Auton Robot (2009) 27: 105–121
DOI 10.1007/s10514-009-9129-8

A novel method for learning policies from variable constraint data

Matthew Howard · Stefan Klanke · Michael Gienger · Christian Goerick · Sethu Vijayakumar

Received: 7 November 2008 / Accepted: 23 July 2009 / Published online: 30 July 2009

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 come from movements under different constraints. As a key ingredient, we introduce a small but highly effective modification to the standard risk functional, allowing us to make a meaningful comparison between the estimated policy and constrained observations. We demonstrate our approach on systems of varying complexity, including kinematic data from the ASIMO humanoid robot with 27 degrees of freedom, and present results for learning from human demonstration.

Keywords Direct policy learning · Constrained motion · Imitation · Nullspace control

1 Introduction

A wide variety of everyday human skills can be framed in terms of performing some task subject to constraints imposed by the physical environment (Ohta et al. 2004; Svinin et al. 2005). Examples include opening a door, pulling out a drawer or stirring soup in a saucepan.

In a more generic setting, constraints may take a much wider variety of forms (Udwadia and Kalaba 1996). For example, in climbing a ladder, the constraint may be on the centre of mass or the tilt of the torso of the climber to prevent over-balancing. Alternatively, in problems that involve control of contacts such as manipulation or grasping solid objects, the motion of fingers is constrained during the grasp by the presence of the object (Sapio et al. 2006; Park and Khatib 2006). In systems designed to be highly competent and adaptive, such as humanoid robots, behaviour may be subject to a wide variety of constraints that are usually non-linear in actuator space and often discontinuous (Sentis and Khatib 2004, 2005, 2006; Gienger et al. 2005; Sapio et al. 2005). Consider the task of running or walking on uneven terrain: the cyclic movement of the legs of the runner is constrained by the impact of the feet on the ground in a dynamic, discontinuous and unpredictable way.

A promising approach to providing robots with such skills as running and opening doors is to take examples of motion from existing demonstrators (e.g., from humans) and attempt to learn a control policy that somehow captures the desired behaviour (Ratliff et al. 2009; Calinon and Billard 2007; Billard et al. 2007; Alissandrakis et al. 2007; Grimes et al. 2006, 2007; Chalodhorn et al. 2006; Takano et al. 2006; Schaal et al. 2003; Inamura et al. 2004; Ijspeert et al. 2003). An important component of this is the ability to deal with the effect of constraints and the apparent variability in movements induced by these constraints. For example, one wishes to learn a policy that allows one not only to open a specific door of a particular size (e.g. constraining the hand to a curve of a particular radius), but rather to open many doors of varying sizes (or radii).

Electronic supplementary material The online version of this article (http://dx.doi.org/10.1007/s10514-009-9129-8) contains supplementary material, which is available to authorized users.

M. Howard (✉) · S. Klanke · S. Vijayakumar
Institute of Perception Action and Behaviour, University of Edinburgh, Edinburgh, Scotland, UK
e-mail: matthew.howard@ed.ac.uk

M. Gienger · C. Goerick
Honda Research Institute Europe (GmBH), Offenbach, Germany

The focus in this paper is on modelling control policies subject to a specific class of constraints on motion, with the aim of finding policies that can generalise *over different constraints*. We take a direct policy learning (DPL) approach (Martinez-Cantin et al. 2009; Vlassis et al. 2009; Stolle and Atkeson 2009; Chalodhorn et al. 2006; Nakanishi et al. 2004; Schaal et al. 2003; Atkeson and Schaal 1997; Mussa-Ivaldi 1997) whereby we attempt to learn a continuous model of the policy from motion data. While DPL has been studied for a variety of control problems in recent years,[1] crucially these problems involved policies that are either directly observable from motion data, i.e. unconstrained policies, or policies subject to identical constraints in every observation, in which case the constraints can be absorbed into the policy itself. The difference here is that we consider observations from policies projected into the nullspace of a set of dynamic, non-linear constraints, and that these constraints may *change between observations*, or even during the course of a single observation.

Our strategy is to attempt to consolidate movement observations under different specific constraints to find the underlying *unconstrained policy* common to all. Learning the latter enables generalisation since we can apply new constraints to predict behaviour in novel scenarios. In general, learning (unconstrained) policies from constrained motion data is a formidable task. This is due to (i) the *non-convexity* of observations under different constraints, and; (ii) *degeneracy* in the set of possible policies that could have produced the movement under the constraint (Howard et al. 2008; Howard and Vijayakumar 2007). However, despite these hard analytical limits, we will show that it is still possible to find a good approximation of the unconstrained policy given observations under the right conditions. Noting that the policy observations are projected into the nullspace of the constraints, our proposal is to reformulate the standard risk functional by introducing a projection of the estimated policy onto the observations before calculating errors. By making this simple, but significant alteration, we show that it is possible to model the unconstrained policy (i) with no explicit knowledge of the constraints, and; (ii) without explicit access to unconstrained policy vectors. Furthermore, we show that using this approach one can fully reconstruct the unconstrained policy given observations under a sufficiently rich set of constraints. To validate the approach we modify standard regression techniques to use the proposed objective function and demonstrate robust learning for several policies on complex, high-dimensional movement systems, subject to realistic constraints. Finally, we demonstrate the use of our approach for learning from human demonstrations and transferring behaviour to the ASIMO humanoid robot.

[1]For a review on DPL, please see (Billard et al. 2007) and references therein.

2 Problem formulation

In this section, we characterise the problem of DPL in general, and discuss the problems encountered when variable constraints are applied to motion.

2.1 Direct policy learning

Following Schaal et al. (2003), we consider the learning of autonomous policies

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

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^d$ are appropriately[2] chosen state- and action-spaces, respectively. The goal of DPL is to approximate the policy (1) as closely as possible (Schaal et al. 2003). It is usually formulated as a supervised learning problem where it is assumed that we have observations of $\mathbf{u}(t)$, $\mathbf{x}(t)$ (often in the form of trajectories), and from these we wish to learn the mapping $\boldsymbol{\pi}$. In previous work this has been done by fitting parametrised models in the form of dynamical systems (Ijspeert et al. 2002, 2003), non-parametric modelling (Peters and Schaal 2008a; Calinon and Billard 2007; D'Souza et al. 2001), probabilistic Bayesian approaches (Grimes et al. 2006, 2007) and hidden Markov models (Takano et al. 2006; Inamura et al. 2004).

An implicit assumption found in DPL approaches to date is that the data used for training comes from behavioural observations of some *unconstrained* or *consistently constrained* policy (Calinon and Billard 2007). By this it is meant that the policy is observed either under no constraint (e.g. movements in free space such as gestures or figure drawing), or under constraints consistent over observations (e.g. interacting with the same objects or obstacles in each case). However, in many everyday behaviours, there is variability in the constraints, such as when opening doors of varying sizes or walking on uneven terrain. This *variability in the constraints* cannot be accounted for by standard DPL approaches.

[2]It should be noted that, as with all DPL approaches, the choice of state- and action-space is problem specific (Schaal et al. 2003) and, when used for imitation learning, depends on the *correspondence* between demonstrator and imitator. For example if we wish to learn the policy a human demonstrator uses to wash a window, and transfer that behaviour to an imitator robot, an appropriate choice of $\mathbf{x}$ would be the Cartesian coordinates of the hand, which would correspond to the end-effector coordinates of the robot. Transfer of behaviour across non-isomorphic state- and action-spaces, for example if the demonstrator and imitator have different embodiments, is also possible by defining an appropriate state-action metric (Alissandrakis et al. 2007).

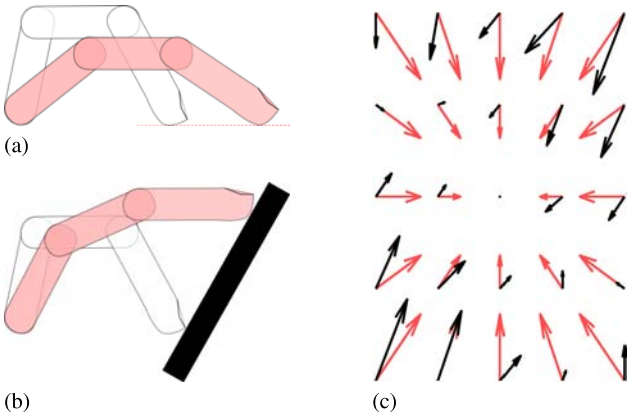


Fig. 1 Illustration of two apparently different behaviours from the same policy: (**a**) unconstrained movement (**b**) movement constrained by an obstacle (*black box*) (**c**) the unconstrained (*red*) and constrained (*black*) policy over two of the joints of the finger (y-axis: base joint; x-axis: first knuckle)

2.1.1 *Example: finger extension with contact constraints*

As an example, consider the learning of a simple policy to extend a jointed finger. In Fig. 1(a) the finger is unconstrained and the policy simply moves the joints towards the zero (outstretched) position. On the other hand, in Fig. 1(b), an obstacle lies in the path of the finger, so that the finger movement is constrained—it is not able to penetrate the obstacle, so moves along the surface. The vector field representation of the two behaviours is shown in Fig. 1(c).

Given the task of learning in this scenario, applying traditional DPL approaches would result in one of two possibilities. The first is that if the observations are *labelled with respect to the constraint* (here, the orientation, position and shape of the obstacle) one could learn a separate policy model for the behaviour in each of the settings. However this is clearly unsatisfactory, since each model would only be valid for the specific setting, and we would need increasing numbers of models as we observed the policy under new constraints (for example different shaped obstacles at different positions and orientations). The second possibility is that the data is *unlabelled* with respect to the constraint. In this case, one might try to perform regression directly on the observations, that is observations from both vector fields (cf. Fig. 1(c), black and red vectors). However, this presents the problem that *model averaging* would occur across observations under different constraints, resulting in a poor representation of the movement in terms of the magnitude and direction of the predictions (see Sect. 2.3).

We can avoid the need for multiple policy models if we relax our assumptions on the form (1) of the observed commands, and allow for an additional transformation of $\boldsymbol{\pi}(\mathbf{x})$. We thus model both the red and black observations as stemming from the same policy ('extend the finger'), and attribute its different appearance to the transformations as induced by the constraints. With a restriction on the class of possible transformations, as will be detailed in the next section, we can model the unconstrained policy even if we only observed constrained movements, and we can apply new constraints to adapt the policy to novel scenarios.

