xi}}_s(t) \sim \mathcal{P}\left(\dot{\boldsymbol{\xi}}_s | \beta\left(\boldsymbol{\xi}_s - \tilde{\boldsymbol{\xi}}_s\right); \boldsymbol{\theta}_{dyn}^g\right)$

                          $\boldsymbol{\xi}_s(t+1) = \boldsymbol{\xi}_s(t) + \alpha\dot{\boldsymbol{\xi}}_s(t)\Delta t$
       $t \leftarrow t+1$
    until: *Convergence* $\left(\|\dot{\boldsymbol{\xi}}_s(t)\| < \epsilon \quad \text{and} \quad \|\dot{\boldsymbol{\xi}}_m(t)\| < \epsilon\right)$

sub-systems controlling the motion of the end-effector and the fingers. For simplicity, we consider 1-D cartesian position ξ_x as the master variable and 1 finger joint angle ξ_f as the slave variable, both expressed in the goal reference frame 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 robot hand instead of ξ_f. For reference with the formal description of the CDS model we re-iterate the following equivalences in this example: $\boldsymbol{\xi}_m \equiv \xi_x$ (master variable), $\boldsymbol{\xi}_s \equiv \xi_f$ (slave variable), $\Psi(\xi_x) \equiv \xi_x$ (coupling function) and $d_m = 1$. Note that in different tasks, depending on the nature of coupling in different dimensions of the master variable, other coupling functions can be used.

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: $\mathcal{P}\left(\xi_x, \dot{\xi}_x \,|\, \boldsymbol{\theta}_x\right)$, $\mathcal{P}\left(\xi_x, \xi_f \,|\, \boldsymbol{\theta}_{inf}\right)$ and $\mathcal{P}\left(\xi_f, \dot{\xi}_f \,|\, \boldsymbol{\theta}_{dyn}\right)$. The resulting mixtures for each of the models is as shown in Fig. 3. For reproducing the task, instead of the earlier
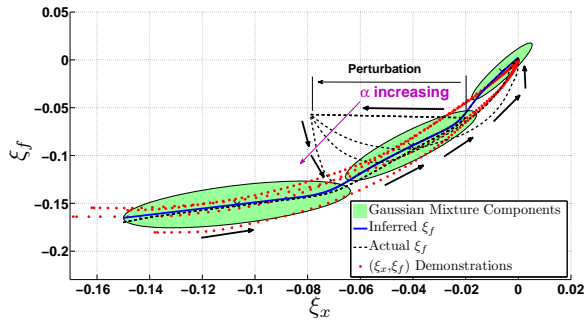
Fig. 4. Reproducing the task under the CDS model. The reproduction (dashed) is overlaid on the demonstrations for reference. The model is run for different α values and the flow of the state values in time is depicted by the arrows. It is evident that the model tries to track the desired ξ_f (blue) values at the current ξ_x by reversing the velocity in ξ_f direction. The tracking is more stringent for larger α.
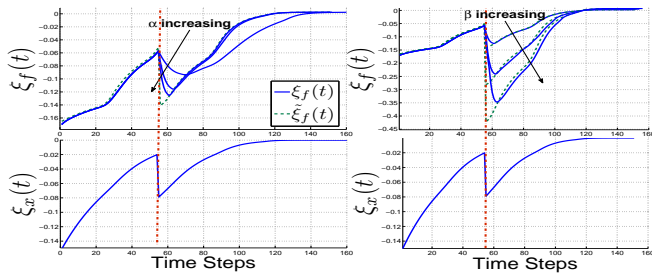

Fig. 5. Variation of obtained trajectories with α and β. Vertical red line shows the instant of perturbation when the target is suddenly pushed away along positive ξ_x direction. Negative velocities are generated in ξ_f in order to track $\tilde{\xi}_f$. Speed of retracting is proportional to α (**left**)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

approach presented in [9] where the system evolves under the velocities computed as $\mathbb{E}\left[(\dot{\xi}_x; \dot{\xi}_f) | (\xi_x; \xi_f)\right]$ 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, viz. 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. $\tilde{\xi}_f$ is nothing but the expected value of ξ_f given ξ_x as seen during the demonstrations. The time variation of ξ_f and its variation 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 retracting. Fig. 6 shows the streamlines of this system in order to visualize the global behavior of trajectories evolving under the CDS model. The CDS model run is compared to uncoupled task execution in Fig. 7. It shows the behavior when the same perturbation is introduced on the abscissa in both coupled and un-coupled executions. Clearly, the unperturbed variable (ξ_f in this case) does not react when there is no coupling and also the order of convergence is not the same as in the demonstrations where ξ_x converges faster than ξ_f.

