Neuroplanners and Their Application To Eyes/Head/Neck Coordination

Daryl H. Graf and Wilf R. LaLonde*
School of Computer Science
Carleton University,
Ottawa, Ontario Canada, K1S 5B6

Abstract

We review a neuroplanner architecture for use in constructing subcognitive controllers and new application that uses it. These controllers have wo important properties: (1) the ability to learn the topology of three continuous spaces: a steering space, a control space, and an observation space, and (2) the ability to integrate the three spaces so that initial and goal steering conditions can suggest a sequence of control states that lead the controlled system to the goal in the presence of obstacles. The result is a rudimentary planner or guidance system that can be used for such subcognitive tasks as robot manipulator control, head/eye coordination, and task sequencing. In this paper, we consider the second domain. The term neuroplanner is intended to convey the impression that the planner is implemented neurally and is more rudimentary than the conventional symbolic planners typical of artificial intelligence research.

1 Introduction

Much recent work is concerned with extending our understanding of neural mechanisms as they relate to subcognitive tasks such as vision [Fischler and Firschein, 1987], head/eye coordination [Grossberg and Kuperstein, 1986], speech processing [Kohonen, 1986, Lippmann and Gold, 1987], and motor control [Poizner *et al.*, 1987]. Most approaches develop domain specific solutions that focus either on neuron-like devices as building blocks or on individual networks such as backpropagation nets [Rumelhart and McClelland, 1986], Hopfield nets [Hopfield, 1982], and Kohonen nets [Kohonen, 1984] to name a few.

* This research has been supported by NSERC.

Our efforts are concerned with developing slightly more complex building blocks that might be used for architectural-level design of neural mechanisms. We review [LaLonde and Graf, 1988] a preliminary design for such a building block by (1) postulating the requirements that we would like it to satisfy, (2) suggesting an implementation that would satisfy those requirements, and (3) providing a new example of its use. In particular, we focus on a class of building blocks that can be used for subcognitive planning and control. We refer to these devices as neuroplanners. Unlike traditional rule-based mechanisms, neuroplanners distribute knowledge within separate self-organizing networks and between the networks in patterns of connections.

2 Neuroplanners

Neuroplanners integrate three vector spaces: a steering space, a control space, and an observation space. We refer to vectors in these spaces as steering states, control states, and observation points respectively. Intuitively, the steering space is used to guide the system being controlled, the control space represents the control states of the system being steered, and the observation space is a constraint space that dictates illegal control states. For a legged vehicle, the steering space might be represented by the angles of a pilot's steering wheel, the control space by the status of the servo-machinery that actually makes the legs move, and the observation space by terrain data about holes and hillocks that constrain the leg positions.

Our neuroplanner currently works with continuous vector spaces. Each vector represents a system state. It does not store all points in the states but uses a self-organizing algorithm to create a quantized representation of the space. Neuroplanners are intended to be used in domains where (1) there is a natural correlation between steering and control spaces, (2) where the observation space can be used to constrain the control space, and (3) where these spaces and correlations can be learned. There are two operational phases: a learning phase where the topology of the spaces are learned and correlated and a planning phase where a simple plan is generated. In this context, a plan is an ordered sequence of legal control states that satisfies a goal by moving the controlled system from an initial control state to a desired control state without violating the observation space constraints.

Figure 1 depicts a neuroplanner interface. It consists of a mode switch, four inputs, and one output. The planning-mode switch determines the mode to be used during the planning stage; it indicates cither a hypothetical initial condition or an actual initial condition. In hypothetical mode, the initial steering signal is used; otherwise, the initial control signal is used. The observation data is a quantized representation of the observation space. The maximum bandwidth of this channel is determined by the number of cells in the representation. An n-dimensional space with r grid divisions per dimension would have r^n cells, for example. The initial control signal consists of a control state and an indication as to whether or not it is legal. During learning, it is used as training data; during the planning stage, it is used as the initial control state when not in hypothetical mode (in that case, the legal indication is ignored). The initial steering signal is a steering state. During learning, it is used as training data; during the planning stage, it is used as the initial steering state when in hypothetical mode. The final steering signal is a steering state that represents a goal. It is used only in the planning stage. Finally, the output plan is the planned sequence of control states.