2.2 Constraint model

In this paper we consider constraints which act as hard restrictions on actions available to the policy. Specifically, we consider policies subject to a set of k-dimensional ($k \leq n$) Pfaffian constraints[3]

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

Under these constraints, the policy is projected into the nullspace of $\mathbf{A}(\mathbf{x}, t)$:

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

where $\mathbf{A}(\mathbf{x}, t) \in \mathbb{R}^{k\times d}$ is some matrix describing the constraint, $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix and $\mathbf{N}(\mathbf{x}, t) \equiv (\mathbf{I} - \mathbf{A}^{\dagger}\mathbf{A}) \in \mathbb{R}^{d\times d}$ is projection matrix,[4] that in general has nonlinear dependence on both time and state. Constraints of the form (2) commonly appear in scenarios where manipulators interact with solid objects, for example when grasping a tool or turning a crank or a pedal, that is, contact constraint scenarios (Park and Khatib 2006; Murray et al. 1994; Mattikalli and Khosla 1992). Such constraints are also common in the control of redundant degrees of freedom in high-dimensional manipulators (Liégeois 1977; Khatib 1987; Peters et al. 2008), where policies such as (3) are used, for example, to aid joint stabilisation (Peters et al. 2008), or to avoid joint limits (Chaumette and Marchand 2001), kinematic singularities (Yoshikawa 1985) or obstacles (Choi and Kim 2000; Khatib 1985) under task constraints. As an example: Setting $\mathbf{A}$ to the Jacobian that maps from joint-space to end-effector position increments would allow any motion in the joint space provided that the end-effector remained stationary. The formalism is generic and can also readily be applied to learning policies based on dynamic quantities such as torques or (angular and linear) momentum subject to constraints (e.g., see Peters et al. 2008 and Kajita et al. 2003, respectively). Such constraints are also not limited to manipulator kinematics and dynamics, for example Antonelli et al. (2005), apply it to team coordination in mobile robots.

In general, the effect of constraints (2)–(3) is to disallow motion in some sub-space of the system, specifically

[3] A thorough treatment of the role of constraints such as (2)–(3) on the dynamics of multibody systems can be found in many standard texts on analytical mechanics, for example see Udwadia and Kalaba (1996).

[4] Here, $\mathbf{A}^{\dagger}$ denotes the (unweighted) Moore-Penrose pseudoinverse of the matrix $\mathbf{A}$.

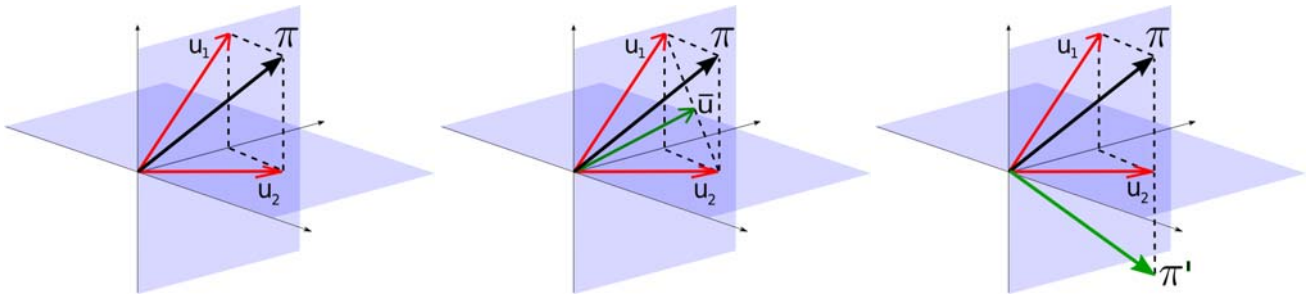


Fig. 2 Illustration of the effect of constraints on the unconstrained policy, the averaging effect of standard DPL and the degeneracy problem. *Left*: Two constraints applied to the policy $\boldsymbol{\pi}$ result in projected observations $\mathbf{u}_1, \mathbf{u}_2$. *Centre*: direct regression results in averaging of the two movements $\bar{\mathbf{u}}$ in a way that cannot explain the observations. *Right*: Two policies $\boldsymbol{\pi}, \boldsymbol{\pi}'$ that both may be constrained in such a way as to produce the observation $\mathbf{u}_2$

the space orthogonal to the image of $\mathbf{N}(\mathbf{x}, t)$. In essence, these components of motion are *projected out* of the observed movement. For example, as illustrated in Fig. 2(a), a policy $\boldsymbol{\pi}$ is constrained in two different ways corresponding to two different projections of the unconstrained movement. In the first observation $\mathbf{u}_1$, the constraint prevents movement in the direction normal to the vertical plane.[5] For the second observation $\mathbf{u}_2$, the constraint only allows movement in the horizontal plane.

2.3 Learning from constrained motion data

From the viewpoint of learning, constraints as described in the previous section present problems for traditional DPL approaches. Specifically there are two issues that must be dealt with; that of *non-convexity* of observations and *degeneracy* between policies (Howard et al. 2008).

The *non-convexity* problem comes from the fact that between observations, or even during the course of a single observation, constraints may change. For example, in Fig. 2(b), the two policy observations under the different constraints, $\mathbf{u}_1$ and $\mathbf{u}_2$, appear different depending on the constraint. To the learner, this means that the data from the two scenarios will appear *non-convex*, i.e. for any given point $\mathbf{x}$ in the input space, multiple outputs $\mathbf{u}$ may exist. This causes problems for supervised learning algorithms since, for example, directly training on these observations may result in model-averaging. Here, averaging of $\mathbf{u}_1, \mathbf{u}_2$ results in the prediction $\bar{\mathbf{u}}$ that clearly does not match the unconstrained policy $\boldsymbol{\pi}$, neither in direction nor magnitude (ref. Fig. 2(b)).

The *degeneracy* problem stems from the fact that for any given set of projected (constrained) policy observations, there exist multiple candidate policies that could have produced that movement. This is due to the projection eliminating components of the unconstrained policy that are orthogonal to the image of $\mathbf{N}(\mathbf{x}, t)$ so that the component of $\boldsymbol{\pi}$ in this direction is undetermined by the observation. For example, consider the constrained observation $\mathbf{u}_2$ in Fig. 2(c), where the restriction of the motion in vertical direction implies that we do not observe that specific component of $\boldsymbol{\pi}$. Given only $\mathbf{u}_2$ we cannot determine if the policy $\boldsymbol{\pi}$ or an alternative, such as $\boldsymbol{\pi}'$ (ref. Fig. 2(c)), produced the observation. In effect, we are not given sufficient information about the unconstrained policy to guarantee that it is fully reconstructed.

Despite these restrictions, we wish to do the best we can with the data available. We adopt a strategy whereby we look for policies that are, at the very least, consistent with the constrained observations $\mathbf{u}$. For example, returning to Fig. 2(c), if we only observe $\mathbf{u}_2$, (that is the policy under a single, specific constraint), the simplest (and safest) strategy would be to use that same vector as our prediction. In this way, we can at least accurately predict the policy under that constraint (albeit only under that particular constraint). If we are given further observations under new constraints, we can recover more information about the unconstrained policy $\boldsymbol{\pi}$. For instance, observing $\mathbf{u}_1$ eliminates the possibility that $\boldsymbol{\pi}'$ underlies the movements since it cannot project onto both $\mathbf{u}_1$ and $\mathbf{u}_2$. Applying this strategy for increasing numbers of observations, our model will not only generalise over the constraints seen, but also come closer to the unconstrained policy $\boldsymbol{\pi}$.

Finally, it should be noted that if in all observations certain components of the policy are always constrained, then we can never hope to uncover those components. However, in such cases it is reasonable to assume that, if these components are always eliminated by the constraints, then they are not relevant for the scenarios in which movements were recorded.

[5] It should be noted that in general the orientation of the constraint plane onto which the policy is projected may vary both with state position and time.

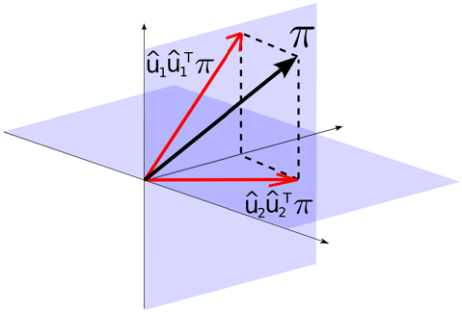


Fig. 3 Illustration of our learning scheme. The projection of the correct policy $\boldsymbol{\pi}$ onto the observations matches those observations

In the following, we propose a method by which we can overcome these problems by reformulating the standard DPL problem. We will show that it is still possible to learn a good model of the policy $\boldsymbol{\pi}$, without need for explicit knowledge of the constraints $\mathbf{N}(\mathbf{x}, t)$, and that is, as a minimum, consistent with all constrained observations. In previous work, we demonstrated the feasibility of this and proposed an algorithm that allowed us to learn potential-based policies from constrained motion data (Howard et al. 2008). Here we remove the restriction to potential-based policies allowing us to learn any generic policy that can be represented as a vector function of state. We turn to the details of the new method in the next section.

3 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 $\mathbf{N}_n$. Furthermore, we assume that the projection matrix for any given observation is not explicitly known, i.e. our data is unlabelled with respect to the constraints in force at the time of observation.

With only $\mathbf{x}_n$ and $\mathbf{u}_n$ given, one may be tempted to estimate a policy $\tilde{\boldsymbol{\pi}}(\cdot)$ by simply minimising

$$E_{naive}[\tilde{\boldsymbol{\pi}}] = \sum_{n=1}^{N} \|\mathbf{u}_n - \tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\|^2. \tag{4}$$

However, this would ignore that constraints might have been in force and result in a naive averaging of commands from different circumstances (cf. Fig. 2). This corresponds to the standard DPL approach.

If we had access to samples of either (i) the (unconstrained) policy $\boldsymbol{\pi}_n = \boldsymbol{\pi}(\mathbf{x}_n)$, or (ii) the projection matrices $\mathbf{N}_n$, we could use standard regression techniques to estimate a policy $\tilde{\boldsymbol{\pi}}(\mathbf{x})$ by minimising an appropriate risk functional. Specifically, in the former case, we could minimise

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

and in the latter case, we could minimise

$$E_{cpe}[\tilde{\boldsymbol{\pi}}] = \sum_{n=1}^{N} \|\mathbf{u}_n - \mathbf{N}_n\tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\|^2, \tag{6}$$

where we refer to the former as the *unconstrained policy error* (UPE) and the latter as the *constrained policy error* (CPE), respectively. However, since by assumption samples of $\boldsymbol{\pi}_n$ and $\mathbf{N}_n$ are not available, these functionals cannot be used to estimate the policy.

