

Dexterous Reaching, Grasp Transfer and Planning Using Electrostatic Representations

Peter Sandilands[†], Vladimir Ivan[†], Taku Komura[†], Sethu Vijayakumar[†]

Abstract—We propose a novel approach to transfer reach and grasp movements while being agnostic and invariant to finger kinematics, hand configurations and relative changes in object dimensions. We exploit a novel representation based on electrostatics to parametrise the salient aspects of the demonstrated grasp. By working in this alternate space that focuses on the relational aspects of the grasp rather than absolute kinematics, we are able to use inference based planning techniques to couple the motion in abstract spaces with trajectories in the configuration space of the robot. We demonstrate that our method computes stable grasps that generalise over objects of different shapes and robots of dissimilar kinematics while retaining the qualitative grasp type – all without expensive collision detection or re-optimisation.

I. INTRODUCTION

Movements that involve reaching and grasping are difficult to synthesize, due to collisions between the hand model and its environment, the self-collisions, and the complex shape of the open space, especially when concave objects are involved. Synthesis of such movements is expensive because collision detection and global path-planning are required. Furthermore, the movement is no longer valid once the geometry of the hand or the shape of the object to be grasped change.

A similar problem happens in applications such as tele-operation [10] or learning from demonstration [12], where the aim is to compute the mapping between the human hand and the robotic hand that have different geometrical or even topological structures. This mapping is usually non-linear and it is computed either using machine learning techniques [3] or it is defined manually by exploiting the task definition using a proxy object [5] or fingertip position correspondence [18]. These ways of mapping the motion require either large amount of sample data (in the former case) or they restrict the mapping in terms of complexity of the motion or dissimilarity of the kinematics of the two systems (in the latter case).

In this paper, we propose to solve these problems by using a novel representation that borrows the concepts from electrostatics [26] that looks to define the spatial relationship between the hand and the object, as well as the envelopment of the hand around the object. Firstly, by virtually charging the object, we compute an object-centric coordinate system that we call electric coordinates. The electric coordinates can canonically parametrise the open space around objects.

[†]Institute of Perception Action and Behaviour, University of Edinburgh, United Kingdom.

*Address correspondence to Peter Sandilands at the address above or his e-mail: Peter.Sandilands@ed.ac.uk

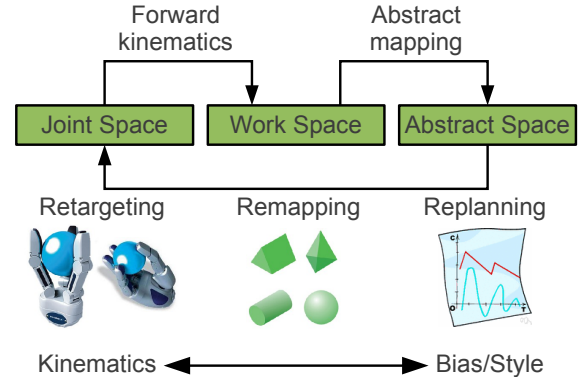


Fig. 1. Structure of motion planning problem using multiple representations: joint space (defined by kinematics), work space (defined by kinematics and grasping style) and abstract spaces (augmented by grasping style). Changes of kinematics result in re-targeting, changes of the shape of the object result in remapping and changing the abstract state are performed by replanning.

The movements of the hand model defined in the electric coordinates can be canonically parametrised even when they involve extensive close interaction between the hand model and the object. Also, due to the harmonic nature of the electric coordinates, the hand model can be easily guided towards the object without suffering from local minima. Secondly, based on Gauss's law, we quantify how much the object is surrounded by the hand model using the electric flux. This allows us to successfully guide the hand to grasp the object using the electric flux as a control parameter.

Using this representation, we can significantly reduce the complexity of motion planning and transfer by offloading the computational effort onto the mapping between spaces, which can be easily done by the electrostatic parameters. The complexity of the space for planning gets lowered by abstracting the movements from that in the joint angle space to that in the electrostatic parameters. This mapping enables us to apply optimisation methods with local motion planning, where exploratory methods such as RRT or PMR would be necessary otherwise. The optimal trajectory is computed within the Approximate Inference Control framework (AICO) that allows combining the electrostatic representation with other representations such as the control effort in joint space. AICO has been used in recent publications for planning in topological spaces that have similarities with our electric coordinate system [11].

Contributions:

- Electric field based representations suitable for transferring reaching and grasping motion.

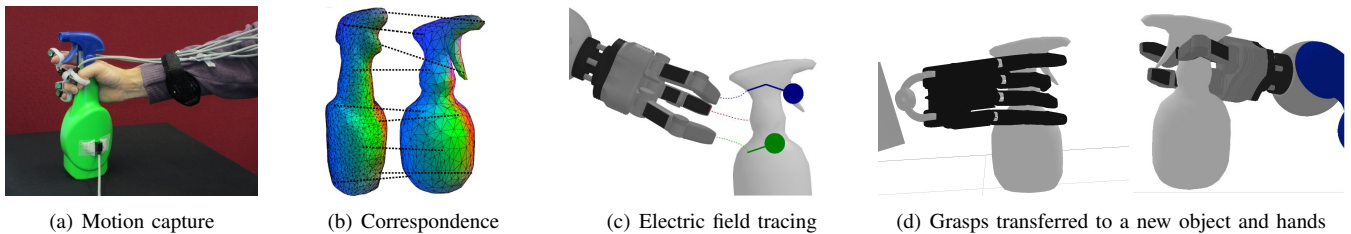


Fig. 2. Synthesizing a grasping motion from motion capture. (a) An interaction is captured of an actor grasping the object. (b) A dense mapping to a new object is produced using the electric field. (c) The interaction points on the hand projected to the object’s surface by following the electric field gradient. The solid line is the path on the surface that the projection must match for the rest of the sequence. (d) The re-targeted motion.