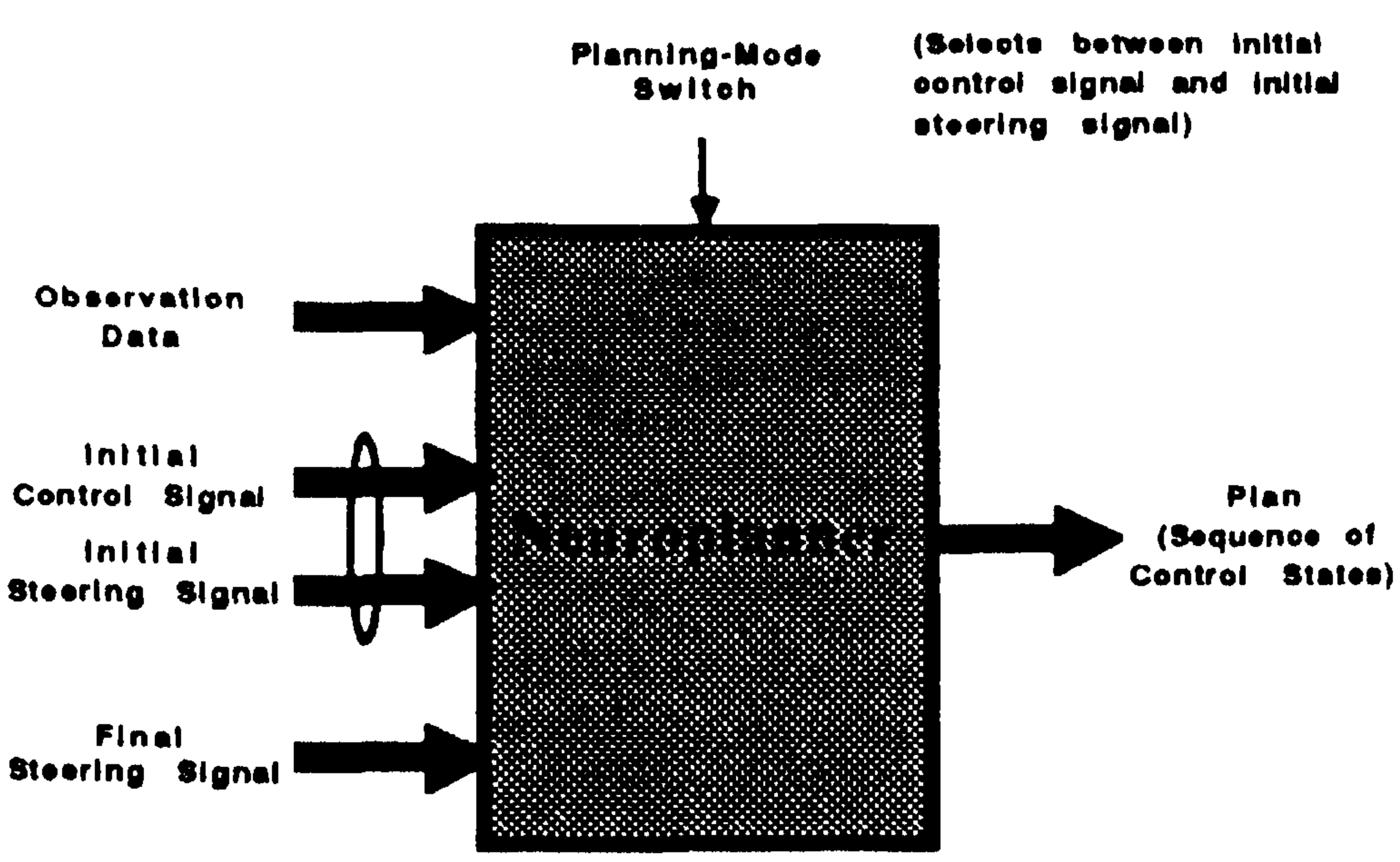


Figure 1: The Interface To The Neuroplanner Architecture

2.1 The Control Space

To be effective, the neuroplanner must learn something about its domain of application. In particular, it must understand which states are included in the control space. For example, an automated vehicle would never have both the power and braking systems fully activated. To limit its choice of states, the neuroplanner must learn where the boundaries are for legal and illegal control states and what the typical control states are.

Illegal states can occur for two reasons: (1) they represent some impossible situation such as a state beyond the physical limits of a joint or a situation that is to be permanently avoided such as a meltdown region in a nuclear plant or (2) they represent some situation that is illegal only in special circumstances; e.g., the front legs in a legged vehicle being prevented from being fully extended because they are resting on a hillock. Permanently illegal states are associated with the control space. Temporarily illegal states are associated with the observation space — they will be considered later.

Figure 2 presents an intuitive 2-dimensional representation of a control space with legal and illegal regions. The control space in this case consists of 2-vectors with x and y components. In the automated vehicle example, they could be braking pressure and fuel injection rate respectively.

