In: Proc. 2009 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS), Oct 11-15, St. Louis, MO, USA (2009)

Robust Constraint-consistent Learning

Matthew Howard, Stefan Klanke, Michael Gienger, Christian Goerick and Sethu Vijayakumar

***Abstract*— Many everyday human skills can be framed in terms of performing some task subject to constraints imposed by the environment. Constraints are usually unobservable and frequently change between contexts. In this paper, we present a novel approach for learning (unconstrained) control policies from movement data, where observations are recorded under different constraint settings. Our approach seamlessly integrates unconstrained and constrained observations by performing hybrid optimisation of two risk functionals. The first is a novel risk functional that makes a meaningful comparison between the estimated policy and constrained observations. The second is the standard risk, used to reduce the expected error under impoverished sets of constraints. We demonstrate our approach on systems of varying complexity, and illustrate its utility for transfer learning of a car washing task from human motion capture data.**

I. Introduction

Many human motor skills involve performing some task subject to constraints imposed either by the environment [8], the task [3] or, more commonly, both. For example, when opening a door, the door acts as an environmental constraint that restricts the movement of one's hand along the opening arc of the door. When stirring soup in a saucepan, the sides of the pan prevent the spoon moving beyond the radius of the pan. Many tasks require self-imposed task constraints to be fulfilled in order to achieve adequate performance. For example when pouring water from a bottle to a cup the orientation of the bottle must be constrained so that the stream of water falls within the mouth of the cup. When wiping a window, one's hand must be constrained to maintain contact with the wiping surface [9].

A promising approach to rapidly providing robots with skills such as opening doors and washing windows (ref. Fig. 1), is to take examples of motion from existing systems, such as humans, and attempt to learn a control policy that somehow captures the essence of the desired behaviour [1], [7], [15]. Such techniques offer (i) a simple, intuitive interface for programming robots, (ii) effective methods for motion recognition and segmentation [7], and; (iii) accelerated optimisation of movements by seeding learning from demonstrations [12]. However, while a wide variety of approaches for learning and representing movements have been proposed in recent years (for a review, see [1] and references therein), few have explicitly considered the effects of constraints on motion and ways to cope with these in learning.

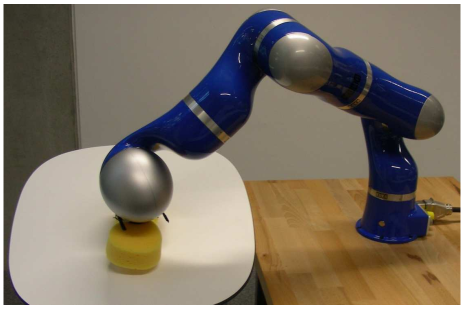


Fig. 1. Anthropomorphic DLR light-weight arm used in our experiments.

In this paper we address the problem of modelling control policies in a way that is consistent with the fact that they may be subject to generic (environmental or task-based) constraints on motion. Our approach is inspired by direct policy learning[1] (DPL) [15] whereby we attempt to learn a continuous model of the policy directly from motion data. However, our method differs from standard DPL in that we consider observations from policies projected into the nullspace of a set of dynamic, non-linear, or even discontinuous constraints, and that these constraints may change between observations, or even during the course of a single observation. In doing this we aim to illustrate how existing DPL approaches (e.g. Dynamic Movement Primitives [13] and other dynamical system-based approaches [6]) that currently rely on traditional supervised learning techniques can be extended to cope with the effect of motion constraints in the data.

In previous work we proposed a reformulation of the risk functional used for learning by introducing a projection of the estimated policy onto the observations before calculating errors [5]. This allowed us to effectively reconstruct policies from constrained movements without explicit knowledge of the constraints, provided the data was 'rich enough' in terms of the different constraints contained in that data. This was found to be highly effective for learning from data containing high variability in the constraints, even for very high dimensional systems such as 22-DOF ASIMO joint space data. However, in its basic form the method presented in [5] tends to prefer to explain variations in observations as variations in constraints instead of as variations in the policy itself. This can result in poor performance when learning on unconstrained data or data where constraints are highly correlated between observations.

In this paper we propose an extension to that method to deal with these problems. As a key ingredient, we partition

M. Howard, S. Klanke and S. Vijayakumar are with the Institute of Perception Action and Behaviour, University of Edinburgh, Scotland, UK. matthew.howard@ed.ac.uk

M. Gienger and C. Goerick are with the Honda Research Institute Europe (GmBH), Offenbach, Germany. michael.gienger@honda-ri.de

[1] To clarify the terminology used, we refer to DPL as the supervised learning of policies from given data. This is in contrast to the learning of policies directly from cost/reward feedback without the use of a value function, which is also sometimes referred to as DPL.

our model optimisation into two parts. The primary part uses the constraint-consistency objective function proposed in [5] to deal with the effect of the constraints in the data. We then perform a secondary optimisation to tighten the fit on the data in regions where there is little variation in the constraints. By extending the method in this way, we are able to seamlessly integrate constraint-consistent learning with optimisation of more standard risk functionals. We demonstrate the utility of our approach for learning a car washing task from human demonstration data.

II. Learning from Constrained Policies

Here, we briefly characterise the problem of direct policy learning when constraints are applied to motion. Following [15], [11], we consider the learning of the autonomous policy mapping

$$\mathbf{u}(t) = \boldsymbol{\pi}(\mathbf{x}(t)) \ , \qquad \boldsymbol{\pi} : \mathbb{R}^n \mapsto \mathbb{R}^d \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^d$ are some appropriately chosen state and action vectors[2]. We consider policies that are constrained in such a way that there are hard restrictions on movement. Analytically [16], this means that, under a set of k-dimensional constraints

$$\mathbf{A}(\mathbf{x}, t)\mathbf{u} = \mathbf{0} \tag{2}$$

the policy is projected into the nullspace of those constraints

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{N}(\mathbf{x}, t)\boldsymbol{\pi}(\mathbf{x}(t)), \tag{3}$$

where $\mathbf{N}(\mathbf{x}, t) \equiv (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A}) \in \mathbb{R}^{d \times d}$ is in general a non-linear, time-varying projection operator[3] and $\mathbf{A}(\mathbf{x}, t) \in \mathbb{R}^{k \times d}$ is some matrix describing the constraint. Constraints of this form are common in scenarios where manipulators interact with the environment, for example when grasping a tool or turning a crank or pedal. They are also common in controlling of redundant degrees of freedom [10], where policies such as (3) are used, for example, to aid joint stabilisation under task constraints.