- Validation that the proposed representations are simple enough so that local optimisation methods such as AICO can be used.
- Experiments showing grasp transfer/teleoperation with different bias/style.

II. RELATED WORK

Our research is related to topics including (1) grasp synthesis, (2) grasp transfer, and (3) potential-field guided path planning.

A. Grasp Synthesis

Grasping is an active area of research within robotics and many methods for achieving stable grasps under various constraints have been proposed in the literature. The authors of [16] propose replacing the object with a combination of grasping primitives such as spheres and cylinders. Stable grasps for these primitives are precomputed which allows for a fast search for a feasible and stable grasp. In [6] a method using decomposition trees is used to prune the intractably large space of possible grasps into a subspace that is likely to contain many good grasps. In [19], the authors propose a method for learning a manifold with high grasp quality in the space of the robot’s and object’s DOFs by using SVM. These methods often consider only the grasping problem and but they ignore the reaching problem. Additionally, the planned grasps do not generalise over robots with dissimilar kinematics. The authors of [3] propose to use dimensionality reduction techniques to compute so called eigengrasps. In [4] an example-based system is described for planning stable grasps that are functionally suitable for certain tasks. These techniques allow for a certain amount of generalisation, but the reaching motion still has to be planned separately.

B. Grasp Transfer

The areas of teleoperation and learning from demonstration address the the issue of grasping from the point of view of mapping the demonstrated grasp onto a robotic system with dissimilar kinematics. There are three distinct approaches in the literature: 1) Joint-to-joint mapping is taking advantage of similarity of human grasp poses and aims to produce grasps of similar quality directly from demonstrated joint angles. Dimensionality reduction methods have been used to effectively map human demonstrated poses onto different robotic manipulators [3]. 2) Cartesian space

mapping has been used to map fingertip positions [18]. These methods focus on preserving geometric relations between the two spaces and allow to transfer precision grasps. The mapping between hands with different number of fingers is, however, not well defined, and these methods do not generalise over different shapes of objects. A recent extension of this approach that maps contact positions of a grasp to similar objects is presented in [9]. This technique warps the surface geometry of a source object to a target object along with the contact points of a grasp, but does not deal with the non-contact relationships between the hand and the object, nor does it consider the approach for the grasp. 3) The pose mapping is an indirect joint space mapping technique. In [17] the authors propose a method based on functional analysis of the human hand and results in an algebraic transformation of human hand configuration into target domain configurations. These three approaches have been later combined [12]. The work presented in [5] shows how the pose mapping problem can be solved using a proxy object by representing the grasp by a minimal ellipsoid containing the fingertips. The inverse kinematics is then computed with respect to the shape of ellipsoid defined by the demonstrated hand pose. Unlike our method, this ellipsoid representation limits the motions that can be generated via this approach.

C. Potential-Field Guided Path Planning

In [13] authors propose to compute a virtual potential field using local areas of attraction and repulsion to guide the robot towards the goal while avoiding obstacles. The authors of [8] applied this idea to perform grid search within the robot’s configuration space with potential field to find stable grasps. In [2] dynamic potential fields have been used to selectively explore regions of interest when planning grasping of unknown objects. [23] use potential fields in controlling decentralised point-based robots in the transport of simple convex objects. Unlike our method, none of these potential-based methods have non-intersecting field lines when dealing with complex geometry.

Our work also falls into the category of potential-field guided methods. We compute an electric field and potential using charge simulation, which can be applied for guiding the hand model around the object. We make use of the field not only for path planning but also for generalization of the grasping behaviour, which is useful for transferring the grasping motion.

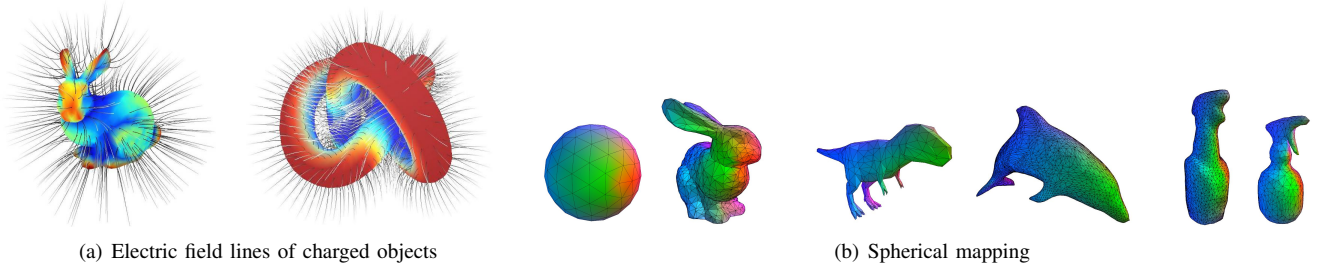


Fig. 3. An example of charged objects. (a) The charge is mapped from blue to red, representing low charge and high charge respectively. (b) Projection of 2D spherical coordinates computed using electric field showing correspondence between different objects.

III. OVERVIEW

We consider the grasp transfer as the problem of transferring the relative configuration of the manipulator and the object. We define the combined hand and arm model as a kinematic system with N number of controllable joints \mathbf{q} where the positions of M number of points $\vec{p}_i(\mathbf{q})$ attached to a link connected to the system by joint j is defined by the forward kinematics function $f_j(\mathbf{q})$:

$$\vec{p}_i(\mathbf{q}) = f_j(\mathbf{q}), \mathbf{q} \in \mathbb{R}^M. \quad (1)$$

We use the points $\vec{p}_i(\mathbf{q})$ to either approximate the desired contact surface of the hand (via W number of triangles, see Fig. 7(a)), or to represent the fingertip positions (see Fig. 2(c)). We then use triangulation to approximate the shape of the object using K number of points \vec{p}_k and L number of triangles (see Fig. 2(b)). Our goal is to abstract the relationship between points $\vec{p}_i(\mathbf{q})$ and \vec{p}_k so that the joint configurations \mathbf{q} produce consistent, stable and semantically similar reaching and grasping motion when the geometry of the hand model changes (changing the forward kinematics function $f_j(\mathbf{q})$) or when the shape of the object changes (changing \vec{p}_k).