Instead, we aim to estimate a policy $\tilde{\boldsymbol{\pi}}(\cdot)$ that is *consistent* with our observations, that is, a policy that can be projected in a way that the observed commands are recovered. To this end, we replace $\mathbf{N}_n$ in (6) by a projection onto $\mathbf{u}_n$ and minimise the *inconsistency* which we define as the functional

$$\begin{aligned} E_i[\tilde{\boldsymbol{\pi}}] &= \sum_{n=1}^{N} \|\mathbf{u}_n - \hat{\mathbf{u}}_n\hat{\mathbf{u}}_n^T\tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\|^2 \\ &= \sum_{n=1}^{N} \big(r_n - \hat{\mathbf{u}}_n^T\tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\big)^2 \\ &\text{with } r_n = \|\mathbf{u}_n\|,\ \hat{\mathbf{u}}_n = \frac{\mathbf{u}_n}{r_n}. \end{aligned} \tag{7}$$

Since $\mathbf{u}_n = \mathbf{N}_n\boldsymbol{\pi}_n$, we can write $\|\mathbf{u}_n - \mathbf{N}_n\tilde{\boldsymbol{\pi}}(\mathbf{x}_n)\|^2 = \|\mathbf{N}_n(\boldsymbol{\pi}_n - \tilde{\boldsymbol{\pi}}(\mathbf{x}_n))\|^2$ and recognise that the CPE is always less than or equal to the UPE, because the projections $\mathbf{N}_n$ can only decrease the norm of the difference between true and predicted policy. The same argument holds for the inconsistency error (7) where the projection onto the 1-D subspace spanned by $\hat{\mathbf{u}}_n$ possibly takes away even more of the error. So we can establish the inequality

$$E_i[\tilde{\boldsymbol{\pi}}] \leq E_{cpe}[\tilde{\boldsymbol{\pi}}] \leq E_{upe}[\tilde{\boldsymbol{\pi}}].$$

Naturally, for estimating the correct policy, we would rather like to minimise an *upper bound* of E_{upe}, but it is unclear how such a bound could be derived from the data we are assumed given. Note that by framing our learning problem as a risk minimisation task, we can apply standard regularisation techniques such as adding suitable penalty terms to prevent over-fitting due to noise.

The proposed risk functional 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 for learning the (unconstrained) policy to demonstrate how the risk functional

can be used. The example function approximators we use are (i) simple parametric models with fixed basis functions (Sect. 3.1), and (ii) locally linear models (Sect. 3.2). In the next section, we describe how these two models can be reformulated to take advantage of the new risk functional.

3.1 Parametric policy models

A particularly convenient model of the policy 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 $\mathbf{c}_i$, $i = 1 \dots M$. With this model, the *inconsistency* error from (7) becomes

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

where we defined $\mathbf{w} \equiv vec(\mathbf{W})$ 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 solve for the optimal weight vector easily:

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

yielding

$$\mathbf{w}^{opt} = \arg\min E_i(\mathbf{w}) = \mathbf{H}^{-1}\mathbf{g} \tag{8}$$

with $\mathbf{g} = \sum_n r_n \mathbf{v}_n$, Hessian $\mathbf{H} = \sum_n \mathbf{v}_n \mathbf{v}_n^T$ and $E_0 = \sum_n r_n^2$. For regularisation, we use a simple weight-decay penalty term, that is, we select $\mathbf{w}_{reg}^{opt} = \arg\min(E_i(\mathbf{w}) + \lambda\|\mathbf{w}\|^2)$. This only requires modifying the Hessian to $\mathbf{H}^{reg} = \sum_n \mathbf{v}_n \mathbf{v}_n^T + \lambda\mathbf{I}$.

Please note that the projection onto $\mathbf{u}$ introduces a coupling between the different components of $\tilde{\boldsymbol{\pi}}$, which prevents us from learning those independently as is common in normal regression tasks. For the same reason, the size of the Hessian scales with $O(d^2M^2)$.

3.2 Locally linear policy models

The basis function approach quickly becomes non-viable 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 (Schaal and Atkeson 1998). For a linear model centred at $\mathbf{c}_m$ with an isotropic Gaussian receptive field with variance σ^2, we would minimise

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

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

$$w_{nm} = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_n - \mathbf{c}_m\|^2\right)$$

weight the importance of each observation $(\mathbf{x}_n, \mathbf{u}_n)$, giving more weight to nearby samples. The optimal slopes $\mathbf{B}_m$ in vector form are retrieved by

$$\mathbf{b}_m^{opt} = \arg\min E_i(\mathbf{b}_m) = \mathbf{H}_m^{-1}\mathbf{g}_m \tag{9}$$

with $\mathbf{H}_m = \sum_n w_{nm}\mathbf{v}_n\mathbf{v}_n^T$ and $\mathbf{g}_m = \sum_n w_{nm} r_n \mathbf{v}_n$.

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

$$\tilde{\boldsymbol{\pi}}(\mathbf{x}) = \frac{\sum_{m=1}^M w_m \mathbf{B}_m \bar{\mathbf{x}}}{\sum_{m=1}^M w_m}$$

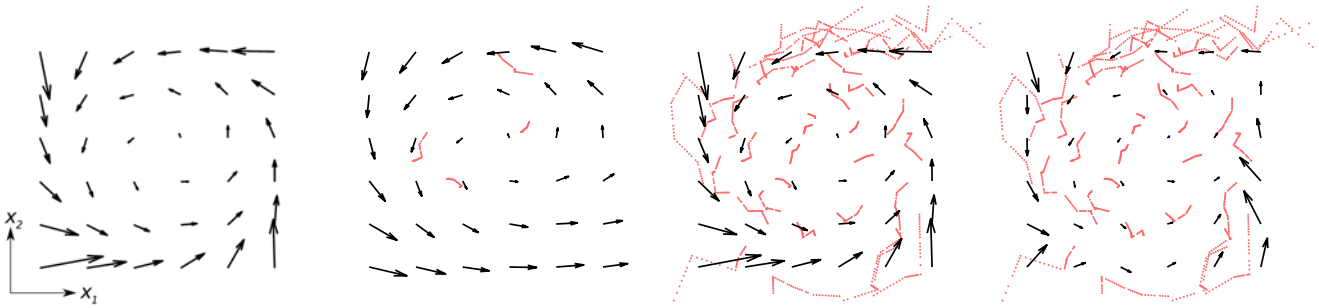
where $w_m = \exp(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{c}_m\|^2)$.

4 Experiments

To explore the performance of our algorithm, we performed experiments on data from autonomous control policies (Schaal et al. 2003) applied to three plants. In our first set of experiments we illustrate the concepts involved on an artificial two-dimensional toy system.[6] We then demonstrate how our method can generalise across constraints on a physically realistic simulation of the 7-DOF DLR lightweight arm (Sect. 4.2). Next, we apply our algorithm to whole body motion control of the humanoid robot ASIMO (Gienger et al. 2005), where we learn policies in both a 6-D task space (Sect. 4.3) and in the 27-DOF joint space (Sect. 4.4).

Having validated the approach on the data where the ground truth is known, we finally explore the utility of our approach for learning in a real imitation learning setting: We demonstrate an application of our approach to enable the ASIMO robot to learn a car washing task from observed human movements (Sect. 4.5).

[6] In fact even these 'simplified' problems are relevant to learning policies in low dimensional task spaces, such as end-effector space.



a) true policy | b) our method, trained on 4 traj. | c) our method, 40 trajectories | d) naive method, 40 trajectories

	b)	c)	d)
nUPE:	0.3788 ± 0.2688	0.0027 ± 0.0087	0.5709 ± 0.0853
nCPE:	0.1276 ± 0.1140	0.0002 ± 0.0002	0.0010 ± 0.0363

Fig. 4 Results on 2D toy data. (**a**) true limit cycle policy, (**b**) learnt policy trained on 4 constrained trajectories, (**c**) learnt policy from 40 constr. traj., (**d**) policy resulting from naive regression on observed commands. Trajectories are shown as *dotted lines*, *black arrows* depict the policy evaluated at a grid of points in the 2-D state space. The normalised CPE and UPE (mean±s.d. over 100 data sets) are given below the figures

4.1 Toy example

Our first experiment demonstrates the learning of unconstrained policies from constrained trajectories in 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 (Fig. 4(a)) of the form

$$\dot{r} = r(\rho - r^2), \qquad \dot{\theta} = \omega \tag{10}$$

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 $(\mathbf{x}_n, \mathbf{u}_n)$ (where $\mathbf{x}_n$ is the Cartesian position and $\mathbf{u}_n \equiv \dot{\mathbf{x}}_n$ the Cartesian velocity) was collected by recording 40 trajectories of length 40 time steps each, generated by the policy from a random start state. During the movement the policy was subjected to random 1-D constraints

$$\mathbf{A}(\mathbf{x}, t) = (\alpha_1, \alpha_2) \equiv \boldsymbol{\alpha}, \tag{11}$$

where the $\alpha_{1,2}$ were drawn from a normal distribution, $\alpha_i = N(0, 1)$. The constraints 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 which can be seen in Fig. 4 (b–d).

We used a parametric model to learn the policy through minimisation of the inconsistency (7) as described in Sect. 3.1. We included the regularisation term and picked the parameter λ by minimising the inconsistency on a validation subset. 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[7] the normalised UPE, CPE and the inconsistency,[8] that is, the functionals from (5), (6) and (7) 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 a naive approach that attempted to perform regression with the same RBF model directly on the constrained observations, i.e., the naive approach attempted to minimise the functional (4).

Figure 4 shows the true policy, the trajectories we trained on, the policies learnt using our and the naive approach, and finally the error statistics below the plots. With an average nUPE of 0.0027, our method outperforms the naive approach by orders of magnitude. Notably, even with only 4 trajectories (Fig. 4(b)), the reconstructed policy already resembles the limit cycle, although large errors still persist in some parts of the state space (e.g., the lower right corner). Further to this, the left panel of Fig. 5 depicts how the nUPE and nCPE evolve with increasing size of the training set, showing a smooth decline (please note the log. scale). In order to further explore the performance of our algorithm, we contaminated the observed commands $\mathbf{u}_n$ with Gaussian noise, the scale of which we varied to match up to 20% of the scale of the data. The resulting nUPE roughly follows the noise level, as is plotted in Fig. 5 (right).

[7] Here, since we wish to evaluate the performance of the approach, we calculate errors in the model against the ground truth, i.e. the policy (10) and constraints (11) from which the data was sampled. Note that these quantities are *not* made available to our algorithm during learning.

[8] 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.