3) Stability and Convergence: We first define the notion of global stability in CDS and then prove that the process
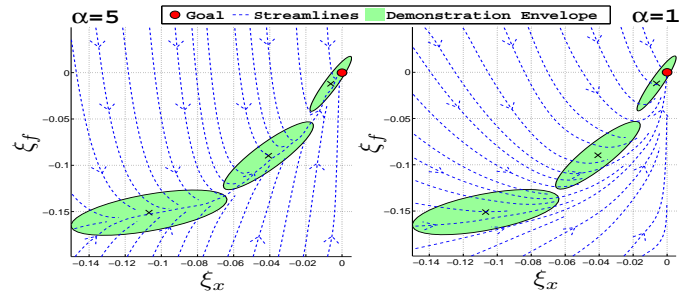

Fig. 6. Change in α affecting the nature of streamlines. Larger α will tend to bring the system more quickly towards the (ξ_x, ξ_f) locations seen during demonstrations.
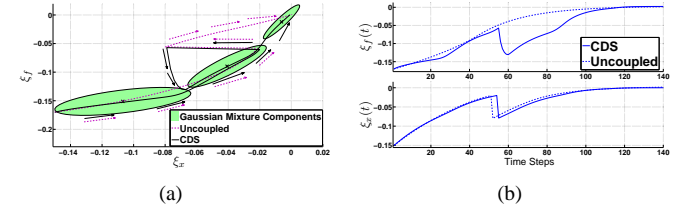

(a)   (b)

Fig. 7. Task reproduction with and without coupling shown in (a) state space, (b) time variation. Dotted lines show the uncoupled task execution. Note the difference in the directions from which the convergence occurs in the two cases. In the coupled execution, convergence is faster in ξ_x than in ξ_f. In the uncoupled case it is just opposite.

evolving under Algorithm 1 is globally stable.

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

$$\lim_{t \to \infty} \boldsymbol{\xi}_m(t) = \mathbf{0} \tag{4a}$$

$$\lim_{t \to \infty} \boldsymbol{\xi}_s(t) = \mathbf{0} \tag{4b}$$

To prove that the CDS model indeed follows the conditions 4, we use the properties of its individual components. The condition 4a holds true due to the global stability of SEDS. To investigate the stability of the coupling, we consider

$$
\begin{aligned}
\lim_{t \to \infty} \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_s | \Psi(\boldsymbol{\xi}_m)\right] &= \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_s \middle| \Psi\left(\lim_{t \to \infty} \boldsymbol{\xi}_m\right)\right] \\
&= \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_s \middle| \lim_{\boldsymbol{\xi}_m \to \mathbf{0}} \Psi(\boldsymbol{\xi}_m)\right] \text{(By 4a)} \\
&= \mathbb{E}\left[\tilde{\boldsymbol{\xi}}_s | \mathbf{0}\right] \quad \text{(By Eq. 2)} \\
&= \mathbf{0} \quad \text{(By Eq. 3)} \tag{5}
\end{aligned}
$$

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

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

Taking the limiting values and using Eq. 5 , we get

$$\lim_{t \to \infty} \dot{\boldsymbol{\xi}}_s = \mathbb{E}\left[\dot{\boldsymbol{\xi}}_s | \beta\boldsymbol{\xi}_s\right] \tag{6}$$

which is again globally asymptotically stable due to SEDS and hence will drive the state $\boldsymbol{\xi}_s$ asymptotically to $\mathbf{0}$. 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 $\alpha > 0$. Why such a condition is required for global stability is proved in detail in [12].

IV. EXPERIMENTS AND RESULTS

In this section, we describe the experiments for recording human demonstrations under random perturbations. We show that the CDS model is indeed qualitatively equivalent to the demonstrated human behavior of grasping under perturbations and it is suited to handle fast perturbations which typically need re-planning and are difficult to handle online. From the experimental data, we identify relationships to infer the free parameters of the CDS model. Unlike the example presented in Section III, we used the coupling function $\Psi(.) = ||.||$ for all the task runs on the robot. We divide the discussion into subsections describing the experimental setup, quantitative results/inferences from the experimental data and task reproductions on the iCub simulator as well as the real robot to validate the model[1].