We now abstract the hand-object interaction using properties of a virtual electrostatic field defined using vertices of the object mesh \vec{p}_k . Once the motion is defined in the abstract representation, it can be easily transferred to different hand models and objects. The semantic similarity is encoded within the representation through fingertip position in the electrostatic field (see Section IV-A) and through hand coverage (see Section IV-C). The trajectory for new hand model is computed using the stochastic optimization framework AICO, such that the hand model follows the desired trajectories of the electrostatic parameters as much as possible, while avoiding collisions and satisfying physical constraints.

IV. SPACE PARAMETRISATION BY ELECTROSTATICS

We briefly review the electrostatic parametrisation proposed by Wang et al. [26], which computes an object-centric curvilinear coordinate system that resembles polar coordinates. Such a coordinate system is useful for defining reaching and grasping motion with respect to the object as it gives a relative spatial relationship between a point and a charged object.

A. Computing the Electric Coordinates

In order to compute this coordinate system we virtually charge the object as a conductor and then parametrise the outer space of the object using the simulated electric field and potential. Using the principle of superposition and the fact that the potential around a charged object is proportional to the charge, we can construct a dense linear system of L equations, each representing the notion that the potential must be 1 volt at some probe point on the surface of the object, with L number of variables denoting the unknown charges Q_l for each of the elements:

$$\begin{cases} V_1(\vec{x}_1)Q_1 + V_2(\vec{x}_1)Q_2 + \dots + V_n(\vec{x}_1)Q_L = 1 \\ \dots \\ V_1(\vec{x}_L)Q_1 + V_2(\vec{x}_L)Q_2 + \dots + V_n(\vec{x}_L)Q_L = 1, \end{cases} \quad (2)$$

where $V_l(\vec{x}_m)Q_l$ denotes the potential V_l at point \vec{x}_m due to the l -th triangle carrying charge Q_l in the analytical form [7]. For the probe points \vec{x}_m we select the barycentres of the mesh triangles (even though any L different points inside or on the surface of the object would do). Equation 2 defines a dense linear system that can be written down in matrix notation as $\mathbf{P}\mathbf{Q} = \mathbf{1}$. This system is typically ill-conditioned, and therefore we solve it using pseudo-inverse of \mathbf{P} in the least squares sense.

Fig. 3(a) visualizes the distribution of charges over the surface of different objects and the resulting electric fields around them by plotting field lines $\vec{E} = -\nabla\varphi_p$ where the potential φ_p is a function of surface charges, object and the query point. In our system we calculate the potential of each point $\vec{p}_i(\mathbf{q})$ attached to the hand (related to the joint angles \mathbf{q} through Equation 1). We compute these from L uniformly charged triangles defined by points \vec{p}_k with surface charges P

$$\phi_p(\mathbf{q}) = f_p(\vec{p}_i(\mathbf{q}), P, \vec{p}_k) \quad (3)$$

as described in [7].

After the charge simulation, the resulting electric field can be used to parametrise the space. Indeed, the field lines emanating from each point on the surface never intersect (or else the conservation of energy would be violated), and since the potential harmonically decreases with distance from the object, each point along a particular field line has a unique potential (see Fig. 4 (a)). By following the field lines from the surface outwards, we can map all the points on

the object to an infinitely large sphere outside, giving us a 2 dimensional parametrisation of the surface of the object. Here we parametrise the 3D space by mapping the spherical coordinates of the infinitely large sphere to every location in the space by following the field lines (see Fig. 3(b)). We compute this parametrisation numerically by evaluating the path integral of the electrostatic field along the field line C that passes through a point \vec{x} via

$$f_{uv}(\vec{x}) = \int_C^\infty \vec{E}(\vec{x}) d\vec{x}, \text{ where } \vec{x} \in C. \quad (4)$$

$$\phi_{uv}(\mathbf{q}) = \begin{bmatrix} f_{uv}(\vec{p}_0(\mathbf{q})) \\ f_{uv}(\vec{p}_1(\mathbf{q})) \\ \vdots \\ f_{uv}(\vec{p}_M(\mathbf{q})) \end{bmatrix} \quad (5)$$

In practice the sphere is not infinite, as we follow the lines only to a low potential, ensuring the endpoints are far from the object. We use the azimuth u and elevation v (from the endpoints of projection $f_{uv}(\vec{x})$) as the first two parameters in our parametrisation as ϕ_{uv} , and the electric potential ϕ_p as the final coordinate. Following Wang et al., we call this parametrisation *electric coordinates*. Additionally, we also compute the inverse map from electric coordinates $\phi_p(\mathbf{q})$ and $\phi_{uv}(\mathbf{q})$ to the position on the surface \vec{s}_i of the object by following the field lines inwards by numerically evaluating the integral $\vec{s}_i(\mathbf{q}) = \int_C -\vec{E}(\vec{x}) d\vec{x}$ (see Fig. 5). This gives us a projection onto the surface of the object for any point in the outer space of the object. The correspondence between different objects is then defined through the mapping onto the sphere. This is a bijective mapping defined for all points on the surface of the object when there are no saddle points. See Fig. 3(b) for examples. This mapping exists even for object of dissimilar shapes and arbitrary topology due to the non-intersection property of the field lines. The projection onto the surface of the object also allows us to compute force closure directly from surface points projected using electric coordinates

$$\phi_{fc}(\mathbf{q}) = f_{fc}(\vec{s}_{1..M}(\mathbf{q})) \quad (6)$$

Function f_{fc} is the force closure measure defined in [15]. This assumes that the projected points will become contact points when distance to surface of the object decreases, which is valid when close to the object. Although these projections can be computed at run-time, we pre-compute the electric coordinates at the vertices of a 3D grid structure surrounding the object in order to decrease computational cost. We look up the electric coordinates for points $\vec{p}_i(\mathbf{q})$ in the 3D grid at runtime.