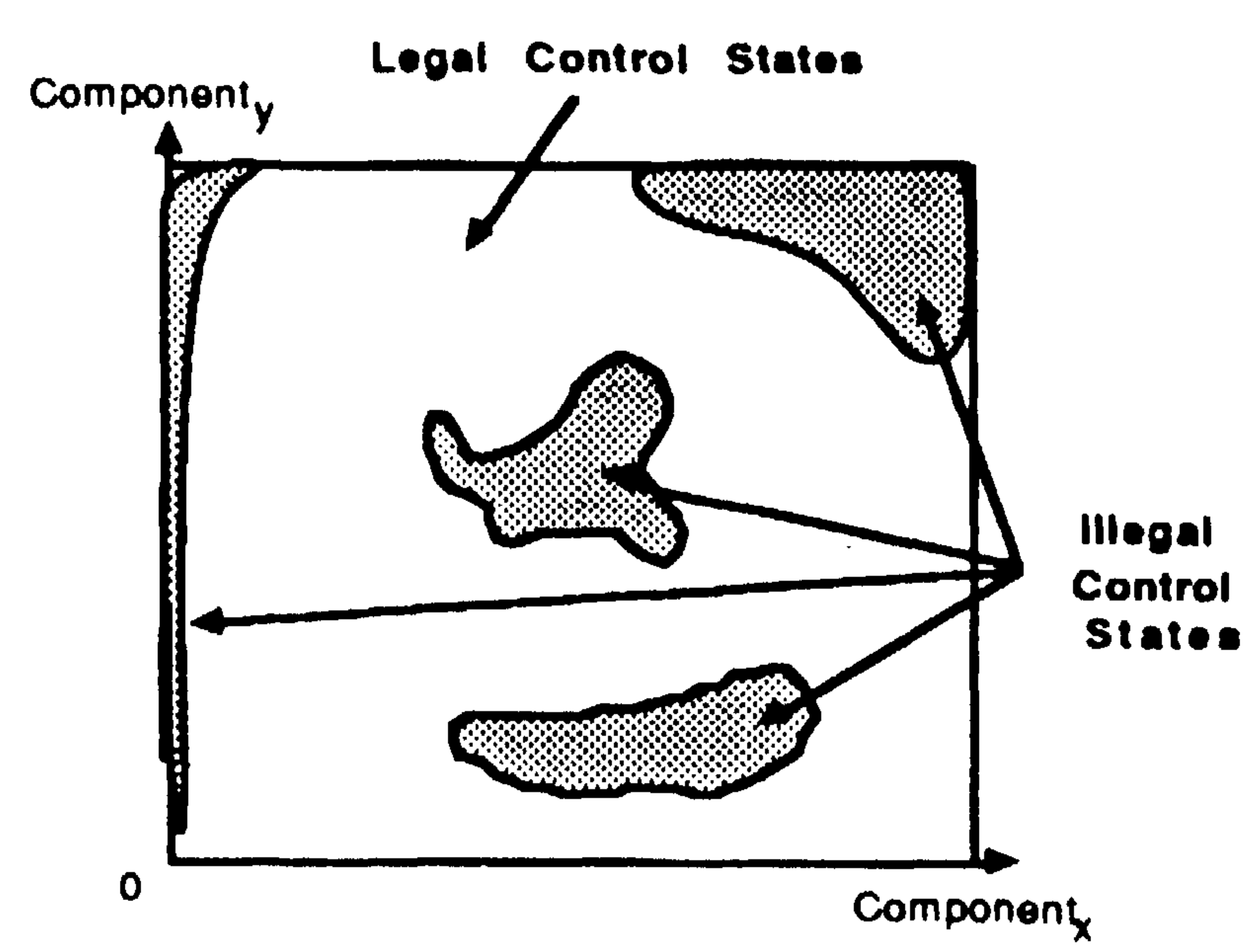


Figure 2: The Control Space

A plan is a path passing through the legal control states the control space. We consider acceptable any plan that ccessfully moves the system from the initial state to any

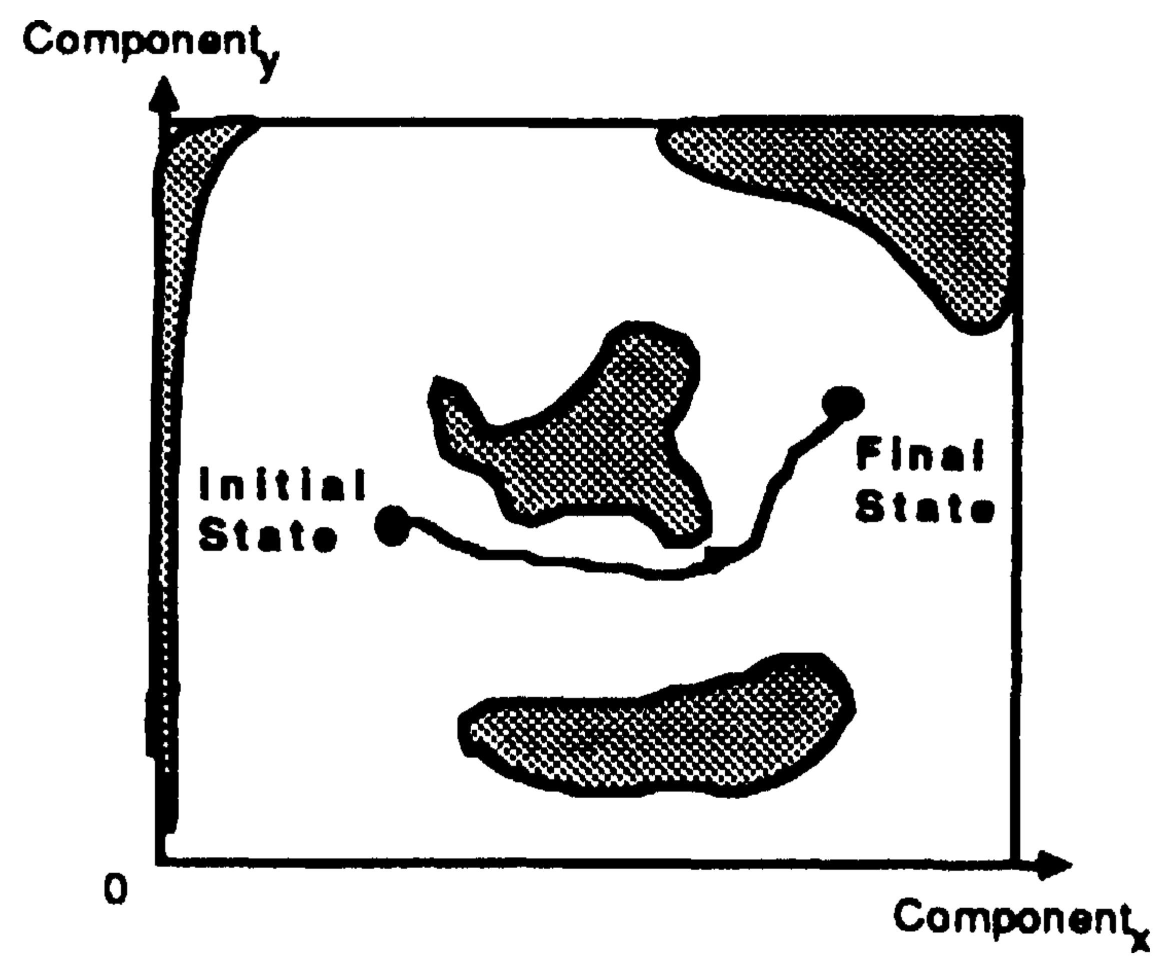


Figure 3: A Plan is a Path Through Control Space

one of the legal final states (see Figure 3). Normally, we choose a path that is minimal or near-minimal. For the approach to work, some mechanism is needed to temporarily make illegal all of those control states that violate the obstacle constraints. We discuss this in more detail in the implementation section.

2.2 The Steering Space

It may not be convenient to represent goals by listing individual control states. Often, there are many — possibly an infinite number of — final control states that satisfy a given goal. It is much simpler to represent a goal as a state in a steering space. The steering space is defined by a set of steering states whose values are provided through steering sensors. In fact, it is quite natural to think of the entire steering space as a goal space.