A. Experiments

The experimental setup to record human demonstrations and create sudden perturbations is shown in Fig. 1. It consists of two stationary targets and the on-screen target selector which prompts the human subject to reach and grasp one of the objects depending on the color shown on the screen. The iCub simulator runs simultaneously while the human is performing demonstrations in order to establish correspondence for the human subject. To start the experiment, one of the targets is switched on and the subject starts to reach towards the corresponding object aiming for a particular grasp depending on the object. In random trials (without the knowledge of the subject), perturbations are introduced by abruptly switching the on-screen target selector. Reacting to this change, the subject starts to adjust the motion of hand and fingers in order to reach the other target. The subject forms a grasp around the final target which marks the end of the trial. All the motion data i.e. hand position, orientation and finger joint angles is recorded throughout the trial at a frequency of 50 Hz. After each trial, the subject is asked to go to a rest posture where a 5 sec. calibration procedure is done for the motion sensors and data-glove. Trials are run until at least 10 unperturbed and 10 perturbed trials are obtained. Learning from perturbations enables the estimation of parameters α and β so that they can be predicted prior to the instant of perturbation at the time of task reproduction. In addition to this, we also learn the CDS model for pinch grasp and populate the set of grasps $\mathcal{G}$. Note that arbitrarily many grasps can be learned and added to this set.

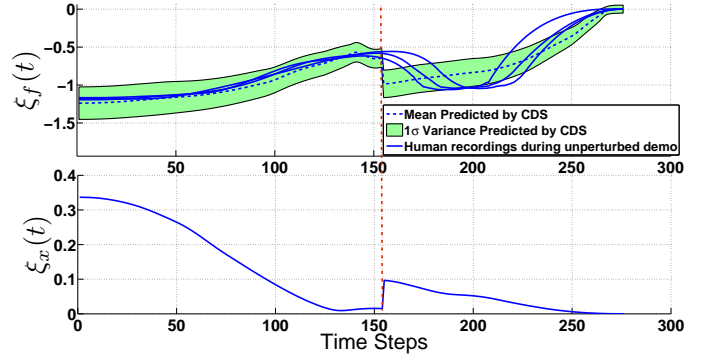[1] Videos are available at http://www.youtube.com/user/TheRoboticsVideos

Fig. 8. A qualitative comparison of CDS model predictions of the inferred finger joint angles and the actual recorded human behavior. Note the coupling of joint angles with the distance from the target. A change in target location triggers a discontinuous shift in the value of $||x - x_{goal}||$ and at the same time, starts reopening of the fingers.
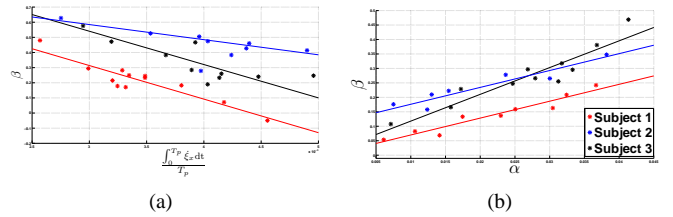
Fig. 9. (a) shows the linear correlation found between mean hand velocity prior to perturbation and β. T_p refers to the time at which perturbation is introduced. (b) shows the linear correlation found between α and β.

After all the data is collected, we learn the CDS model from the data recorded in the unperturbed trials. This serves as the base model which qualitatively mimics human behavior under perturbations. Fig. 8 shows typical human responses of the finger joint angles (only index finger proximal joint is shown) under perturbation of the target. The vertical red line marks the onset of perturbation and the subsequent dip in the joint angles is due to the re-opening of the fingers when the target is suddenly moved away. Note that in all the 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 behavior was found common to all subjects. It is evident from Fig. 8 that the finger joint angle inferred by CDS at each time instant is similar to what is exhibited by humans within the variance of demonstrations.

B. Calculation of model parameters

As illustrated in section III-B, the parameters α and β represent respectively the speed and amplitude of the reaction (in this case, reopening of fingers) to perturbation. Increased α causes the finger trajectories to follow the inferred (desired) values more strictly, hence, a sharp decrease in the joint variables is observed. On the other hand, β modulates the inferred value itself. Hence, a manually tuned combination of the two is sufficient to generate any given behavior after the perturbation. However, as shown in Fig. 9, it was found that there exist correlations which make it possible to predict their values for
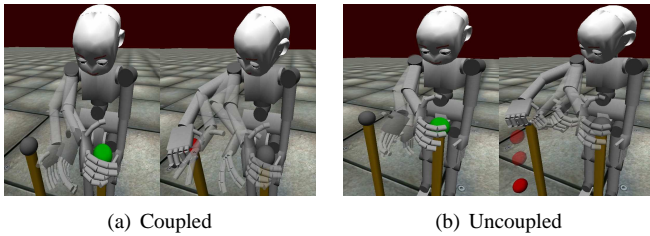
(a) Coupled   (b) Uncoupled