In general, the goal of DPL is to approximate the policy $\boldsymbol{\pi}$ as closely as possible given observations (often in the form of trajectories) of the states and actions $\mathbf{u}(t)$, $\mathbf{x}(t)$. Here, the fact that the observed action is constrained (3) complicates learning in several ways [4], [5]. First there is the fact that commonly the constraint $\mathbf{A}(\mathbf{x}, t)$ (and therefore $\mathbf{N}(\mathbf{x}, t)$ also) is not explicitly known and may be ambiguous. For example when opening a door one might not know the exact radius or opening arc of the door, or might not observe an obstacle behind the door, blocking it. Second, the data set may be *non-convex* (from the point of view of standard DPL approaches) in the sense that there may be multiple observations made at any given point under different constraints. For example when observing wiping on several surfaces, the constraints (and therefore the observed $\mathbf{u}$) will differ between surfaces depending on their orientations in the work space. Finally, there is a *degeneracy* problem due to the fact that, under any given constraint and for any set of observations, there may be multiple policies $\boldsymbol{\pi}$ that could be projected to produce those observations.

While these issues prove problematic for methods that do not take into account the effect of constraints, it was recently shown that an effective strategy for dealing with this is to seek the underlying *unconstrained policy*, $\boldsymbol{\pi}$, rather than directly trying to fit the raw (constrained) data [4], [5]. In previous work we proposed methods to do this for the special case of potential-based policies [4], and later extended this to learning generic policies [5]. For effective learning the latter required rather high variability in the constraints, and its performance sometimes suffered from a tendency to misinterpret variability in the policy (as a function of $\mathbf{x}$) with variability in the constraints, particularly in case the observations were not constrained at all.

Here we further develop the method proposed in [5] in order to (i) improve robustness by avoiding the misinterpretation problem, and (ii) seamlessly integrate constraint-consistent learning with more standard learning approaches. We turn to the details of the approach in the next section.

III. Method

Our method works on data that is given as tuples $(\mathbf{x}_n, \mathbf{u}_n)$ of observed states and constrained actions. We assume that all commands $\mathbf{u}$ are generated from the same underlying policy $\boldsymbol{\pi}(\mathbf{x})$, which for a particular observation might have been constrained, that is $\mathbf{u}_n = \mathbf{N}_n\boldsymbol{\pi}(\mathbf{x}_n)$ for some projection matrix[4] $\mathbf{N}_n$. We assume that the projection matrix for any given observation is not explicitly known, i.e. our data is unlabelled with respect to the active constraints at the time of observation. Our goal is to approximate the unconstrained policy $\boldsymbol{\pi}(\mathbf{x}_n)$ as closely as possible. In the following we briefly review how this can be done by optimisation of the constraint-consistency objective function [5], and then propose an extension to this method through a secondary optimisation approach. We then use the extended method to derive learning rules for two example policy models, based on parametric and local linear regression.

A. Optimisation of the Inconsistency

In [5] a reformulation of the standard risk was proposed for estimating a policy $\tilde{\boldsymbol{\pi}}(\cdot)$ that is *consistent* with our observed $\mathbf{u}_n$, knowing that it may be constrained (projected) by an unknown constraint. For this a key observation is to note that, in order to uncover the unconstrained policy we must find a policy model that can be *projected in such a way that the observed actions are recovered*. That is, we require

$$\mathbf{u}(\mathbf{x}) := \mathbf{P}\boldsymbol{\pi}(\mathbf{x})$$

for an appropriate projection matrix $\mathbf{P}$, that either projects onto the same space as the (unknown) $\mathbf{N}$ (i.e. the image of $\mathbf{N}$), or an (even smaller) subspace of that. Since $\mathbf{N}$ is unknown, we must seek an alternative projection $\mathbf{P}$ that

[2] For example in kinematic control, the state vector could be the joint angles, $\mathbf{x} \equiv \mathbf{q}$, and the action could be the velocities $\mathbf{u} \equiv \dot{\mathbf{q}}$, or in dynamic control a suitable state might be, $\mathbf{x} \equiv \mathbf{q}, \dot{\mathbf{q}}$, with actions corresponding to applied torques, $\mathbf{u} \equiv \boldsymbol{\tau}$.

[3] Here and throughout the paper $\mathbf{A}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of the matrix $\mathbf{A}$ and $\mathbf{I}$ denotes the identity matrix of appropriate dimension.

[4] Note that unconstrained observations are incorporated into this formulation as special case where $\mathbf{N} = \mathbf{I}$.

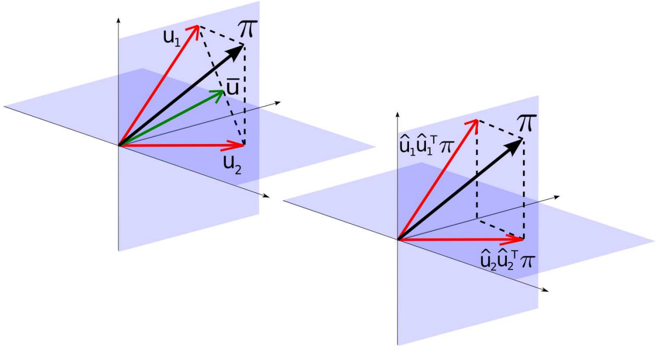


Fig. 2. Illustration of our learning scheme. Left: Direct least-squares regression on constrained commands $\mathbf{u}_1, \mathbf{u}_2$ results in averaging of the observations $\bar{\mathbf{u}}$ in a way that cannot explain the observed actions. Right: The projection of the correct policy $\boldsymbol{\pi}$ onto the observations matches those observations.

approximates it. One such projection, which we know to lie within this subspace, is the 1-D projection onto the observed command itself, that is $\mathbf{P} = \hat{\mathbf{u}}\hat{\mathbf{u}}^T$ (ref. Fig. 2, right). Furthermore, since $\mathbf{u}$ is given, we have all the information we need to calculate this projection and use it for learning, neatly side-stepping the need to explicitly model the full constraint matrix $\mathbf{N}$.

With this as motivation, it was proposed [5] to minimise the *inconsistency*, defined as the functional

$$\begin{aligned} E_i[\tilde{\boldsymbol{\pi}}] &= \sum_{n=1}^{N} \|\mathbf{u}_n - \hat{\mathbf{u}}_n\hat{\mathbf{u}}_n^T\tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\|^2 \\ &= \sum_{n=1}^{N} \left(r_n - \hat{\mathbf{u}}_n^T\tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\right)^2 \\ \text{with} \quad & r_n = \|\mathbf{u}_n\|, \;\; \hat{\mathbf{u}}_n = \frac{\mathbf{u}_n}{r_n}. \end{aligned} \tag{4}$$

Note that this reformulated risk functional avoids the model averaging that would result from using the standard least squares fit to the data $(\mathbf{x}_n, \mathbf{u}_n)$ (cf. Fig. 2, left) [5].