In a road-following vehicle, for example, the steering sensors would provide a self-relative coordinate indicating where the autopilot should go. This coordinate might be the result of specially processed vision data from stereo cameras. When properly correlated with the control space, the steering state can activate a large number of control states all satisfying the same goal.

To learn the correlations between the steering and control spaces, the user of the system must ensure that the steering sensors can be aligned with an observable that results from each control state. This alignment provides the correlations between the steering and control spaces during learning. During the planning phase, the goal is specified with a steering state; the neuroplanner can then find final control states that are correlated with the steering state for use in the planning process.

Distinct control states can of course be associated with the same steering state; i.e., there is a one-to-many mapping from the steering space to the control space. Since illegal goals are not needed, it is not necessary to distinguish between legal and illegal steering states.

2.3 The Observation Space

As we have seen, additional constraints are often imposed on the planning process. Sensory information regarding states that interfere with the planning process further constrains the shapes of valid plans. One must plan to navigate around obstacles, not through them. We refer to observables that are illegal or undesirable as obstacles. Obstacles may represent either physical objects or abstract situations to be avoided. For example, an autopilot is constrained to avoid situations where the control state of a vehicle leads to an observable collision with another vehicle. Similarly, a nuclear power plant might be constrained to avoid observable excesses of heat, radiation, and power output that can be controlled by its damping rods.

The observation space is a sensory space that dynamically tracks the observable state of the domain. Obstacles and a representation of the control state appear in the observation space as regions of points — we will refer to these as the obstacle regions and the self-image regions respectively. For planning purposes, we need to constrain the controlled system to avoid obstacles. In the observation space, this corresponds to preventing obstacle regions from intersecting with the self-image regions. The self-image is not allowed to simultaneously be its own obstacle. In order to do this efficiently, it is sufficient to correlate the points in the self-image regions with their associated control states. Later, when obstacles are observed, it not necessary to intersect these with the self-image regions. Instead, it is sufficient to insure that the plan solution does not include control states associated with obstacle region points. This is achieved by correlating the control states with their own self-image region points during a learning phase.

In the case of the above nuclear reactor example, the observation space might encode heat versus radiation versus power. For a given control state, the self-image region would contain a set of points which represent the amount of heat, radiation, and power produced by the reactor (blurred by a margin of error). In the learning phase, the self-image region points are correlated with their associated control state. In the application, phase, only the obstacle regions need be represented because these points are sufficient to deactivate the correlated undesirable control states.

3 A Neuroplanner Implementation

Neuroplanners can be implemented using a pixelated map for the observation space and two Kohonen maps for the control and steering spaces as shown in Figure 4. Each map is implemented with a network of neurons that effectively quantize the space. Obstacle regions and self-image regions in the pixelated map are represented by neurons in the active state; all others are inactive. Figure 4 actually illustrates the internal structure of a neuroplanner used for non-hypothetical reasoning. For simplicity and space reasons, we will avoid a formal mathematical treatment of the implementation. Such a treatment can be found in [Graf and LaLonde, 1988].

In the planning phase, the current control state activates one neuron in the control map using the standard Kohonen algorithm [Kohonen, 1984]. Similarly, the steering state activates a neuron in the steering map. Correlated with each neuron in the steering map is a set of neurons in the control map. The activated neuron in the steering map causes these correlated control neurons to be activated. At the same time, the observation map has a number (usually fairly large) of neurons that are activated — these are observation neurons.

