Topology Representing Network for Sensor-Based Robot Motion Planning

Michael Zeller, Rajeev Sharma and Klaus Schulten

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 N. Mathews Avenue
Urbana, IL 61801

email: zeller@ks.uiuc.edu - rajeev@cs.uiuc.edu

Abstract

We present a framework for sensor-based motion planning of robotic manipulators using a *topology representing network* (TRN). Exploiting the perfectly topology preserving features of the network, the algorithm learns the representation of the *Perceptual Control Manifold* (PCM), a recently introduced concept for motion planning. This concept allows sensors to be integrated into robot motion planning. Besides a demonstration of the technical feasibility of motion planning through perfectly topology preserving maps the capabilities of this approach within an engineering framework, namely the implementation on a pneumatically driven robot arm (*SoftArm*), are demonstrated.

1 Introduction

An important step toward autonomous robotics is developing ways to generate motion plans for achieving certain goals while satisfying environmental constraints. Classical motion planning is defined on a configuration space (C-space) which is assumed to be known, implying the complete knowledge of both the robot kinematics as well as knowledge of the obstacles in the C-space [3]. Uncertainty, however, is prevalent which makes some of these motion planning techniques quite inadequate for practical purposes. Sensors such as cameras can help in overcoming these uncertainties. To best utilize the sensor feedback, a robot motion plan should incorporate constraints from the sensor system as well as criteria for optimizing the quality of the sensor feedback. Unfortunately, in most motion planning approaches, sensing is completely decoupled from planning. In [8] we present a framework for motion planning that considers sensors as an integral part of the definition of the motion goal. The approach is based on the concept of *Perceptual Control Manifold* (PCM), defined on the product of the robot C-space and sensor space (e.g. a set of image features). The PCM provides a flexible way of developing motion plans that exploit sensors effectively. However, there are robotic systems, such as the pneumatic robot arm we use for our experiments, where the PCM cannot be derived analytically, since the exact mathematical relationship between configuration space, sensor space and control signals is not known (see section 3). Instead of using the analytical expressions for deriving the PCM we therefore propose the use of a self-organizing neural network to learn the topology of this manifold. The learnt representation can then be utilized for motion planning and control of the robot. Experiments on a pneumatic robot system establish the feasibility of this framework for motion planning within an engineering environment.

2 Topology Representing Networks for Motion Planning

Topology representing networks (TRN), as introduced by Martinetz and Schulten [4, 5], can be formulated as a combination of a vector quantization scheme and a competitive Hebb rule. Although related to Self-

Organizing Feature Maps (SOFM) [2], *a priori* knowledge of the input dimensionality is not crucial and the algorithm adjusts to the topological structure of a given input manifold $\mathbf{M}$ forming a perfectly topology preserving mapping. A rigorous definition of the terms 'neighborhood preserving mapping' and 'perfectly topology preserving map' based on Voronoi polyhedra and Delaunay triangulations is given in [5]. In the following, we will outline the implemented algorithm, including the extension of the original sequence to provide additional output weights $\mathbf{w}_i^{out}$ which will be used to link a desired control action to a specific sensory input.

Following the initialization of input weights $\mathbf{w}_i^{in}$, output weights $\mathbf{w}_i^{out}$ for all units $i = 1 \dots N$ with random numbers $\in [0, 1]$ and resetting all connections to $c_{ij} = 0$ the learning cycle reads:

1. Read input vector $\mathbf{u}$ and determine current ranking order.
$$\|\mathbf{w}_0^{in} - \mathbf{u}\| \leq \|\mathbf{w}_1^{in} - \mathbf{u}\| \leq \ldots \leq \|\mathbf{w}_{N-1}^{in} - \mathbf{u}\| \tag{1}$$
2. Update input weights $\mathbf{w}_i^{in}$ and output weights $\mathbf{w}_i^{out}$ according to:
$$\mathbf{w}_i^{in}(t+1) = \mathbf{w}_i^{in}(t) + \gamma(r,t) \cdot (\mathbf{u} - \mathbf{w}_i^{in}(t)) \tag{2}$$
$$\mathbf{w}_i^{out}(t+1) = \mathbf{w}_i^{out}(t) + \gamma(r,t) \cdot (\mathbf{u} - \mathbf{w}_i^{out}(t)) \tag{3}$$
with
$$\gamma(r,t) = \epsilon(t) \cdot e^{-r_i/\lambda(t)} \tag{4}$$
for $i = 1 \dots N$, where r_i is the current rank of neuron i as determined in step 1. $\epsilon(t)$ determines the change in the synaptic weights and $\lambda(t)$ represents a neighborhood function.
3. Update the connection c_{01} between the units currently ranked 0 and 1. If $c_{01} = 0$ then set $c_{01} = 1$ and the age of the connection $t_{01} = 0$; if $c_{01} > 0$ refresh the connection age.
4. Increase the age of all connections c_{0j} to $t_{0j} = t_{0j} + 1$ for all units j with $c_{0j} > 0$. Remove connections c_{0j} which exceed a given lifetime $t_{0j} > T(t)$. Continue with step 1.