B. Secondary Optimisation of the Standard Risk

Optimisation of the inconsistency (4) has been demonstrated to be effective when learning from data containing high variability in the constraints for systems of varying size and complexity [5]. However, in the simple form outlined so far, it can suffer from the problem of degeneracy in the set of models that are optimal with respect to (4). Because the observations $\mathbf{u}$ influence the estimated policy in a more complex way than in direct regression, small variations in the observations may result in large variations of the learnt policy[5], which can become catastrophic when the method is given data with insufficient variability in the constraints to disambiguate the best policy models.

To illustrate the problem, Fig. 3 shows three candidate policy models $\tilde{\boldsymbol{\pi}}_1$, $\tilde{\boldsymbol{\pi}}_2$ and $\tilde{\boldsymbol{\pi}}_3$ as well as data under a single constraint (right) and two different constraints (left). Consider that we have to select one of these candidates as our policy model based on the available data. For the multiple (i.e. variable) constraint case (Fig. 3, left), optimising the inconsistency (4) clearly determines the best model given the available data: In this case we would choose $\tilde{\boldsymbol{\pi}}_1$, since this has the lowest inconsistency error, $E_i[\tilde{\boldsymbol{\pi}}_1] < E_i[\tilde{\boldsymbol{\pi}}_2] < E_i[\tilde{\boldsymbol{\pi}}_3]$.

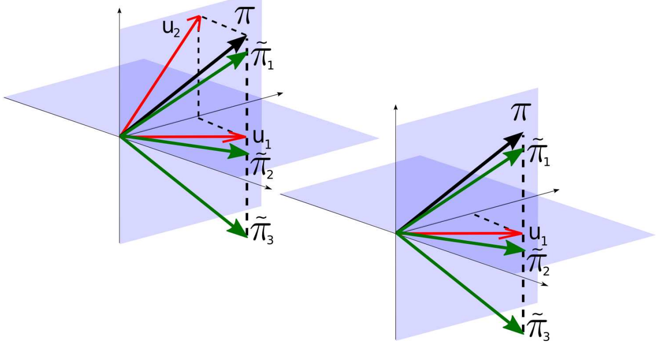


Fig. 3. Illustration of the model degeneracy problem. Shown are three different models with equal inconsistency with respect to the observation $\mathbf{u}_1$. Left: Given observations under different constraints, e.g. $\mathbf{u}_2$, the inconsistency error disambiguates between the three candidate models selecting that which is consistent with both observations (i.e. $\tilde{\boldsymbol{\pi}}_1$). Right: Given only observations under a single constraint there is ambiguity in which is the best model since we cannot be sure about the policy components in the vertical dimension.

However, when there is less variability in the constraints, for example we only see an observation under a single constraint (Fig. 3, right) there may be little difference in the inconsistency for the three models (here, $E_i[\tilde{\boldsymbol{\pi}}_1] = E_i[\tilde{\boldsymbol{\pi}}_2] = E_i[\tilde{\boldsymbol{\pi}}_3]$) resulting in ambiguity as to which model to choose. This is a critical problem, since if we select the wrong model, e.g. $\tilde{\boldsymbol{\pi}}_3$, then it may significantly degrade performance both in terms of prediction of the unconstrained policy (compare $\boldsymbol{\pi}$ and $\tilde{\boldsymbol{\pi}}_3$ in Fig. 3) and also the constrained policy (consider the projection of $\tilde{\boldsymbol{\pi}}_3$ onto the vertical plane, and compare with $\mathbf{u}_2$). Note also that this is a manifestation of the fact that E_i is a lower bound on both the unconstrained policy error (UPE) and the constrained policy error (CPE) [5], since it is precisely these components of the policy that are projected out in the calculation of the inconsistency error that lead to the degeneracy in the models.

In order to deal with this problem, our proposal is to perform an additional *secondary optimisation* to select between models. For this, we propose to optimise the secondary objective

$$E_2[\tilde{\boldsymbol{\pi}}] = \sum_{n=1}^{N} \|\mathbf{u}_n - \tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\|^2 \tag{5}$$

under the constraint that

$$\tilde{\boldsymbol{\pi}} \in \arg\min_{\boldsymbol{\pi}'} \left\{E_i[\boldsymbol{\pi}']\right\}. \tag{6}$$

That is, we propose to optimise the standard risk *subject to the model being consistent with the constrained observations*[6].

By performing this additional secondary optimisation we tighten our fit to the available data and avoid models that are

[5] In machine learning terms, the pure inconsistency-based estimator has high variance.

[6] It should also be noted that in principle we may choose alternative secondary optimisation functions depending on the application. For example, we may wish to bias solutions toward a particular dynamic behaviour, e.g. stabilising movements, subject to consistency with the demonstrated observations.

not strongly supported by the inconsistency. For example, in Fig. 3 (right), optimisation of (5) will result in model $\tilde{\boldsymbol{\pi}}_2$ being chosen since this has the lower E_2. Since we have no information about the vertical component of the policy here, choosing this model is more appropriate since there is little support for $\tilde{\boldsymbol{\pi}}_1$ or $\tilde{\boldsymbol{\pi}}_3$ based on the available data. In effect this acts to regularise our model and improve safety in its performance: In the case that observations are given under an impoverished set of constraints, the model will at worst reproduce the behaviour under those same constraints[7].

Finally, it should be noted that in practice, the hard constraint (6) may need to be softened to improve robustness and avoid numerical instabilities. For this reason, in the following sections we describe how this can be done by looking at eigenvalues derived from gradients of E_i.

The proposed approach can be used in conjunction with many standard regression techniques. However, for the experiments in this paper, we restrict ourselves to two classes of function approximator (i) simple parametric models with fixed basis functions (Sec. III-C), and (ii) locally linear models (Sec. III-D). In the following we describe how these two models can be reformulated to take advantage of the new approach.

C. Parametric policy models

A convenient policy model is given by $\tilde{\boldsymbol{\pi}}(\mathbf{x}) = \mathbf{W}\mathbf{b}(\mathbf{x})$, where $\mathbf{W} \in \mathbb{R}^{d\times M}$ is a matrix of weights, and $\mathbf{b}(\mathbf{x}) \in \mathbb{R}^M$ is a vector of fixed basis functions. This notably includes the case of (globally) linear models where we set $\mathbf{b}(\mathbf{x}) = \bar{\mathbf{x}} = (\mathbf{x}^T, 1)^T$, or the case of normalised radial basis functions (RBFs) $b_i(\mathbf{x}) = \frac{K(\mathbf{x}-\mathbf{c}_i)}{\sum_{j=1}^M K(\mathbf{x}-\mathbf{c}_j)}$ calculated from Gaussian kernels $K(\cdot)$ around M pre-determined centres $\mathbf{c}_i$, $i = 1 \ldots M$. With this model, the *inconsistency* error from (4) becomes