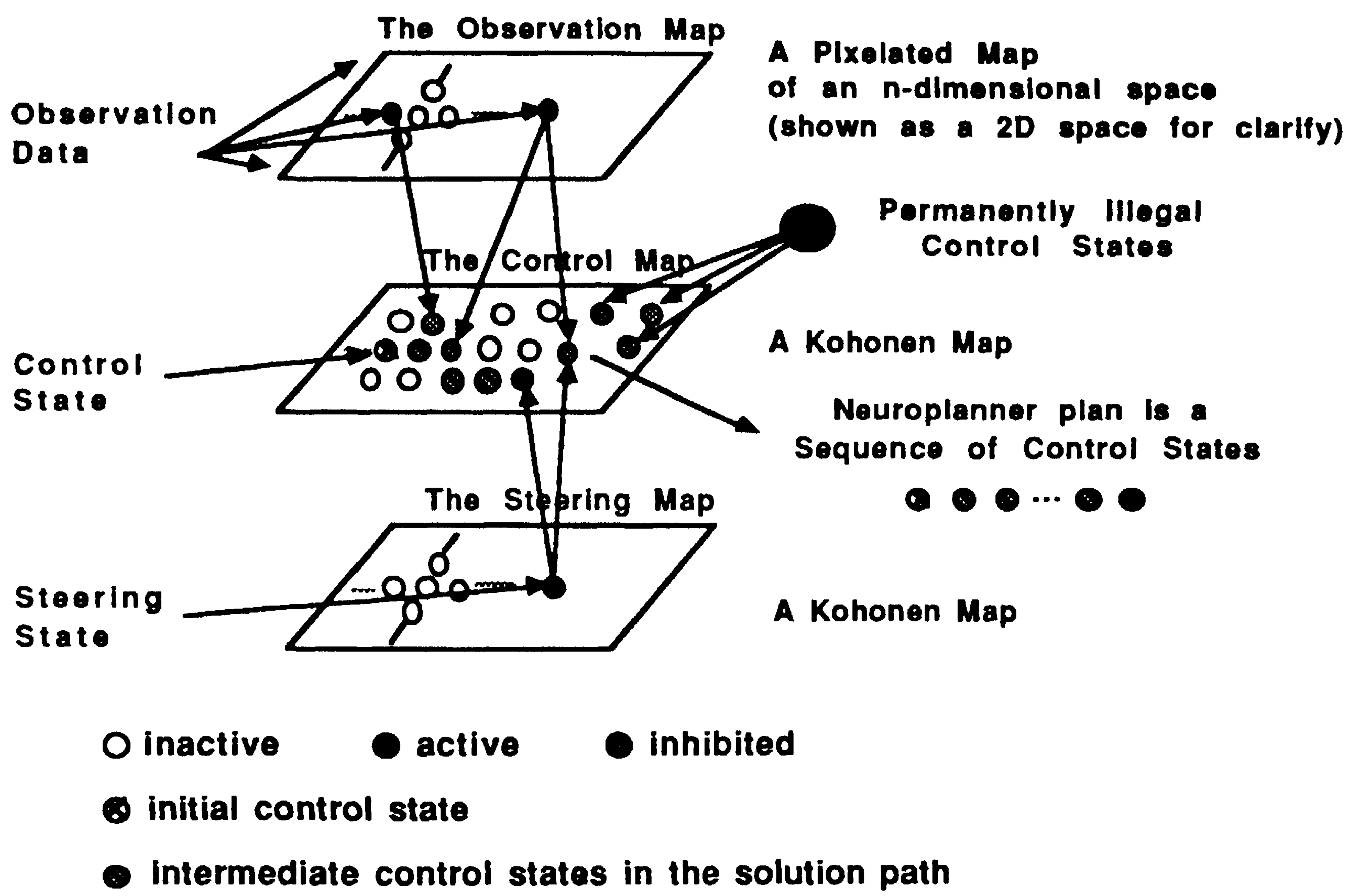


Figure 4: A Feasible Internal Structure For A Neuroplanner

These deactivate the correlated control neurons preventing them from being activated. Note that only two time steps are needed to accomplish the above since most of the activity occurs in parallel. A standard path planning algorithm can then be used to find a minimal path from the current control state neuron to any one of the active goal control state neurons. If each neuron is a processor in a systolic array, a parallel algorithm similar to the algorithm described by [Miller and Stout, 1985] could be used to find a minimal path in time where n is the number of neurons in the map. The control states associated with the neurons in the path can then be used to construct the corresponding plan. Actually, the control states are not stored with the neurons but they can be reconstructed from the weights stored in the Kohonen map.

In the learning phase, two stages are used: (1) the topology learning stage where Kohonen's self-organizing algorithm is independently applied to the control and steering maps and (2) the correlation stage which correlates both the observation and steering maps with the control map as follows. As the control space is sampled, a set of observation states and a steering state is associated with the corresponding control state by growing connections from the observation and steering neurons to the control neuron. By an observation neuron, we mean an active neuron in the observation map. Connections need not be grown if they already exist Because all three maps are quantizations of the spaces they represent, distinct states might be quantized to the same value. Note that the observation and steering information associated with a specific control state is arbitrary. The same observation and steering data might later become associated with a different control state. Once the correlation stage is over, it should be clear that a specific observation state is correlated with an arbitrary number of control states. Similarly, a steering state is correlated with some arbitrary number of control states. In other words, there is a one-to-many mapping from the observation and steering maps to the control map.

This summarizes the operation of the neuroplanner in non-hypothetical mode; i.e., in the mode where the current control state is a meaningful starting point for solutions to the steering goal. The hypothetical mode is used when the initial control state is unknown but an initial steering state is available instead. Given an initial and goal steering state, the neuroplanner finds a sequence of control states leading from any one of the control states associated with the initial steering state to any one of the control states associated with the goal steering state.

4 An Example Using Neuroplanners

In [Graf and LaLonde, 1988], we presented an application dealing with a robot arm — the system could be described as solving the "move the arm to where you look" problem in the presence of obstacles. Simulation results using a 2 degree of freedom arm were presented. In this section, we consider the inverse problem — that of controlling a simplified eyes/head/neck (EHN) system. Although the approach can be generalized to 3-d, we focus on the 2-dimensional model depicted in Figure 5 for simplicity. The problem might colloquially be described as the "look at your finger" problem; i.e., assuming that the arm has been moved to some arbitrary position in a workspace, the goal is to have the EHN system look at the tip of the arm. We assume the robot can already control its arm. Our task here is to provide it with the additional capability of controlling the EHN system. As depicted, the system has more degrees of freedom than necessary (if no obstacles are allowed). In particular, the eyes can rotate, the head can rotate, and the neck can rotate and stretch/shrink. There is a limit to the movement permitted by each component; e.g., 45° for the eyes, 30° for the head, 20° for the neck and a neck length of between one and two head widths.