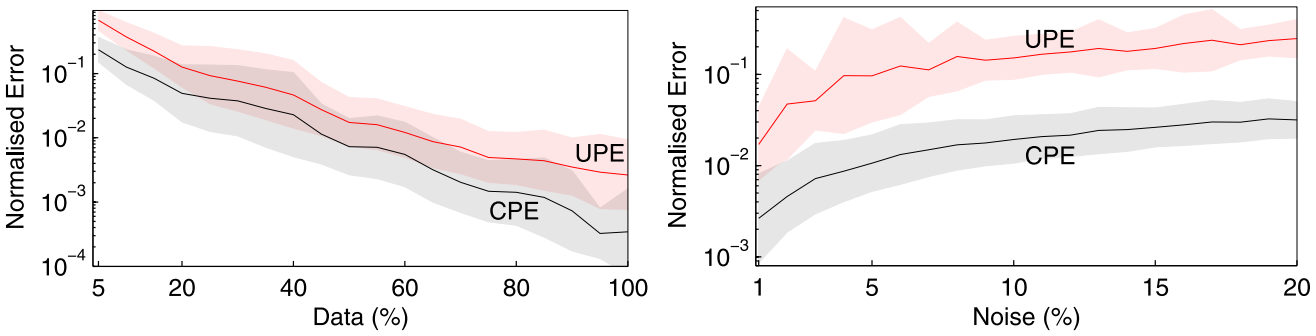


Fig. 5 *Left*: Normalised UPE and CPE versus data set size as a percentage of the full $K=40$ trajectories of length $N=40$. *Right*: Normalised UPE and CPE for increasing noise levels in the observed $\mathbf{u}_n$. For clarity, we do not report the (consistently high) errors of the naive method. Both plots show the (mean±s.d.) over 100 data sets (coloured area indicates one standard deviation from the mean)

4.2 Generalisation over unseen constraints

The two goals of our second set of experiments were (i) to characterise how well the algorithm scaled to more complex, realistic constraints and (ii) to characterise how well the learnt policies generalised over unseen constraints. For this, 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 $\phi(\mathbf{x}) = \sum_{i=1}^{7} |x_i|^p$ for $p = 1.5$, $p = 1.8$, or $p = 2.0$. We then generated 100 trajectories with 100 points each, following the policy under 4 different constraints, which we refer to as 1-2-3, 4-5-6, 1-3-5, and 2-4-6. Here, the three numbers denote which end-effector coordinates in task space[9] we kept fixed, that is, 1-2-3 means we constrained the end-effector position, but allowed arbitrary changes in the orientation (here, orientation was represented as yaw, pitch and roll angles in the inertial frame). Similarly, 2-4-6 means we constrained the y-coordinate and the orientation around the x- and z-axis, while allowing movement in x-z position and around the y-axis. For all 4 constraint types, we estimated the policy from a training subset, and evaluated it on test data from the same constraint, as well as on trajectories from the complementary constraint (e.g., 2-4-6 is complementary to 1-3-5).

For learning in the 7-D state space, we selected locally linear models as described in Sect. 3.2, 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.

While the linear policy $\boldsymbol{\pi}(\cdot)$ corresponding to $p=2.0$ was learnt almost perfectly (all normalised errors were in the order of 10^{-9}), the less linear policies ($p=1.8$ and especially $p=1.5$) turned out to be a much harder problem. This can be seen when comparing both the nUPE and nCPE for the two policies (ref. Table 1). Still, we recovered the constrained policy in all cases to good accuracy (ref. Table 1, 4th column), with good generalisation to the complementary constraints (ref. Table 1, 5th column). We can also see that constraining the end-effector position (1-2-3) made it more difficult to recover the unconstrained policy compared to constraining the orientation (4-5-6), or using mixed constraints (1-3-5 and 2-4-6). It should also be noted that running the same experiment using the naive approach (ref. Sect. 4.1) gave consistently poor results; for example, when training on data under the (1-2-3) constraint, the naive approach gave nUPE of $83.44 \pm 1.20 \times 10^{-2}$ for the $p=1.5$ policy, $80.94 \pm 1.37 \times 10^{-2}$ for $p=1.8$ and $79.62 \pm 1.39 \times 10^{-2}$ for $p=2.0$.

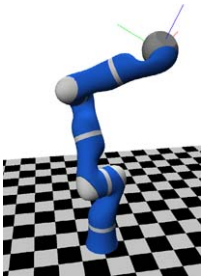
4.3 Reaching for a ball

The goal of our next set of experiments was to illustrate the utility of our approach for learning from observations of an everyday task with realistic constraints. For this, we chose an example scenario, in which a set of observations of a demonstrator performing the task of reaching for a ball on a table are given, and the student is expected to learn a policy to enable it to reproduce this task. The learning problem is complicated however, by the presence of different obstacles on the table for each of the example trajectories, constraining the possible motions of the hands. The goal is to uncover a policy that accurately predicts the demonstrator's (unconstrained) behaviour and generalises to predict the behaviour under novel constraints.

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

Table 1 Normalised UPE, CPE on the training constraints, CPE on complementary constraints and inconsistency error, for data from the DLR arm (*right*). All errors normalised by the variance of the policy. We report (mean ± s.d.)$\times 10^{-2}$ over 100 trials with different data sets

Potential	Constraint	nUPE	nCPE	Compl. nCPE	Norm. Incon.
$p=1.5$	1-2-3	64.338 ± 32.030	2.917 ± 0.368	15.951 ± 6.473	0.755 ± 0.067
	4-5-6	34.753 ± 19.125	2.491 ± 0.228	15.478 ± 7.755	0.388 ± 0.036
	1-3-5	16.179 ± 3.813	3.204 ± 0.276	5.108 ± 1.079	0.706 ± 0.067
	2-4-6	10.355 ± 1.827	2.723 ± 0.237	4.749 ± 0.956	0.401 ± 0.039
$p=1.8$	1-2-3	8.096 ± 5.766	0.477 ± 0.088	2.278 ± 1.133	0.112 ± 0.011
	4-5-6	5.364 ± 2.961	0.352 ± 0.038	2.221 ± 0.984	0.051 ± 0.006
	1-3-5	2.275 ± 0.645	0.455 ± 0.041	0.773 ± 0.171	0.098 ± 0.011
	2-4-6	1.421 ± 0.314	0.401 ± 0.042	0.729 ± 0.174	0.058 ± 0.007



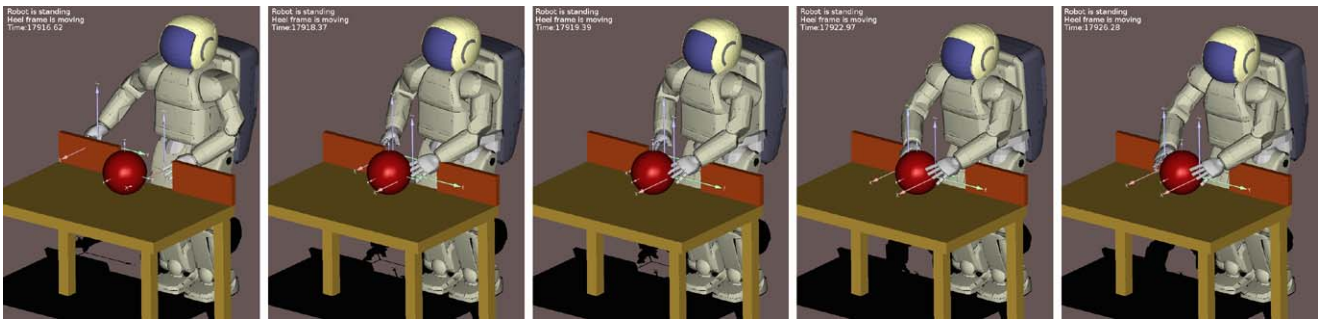


Fig. 6 Example constrained trajectory used as training data in the ball-reaching experiment. Starting with hands at the sides, the demonstrator robot reaches between the barriers to get the ball. Note that the width of the gap in the barriers was randomly altered for each trajectory recorded

The example scenario was implemented using the whole body motion (WBM) controller of the 27-DOF humanoid robot ASIMO (for details see Gienger et al. 2005). For this, data was recorded from a 'demonstrator' robot that followed a policy defined by an inverted Gaussian potential

$$\boldsymbol{\pi}(\mathbf{x}) = -\nabla_{\mathbf{x}}\phi(\mathbf{x}); \quad \phi(\mathbf{x}) = \alpha\left(1 - e^{\|\mathbf{x}-\mathbf{x}_c\|^2/2\sigma^2}\right), \tag{12}$$

where $\mathbf{x} \in \mathbb{R}^6$ corresponds to the Cartesian position of the two hands (hereafter, the 'task space') and the actions $\mathbf{u} = \dot{\mathbf{x}} = \boldsymbol{\pi}(\mathbf{x})$ correspond to the hand velocities. We chose $\sigma^2 = 2$, $\alpha = 0.25$ and the target point $\mathbf{x}_c \in \mathbb{R}^6$ to correspond to a reaching position, with the two hands positioned on either side of the ball. Following the policy (12) with this set of parameters, the demonstrator was able to reach the ball under each of the constraints considered in this experiment (see below). Inverse kinematics via the WBM controller was used to map the desired task space policy motion into the appropriate joint-space velocity commands for sending to the robot.

The demonstrator's movements were constrained by the presence of a barrier on the table with a gap in it, placed so that the demonstrator robot had to reach through the gap to get the ball (ref. Fig. 6). The barriers acted as inequality constraints on each of the hands so that motion in the direction normal to the barrier surface was prevented if a hand came too close. Specifically, the constraints took the form

$$\mathbf{A}(\mathbf{x}, t) = \left(\begin{array}{c|c} \mathbf{A}_{[1,1]} & \mathbf{0} \\ \hline \mathbf{A}_{[1,2]} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{A}_{[2,1]} \\ \hline \mathbf{0} & \mathbf{A}_{[2,2]} \end{array}\right), \tag{13}$$

where

$$\mathbf{A}_{[i,j]}(\mathbf{x}, t) = \hat{\mathbf{n}}_j^T; \quad d_{i,j} \leq d_{min} \text{ and } \hat{\mathbf{u}}_{[i]}^T \hat{\mathbf{n}}_j > 0$$

$$\mathbf{A}_{[i,j]}(\mathbf{x}, t) = \mathbf{0}; \quad \text{otherwise.}$$

Here, $d_{i,j}$ is the distance of the ith hand (where $i \in \{1, 2\}$, i.e. left and right hands respectively) to the closest point on the jth barrier (where $j \in \{1, 2\}$, i.e. left and right barriers respectively), $\hat{\mathbf{n}}_j \in \mathbb{R}^3$ is the normal to the barrier surface[10] at that point and $\hat{\mathbf{u}}_{[i]} \in \mathbb{R}^3$ is the normalised command for the ith hand (i.e. the ith 3-vector block of the command vector $\mathbf{u}$ corresponding to that hand; for example for the right hand ($i = 2$) this was $\mathbf{u}_{[2]} \equiv (u_4, u_5, u_6)^T$ with $\hat{\mathbf{u}}_{[2]} = \mathbf{u}_{[2]}/|\mathbf{u}_{[2]}|$). Here, the full constraint matrix $\mathbf{A}(\mathbf{x}, t) \in \mathbb{R}^{4\times 6}$ was constructed by assigning 3-vectors to the appropriate

[10]Note that in order to ensure smooth, natural-looking trajectories the barriers were modelled as planes with smooth 'swept-sphere' edges, similar to those described in Sugiura et al. (2007).

matrix blocks $\mathbf{A}_{[i,j]}$, according to the system state. For example, if the left hand ($i = 1$) approached the left barrier ($j = 1$) to a distance of $d_{1,1} < d_{min}$, and if the next commanded movement would bring the hand toward that barrier (i.e. $\hat{\mathbf{u}}_{[1]}^T \hat{\mathbf{n}}_1 > 0$), then the elements of the constraint matrix corresponding to that hand/barrier pair were updated (in this example the first row of the matrix would be updated, $\mathbf{A}_{1,:} = (\hat{\mathbf{n}}_1^T, 0, 0, 0)$, constraining the left hand). Note that under this setup the constraints are highly nonlinear (due to the complex dependence on state) and have discontinuously switching dimensionality (i.e. the rank of $\mathbf{A}(\mathbf{x}, t)$ switches) when either of the hands approaches or recedes from the barrier.

