Hierarchical Motion Planning in Topological Representations

Dmitry Zarubin†, Vladimir Ivan*, Marc Toussaint†, Taku Komura*, Sethu Vijayakumar*
* School of Informatics, University of Edinburgh, UK
† Department of Computer Science, FU Berlin, Germany

***Abstract*—Motion can be described in alternative representations, including joint configuration or end-effector spaces, but also more complex *topological* representations that imply a change of Voronoi bias, metric or topology of the motion space. Certain types of robot interaction problems, e.g. wrapping around an object, can suitably be described by so-called writhe and interaction mesh representations. However, considering motion synthesis solely in topological spaces is insufficient since it does not cater for additional tasks and constraints in other representations. In this paper we propose methods to combine and exploit different representations for motion synthesis, with specific emphasis on generalization of motion to novel situations. Our approach is formulated in the framework of optimal control as an approximate inference problem, which allows for a direct extension of the graphical model to incorporate multiple representations. Motion generalization is similarly performed by projecting motion from topological to joint configuration space. We demonstrate the benefits of our methods on problems where direct path finding in joint configuration space is extremely hard whereas local optimal control exploiting a representation with different topology can efficiently find optimal trajectories. Further, we illustrate the successful online motion generalization to dynamic environments on challenging, real world problems.**

I. Introduction

Many relevant robotic tasks concern close interactions with complex objects. While standard motion synthesis methods describe motion in configuration space, tasks that concern the interaction with objects can often more appropriately be described in representations that reflect the interaction more directly. For instance, consider the wrapping of arms around an object, e.g. embracing a human. Described in joint space, such a motion is complex and varies greatly depending on the embraced object. When describing the motion more directly in terms of the interaction of arm segments with object parts (e.g. using the interaction mesh representation that we will introduce below) we gain better generalization to objects of different shape or position and online adaptation to dynamic objects. Similar arguments apply to other scenarios, e.g. multi-link articulated robots reaching through small openings and complex structures, surfaces wrapping around objects and fingers grasping and manipulating objects. In such cases, the alternate representations greatly simplify the problems of motion generalization as well as planning.

There are several formal views on the implication of an alternate abstract representation: 1) In the context of randomized search, such representations alter the Voronoi *bias* and therefore, the efficiency of RRT or randomized road maps. [9] demonstrate this effect in the case of RRTs. 2) An alternate representation may imply a different *metric*, such that a trajectory that is a complex path in one space becomes a simple geodesic in another. 3) An alternate representation may change the topology of the space such that local *optimization* in one space is sufficient for finding a solution whereas global optimization (randomized search) would be needed in the other. 4) Finally, different representations may allow us to express different *motion priors*, for instance, a prior preferring "wrapping-type motions" can be expressed as a simple Brownian motion or Gaussian process prior in one space, whereas the same Brownian motion prior in configuration space renders wrapping motions extremely unlikely.

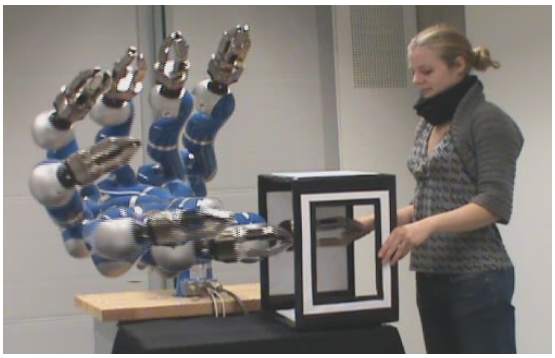


Fig. 1. KUKA LWR 4 robotic arm reaching through a hollow box with task being defined in combined Writhe and interaction mesh space, showing an example of planning and dynamic remapping using topological representations as described in Section V-B.

As opposed to the computer animation domain, where topological representation have recently been used [5], synthesizing motion in such abstract spaces for planning and control of robotic systems come with additional challenges. Typically, control tasks are specified in world (or end effector) coordinates, the obstacles may be observed in visual (or camera) coordinates, and the joint limits of the actuators are typically described in joint coordinates. Therefore, the general challenge is to devise motion synthesis methods that combine the benefits of reasoning in topological coordinates while preserving consistency across the control coordinates and managing to incorporate dynamic constraints from alternate representations seamlessly.

In this paper, we propose methods that can combine the different representations at the abstract and lower level and plan simultaneously at these different levels to efficiently generate optimal, collision free motion that satisfy constraints.

Each representation has their own strength and weaknesses and coupling them can help solve a wider range of problems. More specifically, our contributions are:

- The introduction of topological *representations* tuned to the domain of robot motion synthesis and manipulation, with a strong focus on the interaction of manipulation and the environment.
- An extension of a stochastic optimal *control framework* that combines various representations for motion synthesis. This is expressed in a graphical model that couples motion priors at different levels of representations.
- A method for motion *generalization* (remapping) to novel situations via a topological representation.
- Experiments that *validate* the benefit of Bayesian inference planning with topological representations in problems involving complex interactions.