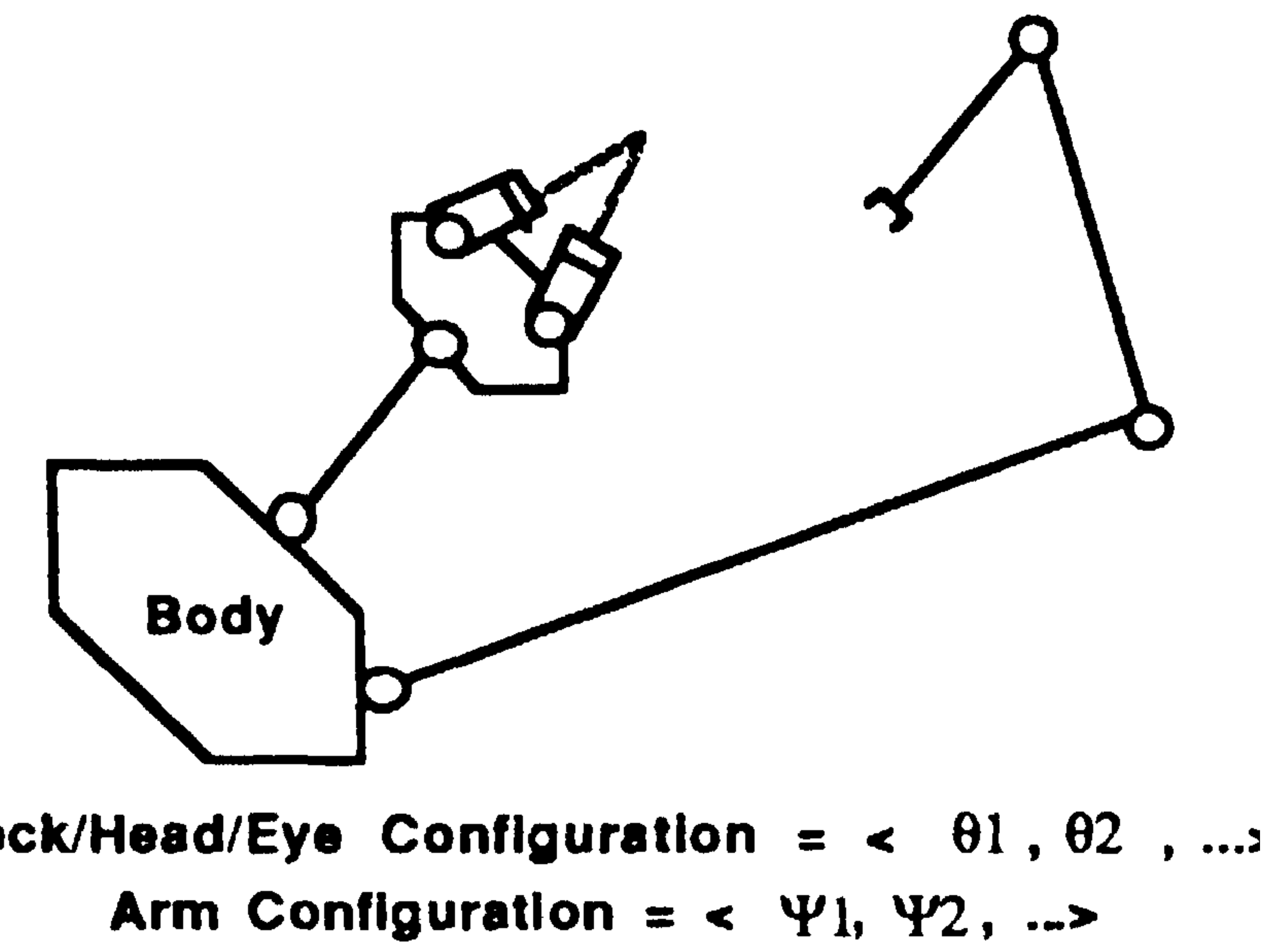


Figure 5: The EHN (Eyes/Head/Neck) System Coupled With An Arm

The EHN system is separate from the arm system and has access only to a vector which represents an arm configuration. The meaning of this vector is unknown to the EHN system; i.e., whether or not it consists of joint angles, polar coordinates, or cartesian coordinates of the arm tip is not germane. However, the system will have to learn the associations between arm configurations and the actual position of the tip of the arm.

To complicate the problem, we permit obstacles to be in the way so that the EHN system may have to "look