Data was collected by recording $K = 100$ trajectories of length $2s$ at 50 Hz, (i.e. $N = 100$ points per trajectory) from the demonstrator following the policy (12) under the constraints (13). Start states were sampled from a Gaussian distribution over joint configurations $\mathbf{q} \sim N(\mathbf{q}_0, 0.1\mathbf{I})$ (where $\mathbf{q}_0$ corresponds to the default standing position) and using forward kinematics to calculate the corresponding hand positions. The joint vector $\mathbf{q}$ was clipped where necessary to avoid joint limits and self collisions, and to ensure the start postures looked natural.

For each trajectory, the constraints were varied by randomly changing the width of the gap in the barriers. The gap widths were sampled from a Gaussian distribution $d_{gap} \sim N(\mu_{gap}, \sigma_{gap})$ where $\mu_{gap} = 0.25m$, $\sigma_{gap} = 0.1m$ and the diameter of the ball was $0.15m$. The hand-barrier distance at which the constraints came into force was fixed at $d_{min} = 0.05m$. Figure 6 shows an example trajectory under this setup.

We used our algorithm to perform learning on 50 such data sets using 150 local linear models, with centres placed using k-means. For comparison, we also repeated the experiment on the same data with the same local linear model (i.e. same number and placement of centres), but using the naive approach for training (i.e. training on $(\mathbf{x}_i, \mathbf{u}_i \equiv \dot{\mathbf{x}}_i)$, $i = 1, \ldots K \times N$ directly, using the risk functional (4)).

To assess the performance for both methods we evaluated the errors in predicting the policy subject to (i) the training data constraints (nCPE), (ii) no constraints (nUPE), and (iii) a novel constraint, unseen in the training data, on a set of test data. For the latter, a barrier was placed centrally between the robot and the ball, so that the robot had to reach around the barrier to reach the ball (see Fig. 8). Specifically, the constraint took a form similar to (13) but this time with only one barrier present (i.e. $j \equiv 1$), so that the constraint matrix $\mathbf{A}(\mathbf{x}, t) \in \mathbb{R}^{2\times 6}$ attained a maximum rank of $k = 2$ when both hands approached the barrier. The width of the new barrier was fixed at $0.5m$.

As expected, learning using the proposed risk functional (7) (the 'non-naive' approach) performed several orders of magnitude better than the naive approach in terms of the numerical error measures (ref. Table 2). However, the real difference in the methods is best highlighted if we compare trajectories generated by the two policies under different constraint settings.

Table 2 Normalised policy errors for predicting the policy under three constraint conditions from the ball-reaching data for the naive and non-naive methods. Values are mean±s.d. over 50 data sets

Constraint	Naive	Non-naive
Training	0.1940 ± 0.0153	0.0056 ± 0.0022
Unseen Barrier	0.4678 ± 0.0264	0.0057 ± 0.0023
Unconstrained	0.7014 ± 0.0430	0.0058 ± 0.0023

Firstly, Fig. 7 shows example trajectories for the *unconstrained* reaching movements produced by the demonstrator ('expert'), and the policies learnt by (i) the naive approach, and; (ii) the non-naive approach; from a number of start states. In the former the hands always take a curved path to the ball (Fig. 7, top), reproducing the average behaviour of the (constrained) demonstrated trajectories. The naive method is unable to extract the underlying task (policy) from the observed paths around the obstacles. In contrast, the policy learnt with the non-naive approach better predicts the unconstrained policy, enabling it to take a direct route to the ball that closely matches that of the demonstrator (Fig. 7, bottom).

Secondly, Fig. 8 shows example trajectories when the policies are again constrained. Figure 8 (top) shows the movement from the non-naive policy under a similar constraint as in the training data. Under this constraint both naive and non-naive policies take a similar path as the demonstrator: The hands move in first, then forward to the ball. Note that under this constraint the movement of the naive policy is noticeably slower due to averaging of the constrained observations.

Finally, under the unseen barrier constraint, there is a marked difference in behaviour. Under this constraint, the demonstrator (still following the policy (12)) reaches around the barrier to get the ball. This behaviour is reproduced by the policy learnt with the proposed approach (Fig. 8, middle). In contrast however, the naive policy does not generalise to the new constraint and gets trapped behind the barrier, eventually dislodging it[11] (Fig. 8, bottom). The behaviour of the three policies (demonstrator, naive and non-naive policies) can be examined in detail in the accompanying video.

[11] Note that the collision of the hands with the barrier in fact violates the constraint. The reason for this is that on the real robot, under this constraint, the naive policy forces the robot into a self-collision (of the robot's arms with the torso). To prevent damage to the robot, an on-board safety mechanism then kicks in and pushes the hands away from the body, causing collision with the barrier.

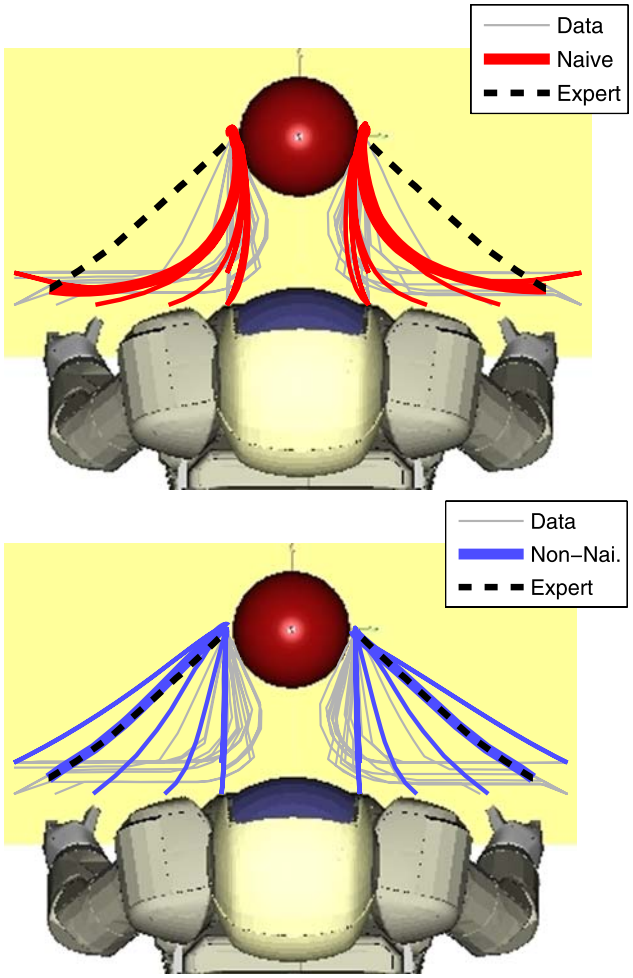


Fig. 7 Reaching movements produced by the policies learnt by the naive approach (*top*) and by optimisation of the inconsistency (*bottom*) when unconstrained. Shown are trajectories of the hands from five start states, with one example highlighted (*thick line*). The demonstrator ('expert') trajectory corresponding to the highlighted example is overlaid (*black dashed line*). Twenty example training data trajectories are also shown (*thin grey lines*)

4.4 Learning from High-dimensional Joint-space Data

In our next experiment we tested the scalability of our approach for learning in very high dimensions. For this we chose a policy defined by a quadratic potential in the joint space (i.e. $\mathbf{x} \equiv \mathbf{q} \in \mathbb{R}^{27}$)

$$\boldsymbol{\pi}(\mathbf{x}) = -\nabla_{\mathbf{x}}\phi(\mathbf{x}); \quad \phi(\mathbf{x}) = (\mathbf{x}-\mathbf{x}_c)^T\mathbf{W}(\mathbf{x}-\mathbf{x}_c), \tag{14}$$

where $\mathbf{x}_c \in \mathbb{R}^{27}$ is a target posture and $\mathbf{W}$ is a weighting matrix. The policy (14) represents an attractor in joint space that pulls the robot into a desired posture at $\mathbf{x}_c$. For the experiments, $\mathbf{x}_c$ was chosen to correspond to a reaching posture with both arms outstretched (ref. Fig. 9, right) and we chose $\mathbf{W} = 0.05\mathbf{I}$.

During data collection, the policy was constrained by the presence of obstacles which took the form of a vertical wall placed directly in front of the robot at different orientations and distances (ref. Fig. 9, left). Specifically, the constraint matrix, $\mathbf{A}(\mathbf{x}, t) \in \mathbb{R}^{2\times 27}$, took the form

$$\begin{aligned} &\mathbf{A}_i(\mathbf{x}, t) = \mathbf{0}; \quad d_i > 0 \\ &\mathbf{A}_i(\mathbf{x}, t) = \hat{\mathbf{n}}^T\mathbf{J}_i(\mathbf{x}); \quad \text{otherwise.} \end{aligned} \tag{15}$$

Here, $\hat{\mathbf{n}} \in \mathbb{R}^2$ is the normal[12] to the wall surface, d_i is the perpendicular distance of the ith hand from the wall surface (with $i \in \{1, 2\}$, i.e. left and right hands respectively), $\mathbf{J}_i(\mathbf{x}) \in \mathbb{R}^{2\times 27}$ is the Jacobian mapping from joint-space to the lateral (i.e. horizontal planar) coordinates of that hand and $\mathbf{A}_i(\mathbf{x}, t) \in \mathbb{R}^{1\times 27}$ is the corresponding row of the constraint matrix. At the start of each trajectory, the orientation of the wall was drawn from a uniform random distribution $\theta \sim U[-\theta^{max}, \theta^{max}]$ where θ is the angle of the wall with respect to the left-right axis of the robot heel frame (horizontal axis in Fig. 9, left), and we chose $\theta^{max} = 27°$. The distance of the wall was adjusted at the start of each trajectory to ensure that the hands were a minimum distance of $0.15m$ from the wall before the onset of movement.