In the rest of this section, we will first review previous work on the use of topological representations for character animation and abstract spaces used for robot motion synthesis. Section II then introduces two specific types of alternate representations, one of which has the origin in topological properties of strings—the writhe—and the other capturing the relative distances between interacting parts. Section III presents our approach to combining topological and configuration space representations in an optimal control setting through the Approximate Inference Control (AICO) [17, 13, 12]) framework. This naturally leads to an extension that includes random variables for both the topological and configuration space representations, with their specific motion priors coupled via the graphical model. Section IV addresses using topological representation to generalize motion to novel situations by "remapping" it using this more abstract space. Finally, in Section V we present experiments on using the proposed methods to solve motion synthesis problems like unwrapping that are infeasible (e.g. for RRT methods) without exploiting alternate representations and demonstrate generalizability to novel, dynamic situations.

A. Previous Work

There is strong interest in reducing the dimensionality of the state space of robots to simplify the control strategy. In [14], the dimensionality is reduced by projecting the states into the task space and biasing the exploration of random trees towards the goal in the task space. Although such an approach is valid if the work space is an open area with low number of obstacles, it is difficult to use when the task involves close interactions between the robot and objects in the scene.

Machine learning techniques have been used to reduce the dimensionality of the task space. In [2], a latent manifold in joint space is computed using Gaussian process from sample configurations produced by an expert. This manifold is, however, defined by samples from a valid trajectory in joint space and it does not capture state of the environment directly.

In order to cope with problems of close interactions, it is necessary to abstract the state space based on the spatial relations between the body parts and objects. Several robotics researchers have developed knotting robots that abstract the status of the strands and plan the maneuvers by probabilistic road maps [15, 18]. These approaches represent the rope state based on how it is overlapping with itself when viewing it from a specific direction [4]. Then, the state transition is achieved by moving the end points toward a specific direction. Such a representation is not very convenient due to the view-dependence and the difficulty of manipulating the rope.

Alternate topological representations that describes the relationships between 1D curves using their original configurations are proposed for motion synthesis [5, 16] and classifying paths into homotopy groups [1]. In [5], a representation called Topology Coordinates that is based on the Gauss Linking Integral is proposed to synthesize tangling movements. Bodies are mapped from the new representation to the joint angles by the least squares principle. In [16], the same representation is applied for controlling the movement of a robot that puts a shirt on a human. The mapping from the new representation to the low level representation is learned through demonstrations by humans. Such an approach is applicable only when the desired movement is consistent and simple, but not when the planning needs to be done between arbitrary sample points in the state space. An idea to abstract the paths connecting a start point and the end point using homotopy classes in 2D based on complex numbers and in 3D based on the Ampere's law [1] are also proposed. Here the paths are only classified into homotopy groups and there is no discussion about the mapping from the topological representation to the low level control coordinates. These representations are only applicable for 1D curves and is not applicable for describing the relationship between 2D surfaces, which is often needed for controlling robots.

Another representation called interaction mesh that abstracts the spatial relationship of the body parts is introduced in [6]. The relationship is quantified by the Laplacian coordinates of the volumetric mesh whose points are sampled on the body parts composing the scene. The advantage of this approach is that it is not restricted to describe the relationship between 1D curves but is extendable to those between 2D surfaces. On the contrary, it is a discrete representation which is only valid in the neighborhood of the posture that the volumetric mesh is computed.

In summary, using a representation based on the spatial relations is a promising direction to reduce the dimensionality of the control, although little work has been done for path planning or optimal control in such state spaces.

II. Topological representations

The topological spaces we will introduce significantly modify the metric and topology of the search space. For instance, points that are near in the topology space might be far in configuration space, or a linear interpolation in topology space may translate to complex non-linear motion in configuration space. The motion synthesis methods proposed in the next section are independent of the specific choice of representation. Here we will introduce two specific examples of topological representations which we will also use in the experiments.

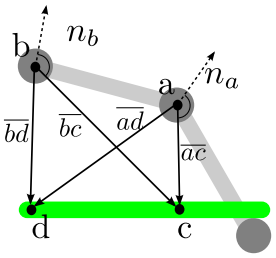


Fig. 2. Illustration of the definition of writhe for two segments. One - ab - belongs to the manipulator and another - cd - is a part of the obstacle.

These representations have previously been used in the context of computer animation, namely the writhe representation [5] that captures the "windedness" of one object around another, and the interaction mesh representation [6] that captures the relative positioning of keypoints between interacting objects.

Both types of representations can be formalized by a mapping $\phi : q \mapsto y$ from configuration space to the topological space, where $q \in \mathbb{R}^n$ is the configuration state with n degrees of freedom. To be applicable within our motion synthesis system, we need to compute ϕ and its Jacobian J for any q.

A. Writhe matrix and scalar

The writhe [7] is a property of the configuration of two kinematic chains (or in the continuous limit, of two strings). Intuitively the writhe describes to what degree (and how and where) the two chains are wrapped around each other.

Let us describe two kinematic chains by positions $p^{1,2}_{1:K}$ of their joints, where $p^i_k \in \mathbb{R}^3$ is the kth point of the ith chain. Using standard kinematics, we know how these points depend on the configuration $q \in \mathbb{R}^n$, that is, we have the Jacobian $J^i_k := \frac{\partial p^i_k}{\partial q}$ for each point. The writhe matrix is a function of the link positions $p^{1,2}_{1:K}$.