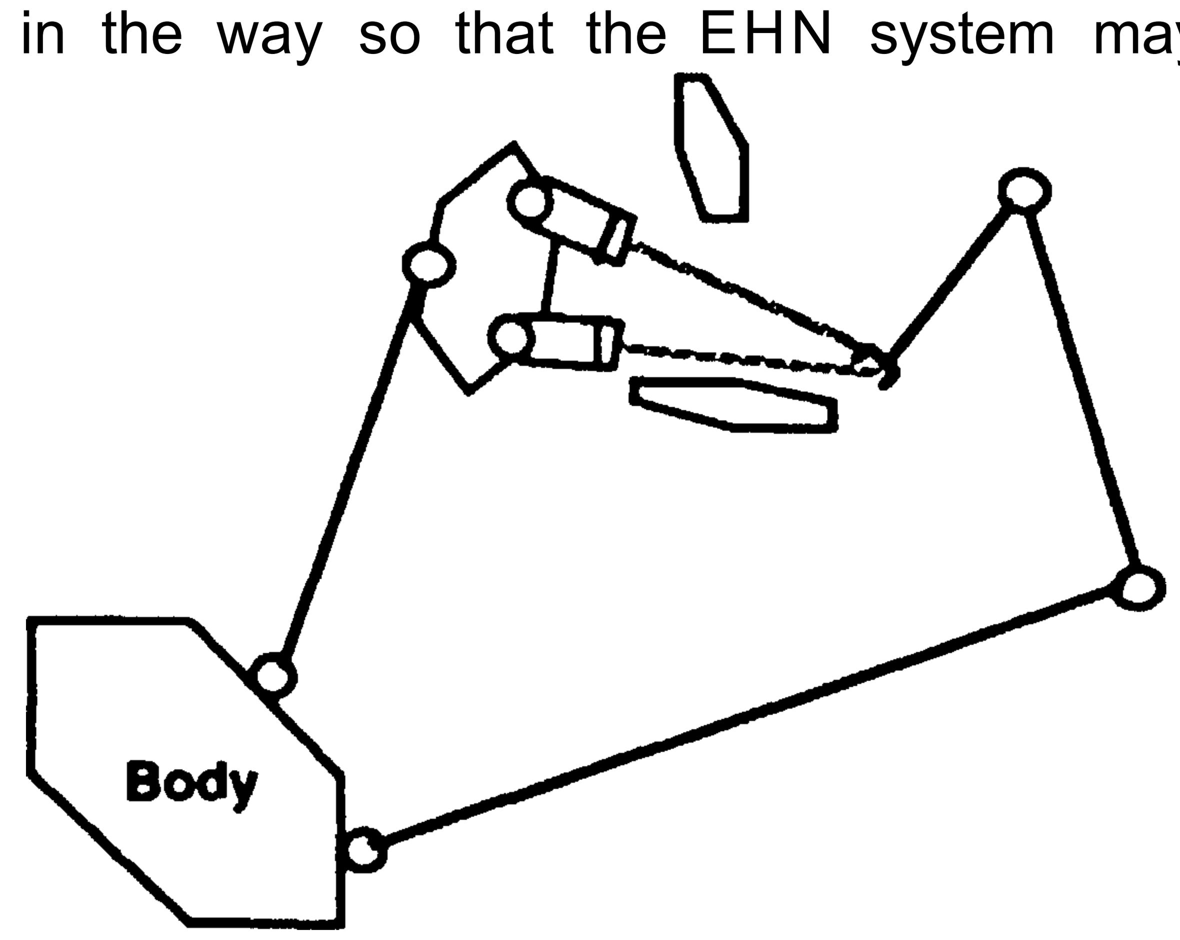


Figure 6: The EHN System Looking At Its Finger

around" the obstacles that are blocking the view (see Figure 6). If it isn't possible for the EHN system to look at the tip of the arm, no movement is initiated. Otherwise, the EHN system looks around whatever obstacles are blocking the view. The EHN system includes the dotted lines from the eyes to the point of focus. To have a clear line of sight, obstacles must not intersect either of these lines.

The EHN system goes through a learning phase without obstacles where it learns two things: associations between (1) EHN configurations and arm configurations and (2) obstacle constraints. After learning, obstacles can be added arbitrarily and the EHN system will endeavour to look at the tip of the arm where possible. Changing the obstacles does not require any additional learning. In fact, if the system is fast enough, it should be possible to track the tip of the arm as the obstacles change dynamically. However, we will not consider this aspect in this paper.

4.1 Mapping The Problem Onto Neuroplanner

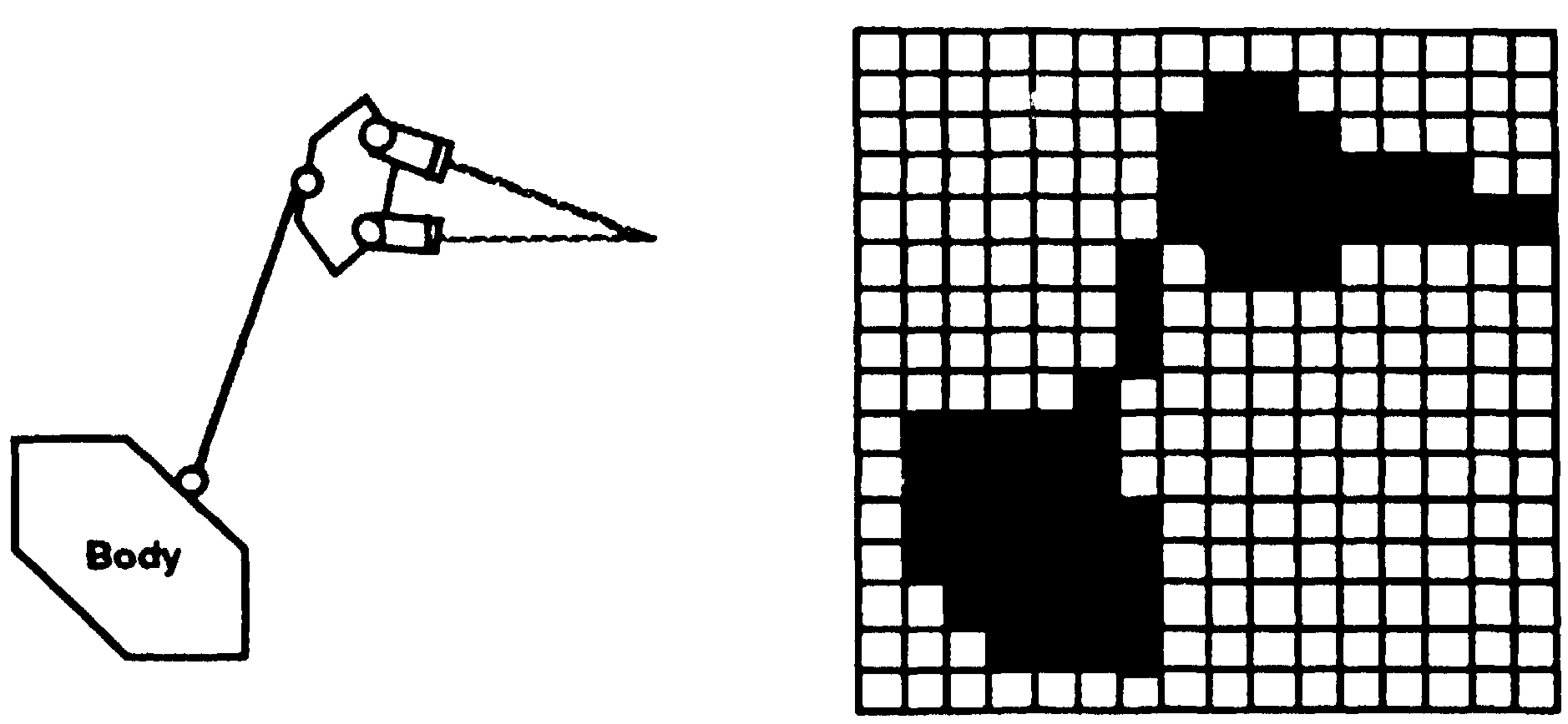
The neuroplanner control space consists of EHN configurations while the steering space consists of arm configurations. The arm is used to steer the EHN system. The arm configuration can be used to find an EHN configuration that focusses on the tip of the arm in the presence of obstacles. In general, the observation space is a 2-d pixclated map of the workspace which could be generated from a 3-d depth map such as might be produced by a sophisticated vision system. The self-image of the EHN system includes the workspace area taken up by the neck, head, eyes, and lines of sight up to the point of focus. It could be produced by the current EHN system if it is controlled and time-shared by some higher level system. For our simplified problem, the observation map can be produced by a high-contrast image (see Figure 7b) obtainable from a camera suspended above the robot workspace. Although we can easily obtain the image from a camera, we must keep in mind that it is really an internal mental image since such a camera defeats the purpose of the eyes/head/neck system in the first place; e.g., why peer around an obstacle if you can sec the arm tip from above.