The effect of the constraints was to restrict the movement of the hands when they approached the wall. This constraint was projected back into the joint space where the policy was operating via the Jacobian. This causes the policy to appear highly complex and non-linear in the state space (joint space), with discontinuous changes to the dimensionality of the constraints as the hands of the robot approached the wall.

Using the formalism from Sect. 3.1 with $\mathbf{b}(x) = \bar{\mathbf{x}}$, we fitted linear models to 100 data sets, each consisting of 100 trajectories of 100 data points. Despite the high dimensionality, our method reached a normalised UPE of $0.291 \pm 0.313 \times 10^{-2}$. It is important to point out that this result can not only be explained by our choice of a linear model where we knew that the true policy (14) was also linear: Direct (naive) linear regression on the observed commands resulted in a normalised UPE of $63.9 \pm 3.1 \times 10^{-2}$ (nCPE was $7.98 \pm 0.66 \times 10^{-2}$), which again is orders of magnitude higher and similar to our results on toy data.

4.5 Washing a car

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')

[12]Note that since the wall was vertical in all example trajectories (and thus did not affect vertical movements) only the normal in the horizontal plane is relevant to calculation of the constraints.

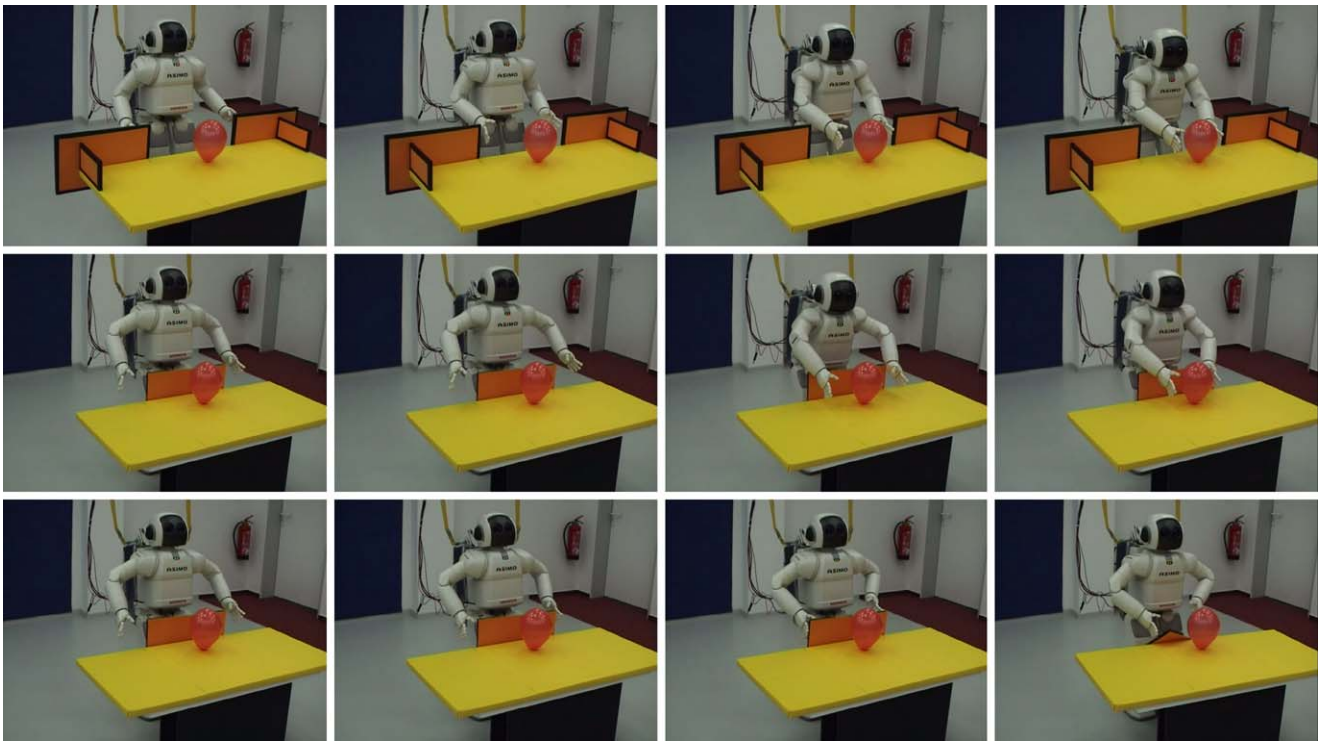


Fig. 8 Reaching movements produced by the learnt policies under different constraints. Shown are trajectories from (**i**) the non-naive policy under a similar constraint as in the training data (*top row*); (**ii**) the non-naive policy under a new, unseen barrier constraint (*middle row*), and; (**iii**) the naive policy under the new constraint

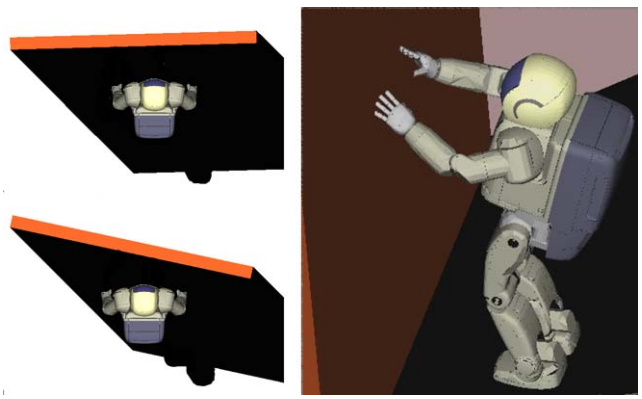


Fig. 9 Data collection for the joint space policy under wall constraints. *Left*: Start states for two example reaching movements with the wall at different distances and orientations with respect to the robot. *Right*: Side view after reaching

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 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, while eliminating artifacts induced by the constraints.

Fig. 10 Human wiping demonstrations on surfaces of varying tilt and rotations. The ASIMO stereo vision system was used to track the 3-D coordinates of the sponge (*coloured rectangles* show the estimated position). Tilts of $\pm 16^{\circ}$ and $+27^{\circ}$ about the x-axis are shown

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. 10). 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 using the on-board stereo cameras of the ASIMO robot at a rate of 20 frames per second (for details on the ASIMO vision system please see Bolder et al. 2007). The recorded trajectories are shown in Fig. 11 (left).

The policy was modelled as the $\mathbb{R}^3 \mapsto \mathbb{R}^3$ mapping from hand (sponge) positions to velocities. Since this is a rela-

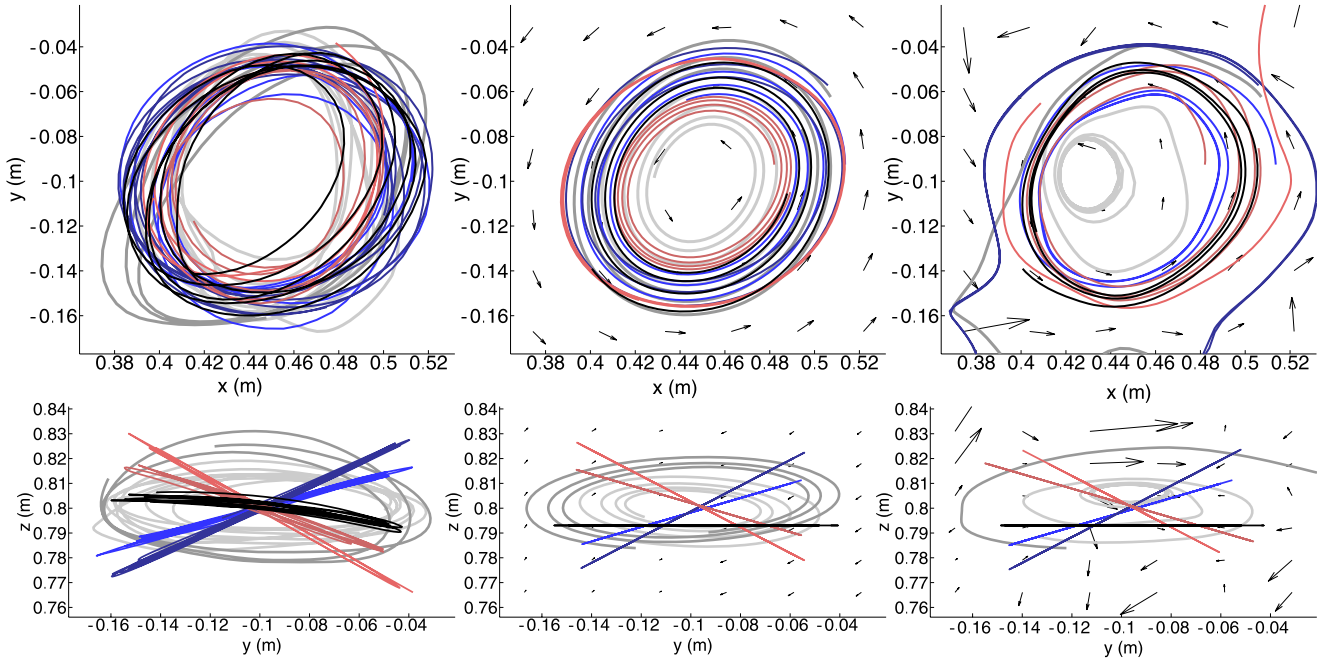


Fig. 11 Learning from human wiping demonstrations. *Left*: Trajectories of the sponge when wiping on the surface when flat (*black*), tilted $+16°$ and $+27°$ about the x-axis (*red*), $-16°$ and $-27°$ about the x-axis (*blue*), and $\pm 16°$ about the y-axis (*grey*). *Centre* and *right*: Reproduced trajectories using the policies (*black arrows*) learnt with the non-naive and naive approaches respectively. In each case the same example trajectory is highlighted (*thick black*). The top and front views are shown (*top* and *bottom rows*)

tively low-dimensional problem, and for ease of comparison with the toy problem (Sect. 4.1), we used RBFs to model the policy. For each of the experiments described below, we used a set of 300 RBFs with centres placed by k-means as our policy model.

Since the ground truth (i.e. the true unconstrained policy and the exact constraints in force) is not known for the human data, performance was evaluated on a behavioural level. In particular, we looked at how the movements produced by the learnt policies compared with those of the human when subject to (what we assumed to be) a similar set of constraints. For this, we implemented the learnt policies on the ASIMO humanoid robot and applied constraints that approximated[13] those contained in the demonstrations.