More precisely, the **writhe matrix** W_{ij} describes the relative configuration of two points (p^1_i, p^1_{i+1}) on the first chain and two points (p^2_j, p^2_{j+1}) on the second where i, j are indexes of points along the first and the second chain respectively. For brevity, let us denote these points by $(a, b) = (p^1_i, p^1_{i+1})$ and $(c, d) = (p^2_j, p^2_{j+1})$, respectively (see Fig. 2). Then

$$W_{ij} = \sin^{-1}\frac{n_a^\top n_d}{|n_a||n_d|} + \sin^{-1}\frac{n_b^\top n_c}{|n_b||n_c|} + \sin^{-1}\frac{n_c^\top n_a}{|n_c||n_a|} + \sin^{-1}\frac{n_d^\top n_b}{|n_d||n_b|} \tag{1}$$

where n_a, n_b, n_c, n_d are normals at the points a, b, c, d with respect to the opposing segment (c.f. Fig. 2),

$$n_a = \overline{ac}\times\overline{ad}\ ,\ n_b = \overline{bd}\times\overline{bc}\ ,\ n_c = \overline{bc}\times\overline{ac}\ ,\ n_d = \overline{ad}\times\overline{bd}\ . \tag{2}$$

The above equations for computing the Writhe are an analytical expression for Gauss integral along two segments which is the solid angle formed by all view directions in which segments (a, b) and (c, d) intersect [7].

Since the writhe matrix is a function of the link positions $p^{1,2}_{1:K}$ we can compute its Jacobian using the chain rule

$$\frac{\partial W_{ij}}{\partial q} = \frac{\partial W_{ij}}{\partial p^1_i} J^1_i + \frac{\partial W_{ij}}{\partial p^1_{i+1}} J^1_{i+1} + \frac{\partial W_{ij}}{\partial p^2_j} J^2_j + \frac{\partial W_{ij}}{\partial p^2_{j+1}} J^2_{j+1}\ . \tag{3}$$

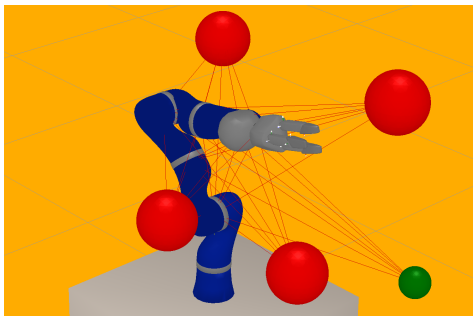


Fig. 3. Interaction mesh created from the obstacles (red), the goal (green) and the KUKA LWR 4 arm. Edges of the interaction mesh are shown as red lines. Edges between the obstacles have been removed.

The writhe matrix W is a detailed description of the relative configuration of two chains. Fig. 5 illustrates 2 configurations together with their writhe matrix representation. Roughly, the amplitude of the writhe (shading) along the diagonal illustrates which segments are wrapped around each other.

From the full writhe matrix we can derive simpler metrics, usually by summing over writhe matrix elements. For instance, the Gauss linking integral, which counts the mean number of intersections of two chains when projecting from all directions, is the sum of all elements of the writhe matrix. In our experiments, we will also use the vector $w_j = \sum_i W_{ij}$ as a representation of the current configuration. Writhe, however, does not provide a unique mapping to joint angles which is why we require additional constraints and cost terms especially in scenarios where wrapping motion is not dominant.

B. Interaction mesh

An interaction mesh describes the spatial interaction of the robot with its environment [6]. The interaction mesh is a function of a graph connecting a set of landmark points on the robot and in the environment. More precisely, assume we have a set of points $P = \{p_i\}_i$ including landmarks on a kinematic robot configuration and on objects in the environment. Let G be a (fully or partially connected) graph on P. The selection of landmarks and the graph connectivity should reflect the desired interaction of the environment and the robot. To each vertex $p \in G$ in the graph, we associate the Laplace coordinate

$$L_G(p) = p - \sum_{r \in \partial_G p} \frac{r}{|r - p|W} \tag{4}$$

where $\partial_G p$ is the neighborhood of p in the graph G and $W = \sum_{s\in\partial_G p} |s - p|^{-1}$ is the normalization constant for the weighting inversely proportional to $|r - p|$. This constant ensures that the weights sum up to one and therefore make the representation invariant to scale. The collection of Laplace coordinates of all points,

$$M = (L_G(p))_{p\in P}\ , \tag{5}$$

is a $3|P|$-dimensional vector which we denote as *interaction mesh*. As with the writhe matrix, we assume the Jacobian of all robot landmarks in P is given and the Jacobian of other environmental landmarks is zero. The Jacobian $\frac{\partial M}{\partial q}$ of the interaction mesh is given via the chain rule.