Fig. 10. Reach-grasp task executions with and without coupling. In the coupled execution (a), fingers maintain the correlations seen during the demonstrations which prevents premature finger closure. The uncoupled execution (b), fingers close early and the grasp fails.
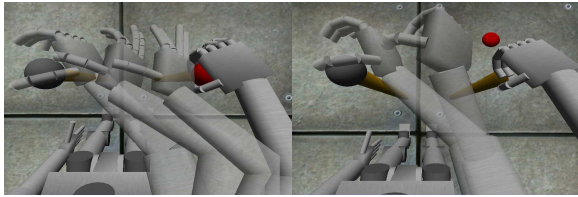

Fig. 11. Closeup of hand motion post perturbation in coupled (**left**) and uncoupled (**right**) executions. Note the re-opening of fingers leading to a successful grasp in the coupled case.

a human just by observing the motion of the hand prior to the perturbation. Fig. 9(a) shows there is a linear relationship between the velocity of the hand prior to perturbation and the parameter β. It shows that the faster a subject moves towards the target, the less they reopen the fingers upon perturbation. Fig. 9(b) shows the linear correlation between the parameters α and β. It shows that a faster reopening of the fingers is accompanied by a larger amount of reopening. Consequently, both the parameters α and β can be predicted based on the motion of the hand prior to perturbation. Hence, they no longer need to be specified manually in Algorithm 1.

C. Validation

We validate the CDS model by executing the learned grasping strategies on the iCub robot. First we show that grasping under perturbations using the uncoupled approach fails due to lack of knowledge of the correlations between the reach and grasp sub-systems. Fig. 10 shows such comparison. The robot starts moving towards a fixed target which is suddenly shifted to the right just before the completion of the task. In the uncoupled case (Fig. 10(b)), the fingers close too early and the task fails. In the coupled case (Fig. 10(a)), the finger motion is delayed due to the coupling and the fingers close according to the correlations learned during the demonstrations. Fig. 11 clearly shows the re-opening of fingers.

Next, we show that the presented scheme also enables online adaptation against fast perturbations of grasp-type. We learn power grasps in palm-up and palm-down configurations of the hand from unperturbed demonstrations and populate the set of grasps $\mathcal{G}$ with them. In the reproduction phase, the robot starts moving towards the target aiming for the palm-down configuration grasp. After a certain period, the object is made to reappear at another location, however not supported
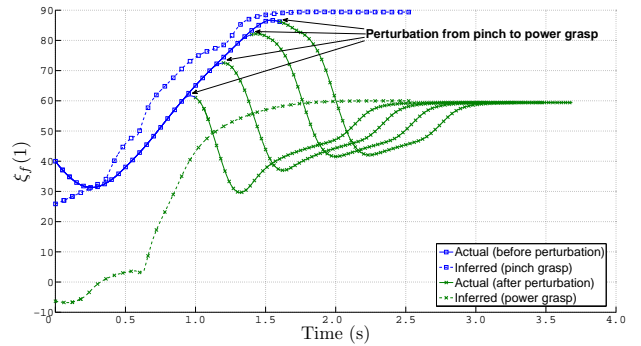

Fig. 13. Motion of one finger joint angle under grasp-type perturbation. Demonstrated models for the two grasps recorded during the unperturbed demonstrations are shown. The CDS model switches smoothly between the models when the perturbation occurs.

against gravity so that it starts falling. This requires a fast adaptation from palm-down to palm-up grasp, while the target keeps on moving. Note that we use the approach presented in previous work on ball catching by Kim et al. [13] to ensure that the robot intercepts the falling object in its workspace. Here the contribution of the CDS model is to re-adjust the hand orientation and finger curl online so that the grasp is completed successfully on the falling object. The task being performed by the iCub in simulation is shown in Fig. 12. Note that the fingers close proportionately as the distance between the falling ball and the robot hand decreases, maintaining the correlations seen during the demonstrations. The time elapsed is indicated on the figure in seconds.

We perform another task showing the ability of the CDS model to adapt between pinch and power grasps. We learn pinch and power grasps from demonstrations as shown previously and change the target object while the robot goes for the pinch grasp. Fig. 13 shows the motion of robot's index finger proximal joint. It is evident that the hand aperture decreases quickly when the robot aims for the pinch grasp but after the switch, 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). The model can also be seen to be robustly handling different instants of perturbation.