Specifically, we assumed the constraints in the car wash task to arise from two sources, namely (i) environmental (i.e. physical) constraints and (ii) constraints self-imposed by the demonstrator to ensure task success. In this experiment, the former can be clearly identified as an inequality constraint preventing the hand from penetrating the wiping surface, i.e.

$$\mathbf{A}(\mathbf{x}, t) = \hat{\mathbf{n}}_s(\mathbf{x}); \quad d = 0 \text{ and } \hat{\mathbf{u}}^T \hat{\mathbf{n}}_s(\mathbf{x}) > 0 \tag{16}$$

where d is the distance of the hand from the surface and $\hat{\mathbf{n}}_s(\mathbf{x})$ is the normal to the surface s at point $\mathbf{x}$. In addition, we can also identify a self-imposed constraint in force. In the car wash setting, successful performance of the task (i.e. wiping) requires the sponge to maintain contact with the surface at all times so that motion of the hand away from the surface (i.e. lifting the sponge) is not permitted. To capture this, we therefore assumed a further constraint of the form

$$\mathbf{A}(\mathbf{x}, t) = \hat{\mathbf{n}}_s(\mathbf{x}); \quad d = 0 \text{ and } \hat{\mathbf{u}}^T \hat{\mathbf{n}}_s(\mathbf{x}) < 0. \tag{17}$$

Note that in combination, the effect of the two constraints (16)–(17), when considered on the wiping surface ($d = 0$), amounts to the single equality constraint

$$\mathbf{A}(\mathbf{x}, t) = \hat{\mathbf{n}}_s(\mathbf{x}); \quad d = 0. \tag{18}$$

This constraint was applied to the learnt policies as a reasonable approximation of the true constraints contained in the data, in order to compare the demonstrated and reproduced movements for any given surface s and assess the generalisation across constraints.

Under this set-up, we first compared learning with our approach against learning with the naive approach. For this, we trained two RBF models on the full data set of seven demonstrations (i.e. wiping data for each of the surfaces). The first model was trained with the approach described in Sect. 3.1, the second with the standard (naive) approach to regression. We then used the policies learnt by the two ap-

[13] Please note that for training the policy models, the constraints were not explicitly modelled.

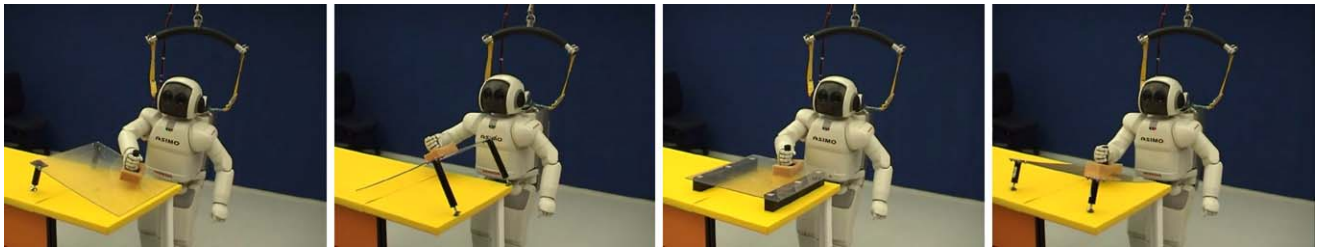


Fig. 12 Reproduced movements on the ASIMO robot for the surface tilted 0°, +16°, −27° about the x-axis, and +16° about the y-axis

proaches to reproduce the movements under each of the surface constraints (i.e. constraint (18) for $s = 1, \ldots, 7$). The results are shown in Fig. 11, where we show the demonstrated trajectories (left), those produced by the non-naive policy (centre) and those learnt by the naive approach (right) under the different constraints (tilts of the surface).

Looking at the learnt policies, we see that our approach learns a smooth policy that resembles the limit cycle of Sect. 4.1. The trajectories under each of the constraints are smooth periodic movements, similar to those of the human. These were implemented on the ASIMO robot to produce natural wiping movements (see Fig. 12). The policy learnt with the naive approach also captures the periodicity to some extent. However, it appears highly irregular in several regions and the trajectories are unstable, with some spiralling in to the centre, and others diverging to other parts of the state space. By attempting to learn all of the artifacts induced by the constraints, the naive approach learns an unstable policy that cannot be safely used for movement reproduction on the robot.[14]

Finally, to confirm that our approach is able to generalise well over unseen constraints, we repeated the experiment, but this time training the model on a subset of the data containing one set of constraints, then testing on a different subset containing different constraints. Specifically, we used our approach to train a model on the three demonstrations corresponding to the surface tilted by 0°, +16° and +27° about the x-axis (Fig. 13, left). We then took the demonstrated movements for the surface tilted at −16° and −27° about the x-axis (Fig. 13, right) as our test set and compared the movement reproduction.

In Fig. 13 we show the demonstrated (grey) and reproduced (black) trajectories for the training data constraints (left) and the test data constraints (right). Though we train on a smaller data set here, the policy learnt by our approach again produces a stable wiping movement that reproduces the human movement well, both under the training data constraints and under the unseen test constraints.

[14]The behaviour produced by the two methods can be examined in detail in the second video accompanying this paper.

5 Discussion

In this work, we introduced a novel approach to direct policy learning in cases where demonstrated movements are subject to variable, dynamic, non-linear constraints. Due to a small but very effective modification in the calculation of an empirical risk, our method can recover the unconstrained policy from arbitrarily constrained observations, without the need for explicit knowledge of the constraints. This allows us to learn policies that generalise over constraints, including novel constraints, unseen in the training data. We demonstrated our method using parametric (RBF) and locally linear function approximators to learn policies for problems of varying size and complexity.

Our results clearly show the efficacy of learning from constrained demonstrations using our approach, and then applying the resultant policies to new constraint scenarios. However, in terms of lessons learnt from these experiments there are also some bigger issues raised. One such issue is the question of when, faced with a new constraint, the learnt policy will fail at the desired task. For example, in the ball grasping experiment, under certain configurations of the constraints (e.g. if the barriers were placed exactly on either side of the ball, or a much larger barrier was placed between the robot and the ball) the learnt policy would fail at the task of grasping. This may be due to several factors, for instance if the control vector happens to be orthogonal to the nullspace of the constraint, deadlock would occur (this is similar to the problem of local minima in many gradient-based controllers, e.g. see Conner et al. 2003). While problems such as these are in general unavoidable when dealing with constrained systems, one of the nice properties of our approach is that it learns a policy that is successful *under the same constraints that the demonstrator is successful*. So, although the learnt policy for the grasping task is not guaranteed to successfully get the ball in the presence of any arbitrary barrier (constraint), it successfully reaches the ball whenever (i.e. with whatever barriers) the demonstrator does. In some sense we can say the *robustness* of the demonstrator's policy against different constraints was transferred to the learner.

A second, related issue concerns the role of adaptation of policies in response to constraints. Clearly there are circum-

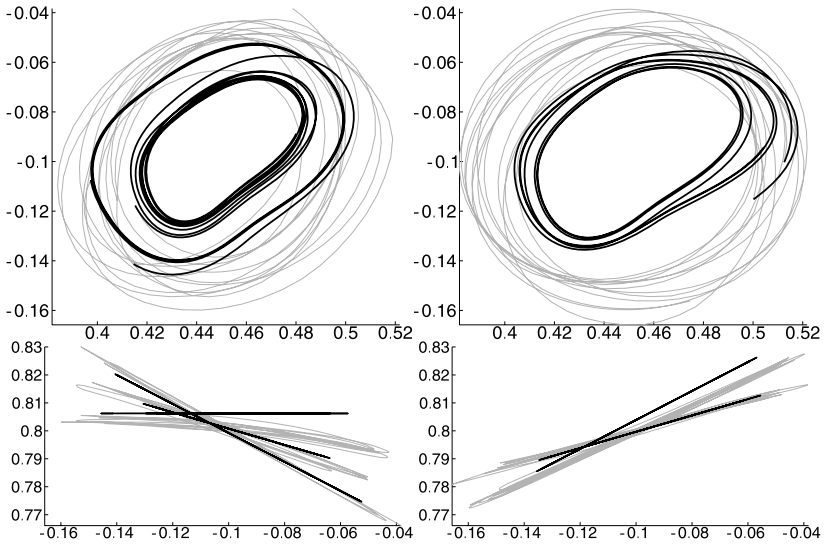


Fig. 13 Generalisation over constraints when learning from human wiping data. *Left*: Three demonstrated trajectories with surface tilt 0°, +16° and +27° (*grey lines*) used to train the model. *Right*: Two trajectories with tilt −16° and −27° (*grey lines*) held out for testing. Reproduced trajectories from the learnt policy under the corresponding constraints (both train and test) are overlaid in black

stances in which it is desirable to re-plan the policy to cope with certain sets of constraints, especially if the learner's existing policy (here, learnt from demonstration) fails under those constraints (and in some cases the learner may even take advantage of certain types of constraint to improve performance). However, here a balance must be struck. On the one hand re-planning the policy will likely improve performance under any given set of constraints; but on the other hand the adapted policy will also become more specialised to that particular set of constraints (and may even lead to degraded performance for other constraints). In other words we lose the *generalisation to other constraints* that here we attempt to extract from the demonstrator. Furthermore, due to the inherent uncertainty in the constraints in most real world problems, it may not be feasible to explicitly incorporate all of the constraints when re-planning. For example consider planning a policy for walking on uneven terrain; to explicitly incorporate the constraints involved here would require a detailed model of the terrain, which is rarely available. The proposed approach, however, allows us to sidestep this, providing a shortcut to uncovering the policy used by the demonstrator[15] (who, if observed to use the same policy under a number of constraint settings, presumably finds it sufficient successful for those and similar settings). Therefore in this sense, we envisage a move away from the traditional approach of planning explicitly with respect to all possible constraints that is typically only possible in highly controlled, structured environments.

In future work we intend to continue our analysis of learning from variable constraint data. An interesting direction would be to test the approach for learning policies that incorporate forces as well as positions and velocities. For example in the car wash experiment, one might consider using kinesthetic demonstrations to generate data including, for example, the normal force to the surface. One might also use such data to model the forces that constrain the policy (as distinct from the forces applied by the policy to generate the movement), and thus potentially lead to automated methods for explicitly decomposing observations into the policy actions and the (environmental or self-imposed) constraints affecting those actions.

[15] It should also be noted that our approach may also be combined with adaptive methods, for example using the policy learnt from demonstration to initialise further optimisation of the policy (e.g. through reinforcement learning, Peters and Schaal 2008b; Riedmiller et al. 2009), similar to, e.g., Guenter et al. (2007).