We would like to point out that the squared metric in M-space has a deformation energy interpretation [6]. To see this, consider a change of position of a single vertex p to a new position p'. The deformation energy associated to such a change in position is defined based on the neighborhood in a tetrahedronisation T of the point set:

$$E_T(p') = \frac{1}{2}\|L_T(p') - L_T(p)\|^2 \tag{6}$$

where $L_T(p)$ are the Laplace coordinates of p w.r.t. the tetrahedronisation T. The difference to our definition of the interaction mesh is that we consider Laplace coordinates L_G w.r.t. the fully connected graph G instead of only T, which is a subgraph of the fully connected graph G. Since different configurations lead to topologically different T, using L_G has the benefit of more continuous measures (deformation energies as well as Jacobian). Neglecting this difference, minimizing squared distances in M-space (as is implicit, e.g., in inverse kinematics approaches as well as the optimal control approaches detailed below) therefore corresponds to minimizing deformation energies.

III. Optimal control combining topological and configuration space representations

The motivation for introducing topological spaces is that they may provide better metrics or topology for motion synthesis, ideally such that local optimization methods within topological space can solve problems that would require more expensive global search in configuration space. In this section, we describe our method for exploiting topological spaces for motion synthesis in an optimal control context. We formulate the approach within the framework of Approximate Inference Control[13, 17], which is closely related to differential dynamic programming [10] or iLQG [8, 11, 3] (see details below), and will allow us to use a graphical model to describe the coupling of motion estimation on both representations. The main difference between iLQG and AICO is that iLQG (and other classical methods) do not have have an counterpart of the forward messages (see [17] for more details). AICO also allows for easy definition of the problem by extending the graphical model. In the following, we first briefly introduce this framework before we explain how to couple additional representations in the probabilistic inference framework.

A. Approximate Inference Control

Approximate Inference Control (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

$$C(x_{0:T}, u_{0:T}) = \sum_{t=0}^{T} [c_x(x_t) + c_u(u_t)] \tag{7}$$

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

$$p(x_{0:T}, u_{0:T}) \propto P(x_0) \prod_{t=0}^{T} P(u_t) \prod_{t=1}^{T} P(x_t \,|\, u_{t-1}, x_{t-1}) \cdot \prod_{t=0}^{T} \exp\{-c_x(x_t)\} \ . \tag{8}$$

The control prior $P(u_t) = \exp\{-c_u(u_t)\}$ 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 may 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 $p(x_{0:T}, u_{0:T}) = P(x_{0:T}, u_{0:T} \,|\, z_{0:T} = 1)$. AICO in general tries to estimate p, in particular the posterior trajectory and controls. In [17], this is done using Gaussian message passing (comparable to Kalman smoothing) based on local Gaussian approximations around the current belief model. In [13], theory on the general equivalence of this framework with stochastic optimal control is detailed. Generally, the approach is very similar to differential dynamic programming [10] or iLQG [8] methods with the difference that not only backward messages or cost-to-go functions are propagated but also forward messages ("cost-to-reach functions"), which allows AICO to compute a local Gaussian belief estimate $b(x_t) \propto \alpha(x_t)\beta(x_t)$ as the product of forward and backward message and utilize it to iterate message optimization within each time slice.

B. Expressing motion priors in topological spaces and coupling spaces

To estimate the posterior, the controls u_t can be marginalized, implying the following **motion prior**:

$$P(x_{t+1} \,|\, x_t) = \int_u P(x_t \,|\, u_{t-1}, x_{t-1}) \ P(u_t) \ du \ . \tag{9}$$

This motion prior arises as the combination of the system dynamics and our choice of control costs $c_u(u_t)$ in x-space; for LQ systems, it is a linear Gaussian.

This motion prior is a unique view on our motivation for topological representations. In the introduction, we mentioned the impact of representations on the Voronoi bias, the metric, or the topology. In other words, successful trajectories are likely to be "simpler" (easier to find, shorter, local) in an appropriate space. In Machine Learning terms, this is expressed in terms of a prior. In this view, topological spaces are essentially a means to express priors about potentially successful trajectories—in our case we employ the linear Gaussian prior in a topological space to express the belief that trajectories may appear "simpler" in a suitable topological space.

However, using AICO with a linear Gaussian motion prior in topology space is not sufficient to solve general motion synthesis problems: 1) The computed posterior in topology

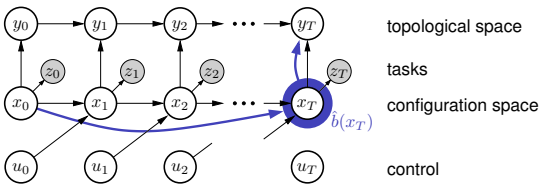


Fig. 4. AICO in configuration and topological space. The blue arcs represent the approximation used in the end-state posterior estimation.

does not directly specify an actual state trajectory or control law on the joint level. 2) We neglect the problem of minimization of control and task costs originally defined on the joint level. To address these issues, we need mechanisms to couple inference in topological and state space. We do so by coupling topological and joint state representations in AICO's graphical model framework.

Fig. 4 displays a corresponding graphical model. The bottom layer corresponds to the standard AICO setup, with the motion prior $P(x_{t+1} \,|\, x_t) = \int_u P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; P(u_t)\; du$ 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 topological space with an a priori given linear Gaussian motion prior $P(y_{t+1}|y_t)$. Both layers are coupled by introducing additional factors $f(x_t, y_t) = \exp\{-\frac{1}{2}\rho\|\phi(q_t) - y_t\|^2\}$, which essentially aim to minimize the squared distance between the topological state y_t and the one computed from the joint configuration $\phi(q_t)$, weighted by a precision constant ρ. Note that for Gaussian message passing between levels using a local linearization of ϕ (having the Jacobian of the topological space) is sufficient. These factors essentially treat the topological state y_t as an additional task variable for the lower level inference, analogous to other potential task variables like end-effector position or orientation.