Both $\epsilon(t)$ and $\lambda(t)$ as well as $T(t)$ are a function of time and depend on the current learning step t in the same manner[1].

After the topology preserving map of the input manifold $\mathbf{M}$, which in our case is equivalent to the PCM, has been established, a locally optimized path can be determined by minimizing the Euclidean distance d_E from the current position to a given target. The motion plan can be generated as follows:

1. Read current position $\mathbf{u}_{current}$ and target position $\mathbf{u}_{target}$
2. Find best matching neurons $\mathbf{w}_{current}^{in}$ and $\mathbf{w}_{target}^{in}$
3. Move from current unit $\mathbf{w}_{current}^{in}$ to a neighboring unit i with $c_{current,i} > 0$ that satisfies
$$d_E(\mathbf{w}_i^{in}, \mathbf{w}_{target}^{in}) = min\{d_E(\mathbf{w}_i^{in}, \mathbf{w}_{target}^{in})\} \tag{5}$$
4. If $\mathbf{w}_{current}^{in} = \mathbf{w}_{target}^{in}$ then stop, otherwise continue with step 1.

In the presence of obstacles within the workspace, step 3 has to check if a move will result in a collision and avoid it. Finally, if the motion plan meets a given goal, movement can be initiated using the corresponding output values $\mathbf{w}_i^{out}$ of the map to generate the sequence of commands necessary to navigate the robot from start to target. Other global optimization strategies can be applied to the learnt representation of PCM as well. However, these will be computationally expensive especially when complex obstacles are taken into account. A promising algorithm which we plan to explore in this regard is described in [6].

As means of demonstrating the practical capabilities within an engineering framework for motion planning and control, the following section will describe the implementation of the presented neural algorithm on a pneumatic robot arm.

[1] $\epsilon(t) = \epsilon_i(\epsilon_f/\epsilon_i)^{t/t_{max}}$ $\lambda(t) = \lambda_i(\lambda_f/\lambda_i)^{t/t_{max}}$ $T(t) = T_i(T_f/T_i)^{t/t_{max}}$
with $\epsilon_i = 0.3$, $\epsilon_f = 0.05$, $\lambda_i = 0.2N$, $\lambda_f = 0.01$, $T_i = 0.1N$, $T_f = 2N$

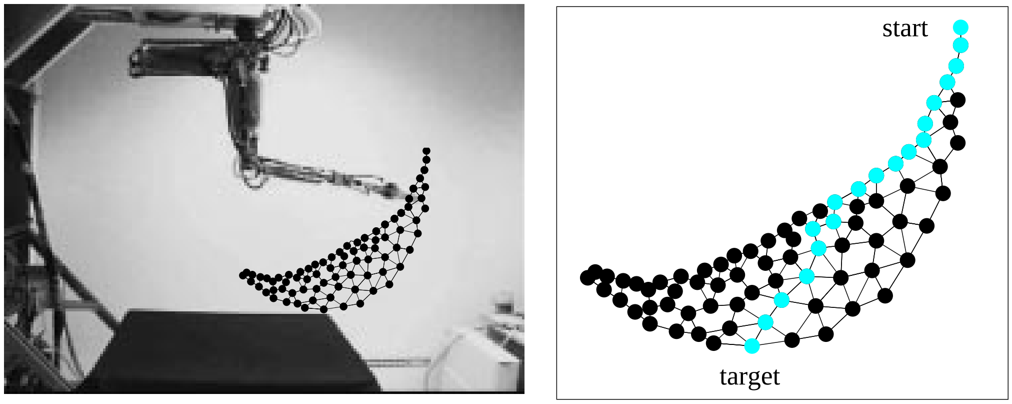

Figure 1: (*left*) *SoftArm* robot system and network structure in the workspace as seen by the camera. The learning has been accomplished and the network represents the topology of the PCM. (*right*) Visual components of the mapping and a motion plan (grey units) generated in configuration (encoder) space after start and target have been defined in vision space.

3 Motion Planning for the *SoftArm* Robotic System

The *SoftArm* is a pneumatically driven robotic manipulator, modeled after the human arm. It exhibits the essential mechanical characteristics of skeletal muscle systems employing agonist-antagonist pairs of *rubbertuators* which are mounted on opposite sides of rotating joints. Pressure difference drives the joints, average pressure controls the force (complience) with which the motion is executed. This latter feature allows operation at low average pressures and, thereby, allows one to carry out a complient motion of the arm. This makes such robots suitable for operation in a fragile environment, in particular, allows direct contact with human operators. The price to be paid for this design is that the response of the arm to pressure signals $(\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_N)^T$ and $(\Delta p_1, \Delta p_2, \ldots, \Delta p_N)^T$ cannot be described by *a priori* mathematical equations, but rater must be acquired heuristically. Furthermore, one expects that the response characteristics change during the life time of the arm through wear, after replacement of parts and, in particular, through hysteretic effects. In consequence, accurate positioning of the *SoftArm* presents a challenging problem and can only be achieved by an adaptive control mechanism. For a more detailed introduction to the mechanics of the *SoftArm* see [1].