The electric field computational complexity is $O(L)$ (where L is the number of triangles comprising the object), as defined by the superposition principle and discussed in [26]. This lookup grid reduces the computational complexity of the potential calculation for a single point from $O(L)$ to an $O(1)$ lookup and trilinear interpolation. It also reduces the complexity of computing the corresponding point on the surface of the object given a point in space, as this is

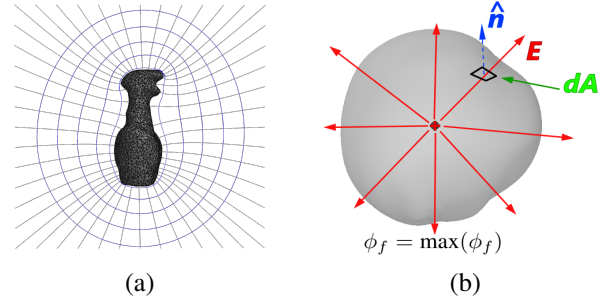


Fig. 4. (a) The grid (field lines and equipotentials) of our curvilinear electric coordinate system. (b) Illustration of Gauss's law for a charged point (red) entirely surrounded an arbitrary surface (grey).

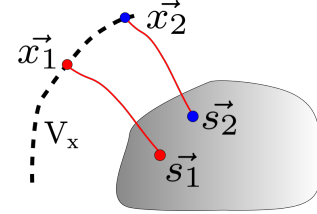


Fig. 5. An illustration of the projection of points \vec{x}_i onto the surface of an object using the electric coordinates. V_x is the isosurface (surface at equal potential, dashed line) of the electric field that \vec{x}_i lies on, \vec{s}_i is the point on the surface of the object corresponding to \vec{x}_i obtained by increasing the potential along the electric field line that intersects \vec{x}_i (shown as a solid red line).

performed by using the Euler method of integration along the field line until the surface is reached. This means that ϕ_{uv} and ϕ_p are calculated during runtime in $O(M)$ time, where M is the number of points attached to the hand as defined in Section III.

B. Combining The Electric Field with the Interior Distance Field

The calculations presented so far as an approximation of the electric field are only valid for the outer space of the object. Although theoretically the field is a consistent 1 volt both on and inside the surface of the object, sampling our approximation of this field inside the object leads to values less than the surface value of 1 volt. This not only causes a local maxima at the surface of the object but means that the local potential value does not define whether we are ‘inside’ the object or not.

To alleviate this problem we post-process our computed voxel grid to store the values of a distance field as the potential for interior points, which computes $(1 + \|\vec{x} - \vec{b}\|)^3$ where \vec{x} is the point in question and \vec{b} is the closest surface point. In this way the field increases in the direction of the medial axis of the object and decreases in the direction of the nearest surface. A point that finds itself inside the object would be able to escape the object by moving along this field towards a potential value of 1, bringing it to the closest point on the surface of the object.

In practise we compute this by using a physics engine to emit a ray in an arbitrary direction from the point and return

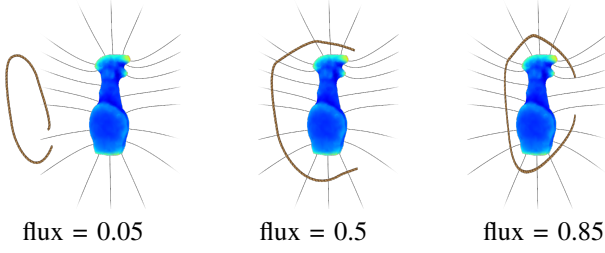


Fig. 6. A 2 dimensional illustration of different configurations of a deformable object and a charged reference object labelled with the corresponding flux value of the deformable object.

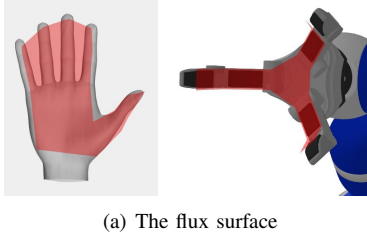


Fig. 7. Here we show the flux surface as specified on the original hand model and the Schunk robotic hand model. The coverage of the object by these hands is reported using the flux surface.

the first point on the object we hit. If no object is hit, we can safely ignore the point as it is outside of the object (with the assumption that the object is closed). If the object is hit, we compute the dot product between the ray and the normal vector for the triangle at the point we hit. If it is positive (the ray has hit on the inside of the triangle), we then recalculate the potential of this point using the interior distance field calculation as previously specified.

C. Using Electrostatics to Compute Coverage

This electrostatic simulation on the object can also be used to evaluate the amount the object is surrounded by the hand model using a physical property called flux, which is the surface integral of an electric field through a surface. In terms of a representation for planning this is interesting as it encodes the relative ‘envelopment’ of a surface around an electrically charged object.