C. End-state posterior estimation

Probabilistic inference in the full factor graph given in Fig. 4 would require joint messages over configuration and topological variables or loopy message passing. We approximate this inference problem by a stage-wise inference process:

1) We first focus on approximating directly an end-state posterior $\hat{b}(x_T)$ for the end-state x_T which fulfils the task defined in configuration space. We explain the method used for this below.
2) Accounting for the coupling of this end-state posterior to the topological representation, we compute a trajectory posterior in topological space.
3) We then project this down to the joint level, using AICO in configuration space coupled to the topological space via factors introduced above.

Clearly this scheme is limited in that the initial inference in topology space only accounts for the task at the final time step. To overcome this limitation, we would have to iterate inference between levels. For the problems investigated in our experiments, the approximation scheme above is sufficient.

End-state posterior estimation computes an approximate belief $\hat{b}(x_T) \approx P(x_T \mid x_0, z_{0:T} = 1)$ about the final state given the start state and conditioned on the task. This approximation neglects all intermediate task costs and assumes linear Gaussian system dynamics of the form

$$P(x_t \mid x_{t\text{-}1}) = \mathcal{N}(x_t \mid A_t x_{t\text{-}1} + a_t, W_t) \;. \tag{10}$$

We integrate the system dynamics,

$$P(x_T \mid x_0) = \sum_{x_{1:T\text{-}1}} \prod_{t=1}^{T} P(x_t \,|\, x_{t\text{-}1}) \;, \tag{11}$$

which corresponds to the blue arc in Fig. 4. For stationary linear Gaussian dynamics, we have

$$P(x_T|x_0) = \mathcal{N}(x_T|A^T x_0 + \sum_{i=0}^{T\text{-}1} A^i a, \sum_{i=0}^{T\text{-}1} A^i W A^{'i}) \;, \tag{12}$$

where superscript on A states for a power of matrix, defined iteratively $A^i = A * A^{i-1}$. To estimate $\hat{b}(x_T)$, we condition on the task,

$$P(x_T \mid x_0, z = 1) = \frac{P(z_T = 1 \mid x_T)\; P(x_T \mid x_0)}{P(z_T = 1 \mid x_0)} \;. \tag{13}$$

Since we assume $P(x_T \mid x_0)$ to be Gaussian, using a local Gaussian approximation of the task $P(z_T = 1 \mid x_T)$ around the current mode of $\hat{b}(x_T)$, $P(x_T \mid x_0, z = 1)$ can be approximated with a Gaussian as well. We iterate this by alternating between updating the end-state estimate $\hat{b}(x_T)$ and re-computing the local Gaussian approximation of the task variable. A Levenberg-Marquardt type damping (depending on monotonous increase of likelihood) ensures robust and fast convergence.

IV. Generalization and Remapping using a Topological Space

The optimal control approach presented in the previous section exploits the additional topological space to generate an optimal trajectory (and controller). However, the computed optimal trajectories may no longer be valid when there is a change in the environment, e.g. the obstacles have moved. In order to cope with this issue, we propose a per-frame remapping approach in which the optimal trajectory in the topological representation $y^*_{0:T}$ is inverse-mapped to the configuration space according to the novel condition of the environment.

Technically, this is done by computing the configuration of the system per-frame such that the remapping error from the original optimal topological trajectory $y^*_{0:T}$ is minimized. However, topological representations such as the writhe matrix or the interaction mesh are very high-dimensional—often higher dimensional than the configuration space itself. This is in strong contrast to thinking of $y^*_{0:T}$ as a lower dimensional task space like an endeffector space. Therefore, following $y^*_{0:T}$ exactly is generally infeasible and requires a regularisation

procedure that minimizes the 1-step cost function:

$$f(q_{t+1}) = \|q_{t+1} - q_t - h\|^2 + \|\phi(q_{t+1}) - y^*\|^2_C \ , \quad (14)$$

$$\underset{q_{t+1}}{\text{argmin}}\, f(q_{t+1}) = q_t + J^\sharp(y^* - y_t) + (I - J^\sharp J)h \quad (15)$$

$$\text{with } J^\sharp = J^\top (JJ^\top + C^{-1})^{-1}$$

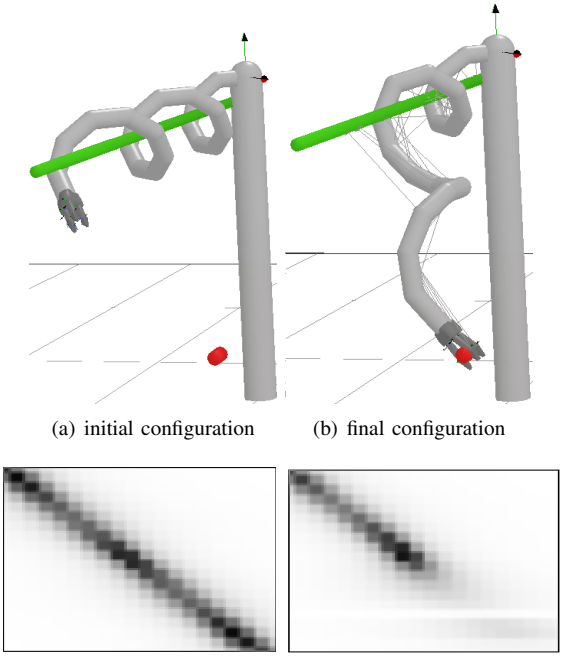
where C describes a cost metric in y-space.