In addition to the previous work on topology representing networks (TRN) in robotics [9, 1], where neighborhood preservation has been used to average over the output of several adjacent units in order to achieve a more accurate positioning, in the present study we focus on exploiting the topology to generate a motion plan from a current position to a given target in a 2-dimensional plane satisfying several contstraints. These contstraints can include obstacles defined in C-space, obstacles given through vision space and limitations of the camera feedback [7].

The PCM is defined as the product of C-space and sensor space. Therefore, two different types of information converge upon neurons within the network. Visual input $\mathbf{r} = (x_1, x_2)^T$ is derived from a video camera; vision preprocessing resolves the gripper location in the video frames. Angular position of the manipulator, denoted by $\mathbf{\Theta} = (\theta_1, \theta_2)^T$, is derived from the feedback of optical encoders mounted on each joint. Following a suitable training period, the topology of the network resembles the PCM. In addition, the network provides the nonlinear mapping between the position in work space $\mathbf{u} = (x_1, x_2, \theta_1, \theta_2)^T$ and the corresponding pressure commands $\mathbf{p} = (p_1, p_2)^T$ to achieve this configuration.

A sample network is depicted in Figure 1 by plotting the visual components $\mathbf{r}$ of the 4-dimensional vectors $\mathbf{w}_i^{in}$. This network was trained with a dataset of 800 random moves within a subset of the workspace and consists of 75 neural units. The left side shows the actual position in the robot's workspace. On the right side, we use the learnt representation to generate a motion plan from a start point to a given target. Both, start and target, are only given in visual space $\mathbf{r}$ (as would be obstacles), the corresponding encoder readings need not to be known. By selecting the best matching neurons for current position and target position in vision space the resulting neurons also provide the values for the encoder readings. This is possible, because $\mathbf{r}$ and $\mathbf{\Theta}$ represent redundant information. The motion plan, shown in Figure 1 on the right hand side, finally is generated exclusively in C-space to ensure a smooth motion in terms of joint angles.

Instead of learning a static topology including obstacles, one may initially present the complete workspace during the training stage and map obstacles into the PCM after the representation of the workspace has been accomplished. This approach is more suited for a robotic manipulator operating in a changing environment, e.g. with obstacles placed at different locations within the workspace.

4 Conclusions

Learning the representation of the *perceptual control manifold* (PCM) provides a very general framework for robot motion planning in which the sensing (in the form of video feedback) is factored automatically into the planning process, leading to a flexible way of visually controlling a robot manipulator. The 2d implementation on a pneumatically driven robot manipulator proves the technical feasibility of our method. It can be generalized to control robotic systems with more degrees of freedom in a 3d environment. The discretizing effect that results from the use of small numbers of neurons to map a high dimensional input space can be alleviated by introducing interpolation strategies [1] which also improve fine motion control.

Acknowledgements: This work was supported by the Carver Charitable Trust.

References

[1] T. Hesselroth, K. Sarkar, P. van der Smagt, and K. Schulten. Neural network control of a pneumatic robot arm. *IEEE Transactions of System, Man and Cybernetics*, 24(1):28–37, 1994.

[2] T. Kohonen. Analysis of a simple self-organizing process. *Biol. Cybern.*, 44:135–140, 1982.

[3] J.C. Latombe. *Robot motion planning.* Kluwer Academic Publishers, Boston, 1991.

[4] T. Martinetz and K. Schulten. A 'neural gas' network learns topologies. In *Proceedings of the International Conference on Artificial Neural Networks, Helsinki, 1991.* Elsevier Amsterdam, 1991.

[5] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[6] H. Ritter and K. Schulten. Planning a dynamic trajectory via path finding in discretized phase space. In *Parallel Processing: Logic, Organization, and Technology*, volume 253 of *Lecture Notes in Computer Science*, pages 29–39. Springer, 1987.

[7] R. Sharma and S. Hutchinson. Optimizing hand/eye configuration for visual-servo systems. In *Proc. IEEE International Conference on Robotics and Automation*, pages 172–177, May 1995.

[8] R. Sharma and H. Sutanto. Integrating configuration space and sensor space for vision-based robot motion planning. *IEEE Transactions on Robotics and Automation*, 1996. (to appear).

[9] J.A. Walter and K. Schulten. Implementation of self-organizing neural networks for visuo-motor control of an industrial robot. *IEEE Transactions on Neural Networks*, 4(1):86–95, 1993.