$$\begin{aligned} E_i(\mathbf{W}) &= \sum_{n=1}^N \left(r_n - \hat{\mathbf{u}}_n^T \mathbf{W}\mathbf{b}(\mathbf{x}_n)\right)^2 \\ &= \sum_{n=1}^N \left(r_n - \mathbf{v}_n^T \mathbf{w}\right)^2 = E_i(\mathbf{w}), \end{aligned}$$

where we defined[8] $\mathbf{w} \equiv vec(\mathbf{W})$ and $\mathbf{v}_n \equiv vec(\hat{\mathbf{u}}_n \mathbf{b}(\mathbf{x}_n)^T) = \mathbf{b}(\mathbf{x}_n) \otimes \hat{\mathbf{u}}_n$ in order to retrieve a simpler functional form. Since our objective function is quadratic in $\mathbf{w}$, we can rearrange to give

$$\begin{aligned} E_i(\mathbf{w}) &= \sum_n r_n^2 - 2\sum_n r_n \mathbf{v}_n^T \mathbf{w} + \mathbf{w}^T \sum_n \mathbf{v}_n \mathbf{v}_n^T \mathbf{w} \\ &= E_0 - 2\mathbf{g}^T\mathbf{w} + \mathbf{w}^T\mathbf{H}\mathbf{w} \end{aligned}$$

with $\mathbf{H} = \sum_n \mathbf{v}_n \mathbf{v}_n^T$ and $\mathbf{g} = \sum_n r_n \mathbf{v}_n$. Now, to solve for the optimal weight vector, we could take the direct inverse

$$\mathbf{w}_1 = \arg\min E_i(\mathbf{w}) = \mathbf{H}^{-1}\mathbf{g}.$$

However, this would ignore degeneracy in the solutions and may result in over-fitting. To avoid this we instead only optimise on elements of the weight vector that make a significant contribution to E_i. For this we perform an eigendecomposition for the inversion

$$\mathbf{w}_1 = \mathbf{V}_1 \mathbf{\Lambda}^{-1} \mathbf{V}_1^T \mathbf{g} \qquad (7)$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the large eigenvalues of $\mathbf{H}$ (i.e. eigenvalues above some minimum threshold $\lambda \geq \lambda_t$) and the columns of $\mathbf{V}_1$ are the corresponding eigenvectors.

In the part of the parameter space spanned by the remaining small eigenvectors[9] ($\lambda \leq \lambda_t$) we then perform the secondary optimisation. For the parametric model, we wish to minimise

$$E_2(\mathbf{W}) = \sum_{n=1}^N \|\mathbf{u}_n - \mathbf{W}\mathbf{b}(\mathbf{x}_n)\|^2 \qquad (8)$$

subject to the solution being optimal with respect to the inconsistency. We therefore look for a solution that has the form

$$\mathbf{w} = \mathbf{w}_1 + \mathbf{V}_2\mathbf{z}. \qquad (9)$$

where the columns of $\mathbf{V}_2$ contain the remaining eigenvectors of $\mathbf{H}$ and $\mathbf{z}$ is a vector. Using a solution of this form means that our optimisation of the model with respect to the secondary objective does not affect the primary optimisation of the inconsistency error.

Rearranging (8), we have

$$E_2(\mathbf{W}) = \sum_n \mathbf{u}_n^T\mathbf{u}_n - 2\sum_n \mathbf{u}_n^T \mathbf{W}\mathbf{b}_n + \sum_n \|\mathbf{W}\mathbf{b}_n\|^2 \quad (10)$$

which can be written in terms of $\mathbf{w}$ as

$$\begin{aligned} E_2(\mathbf{w}) &= \sum_n \mathbf{u}_n^T\mathbf{u}_n - 2\sum_n (\mathbf{b}_n \otimes \mathbf{u}_n^T)\mathbf{w} \\ &\quad + \mathbf{w}^T \left(\sum_n \mathbf{b}_n \mathbf{b}_n^T \otimes \mathbf{I}\right)\mathbf{w} \qquad (11) \\ &= E_{0,2} - 2\mathbf{m}^T\mathbf{w} + \mathbf{w}^T\mathbf{M}\mathbf{w}. \end{aligned}$$

where $E_{0,2} = \sum_n \mathbf{u}_n^T\mathbf{u}_n$, $\mathbf{m} \equiv \sum_n (\mathbf{b}_n \otimes \mathbf{u}_n^T)^T = vec(\mathbf{U}\mathbf{B}^T)$ and $\mathbf{M} \equiv \left(\sum_n \mathbf{b}_n\mathbf{b}_n^T \otimes \mathbf{I}\right) = \mathbf{B}\mathbf{B}^T \otimes \mathbf{I}$.

Substituting (9) and differentiating, we can then retrieve the optimal $\mathbf{z}$:

$$\mathbf{z}^{opt} = (\mathbf{V}_2^T\mathbf{M}\mathbf{V_2})^{-1}\mathbf{V}_2^T(\mathbf{m} - \mathbf{M}\mathbf{w}_1). \qquad (12)$$

We then combine (7) and (12) to find the optimal weights for our model

$$\mathbf{w}^{opt} = \mathbf{V}_1\mathbf{\Lambda}^{-1}\mathbf{V}_1^T\mathbf{g} + \mathbf{V}_2\mathbf{z}^{opt}. \qquad (13)$$

Finally, in order to automatically select the minimum eigenvalue threshold λ_t we perform a line search, repeating the above optimisation for a series of values of λ_t on a subset of the data, and picking the λ_t which minimises the quantity

$$E_\lambda[\tilde{\boldsymbol{\pi}}] = E_i[\tilde{\boldsymbol{\pi}}] + \alpha E_2[\tilde{\boldsymbol{\pi}}].$$

[7] This is similar to the minimum performance guarantee reported in [4] for the special case of potential-based policies, now extended to the learning of any arbitrary policy.

[8] To clarify notation: We denote the vector version of a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ as $vec(\mathbf{A}) = \mathbf{a} \in \mathbb{R}^{1\times nm}$ where the vector $\mathbf{a}$ is formed by stacking the columns of $\mathbf{A}$ on top of one another. Additionally, the notation $\mathbf{A}\otimes\mathbf{B}$ is used to denote the Kronecker product of the two matrices $\mathbf{A}$ and $\mathbf{B}$.

[9] Note that in the limit that $\lambda_t = 0$, (6) acts as a hard constraint on the secondary optimisation so that it only effects on model components that are strictly undetermined by the primary optimisation of E_i.

Here α is a weighting factor that reflects our prior belief on whether the data contains variable constraints. For example one would choose a very low α for data containing very high variance in the constraints.