For the case of the interaction mesh, we mentioned the relation of a squared metric C in M-space to the deformation energy. Therefore, using the per-frame remapping to follow an interaction mesh reference trajectory $M^*_{0:T}$ essentially tries to minimize the deformation energy between the reference M^*_t and the actual $\phi(q_t)$ at each time step. This implies generalizing to new situations by approximately preserving relative distances between interacting objects instead of directly transferring joint angles. In conjunction with the use of feedback gains, the methodology proposed here is able to cope with dynamic environments (see Section V-C) and bounded unpredictable changes.

V. Experiments

A. Rope unwrapping and reaching using writhe space

As an example of a possible application of the writhe space abstraction we simulated a manipulator, consisting of 20 segments and a hand with three fingers, making in total 29 DoF. Initially this rope-like manipulator is twisted two and a half times around a green pole, giving us approximately 900° of writhe density (See Fig. 5(a)).



(a) initial configuration (b) final configuration

(c) initial writhe matrix (d) final writhe matrix

Fig. 5. The experimental task is to grasp the object without collisions. Corresponding writhe matrices are depicted on lower plots - the darkness corresponds to the amplitude of the writhe value.

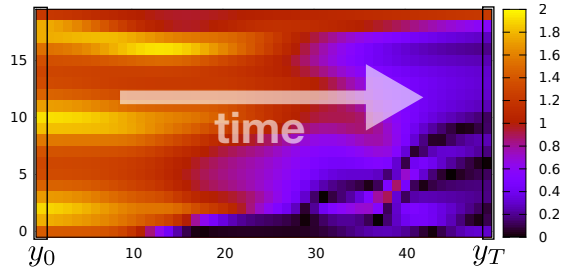


Fig. 6. Each column represents a state in writhe space - values show how much each segment of the manipulator is twisted around the stick at each time step.

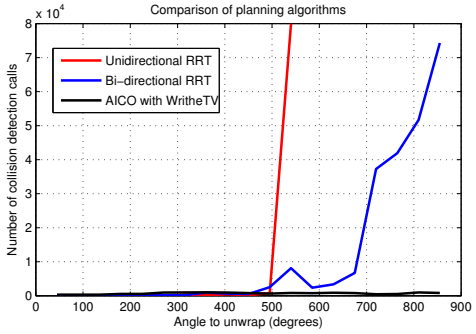


Fig. 7. Performance of planning algorithms for unwrapping task. Computational time is proportional to the number of collision detection calls.

The task is to plan a trajectory which should grasp the red cylinder without colliding with the stick (Fig. 5(a),5(b) and video available at `http://goo.gl/fwSxG`). Clearly, a local feedback approach using Inverse Kinematics will experience failure in this task. On the other hand, a successful trajectory can be well captured as a linear interpolation in writhe space and projected back to the configuration space using the coupling described in Section II-A. Fig. 6 illustrates an example of a unwrapping trajectory in topological space when all rows of writhe matrix are summed up into one column, representing the current state.

Hierarchical AICO conditioned on the end-state in writhe space y_T was able to generate locally optimal trajectories, consisting of 50 time steps, in only few iterations, requiring a relatively small number of expensive collision checks (less than 1000). Comparison with Rapidly-exploring Random Tree (RRT) planning for this reaching task revealed a dependence of the performance on distance between end-effector and object position. Moreover, total costs of obtained trajectories were on average 100 times higher than those generated with the local optimizer. Here, the end-state in configuration space q_T was given as a target for RRTs.

For more systematic evaluation of our planning platform we have designed a sequence of unwrapping trajectories - gradually increasing the relative angle to be unwrapped. This sequence of final states was given as goals to uni- and bi-directional RRT planners. The results demonstrate that for simple trajectories (e.g. in case of nearby lying objects) all methods have no difficulties, whereas starting with one and a half of full twist, unidirectional search fails and bi-directional

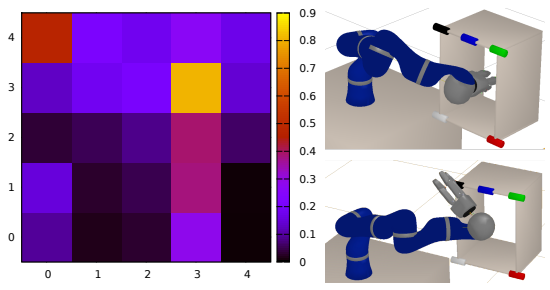
significantly slows down. (See Fig. 7.)