According to Gauss’ law, a closed surface surrounding a charged object will always have a constant flux value, no matter the relative transformation or deformation of the outer surface:

$$\oint_S \vec{E} \cdot d\vec{A} = \oint_S \vec{E} \cdot \vec{n} dA = \frac{Q}{\epsilon_0} = \text{const}, \quad (7)$$

where \vec{E} is the electric field being integrated over the surface S surrounding the charged object (with charge Q). $d\vec{A}$ is a small region of S (a vector with magnitude dA pointing in the normal direction \vec{n}), and ϵ_0 is the electric constant (see Fig. 4(b)). This feature makes the flux suitable for representing the coverage of the hand around the object.

Examples of different configurations in which a deformable object is surrounding the reference object are shown together with the flux value in Fig. 6. It can be

observed that the more the object is surrounded, the larger the flux is. For polygonal meshes, the flux can be computed by summing those of all the triangles in the mesh, for which an analytical expression exists [25]. We compute flux for our system by defining a proxy mesh of W triangles that covers the contact surface of the hand (as mentioned in Section III, defined by $\vec{p}_i(q)$ and shown in Fig. 7(a)), then summing the flux across them caused by the individual triangles making up the object (defined by \vec{p}_k):

$$\phi_f(q) = \sum_{i=1}^W \sum_{j=1}^L f_{\text{flux}}(t_i^{\text{surf}}, t_j^{\text{obj}}) \quad (8)$$

where t_i^{surf} is the i th triangle in the proxy surface on the hand, t_j^{obj} is the j th triangle defined by the object. W, L are the number of hand surface triangles and the number of object triangles respectively. The flux function between a charged (object) and an uncharged (hand) triangle, f_{flux} , is described in [25] and further explained in [26] (Appendix A.3).

In summary, the advantages of this Electrostatics representation when there are no saddle points are: a continuous mapping from the object’s surface to the outer space, non-intersecting field lines, a definition of a coverage measure, and a continuous mapping between objects. The non-intersecting property of the field lines is a well known homology invariant that has been studied in the fields of physics and topology [1].

V. MOTION PLANNING

Given the abstract representations introduced in Section IV one would wish to compute the trajectory in the joint space $q_{0:T}$ and the controls $u_{0:T}$ to execute this trajectory to perform a reaching and grasping motion using a robotic manipulator. We will now introduce the Approximate Inference Control (AICO)[20] to perform planning in the above defined abstract space. AICO frames the problem of optimal control as a problem of inference in a dynamic Bayesian network. Let x_t be the state of the system—we will always consider the dynamic case where $x_t = (q_t, \dot{q}_t)$. Consider the problem of minimizing (the expectation of) the cost

$$C(x_{0:T}, u_{0:T}) = \sum_{t=0}^T c_x(x_t) + c_u(u_t) \quad (9)$$

where c_u describes costs for the control and c_x describes task costs depending on the state (usually a quadratic error in some task space). We describe the robot dynamics by

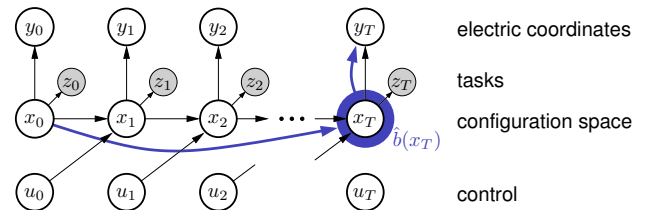


Fig. 9. AICO in configuration and electrostatic space. The grey arcs represent the approximation used in the end-state posterior estimation.

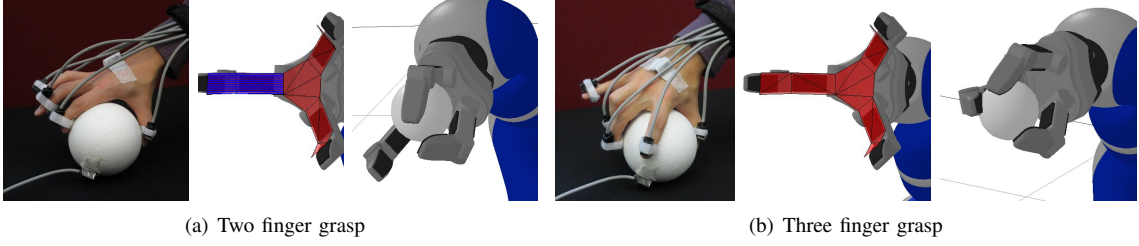


Fig. 8. Grasping objects using electric flux. A two finger grasp (a) and three finger grasp (b) achieved through weighting. An illustration of the weightings are shown on the flux surfaces here. Blue striped surfaces have lower weights. The motion is semantically similar and adapts gracefully to new relative positioning of the hand and the object.

the transition probabilities $P(x_{t+1} | u_t, x_t)$. The AICO framework translates this to the graphical model

$$p(x_{0:T}, u_{0:T}) \propto P(x_0) \prod_{t=0}^T P(u_t) \prod_{t=1}^T P(x_t | u_{t-1}, x_{t-1}) \cdot \prod_{t=0}^T \exp\{-c_x(x_t)\}. \quad (10)$$

The control prior $P(u_t)$ reflects the control costs, whereas the last term $\exp\{-c_x(x_t)\}$ reflects the task costs and can be interpreted as “conditioning on the tasks” in the following sense: We introduce an auxiliary random variable z_t with $P(z_t = 1 | x_t) \propto \exp\{-c_x(x_t)\}$, that is, $z = 1$ if the task costs $c_x(x_t)$ are low in time slice t . The above defined distribution is then the posterior $p(x_{0:T}, u_{0:T}) = P(x_{0:T}, u_{0:T} | z_{0:T} = 1)$.