D. Locally linear policy models

The basis function approach quickly becomes nonviable in high-dimensional input spaces. Alternatively, we can fit multiple locally weighted linear models $\tilde{\boldsymbol{\pi}}_m(\mathbf{x}) = \mathbf{B}_m\bar{\mathbf{x}} = \mathbf{B}_m(\mathbf{x}^T, 1)^T$ to the data, learning each local model independently [14]. For a linear model centred at $\mathbf{c}_m$ with an isotropic Gaussian receptive field with variance σ^2, we can write the inconsistency error

$$\begin{aligned} E_i(\mathbf{B}_m) &= \sum_{n=1}^{N} w_{nm}\left(r_n - \hat{\mathbf{u}}_n^T\mathbf{B}_m\bar{\mathbf{x}}_n\right)^2 \\ &= \sum_{n=1}^{N} w_{nm}\left(r_n - \mathbf{v}_n^T\mathbf{b}_m\right)^2 = E_i(\mathbf{b}_m) \end{aligned} \tag{14}$$

where we defined $\mathbf{b}_m = vec(\mathbf{B}_m)$ and $\mathbf{v}_n \equiv vec(\hat{\mathbf{u}}_n\bar{\mathbf{x}}_n^T)$ similarly to the parametric case. The factors $w_{nm} = \exp(-\frac{1}{2\sigma^2}\|\mathbf{x}_n - \mathbf{c}_m\|^2)$ weight the importance of each observation $(\mathbf{x}_n, \mathbf{u}_n)$, giving more weight to nearby samples. The optimal slopes $\mathbf{B}_m$ with respect to (14) are again retrieved using an eigendecomposition:

$$\mathbf{b}_{1,m} = \arg\min E_i(\mathbf{b}_m) = \mathbf{V}_{1,m}\boldsymbol{\Lambda}_m^{-1}\mathbf{V}_{1,m}^T\mathbf{g}_m \tag{15}$$

where $\boldsymbol{\Lambda}_m$ and $\mathbf{V}_{1,m}$ are the large eigenvalues and corresponding eigenvectors of the Hessian $\mathbf{H}_m = \sum_n w_{nm}\mathbf{v}_n\mathbf{v}_n^T$ for the mth local model and $\mathbf{g}_m = \sum_n w_{nm}r_n\mathbf{v}_n$. We select the number of eigenvalues used for the primary optimisation of the inconsistency using a subset-validation approach similar to the parametric case.

The secondary objective for this model is

$$\begin{aligned} E_2(\mathbf{B}_m) &= \sum_{n=1}^{N} w_{nm}\|\mathbf{u}_n - \mathbf{B}_m\bar{\mathbf{x}}_n\|^2 \\ &= E_{0,2} - 2\mathbf{m}_m^T\mathbf{b}_m + \mathbf{b}_m^T\mathbf{M}_m\mathbf{b}_m = E_2(\mathbf{b}_m) \end{aligned}$$

where $E_{0,2} = \sum_n w_{nm}\mathbf{u}_n^T\mathbf{u}_n$, $\mathbf{m}_m \equiv \sum_n w_{nm}(\bar{\mathbf{x}}_n \otimes \mathbf{u}_n^T)^T$ and $\mathbf{M}_m \equiv \left(\sum_n w_{nm}\bar{\mathbf{x}}_\mathbf{n}\bar{\mathbf{x}}_\mathbf{n}{}^T \otimes \mathbf{I}\right)$. Similar to the parametric case, we look for a solution of the form $\mathbf{b}_m = \mathbf{b}_{1,m} + \mathbf{V}_{2,m}\mathbf{z}_m$. This yields optimal weights

$$\mathbf{b}_m^{opt} = \mathbf{V}_{1,m}\boldsymbol{\Lambda}_m^{-1}\mathbf{V}_{1,m}^T\mathbf{g}_m + \mathbf{V}_{2,m}\mathbf{z}_m^{opt} \tag{16}$$

with

$$\mathbf{z}_m^{opt} = (\mathbf{V}_{2,m}^T\mathbf{M}_m\mathbf{V}_{2,m})^{-1}\mathbf{V}_{2,m}^T(\mathbf{m}_m - \mathbf{M}_m\mathbf{b}_{1,m}). \tag{17}$$

Finally, for predicting the global policy, we combine the local linear models using the convex combination

$$\tilde{\boldsymbol{\pi}}(\mathbf{x}) = \frac{\sum_{m=1}^{M} w_m\mathbf{B}_m\bar{\mathbf{x}}}{\sum_{m=1}^{M} w_m}; \quad w_m = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{c}_m\|^2\right).$$

IV. Experiments

In this section we report experiments exploring the performance of the new approach when learning on data from systems of varying complexity and size. First, in order to illustrate the concepts involved, we apply our method to data from a simulated 2-D toy system. We then test the scalability of the method to higher dimensional systems with more complex constraints using data from the joint-space of the 7-DOF DLR lightweight arm (Fig. 1). Finally we demonstrate the utility of our approach for learning a car-washing task from human motion capture data.

A. Toy Example

Our first experiment demonstrates the robustness of our approach for learning unconstrained policies from variable-constraint data. For this we set up a simple toy example consisting of a two-dimensional system with discontinuously switching motion constraints. As an example policy, we used a limit cycle attractor of the form

$$\dot{r} = r(\rho - r^2), \qquad \dot{\theta} = \omega \tag{18}$$

where r, θ are the polar representation of the Cartesian state space coordinates (i.e. $x_1 = r\sin\theta$, $x_2 = r\cos\theta$), ρ is the radius of the attractor and $\dot{\theta}$ is the angular velocity. For the experiments we set $\rho = 0.5\ m$ and $\omega = 1\ rad\ s^{-1}$ with a sampling rate of 50 Hz. Data was collected by recording 40 trajectories with random start states, of length 40 time steps, generated by (i) the unconstrained policy and (ii) the policy subject to random 1-D constraints. The latter had the form

$$\mathbf{A}(\mathbf{x}, t) = (\alpha_1, \alpha_2) \equiv \boldsymbol{\alpha} \tag{19}$$

where the $\alpha_{1,2}$ were drawn from a normal distribution, $\alpha_i = N(0,1)$. The constraints (19) mean that motion is constrained in the direction orthogonal to the vector $\boldsymbol{\alpha}$ in state space. These were randomly switched by generating a new $\boldsymbol{\alpha}$ twice at regular intervals during the trajectory, inducing sharp turns in the trajectories as can be seen in Fig. 4.

We used a parametric model to learn the policy through the hybrid optimisation approach as described in section III-C. For this toy problem, we chose our function model as a set of 36 normalised RBFs centred on a 6×6 grid, and we simply fixed the kernel width to yield suitable overlap. We repeated this experiment on 100 data sets and evaluated the normalised UPE and CPE (i.e. the prediction error with no constraints, and that under the training data constraints [4], [5]) and the inconsistency[10], divided by the number of data points and the variance of the policy $\boldsymbol{\pi}_n$ on a subset held out for testing. For comparison, we repeated the experiment using (i) direct regression on the observations (i.e. minimising the standard risk) and (ii) optimisation of the inconsistency alone (i.e. minimising the functional (4) without the secondary optimisation step) with the same RBF model.