4.2 Reviewing The Learning Stage

Once the self-organization phase is complete, the second phase proceeds to learn the boundaries of the legal control regions, the correlation between the observation space and the control space, and the correlation between the steering space and the control space. Note: no obstacles are present during the learning phase. The system must learn that obstacles cannot intersect with the self-image. This is achieved by associating each neuron in a workspace area of a self-image (observation neuron) with the corresponding EHN configuration control neuron. After learning is complete, actual obstacles may be added to the workspace. Each neuron which is activated by such an obstacle inhibits all EHN configurations associated with self-images that contained that neuron during learning.

The boundaries are delineated by the illegal control states provided when the components of the EHN configurations reach their limits of movement as described previously. The illegal control states prevent the path planner from planning paths that cross illegal state regions.

When learning the correlation between the observation space and the control space, each active neuron in the self-image is correlated with its associated EHN configuration



(a) (b)

Figure 7: The Observation Map For An Arbitrary Workarea

(control state) neuron. Later, when obstacles are introduced during the planning phase, the neurons in the obstacle regions will inhibit EHN configurations that intersect any one of these obstacles. This prevents the EHN system from colliding with obstacles or the arm as it moves to keep the tip of the end-effector in sight.

When learning the correlation between the steering space and the control space, the control space (the EHN system) is randomly moved around. The steering space (the arm) is then adjusted so that the tip can be seen by focal center of the two eyes (where possible). In that case, the association is made from the steering space to the control space (the arm to the eyes).

4.3 Reviewing The Planning Stage

During the planning stage, the arm (steering sensors) is moved to some arbitrary point. Subsequently, each workspace point occupied by an obstacle (this includes the arm) activates a neuron in the observation map, the current EHN configuration activates a neuron in the control map, and the current arm configuration activates a neuron in the steering map. Each arm configuration (steering state) corresponds to one or more target EHN configurations (control states) that can *see it.* Therefore the active steering state neuron will activate a set of final control state neurons. Similarly, each active observation neuron deactivates any active control state neurons that are *in collision situations.* All remaining active and uninhibited neurons are candidates for the collision-free EHN system movement plan. The plan is found by a standard path finding algorithm as described previously. In this way, knowledge about all possible solution movements is represented simultaneously by uninhibited neurons.

5 Conclusions and Future Work

We have reviewed a preliminary design for a new class of neural planners and an application that is the dual of that presented in [LaLonde and Graf, 1988]. Neuroplanners possess four important characteristics. First, they are applicable to a wide variety of domains. Second, they are capable of self-organization and learning. Third, they lend themselves to implementations in massively parallel hardware and are therefore potentially very fast. Fourth, goals are specifiable at a higher level than the control states of the system — this includes the ability to specify large sets of final control states as a single steering state.

The EHN system is unique in several respects. It will adjust itself with minimal movement to keep the tip of the arm in view (where possible) regardless of the obstacle arrangement. In particular, it will avoid colliding with obstacles or the arm. Changing the obstacles will simply cause the EHN system to re-adjust its position to maintain its view of the arm tip. This occurs without any additional learning. Although the behavior is seemingly intelligent (as seen from an outside observer), it nevertheless arises from a simple interaction among neurons in an appropriately organized network architecture.

References

[Fischlerand Firschein, 1987] Fischler, M.A. and Firschein, O. Readings in Computer Vision, Morgan Kaufmann Pub., 1987.

[Graf and LaLonde, 1988] Graf, D.H., LaLonde, W.R., A Neural Controller For Collision-Free Movement of General Robot Manipulators, Proc. IEEE First Intl. Conf. on Neural Networks, San Diego, Cal., July 1988.

[Grossberg and Kuperstein, 1986] Grossberg, S., Kuperstein, M., *Neural Dynamics of Adaptive Sensory-Motor Control,* North-Hoi land, 1986.

[Hopfield, 1982] Hopfield, J.J., Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, USA, 79, pp. 2554-2558, 1982.

[Kohonen, 1984] Kohonen, T., *Self-Organization and Associative Memory,* Springer-Verlag, 1984.

[Kohonen, 1986] Kohonen, T., Learning Vector Quantization for Pattern Recognition, Helsinki University of Technology, Report TKK-F-A601, November 1986.

[LaLonde and Graf, 1988] LaLonde, W.R. and Graf, D., Neuroplanners: Mechanisms for Subcognitive Control, Neural Networks and their Applications, Nimes, France, November 1988.

[Lippmann and Gold, 1987] Lippmann, R.P. and Gold, B. Neural-Net Classifiers Useful for Speech Recognition, Proc. IEEE First Intl. Conf. Neural Networks, San Diego, Cal., June 1987, vol. 4, pp. 417-426.

[Miller and Stout, 1985] Miller, R. and Stout, Q.F., Geometric Algorithms for Digitized Pictures on a Mesh-Connected Computer, IEEE Transactions on PAMI, vol. 7, no. 2, March 1985, pp. 216-228.

[Rumelhart and McClelland, 1986] Rumelhart, D.E. and McClelland, J.L. Parallel Distributed Processing, Vols 1 and 2, The MIT Press, 1986.

[Poizner *et aL,* 1987] Poizner, H., Klima, E.S, and Bellugi U. What the Hands Reveal about the Brain, MIT Press, 1987.