A. Computing novel trajectories in electrostatic spaces

Fig. 9 displays a corresponding graphical model. The bottom layer corresponds to the standard AICO setup, with the motion prior $P(x_{t+1} | x_t) = \int_u P(x_t | u_{t-1}, x_{t-1}) P(u_t) du$ implied by the system dynamics and control costs. Additionally it includes the task costs represented by $P(z_t = 1 | x_t) = \exp\{-c_x(x_t)\}$. The top layer represents a process in arbitrary task space with a linear Gaussian motion prior $P(y_{t+1} | y_t)$. Both layers are coupled by introducing additional factors

$$f(x_t, y_t) = \exp\left\{-\frac{1}{2}\rho\|\phi(\mathbf{q}_t) - y_t\|^2\right\}, \quad (11)$$

which essentially aim to minimize the squared distance between the state y_t (e.g. in the electric coordinate space) and the one computed from the joint configuration $\phi(\mathbf{q}_t)$ (as defined in Equation 2 and for electric potential and flux respectively), weighted by a precision constant ρ . Note that using a local linearisation of ϕ (having the Jacobian of the electric coordinates) is sufficient. Here we make the assumption that trajectories in these task spaces can be effectively represented using multivariate Gaussian distributions. We justify this assumption by the fact the electric field is homogeneous and it contains no or only few local minima (for obstacles with genus larger than 1).

These factors treat the electric coordinates y_t as an additional task variable for the lower level inference. Additional

nodes can be added to the graphical model the same way as y_t to extend the model with another task variable, such as electric flux or collision avoidance. We then define the task cost as

$$c_x(x_t) = \rho_p \|\phi_p(\mathbf{q}_t) - y_p\|^2 + \rho_{uv} \|\phi_{uv}(\mathbf{q}_t) - y_{uv}\|^2 + \rho_f \|\phi_f(\mathbf{q}_t) - y_f\|^2 + \rho_{fc} \|\phi_{fc}(\mathbf{q}_t)\|^2 + \rho_c \|\phi_c(\mathbf{q}_t)\|^2 \quad (12)$$

where $\phi_p(\mathbf{q}_t)$, $\phi_{uv}(\mathbf{q}_t)$, $\phi_{fc}(\mathbf{q}_t)$ and $\phi_f(\mathbf{q}_t)$ are defined in equations 3, 5, 6 and 8 respectively, and $\phi_c(\mathbf{q}_t)$ is the collision cost computed as reciprocal distance to the closest obstacle. We use the chain rule to compute the Jacobian of electric flux and the collision measure and the finite differences method to approximate the Jacobian of the electric coordinates and force closure. Refer to [24] for further details about how to extend the graphical model and how to perform the inference using message passing algorithm. The control cost is implicitly added through the inference process.

VI. EXPERIMENTS

To validate our representation we performed experiments in motion transferral and planning using the electric coordinates. We used the KUKA LWR4 robotic arm in combination with the Schunk hand, Shadow hand and KCL Metamorphic hand [27] in our experiments to show transferral of human motion to robotic hands with different capabilities.

An important part of the transferral of motion is capturing the original scene accurately, as the transferral process relies heavily on the initial motion. In order to do this we use a magnetic motion-capture system called the Polhemus Liberty [14], a Microsoft Kinect, and the technique in [22] for capturing both the hand configuration and the object’s location for the duration of an interaction. This involves capturing the approximate object geometry using aligned pointcloud data from the Kinect, applying magnetic markers and computing transformations for each rigid body in the scene (see Fig. 2(a) and Fig. 2(b)), then using IK to reconstruct the hand motion with the magnetic marker transformations as targets for each finger. This results in an digitized scene with both geometry and motion data. See [21], [22] for further details.

A. Motion Transfer using Electric Coordinates

We now demonstrate grasp transfer between a human hand and a robotic hand. The object we use here is a spray bottle,

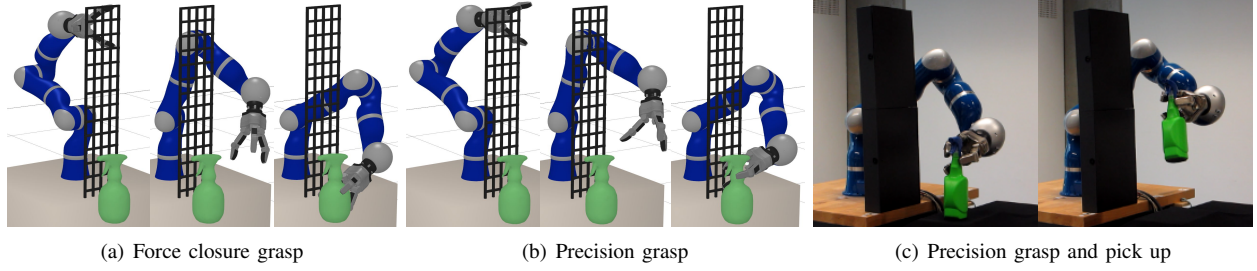


Fig. 12. Planning successful grasps while reaching around an obstacle. (a) Force-closure grasp maximising stability. (b) Precision grasp with style defined using electric coordinates. (c) Precision grasp implemented on hardware and picking up the spray bottle.