Table I shows the results of learning with the different methods under the different constraint settings. Looking at the first row, we see that the direct regression approach is effective for learning on unconstrained data, but performs

[10] Actually, for $\mathbf{u} \in \mathbb{R}^2$ the inconsistency is exactly equivalent to the CPE, since both necessarily involve the same 1-D projection.

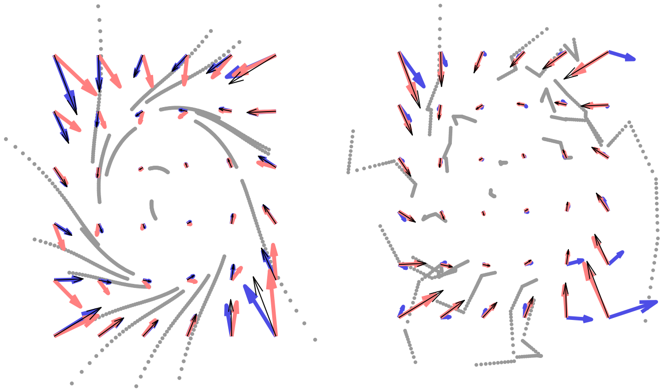


Fig. 4. Policy learnt with the direct approach (blue) and pure inconsistency approach (red) when training on unconstrained (left) and randomly constrained (right) data. The true policy (thin black arrows) and training data (grey trajectories) are overlaid.

Method	Constr.	nUPE	nCPE	Norm. Incon.
Direct	None	0.034 ± 0.044	0.034 ± 0.044	0.026 ± 0.039
	Rand.	58.338 ± 9.556	8.596 ± 2.813	8.596 ± 2.813
Incon.	None	26.640 ± 52.737	26.640 ± 52.737	0.014 ± 0.031
	Rand.	0.118 ± 0.162	0.007 ± 0.010	0.007 ± 0.010
Hybrid	None	0.065 ± 0.268	0.065 ± 0.268	0.042 ± 0.143
	Rand.	0.373 ± 1.109	0.011 ± 0.017	0.011 ± 0.017

TABLE I

ERROR FOR THE DIRECT, INCONSISTENCY AND HYBRID OPTIMISATION APPROACHES WHEN LEARNING ON $K = 40$ TRAJECTORIES OF LENGTH $N = 40$ POINTS, SAMPLED FROM THE LIMIT CYCLE POLICY. ALL VALUES GIVEN AS (MEAN±S.D.)$\times 10^{-2}$

poorly on data containing random constraints. This is in line with expectations since for the former the data is unaffected by constraints and is thus already consistent (i.e. a unique output is observed at each point in the input space), whereas for the latter the variability in the constraints causes model averaging. In contrast, looking at the second row we see that optimisation of the inconsistency is highly effective for learning the unconstrained policy when there is high variation in the constraints. However, on the unconstrained data, though the normalised inconsistency (5th column) is low, the policy errors are relatively large. The pure inconsistency approach misinterprets the variation in the policy as variation in the constraints, and fits an incorrect model (shown in red in Fig. 4).

In contrast, the proposed hybrid approach achieves very low errors both on the unconstrained and the constrained data. With this approach we get the best of both of the other approaches: For data that is already self-consistent it benefits from the tight fit offered by direct least-squares regression. Conversely if data contains variable constraints a model that is consistent with the observations under the different constraints is learnt.

To further test this, we repeated the experiment on data containing several levels of variability in the constraints. For this we again sampled a set of $K = 40$ trajectories of length $N = 40$ points from the limit cycle policy, however this time we applied the constraints

$$\mathbf{A}(\mathbf{x}, t) = \mathbf{I} - \hat{\boldsymbol{\alpha}}_{\boldsymbol{\pi}}^T \hat{\boldsymbol{\alpha}}_{\boldsymbol{\pi}} \qquad (20)$$

where $\hat{\boldsymbol{\alpha}}_{\boldsymbol{\pi}} \equiv \boldsymbol{\alpha}_{\boldsymbol{\pi}} / \|\boldsymbol{\alpha}_{\boldsymbol{\pi}}\|$, $\boldsymbol{\alpha}_{\boldsymbol{\pi}} \equiv \mathbf{R}(\theta)\boldsymbol{\pi}(\mathbf{x})$ and $\mathbf{R}(\theta)$ is a rotation matrix with rotation angle θ. The latter was drawn uniform randomly with increasing angular range, that is $\theta \sim U[-\theta^{max}, \theta^{max}]$ for increasing θ^{max}. This constraint was chosen since it allows us to smoothly vary the effect of the constraints on the observations. For example, for $\theta = 0$ the direction of the constraint is exactly orthogonal to the policy at that point so that the resultant projection has no effect on the policy. As the range of θ increases however, the observations of the unconstrained policy are increasingly corrupted by the projections induced by the constraints.

Fig. 5 depicts how the UPE and CPE evolve with increasing constraint variance (i.e. increasing θ^{max}) for the direct, pure inconsistency and hybrid optimisation approaches. For the direct approach, the UPE and CPE are low when the constraint variance is low, but rapidly increase as the variance grows due to increased model-averaging. In contrast, the pure inconsistency approach deals well with constraints of high variance since this increases the span of the observations, resulting in most of the components of the policy being picked up by the inconsistency error. However when the variance in constraints decreases, the pure inconsistency approach misinterprets the remaining variability in the observations (due to variation in the policy) as variation in the constraints, causing an increase in error. Finally, the proposed hybrid approach achieves consistently low errors irrespective of the variance in the constraints, by automatically finding the direct least-squares fit for low-variance in the constraints, and increasingly using the constraint-consistent fit for high-variance constraints.

B. Higher Dimensional Policies and Constraints

The goal of our second set of experiments was to evaluate the scalability of the approach to higher dimensional systems with constraints of varying dimensionality. This is important when considering systems where the number of constraints is near to the number of degrees of freedom of the system, for example constraining the position and orientation of the end-effector of a manipulator such as an anthropomorphic 7-DOF arm. It is also the case that with increasing numbers of dimensions there are increasing numbers of ways in which the system can be constrained, in terms both of the different dimensionalities of the constraints (i.e. rank of the constraint matrix) and the ways in which constraints can be combined.