V. CONCLUSION

In this paper we presented a model for encoding and reproducing grasping strategies which is capable of handling fast real-time perturbations. Biological inspiration was taken by observing the correlations between arm and finger motion during human experiments of reach-to-grasp tasks under sudden unpredicted perturbations. We showed that once the grasping strategies are taught offline by a human demonstrations, the model is able to reliably switch between those under very fast perturbations. Re-opening of fingers under perturbation has been found to be present in all human trials and is shown to be critical to the success of reach-grasp tasks. Our model ensures this behavior even in the presence of arbitrary perturbations.
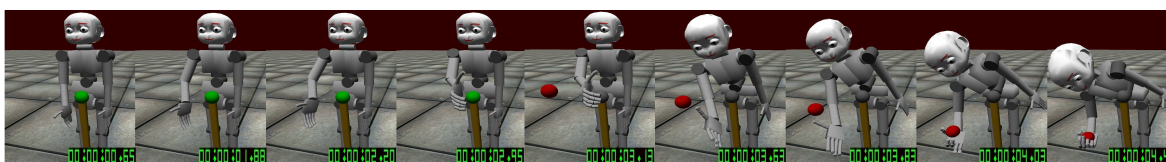
Fig. 12. Fast adaptation under perturbation from palm-down to palm-up power grasp. Torso is included in the inverse kinematics to increase the workspace of the robot so as to highlight the effectiveness of the model in adapting the grasping motion under very fast perturbations.

REFERENCES

[1] Ji-Hun Bae, Suguru Arimoto, Yuuichi Yamamoto, Hiroe Hashiguchi, and Masahiro Sekimoto. Reaching to grasp and preshaping of multi-dofs robotic hand-arm systems using approximate configuration of objects. In *Intelligent Robots and Systems. IEEE/RSJ International Conference on*.

[2] D. Berenson, S.S. Srinivasa, D. Ferguson, A. Collet, and J.J. Kuffner. Manipulation planning with workspace goal regions. In *Robotics and Automation. ICRA. IEEE International Conference on*, pages 618–624. IEEE, 2009.

[3] G.M. Bone, A. Lambert, and M. Edwards. Automated modeling and robotic grasping of unknown three-dimensional objects. In *Robotics and Automation. ICRA. IEEE International Conference on*, pages 292–298. IEEE, May 2008. doi: 10.1109/ROBOT.2008.4543223.

[4] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298, 2007. ISSN 1083-4419.

[5] U. Castiello, K. Bennett, and H. Chambers. Reach to grasp: the response to a simultaneous perturbation of object position and size. *Experimental Brain Research*, 120(1):31–40, 1998. ISSN 0014-4819.

[6] A. Gasparri, G. Oliva, and S. Panzieri. Path planning using a lazy spatial network prm. In *Control and Automation. MED. 17th Mediterranean Conference on*, pages 940–945. IEEE, 2009.

[7] M. Gentilucci, S. Chieffi, M. Scarpa, and U. Castiello. Temporal coupling between transport and grasp components during prehension movements: effects of visual perturbation. *Behavioural Brain Research*, 47(1):71–82, 1992. ISSN 0166-4328.

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

[9] E. Gribovskaya and A. Billard. Learning nonlinear multivariate motion dynamics for real-time position and orientation control of robotic manipulators. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pages 472–477. IEEE, 2009.

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

[11] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation. Proceedings. ICRA. IEEE International Conference on*, volume 2, pages 1398–1403. IEEE, 2002. ISBN 0780372727.

[12] S.M. Khansari-Zadeh and A. Billard. Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pages 2676–2683. IEEE, 2010.

[13] Seungsu Kim, Elena Gribovskaya, and Aude Billard. Learning Motion Dynamics to Catch a Moving Object. In *10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010.

[14] S. LaValle and 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*, page 293. AK Peters, Ltd., 2001. ISBN 156881125X.

[15] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *Robotics and Automation. Proceedings. ICRA. IEEE International Conference on*, volume 2, pages 1824–1829. IEEE, 2003. ISBN 0780377362.

[16] A.R. Mitz, M. Godschalk, and S.P. Wise. Learning-dependent neuronal activity in the premotor cortex: activity during the acquisition of conditional motor associations. *Journal of Neuroscience*, 11(6): 1855, 1991.

[17] J-P. Saut, A. Sahbani, S. El-Khoury, and 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*, pages 2907–2912. IEEE, 2007.

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