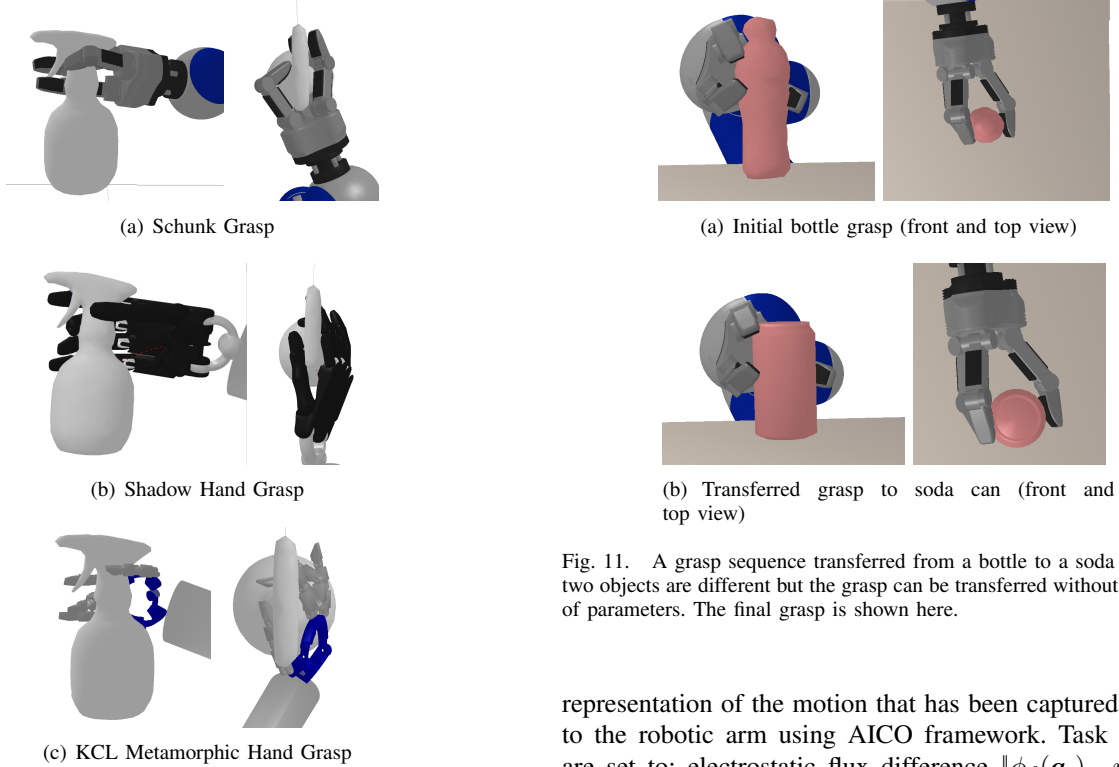


Fig. 11. A grasp sequence transferred from a bottle to a soda can. These two objects are different but the grasp can be transferred without any tuning of parameters. The final grasp is shown here.

Fig. 10. Transferred motion is applicable to different robot hand morphologies. The same electric coordinate sequence is applied to the Schunk hand (a), the Shadow hand (b), and the KCL Metamorphic hand (c). The final grasp is shown here.

which has a standard way of holding (the fingers being wrapped around the bottle neck and one placed finger on the trigger) which we intend to transfer. First, significant points are manually selected on the surface of the hand, and their electric coordinates are recorded during playback of the motion. During our experiments, we found that selecting the finger tips, interphalangeal joints, and the wrist gave a good representation of the whole hand motion. This gives us a per-frame sequence of points in the space of electric coordinates. In addition to this, we also compute the total flux passing through a mesh defined over the surface of the hand, giving us a measure of coverage and overall orientation towards the object. A mapping between the fingers and the Schunk hand is given. This allows us to use the electric coordinate

representation of the motion that has been captured to apply to the robotic arm using AICO framework. Task variables are set to: electrostatic flux difference $\|\phi_f(\mathbf{q}_t) - y_f\|^2$ (the flux passing through the triangles in Fig. 7(a)), electrostatic potential difference $\|\phi_p(\mathbf{q}_t) - y_p\|^2$ and electrostatic UV projection difference $\|\phi_{uv}(\mathbf{q}_t) - y_{uv}\|^2$. We minimise these costs, and so minimize the difference in the electric coordinates between the original motion given by the human hand and the newly generated motion for the Schunk hand and the KUKA LWR4 robotic arm. To show that our method generalizes over robots of dissimilar kinematics we transfer the motion to Shadow Dexterous Hand, KCL Metamorphic Hand and Schunk Robot Arm (see Fig. 10). The resulting motion is semantically similar in terms of relative final grasp locations and approach to the object, which allows us to demonstrate kinematic tasks using the natural interface of our own body. We also performed simplified experiments on a sphere to show transfer of different grasp styles. Results are shown in Fig. 8(a) and Fig. 8(b). These grasps are valid when applied to the hardware.

We are also able to transfer the same grasp between objects. We do this in a similar manner to the transfer between manipulators, by mapping the input motion to the electric

coordinates and then applying the motion to a new object using the task variables as specified above. An example is shown in Fig. 11.

B. Motion Planning using Electric Coordinates

We now add an obstacle into the environment and perform motion planning in the electric coordinate space (see Fig. 12). We have chosen to use AICO to perform the necessary planning. We have set up the system with the following task variables: collision avoidance $\|\phi_c(\mathbf{q}_t)\|^2$ and force closure of electrostatic projection of the fingertip positions $\|\phi_{fc}(\mathbf{q}_t)\|^2$ (for a stable grasp). We have used the original transferred trajectory as initialisation. Since all of the task variables are well defined everywhere in the workspace, we were able to iterate AICO till convergence. Fig. 12 shows the result of planning a reaching and grasping for a scenario where naive motion transfer fails. The planner computed a stable generic grasp for the spray bottle (see Fig. 12(a)). We have then added a task variable using spherical coordinates similar to the previous experiment. In this case, however, we specify a target in spherical coordinates only at the last time step of the trajectory. This allows us to encode a bias (or a style) while the remainder of the trajectory relies predominantly on potential and force closure priors. Fig. 12(b) shows the resulting precision grasp around the neck of the spray bottle and Fig. 12(c) shows the robot executing this plan and subsequently picking up the spray bottle.