For our experiment, we used a kinematic simulation of the 7-DOF DLR lightweight robot (LWR-III). The experimental procedure was as follows: We generated a random initial posture by drawing 7 joint angles uniformly from half the range of each joint, that is $x_i \sim U[-0.5x_i^{max}; 0.5x_i^{max}]$, where for example $x_1^{max} = 170°$. We set up a joint limit avoidance type policy as $\boldsymbol{\pi}(\mathbf{x}) = -0.05\nabla\Phi(\mathbf{x})$, with the potential given by $\Phi(\mathbf{x}) = \sum_{i=1}^{7} |x_i|^{1.8}$. We then generated 100 trajectories with 100 points each following the policy under 6 different constraints of differing dimensionality, which we refer to as 1, 1-2, 1-2-3, etc. Here, the numbers denote which end-effector coordinates in task space[11] we kept fixed, that is, 1-2-3 means we constrained the end-effector

[11] The numbers can also be read as row indices of the 6×7 Jacobian matrix.

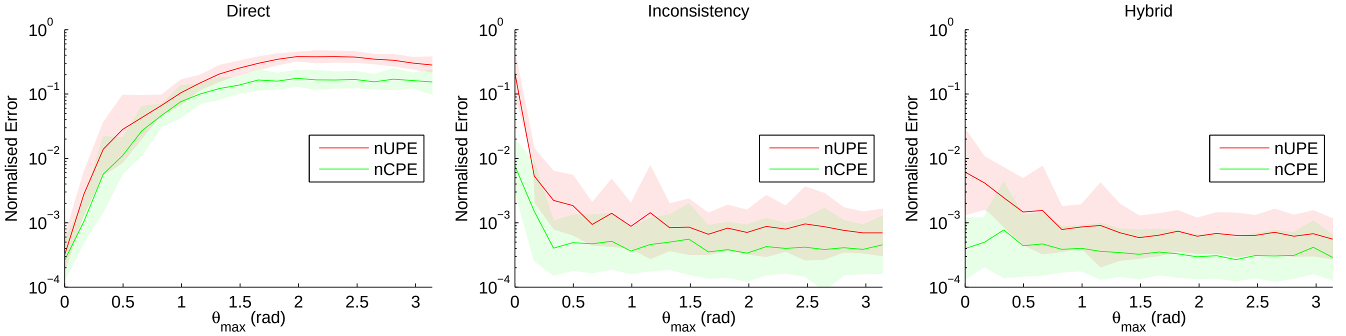


Fig. 5. Normalised UPE and CPE versus variance in the constraints for learning with the direct (left), pure inconsistency (centre) and hybrid optimisation (right) approaches.

position, but allowed arbitrary changes in the orientation. Similarly, 1-2-3-4 means we constrained the end-effector position and the orientation around the x-axis, while allowing movement around the y and z axes. For all constraint types, we estimated the policy from a training subset and evaluated the normalised CPE on test data from the same constraint, as well as the normalised UPE.

For learning in the 7-D state space, we selected locally linear models as described in Sec. III-D, where we chose rather wide receptive fields (fixing $\sigma^2 = 3$) and placed the centres $\{\mathbf{c}_m\}$ of the local models such that every training sample $(\mathbf{x}_n, \mathbf{u}_n)$ was weighted within at least one receptive field with $w_m(\mathbf{x}_n) \geq 0.7$. On average, this yielded about 50 local models.

The results are shown in Table II where we can see the following trends. First, as the constraint dimension increases, learning with the direct approach yields increasingly poor performance in terms of UPE and roughly consistent performance in terms of CPE. This is to be expected since, being naive to the effect of constraints, the direct approach attempts to find the closest fit to the constrained observations. Further, as the number of constraints increases the difference between the constrained and unconstrained policy vectors increases (since the number of components of the unconstrained policy projected out by the constraints increases). As a result the directly learnt model, while fitting the constrained policy closely, performs increasingly poorly in terms of UPE.

Second, for the pure inconsistency approach, we see that the CPE is worse for the 1-D constraint compared to the direct approach, but much better for the higher dimensional constraints. We also see much better performance in terms of the UPE for the intermediate constraints, but very large errors for the 6-D constraint. For the hybrid approach the UPE is uniformly better, and the CPE lower in all but the 1-D constraint case.

The improved UPE performance for these methods may be surprising given that the same constraint is applied for each observation. This would suggest that certain components of the policy are undetermined by the observations since they are never unconstrained. However, here the constraint matrix (i.e the Jacobian) is state-dependent, yielding some *spatial variability* in the constraints, and thereby sufficient information to improve the reconstruction of the unconstrained policy.

Looking at the inconsistency and hybrid approaches, we see that performance (especially in terms of CPE) increases with constraint dimensionality which can be explained by the approximation of the projection (as discussed in Sec. III-A) becoming increasingly accurate. In fact, for the 6-D constraint the approximation is exact.

Method	Constr.	nUPE	nCPE
Direct	1	26.94 ± 3.02	3.63 ± 0.54
	1-2	70.51 ± 2.22	5.72 ± 0.66
	1-2-3	80.70 ± 1.59	4.09 ± 0.33
	1-...-4	86.63 ± 1.36	4.66 ± 0.44
	1-...-5	91.47 ± 0.91	3.59 ± 0.39
	1-...-6	96.78 ± 0.78	1.85 ± 0.27
Incon.	1	18.30 ± 5.46	14.53 ± 5.08
	1-2	6.53 ± 2.90	1.04 ± 0.37
	1-2-3	6.93 ± 2.79	0.50 ± 0.11
	1-...-4	4.57 ± 2.49	0.27 ± 0.02
	1-...-5	5.28 ± 3.40	0.16 ± 0.02
	1-...-6	233.37 ± 136.97	0.04 ± 0.01
Hybrid	1	10.54 ± 4.56	6.98 ± 3.90
	1-2	5.85 ± 1.94	1.00 ± 0.30
	1-2-3	18.17 ± 8.00	0.55 ± 0.14
	1-...-4	8.04 ± 4.16	0.28 ± 0.03
	1-...-5	8.98 ± 5.25	0.18 ± 0.03
	1-...-6	41.30 ± 3.93	0.05 ± 0.01

TABLE II

NORMALISED UPE AND CPE FOR THE THREE METHODS WHEN TRAINING ON DATA FROM THE DLR ARM. ALL ERRORS NORMALISED BY THE VARIANCE OF THE POLICY. WE REPORT (MEAN ± S.D.)$\times 10^{-2}$ OVER 50 TRIALS WITH DIFFERENT DATA SETS.

However, for this latter constraint, we see an explosion in UPE for the pure inconsistency approach which is not seen for the hybrid approach. We attribute this to the combined spatial variation in the policy and the constraints in this particular case, to which the inconsistency approach is overly sensitive. On inspection we noted that the Hessian matrices of the local models had become ill-conditioned in this case. The secondary optimisation in the hybrid approach avoids this problem and emphatically outperforms the two other approaches.