References

Alissandrakis, A., Nehaniv, C., & Dautenhahn, K. (2007). Correspondence mapping induced state and action metrics for robotic imitation. *IEEE Transactions on Systems, Man and Cybernetics*, *37*(2), 299–307.

Antonelli, G., Arrichiello, F., & Chiaverini, S. (2005). The null-space-based behavioral control for soccer-playing mobile robots. In *IEEE int. conf. advanced intelligent mechatronics*, 2005.

Atkeson, C., & Schaal, S. (1997). Robot learning from demonstration. In *Int. conf. machine learning*, 1997.

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2007). Robot programming by demonstration. In *Handbook of robotics*. Cambridge: MIT Press.

Bolder, B., Dunn, M., Gienger, M., Janssen, H., Sugiura, H., & Goerick, C. (2007). Visually guided whole body interaction. In *IEEE int. conf. robotics and automation*, 2007.

Calinon, S., & Billard, A. (2007). Learning of gestures by imitation in a humanoid robot. In *Imitation and social learning in robots, humans & animals: behavioural, social & communicative dimensions*, 2007.

Chalodhorn, R., Grimes, D. B., Maganis, G. Y., Rao, R. P., & Asada, M. (2006). Learning humanoid motion dynamics through sensory-motor mapping in reduced dimensional space. In *IEEE int. conf. robotics and automation*, 2006.

Chaumette, F., & Marchand, A. (2001). A redundancy-based iterative approach for avoiding joint limits: application to visual servoing. *IEEE Transactions on Robotics and Automation*, *17*, 719–730.

Choi, S., & Kim, B. (2000). Obstacle avoidance control for redundant manipulators using collidability measure. *Robotica*, *18*, 143–151.

Conner, D., Rizzi, A., & Choset, H. (2003). Composition of local potential functions for global robot control and navigation. In *IEEE int. conf. intelligent robots and systems*, 2003.

D'Souza, A., Vijayakumar, S., & Schaal, S. (2001). Learning inverse kinematics. In *IEEE int. conf. intelligent robots and systems*, 2001.

Gienger, M., Janssen, H., & Goerick, C. (2005). Task-oriented whole body motion for humanoid robots. In *IEEE int. conf. humanoid robots*, 2005.

Grimes, D., Chalodhorn, R., & Rao, R. (2006). Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Robotics: science and systems*, 2006.

Grimes, D., Rashid, D., & Rao, R. (2007). Learning nonparametric models for probabilistic imitation. In: *Adv. neural information processing systems*, 2007.

Guenter, F., Hersch, M., Calinon, S., & Billard, A. (2007). Reinforcement learning for imitating constrained reaching movements. *RSJ Advanced Robotics*, *21*, 1521–1544 Special Issue on Imitative Robots.

Howard, M., & Vijayakumar, S. (2007). Reconstructing null-space policies subject to dynamic task constraints in redundant manipulators. In *W.S. robotics and mathematics*, 2007.

Howard, M., Klanke, S., Gienger, M., Goerick, C., & Vijayakumar, S. (2008). Learning potential-based policies from constrained motion. In: *IEEE int. conf. on humanoid robots*, 2008.

Ijspeert, A., Nakanishi, J., & Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE int. conf. robotics and automation*, 2002.

Ijspeert, A., Nakanishi, J., & Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *Adv. neural information processing systems*, 2003.

Inamura, T., Toshima, I., Tanie, H., & Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, *23*, 363–377.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., & Hirukawa, H. (2003). Resolved momentum control: humanoid motion planning based on the linear and angular momentum. In *IEEE int. conf. intelligent robots and systems*, 2003.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *IEEE int. conf. robotics and automation*, 1985.

Khatib, O. (1987). A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, *RA-3*, 43–53.

Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man and Cybernetics*, *7*, 868–871.

Martinez-Cantin, R., de Freitas, N., Castellanos, J. A., & Docet, A. (2009). A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, *27* (this issue).

Mattikalli, R., & Khosla, P. (1992). Motion constraints from contact geometry: representation and analysis. In *IEEE int. conf. robotics and automation*, 1992.

Murray, R., Li, Z., & Sastry, S. (1994). *A mathematical introduction to robotic manipulation*. Boca Raton: CRC Press.

Mussa-Ivaldi, F. (1997). Nonlinear force fields: A distributed system of control primitives for representing and learning movements. In *IEEE int. sympos. computational intelligence in robotics and automation*, 1997.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., & Kawato, M. (2004). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, *47*, 79–91.

Ohta, K., Svinin, M., Luo, Z., Hosoe, S., & Laboissiere, R. (2004). Optimal trajectory formation of constrained human arm reaching movements. *Biological Cybernetics*, *91*, 23–36.

Park, J., & Khatib, O. (2006). Contact consistent control framework for humanoid robots. In *IEEE int. conf. robotics and automation*, 2006.

Peters, J., & Schaal, S. (2008a). Learning to control in operational space. *International Journal of Robotics Research*, *27*, 197–212.

Peters, J., & Schaal, S. (2008b). Natural actor-critic. *Neurocomputing*, *71*(7–9), 1180–1190.

Peters, J., Mistry, M., Udwadia, F., Nakanishi, J., & Schaal, S. (2008). A unifying framework for robot control with redundant DOFs. *Autonomous Robots Journal*, *24*, 1–12.

Ratliff, N. D., Silver, D., & Bagnell, J. A. (2009). Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, *27*(1), 25–53.

Riedmiller, M., Gabel, T., Hafner, R., & Lange, S. (2009). Reinforcement learning for robot soccer. *Autonomous Robots*, *27*(1), 55–73.

Sapio, V. D., Warren, J., Khatib, O., & Delp, S. (2005). Simulating the task-level control of human motion: A methodology and framework for implementation. *The Visual Computer*, *21*(5), 289–302.

Sapio, V. D., Khatib, O., & Delp, S. (2006). Task-level approaches for the control of constrained multibody systems. *Multibody System Dynamics*, *16*, 73–102.

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

Schaal, S., Ijspeert, A., & Billard, A. (2003). Computational approaches to motor learning by imitation. *Actions of the Royal Society B: Biological Sciences*, *358*, 537–547.

Sentis, L., & Khatib, O. (2004). Task-oriented control of humanoid robots through prioritization. In *IEEE int. conf. on humanoid robots*, 2004.

Sentis, L., & Khatib, O. (2005). Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, *2*, 505–518.

Sentis, L., & Khatib, O. (2006). A whole-body control framework for humanoids operating in human environments. In *IEEE int. conf. robotics and automation*, 2006.

Stolle, M., & Atkeson, C. (2009). Finding and transferring policies using stored behaviors. *Autonomous Robots*, *27* (this issue).

Sugiura, H., Gienger, M., Janssen, H., & Goerick, C. (2007). Real-time collision avoidance with whole body motion control for humanoid robots. In *IEEE int. conf. intelligent robots and systems*, 2007.

Svinin, M., Odashima, T., Ohno, S., Luo, Z., & Hosoe, S. (2005). An analysis of reaching movements in manipulation of constrained dynamic objects. In *IEEE int. conf. intelligent robots and systems*, 2005.

Takano, W., Yamane, K., Sugihara, T., Yamamoto, K., & Nakamura, Y. (2006). Primitive communication based on motion recognition and generation with hierarchical mimesis model. In *IEEE int. conf. robotics and automation*, 2006.

Udwadia, F., & Kalaba, R. (1996). *Analytical dynamics: a new approach*. Cambridge: Cambridge University Press.

Vlassis, N., Toussaint, M., Kontes, G., & Piperidis, S. (2009). Learning model-free robot control using a Monte Carlo em algorithm. *Autonomous Robots*, *27* (this issue).

Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotics Research*, *4*, 3–9.

Matthew Howard is currently a Ph.D. student in robotics and machine learning at the Institute for Perception, Action and Behaviour, a part of the School of Informatics at Edinburgh University. He received his M.Sc. in Artificial Intelligence from Edinburgh University in 2005, and his M.Sci. in Physics from Imperial College, London in 2004. He works in collaboration with the Honda Research Institute Europe, in Offenbach, Germany, and his research interests span machine learning, robotics and control.

Stefan Klanke received a Diploma in computer science from the University of Bielefeld in 2003, where he then became a member of the graduate programme "Strategies and Optimization of Behavior" before receiving his Ph.D. degree in 2007. He is currently a post-doctoral research fellow at the University of Edinburgh. His broad research interests are centered around machine learning technology and its application to robotics.

Michael Gienger studied mechanical engineering and graduated as Diplom-Ingenieur in 1998 at the Technical University of Munich, Germany. From 1998 to 2003, he was research assistant at the Institute of Applied Mechanics of the TUM where he received his Ph.D. in 2003. Within the research topic "design and realization of a humanoid biped robot," he gained experience in mechanical design, sensors and actuators, computer hardware as well as system theory. Currently Michael Gienger is a Senior Scientist at the Honda Research Institute Europe, Germany. His research interests include mechatronics, robotics, control and cognitive systems.

Christian Goerick studied electrical engineering at the Ruhr-Universität Bochum, Bochum, Germany, and at the Purdue University, West Lafayette, IN, USA. He received a Dipl.Ing. degree in electrical engineering and a Ph.D. degree in electrical engineering and information processing from the Ruhr-Universität Bochum in 1993 and 1998. From 1993 to 2000 he was with the Institute for Neural Computation, Ruhr-Universität Bochum, where he became a Research Assistant, Doctoral Worker, and Project Leader in 1993, and a Post-Doctoral Worker, Project Leader, and Lecturer in 1998. The research was concerned with biologically motivated computer vision for autonomous systems and learning theory of neural networks. Since 2000 he is a Chief Scientist at the Honda Research Institute Europe, Offenbach, Germany. His special interest is in embodied brain-like intelligence covering research on behavior based vision, audition, behavior generation, cognitive robotics, car assistant systems, systems architecture as well as hard- and software environments.

Sethu Vijayakumar is the Director of the Institute of Perception, Action and Behavior in the School of Informatics at the University of Edinburgh. Since 2007, he holds a fellowship of the Royal Academy of Engineering in Learning Robotics, co-sponsored by Microsoft Research Cambridge. He also holds additional appointments as an Adjunct Faculty of the University of Southern California, Los Angeles, a Research Scientist of the ATR Computational Neuroscience Labs, Kyoto-Japan and a Visiting Research Scientist at the RIKEN Brain Science Institute, Tokyo. He has a Ph.D. ('98) in Computer Science and Engineering from the Tokyo Institute of Technology. His research interests span a broad interdisciplinary curriculum ranging from statistical machine learning, robotics, planning and optimization in autonomous systems to motor control and computational neuroscience.