VII. CONCLUSIONS

We have exploited a novel representation for grasp transfer based on electrostatics to parametrise the salient aspects of the demonstrated grasp in terms of spatial relationship by using the flux as an envelopment measure and the electric coordinates of multiple points per link as the relative position and orientation. By working in this alternate space that focuses on the relational aspects of the grasp rather than absolute kinematics, we were able to use Approximate Inference Control, a local optimisation technique, to couple the motion in abstract spaces with trajectories in the configuration space of the robot. We have demonstrated that our method computes stable grasps that generalise over objects of different shapes and robots of dissimilar kinematics while visually retaining the qualitative grasp type. Although the pre-processing stage takes a significant amount of time to generate the field we have also presented optimisations that made it possible to use these representations for near real-time applications.

Acknowledgements: This work was funded by the EU FP7 project TOMSY and EPSRC project EP/H012338/1.

REFERENCES

- [1] S. Bhattacharya, D. Lipsky, R. Ghrist, and V. Kumar. Invariants for Homology Classes with Application to Optimal Search and Planning Problem in Robotics. *ArXiv e-prints*, 2012.
- [2] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. Grasp affordances from multi-fingered tactile exploration using dynamic potential fields. In *Proc. of Humanoids*, pages 168–174, 2009.
- [3] M. Ciocarlie, C. Goldfeder, and P. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *Proc. of IROS*, pages 3270–3275, 2007.
- [4] Hao Dang and P. K. Allen. Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *IROS*, pages 1311–1317, 2012.
- [5] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo. An Object-Based Approach to Map Human Hand Synergies onto Robotic Hands with Dissimilar Kinematics. In *Proc. of R:SS*, 2012.
- [6] C. Goldfeder, P.K. Allen, C. Lackner, and R. Pelossof. Grasp Planning via Decomposition Trees. In *Proc. of ICRA*, pages 4679–4684, 2007.
- [7] E. Goto, Y. Shi, and N. Yoshida. Extrapolated Surface Charge Method for Capacity Calculation of Polygons and Polyhedra. In *Journal of Computational Physics*, volume 100, pages 105–115, 1992.
- [8] M. Haex and L. M. Gambardella. Stable Grasps by Path Planning using Artificial Fields, 1992.
- [9] U. Hillenbrand and M.A. Roa. Transferring functional grasps through contact warping and local replanning. In *Intelligent Robots and Systems (IROS)*, 2012 *IEEE/RSJ International Conference on*, pages 2963–2970, 2012.
- [10] Haiying Hu, Jiawei Li, Zongwu Xie, Bin Wang, Hong Liu, and G. Hirzinger. A robot arm/hand teleoperation system with telepresence and shared control. In *Proc. of AIM*, pages 1312–1317, 2005.
- [11] Vladimir Ivan, Dmitry Zarubin, Marc Toussaint, Taku Komura, and Sethu Vijayakumar. Topology-based representations for motion planning and generalization in dynamic environments with interactions. *International Journal Of Robotics Research*, 32:1151 – 1163, 2013.
- [12] S. B. Kang and K. Ikeuchi. Robot Task Programming by Human Demonstration: Mapping Human Grasps to Manipulator Grasps. In *Proc. of IROS*, volume 1, pages 97–104, September 1994.
- [13] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.
- [14] J. Krieg. Motion tracking: Polhemus technology. *Virtual Reality Systems*, 1(1):32–36, 1993.
- [15] A.T. Miller and P.K. Allen. Examples of 3D grasp quality computations. In *Proc. of ICRA*, volume 2, pages 1240–1246, 1999.
- [16] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic grasp planning using shape primitives. In *Proc. of ICRA*, volume 2, pages 1824–1829 vol.2, 2003.
- [17] L. Pao and T. H. Speeter. Transformation of human hand positions for robotic hand control. In *Proc. of RA*, pages 1758–1763 vol.3, 1989.
- [18] A. Peer, S. Eienkel, and M. Buss. Multi-fingered telemanipulation - mapping of a human hand to a three finger gripper. In *Proc. of RO-MAN*, pages 465–470, 2008.
- [19] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An SVM learning approach to robotic grasping. In *Proc. of ICRA*, volume 4, pages 3512–3518 Vol.4, 2004.
- [20] K. Rawlik, M. Toussaint, and S. Vijayakumar. On Stochastic Optimal Control and Reinforcement Learning by Approximate Inference. In *Proc. of R:SS*, 2012.
- [21] P. Sandilands, Myung-Geol Choi, and T. Komura. Capturing Close Interactions with Objects Using a Magnetic Motion Capture System and a RGBD Sensor. In *Motion in Games*, volume 7660, pages 220–231. Springer Berlin Heidelberg, 2012.
- [22] P. Sandilands, Myung-Geol Choi, and T. Komura. Interaction Capture using Magnetic Sensors. *Computer Animation and Virtual Worlds*, 2013.
- [23] Peng Song and V. Kumar. A potential field based approach to multi-robot manipulation. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1217–1222, 2002.
- [24] M. Toussaint. Robot Trajectory Optimization using Approximate Inference. In *Proc. of ICML*, pages 1049–1056. ACM, 2009.
- [25] A. Van Oosterom and J. Strackee. The Solid Angle of a Plane Triangle. *IEEE Transactions on Biomedical Engineering*, BME-30(2):125–126, February 1983.
- [26] He Wang, K. Sidorov, P. Sandilands, and T. Komura. Harmonic Parameterization by Electrostatics. *ACM Transactions on Graphics (TOG)*, 2013.
- [27] Guowu Wei, Jian S. Dai, Shuxin Wang, and Haifeng Luo. Kinematic Analysis and Prototype of a Metamorphic Anthropomorphic Hand with a Reconfigurable Palm. *International Journal Of Humanoid Robotics*, 8(3):459 – 479, 2011.