C. Car Washing Experiment

Having validated our approach on data where the ground truth (true unconstrained policy) was known, in this section we report experiments on learning from human demonstrations for seeding the robot motion. For this experiment we chose to investigate the problem of learning to wash a car. This is an example of a task which can be intuitively described in terms of a simple movement policy ('wiping') subject to contact constraints that vary depending on the different surfaces of the car to be wiped. Due to the different shapes and orientations of the car surfaces, complex, non-linear contact constraints are imposed on the motion. The resultant trajectories appear periodic, but are perturbed in different ways by the constraints. The goal of our experiments was to learn a policy that captured the periodic nature of the movements, and generalised well over the constraints, i.e. to unseen surfaces.

The experimental setup was as follows. Seven demonstrations of a human wiping different surfaces with a sponge were given to the robot. To simulate observations of washing different surfaces of the car, the wiping was performed on a perspex sheet placed at different tilts and rotations with respect to the robot (see Fig. 6). Specifically, the sheet was oriented to be flat (horizontal), tilted $\pm 16^\circ$ and $\pm 27^\circ$ about the x-axis (horizontal axis pointing directly ahead from the robot) and $\pm 16^\circ$ about the y-axis (horizontal right-left axis). The three-dimensional coordinates of the sponge were tracked by a stereo vision system at a rate of 20 frames per second (for details on the vision system see [2]).

We selected the local linear model for learning, with a fixed kernel width of $\sigma^2 = 0.025$, and centres placed so that every data point was weighted with at least $w_m(\mathbf{x}_n) \geq 0.7$. For this data set this yielded about 22 local models. We trained this model with the three approaches on the five trajectories corresponding to surface rotation about the x-axis, holding the remaining two trajectories out for testing.

To evaluate performance we compared the policy predictions from the three models under different constraints with the observed data. Specifically, since the ground truth (including the true constraints) is unknown, we assumed constraints of the form $\mathbf{A}_j(\mathbf{x}, t) = \hat{\mathbf{n}}_j$ where $\hat{\mathbf{n}}_j$ is the normal to the jth surface, i.e. that the sponge did not penetrate, and could not be lifted from the surface.

Under this approximation of the constraints, we found that the policy learnt with the hybrid approach produced smooth, periodic trajectories when implemented on the DLR arm both under the test and training constraints (see accompanying video). We regard this as remarkably good performance on this very noisy data set.

V. Conclusion

In this paper, we described a method for robust learning of policies from constrained observations. Building upon earlier work [5] we introduced a two-stage optimisation approach which seamlessly combines standard direct policy learning with our idea of fitting a model that is consistent with variable constraint data. Although the previous approach could handle cases where demonstrated movements are subject to variable, dynamic, non-linear and even discontinuous constraints, it suffered from poor performance on data containing highly correlated constraints or purely unconstrained data. The novel approach proposed here avoids these problems as demonstrated in our experiments. We illustrated the utility of our method for learning a car washing task from human demonstration data.

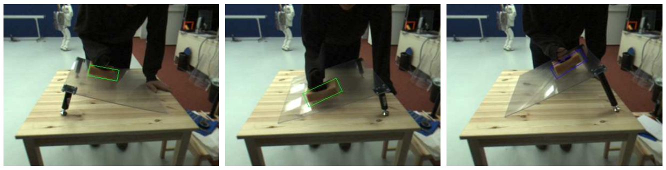
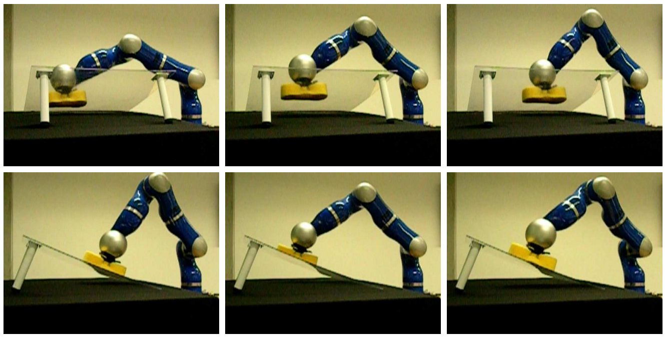


Fig. 6. Above: Human wiping demonstrations on surfaces of varying tilt and rotations. A stereo vision system was used to track the 3-D coordinates of the sponge (coloured rectangles show the estimated position). Tilts of $\pm 16^o$ and $+27^o$ about the x-axis are shown. Below: Reproduction of the movement on the DLR Lightweight arm on a training constraint (top row) and an unseen test constraint (bottom row).



References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In *Handbook of Robotics*. MIT Press, 2007.

[2] B. Bolder, M. Dunn, M. Gienger, H. Janssen, H. Sugiura, and C. Goerick. Visually guided whole body interaction. In *ICRA*, 2007.

[3] S. Calinon and A. Billard. Learning of gestures by imitation in a humanoid robot. In *Imitation & Social Learning in Robots, Humans & Animals: Behavioural, Social & Communicative Dimensions*, 2007.

[4] M. Howard, S. Klanke, M. Gienger, C. Goerick, and S. Vijayakumar. Behaviour generation in humanoids by learning potential-based policies from constrained motion. *Appl. Bionics and Biomechanics*, 5:195–211, 2008.

[5] M. Howard, S. Klanke, M. Gienger, C. Goerick, and S. Vijayakumar. A novel method for learning policies from constrained motion data. In *ICRA*, 2009.

[6] A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *NIPS*, 2003.

[7] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura. Embodied symbol emergence based on mimesis theory. *Int. J. Robotics Research*, 23:363–377, 2004.

[8] K. Ohta, M. Svinin, Z. Luo, S. Hosoe, and R. Laboissiere. Optimal trajectory formation of constrained human arm reaching movements. *Biol. Cybern.*, 91:23–36, 2004.

[9] J. Park and O. Khatib. Contact consistent control framework for humanoid robots. In *ICRA*, 2006.

[10] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal. A unifying framework for robot control with redundant DOFs. *Autonomous Robots J.*, 24:1–12, 2008.

[11] J. Peters and S. Schaal. Learning to control in operational space. *Int. J. Robotics Research*, 27:197–212, 2008.

[12] S. Schaal. Learning from demonstration. In *NIPS*, 1997.

[13] S. Schaal. Dynamic movement primitives - a framework for motor control in humans and humanoid robotics. In H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witte, editors, *Adaptive Motion of Animals and Machines*, pages 261–280. Springer Tokyo, 2006.

[14] S. Schaal and C. Atkeson. Constructive incremental learning from only local information. *Neural Computation*, 10:2047–2084, 1998.

[15] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Phil. Trans.: Biological Sciences*, 358:537–547, 2003.

[16] F. Udwadia and R. Kalaba. *Analytical Dynamics: A New Approach*. Cambridge University Press, 1996.