In this comparison, the RRTs solved a somewhat simpler problem than our system: For the RRTs we assumed to know the final state q_T in configuration space – we take our final pose estimate from Section III-C as the target q_T for RRTs. This is in contrast to our planning platform, where we use the final pose estimate only to estimate a final topological state y_T and then use the hierarchical AICO to compute an optimal trajectory (including an optimal q_T) conditioned on this final topological state. Therefore, the RRT's problem is reduced to growing to a specific end state. We applied the standard method of biasing RRT search towards q_T by growing the tree 10% of the time towards q_T instead of a random sample of the configuration space. Knowing q_T also allowed us to test bi-directional RRTs, each with 10% bias to grow towards a random node of the other tree. Further, RRTs output non-smooth paths whereas AICO produces (locally) optimal dynamic trajectories since it minimizes dynamic control costs[1].

B. Dynamic reaching through a loop using writhe and interaction mesh

Writhe space is a suitable representation for tasks that involve interactions with chains—or loops—of obstacles. As a second demonstration, we have altered the task from Experiment V-A to reaching through a hollow box, where the rim of the box forms a loop of segments, see Fig. 8(b). The interaction of the manipulator with the box can be described by the writhe representation between this loop of box segments and the manipulator. Classically this problem would be addressed using only an end-effector task variable and collision checks in joint configuration space. The advantage of using writhe as a description of the interaction is in defining the task as a *relative* configuration of the robot and the loop—this relative description remains effective also when the situation changes or the box is moved dynamically. The writhe matrix corresponding to the final configuration contains a peak around the last link which passes through the box (see Fig. 1 and 8). In addition to this, target in writhe space does not uniquely define the task for all arbitrary positions of the box (unlike the unwrapping task in Experiment V-A). Fig. 8(b) bottom shows configuration that does not reach through the loop but reaches the writhe space target. The writhe is computed as absolute value of Gauss linking integral which means that direction of wrapping is not clearly defined. This is the reason why one of the arm segments wraps around the rim of the box in wrong direction. In this demonstration we use an additional interaction mesh representation which is well suited to represent movement in a way that keeps appropriate distance between robot links and the obstacle and is mostly invariant w.r.t. the concrete position of the box. Explicit collision avoidance is then superfluous. We coupled all three representations—writhe, interaction mesh, and joint configuration space—using AICO by extending the graphical model to efficiently generate motions for varying

[1]AICO can also be applied on the kinematic level, where $x_t = q_t$ and with the control $u_t = q_{t+1} - q_t$ being just the joint angle step. This would reduce the computational cost further.



(a) (b)

Fig. 8. (a) writhe space target for passing the last link of a robotic arm through a loop. (b) the configuration corresponding to the target in Writhe space. The loop was built by connecting the colored cylinders attached to the rim of the hollow box.

positions of the obstacles. For this the position of a hollow box defining the loop was tracked using magnetic motion tracking system in our experiment on the real robot. The computation of locally optimal trajectories using this methods required 3 to 6 AICO iterations. We compared to generating trajectories using only classical end-effector position and collision avoidance methods. This required 20 to 30 iterations which shows that combined writhe and interaction mesh space is a better choice of representation than end-effector position with collision avoidance.

We also tested online remapping as described in Section IV using both, the writhe and interaction mesh space, to test behavior of the system when the box position is changed dynamically on the fly. A video of this experiment can be found at `http://goo.gl/fwSxG`.

C. Dynamic obstacle avoidance using interaction mesh space

Finally we present an experiment in which the robot manipulator reaches a target object in an environment where spherical obstacles dynamically move around. We show that the cost for the movement stayed close to the optimal value when the movement is adjusted through the local re-mapping scheme described in Section IV in response to the unpredictable random displacement of the obstacles.

The experiment proceeded as follows. We used AICO with work-space target for the end-effector and collision avoidance task variables to compute the optimal trajectory under a given obstacle configuration. The obstacles are then randomly moved in the workspace and the trajectories were recomputed by (1) re-mapping using Interaction mesh and (2) re-planning using AICO with complete knowledge of the obstacle motion. The costs of the two methods with respect to the amplitude of the obstacle displacement are compared in Fig. 9. While re-mapping approach (which does not have the benefit of knowledge of the updated obstacle locations in remaining time steps) results in a higher cost, it stays within a comparable range of the full AICO re-planning. If we simply rely on a generic collision detection / response scheme based on inverse kinematics to amend the movements, close to an imminent collision, the cost increases rapidly as shown in the black plot in Fig. 9. This illustrates that the topological representation generalizes the state space well such that the local re-mapping

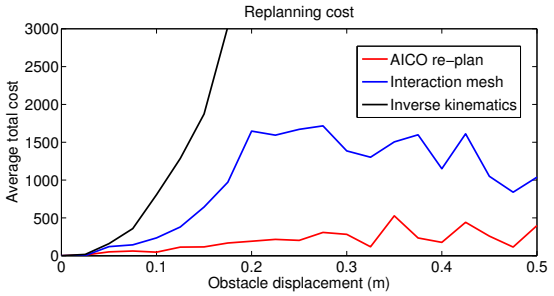


Fig. 9. The average total cost of (1) AICO re-planning, (2) re-mapping the trajectory using interaction mesh and (3) reaching using inverse kinematics to control the end-effector position. The running cost consists of end-effector position term, a collision term and a control cost term.

scheme can produce an adequate response in real-time with low computational effort while achieving the overall task aims.

We demonstrate the results on a real time manipulation using the KUKA LWR 4 robotic manipulator and a vision tracking system. The movement of the obstacles was individually controlled by three uncorrelated individuals, showing that this method can deal with unpredictable movement of obstacles in real time (Video is available at `http://goo.gl/fwSxG`).

VI. Conclusions

Different motion representations have different strengths and weaknesses depending on the problem. For certain interaction problems there exist suitable topological representations in which the interaction can be described in a way that generalizes well to novel or dynamic situations (as with the interaction mesh), or where local optimization methods can find solutions that would otherwise require inefficient global search (as with the writhe representations). However, considering motion planning only in a topological representation is insufficient for additionally accounting for tasks and constraints in other representations.

Previous work with such representations has only tested basic approaches for inverse mapping of fixed topological trajectories to the joint configuration [4, 5, 16]. In contrast, in this paper we presented methods that combine the different representations at the abstract and lower level for motion synthesis. For instance, the reaching task in an endeffector space is coupled with a writhe space that allows a local optimization method to generate an unwrapping-and-reaching motion. Considering such a problem only in joint configuration and endeffector space leads to "deep local minima" that are practically infeasible to solve—as our comparison to RRTs in Section V-A demonstrated. Considering such a problem only in writhe space would not address the actual reaching task.

We chose to formulate our approach in the framework of optimal control as an approximate inference problem since this allows for a direct extension of the graphical model to incorporate multiple representations. Alternative formulations are possible, for instance as a structured constraint optimization problem (MAP inference in our graphical model) that could be solved by methods such as SNOPT. What we coined as a motion prior in topological spaces would here correspond to pseudo control costs for transitions in topological space. Which formulation will eventually lead to computationally most efficient algorithms is a matter of future research. As an outlook, we aim to apply the proposed methods for dexterous robot manipulation (including grasping) of more complex, articulated or flexible objects, where we believe that multiple parallel representations will enable more robust and generalizing motion synthesis strategies.

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

References

[1] S Bhattacharya, M Likhachev, and V Kumar. Identification and representation of homotopy classes of trajectories for search-based path planning in 3D. *Proc.of RSS*, 2011.

[2] S Bitzer and S Vijayakumar. Latent spaces for dynamic movement primitives. In *Proc. of 9th IEEE-RAS International Conference on Humanoid Robots*, pages 574–581. IEEE, 2009.

[3] D J Braun, M Howard, and S Vijayakumar. Optimal variable stiffness control: Formulation and application to explosive movement tasks. *Autonomous Robots (in press)*, 2012.

[4] C H Dowker and B T Morwen. Classification of knot projections. *Topology and its Applications*, 16(1):19–31, 1983.

[5] S L Ho Edmond and T Komura. Character motion synthesis by topology coordinates. In *Computer Graphics Forum (Proc. Eurographics 2009)*, volume 28, Munich, Germany, 2009.

[6] Edmond S L Ho, T Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. *ACM Transactions on Graphics*, 29(4):1–8, 2010.

[7] K Klenin and J Langowski. Computation of writhe in modeling of supercoiled DNA. *Biopolymers*, 54(5):307–317, 2000.

[8] Weiwei Li and E Todorov. An iterative optimal control and estimation design for nonlinear stochastic system. In *Proc. of the 45th IEEE Conference on Decision and Control*, 2006.

[9] S R Lindemann and S M LaValle. Incrementally reducing dispersion by increasing voronoi bias in RRTs. In *Proc. of International Conference on Robotics and Automation*, 2004.

[10] D M Murray and S J Yakowitz. Differential dynamic programming and newtons method for discrete optimal control problems. *Journal of Optimization Theory and Applications*, 43(3):395–414, 1984.

[11] J. Nakanishi, K. Rawlik, and S. Vijayakumar. Stiffness and temporal optimization in periodic movements: An optimal control approach. In *Proc. of IROS*, pages 718 –724, 2011.

[12] K Rawlik, M Toussaint, and S Vijayakumar. An approximate inference approach to temporal optimization in optimal control. In *Proc. of Neural Information Processing Systems*, 2010.

[13] K Rawlik, M Toussaint, and S Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Proc. of RSS*, 2012.

[14] A Shkolnik and R Tedrake. Path planning in 1000+ dimensions using a task-space Voronoi bias. In *Proc.of ICRA*, pages 2892–2898. IEEE, 2009.

[15] J Takamatsu and et.al. Representation for knot-tying tasks. *IEEE Transactions on Robotics*, 22(1):65–78, 2006.

[16] T Tamei, T Matsubara, A Rai, and T Shibata. Reinforcement learning of clothing assistance with a dual-arm robot. *Proceedings of Humanoids*, pages 733–738, 2011.

[17] M Toussaint. Robot trajectory optimization using approximate inference. In *Proc. of ICML*, pages 1049–1056. ACM, 2009.

[18] H Wakamatsu, E Arai, and S Hirai. Knotting/unknotting manipulation of deformable linear objects. *International Journal of Robotics Research*, 25(4):371–395, 2006.