Topological Motion Planning based on Perceiving-Acting Cycle with Attention Mechanism for Robot Manipulators

TAKUMI MIYA[1], **TAKENORI OBO**[1], **NAOYUKI KUBOTA**[1]

[1]Department of mechanical systems engineering, Tokyo Metropolitan University, Tokyo, Japan
E-MAIL: miya-takumi@ed.tmu.ac.jp, t.obo@tmu.ac.jp, kubota@tmu.ac.jp

Abstract:

Robot manipulators performing assistive tasks in unstructured environments require both feasibility and applicability, especially when operating around humans in rehabilitation and healthcare settings. This study considers motion feasibility in configuration space from the perspective of affordance and effectivity in the perceiving-acting cycle. A Growing Neural Gas (GNG) method based on internal focus is proposed to construct a topological structure in configuration space for safe motion planning of manipulators. The perceiving-acting cycle is introduced as a sampling mechanism that dynamically defines focus regions based on collision and proximity information. Learning local covariance matrices in GNG improves the representation of configuration-space topology. Experimental results demonstrate that projecting the learned topology onto task space enables safe and efficient end-effector motion. These results suggest that the proposed method supports both feasibility and applicability in assistive and rehabilitation scenarios.

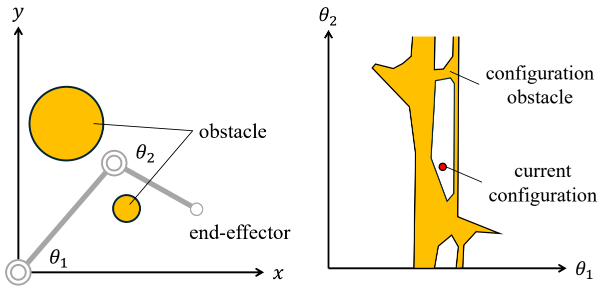
Keywords:

Manipulator; Motion Planning; Perceiving-Acting Cycle; Topological Structure;

1. Introduction

Robot manipulators are increasingly used in healthcare and rehabilitation systems, such as assistive robots and rehabilitation devices. Our laboratory has developed the Trailer Living Lab (TLL), a facility for testing assistive technologies in realistic environments. In these applications, safety and task feasibility are critical because robots operate in close proximity to patients. Therefore, motion planning must ensure not only path generation but also safe task execution.

The research on intelligent robotics has a long history. Various motion planning methods have been proposed [1]–[7]. Motion planning is the process of determining a sequence of intermediate postures from the manipulator's initial posture to its target posture. These methods can be broadly classified into two categories based on the planning space: task space and configuration space. The task space is the three-dimensional space in which the robot operates, whereas the configuration space is a multidimensional space whose coordinates are the manipulator's joint angles.



(a) Task space **(b)** Configuration space

FIGURE 1. Identical manipulator in different expressions

The perceiving-acting cycle [8] describes a cognitive process in which perception and action mutually influence each other for environmental adaptation. Some studies have applied this concept to robotics [9], [10]. In this study, we propose a method for manipulator motion planning based on the perceiving-acting cycle. We consider both applicability and feasibility as key factors for safety. The objective of this paper is to establish theoretical guarantees by jointly considering these factors in manipulator motion planning. Here, applicability refers to the ability to adapt to diverse environments and tasks, while feasibility refers to the ability to perform tasks while avoiding obstacles.

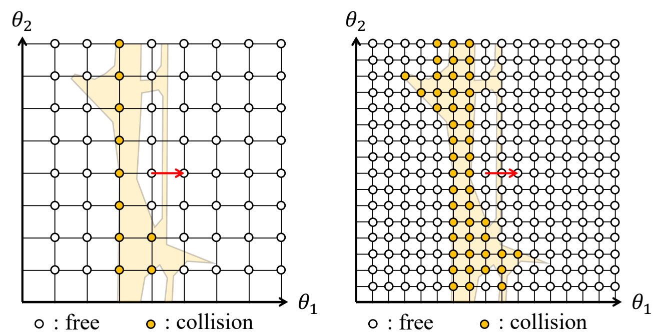
2 Related Works and Issues

In task-space, a common strategy is to design an end-effector trajectory and then generate intermediate postures that follow it while avoiding obstacles. Representative methods include potential-field-based optimization and inverse-kinematics extensions [1], as well as approaches that co-optimize trajectories and postures [2]. However, these methods are still exploratory and iterative, and they do not provide a theoretical criterion for determining task feasibility, especially in dynamic environments where collision-free plans may not exist.

Grid search discretizes the configuration space into cells and solves path planning as a graph search problem [3]. Although coarse discretization is computationally efficient, it cannot represent complex configuration obstacles accurately (Fig. 2a); finer discretization improves accuracy (Fig. 2b) but rapidly increases computational cost. Adaptive-resolution variants [4] mitigate this trade-off, yet quantization error still distorts the underlying topological structure.

PRM constructs a graph by sampling collision-free configurations and connecting neighboring nodes [5]. A path is then searched by adding start/goal nodes to the roadmap. However, edge generation is heuristic, so collision-free connectivity is not guaranteed, and dynamic environments require expensive reconstruction. Even with improved sampling strategies [11], redundant nodes and edges remain a practical concern.

RRT incrementally grows a tree in configuration space through random sampling and nearest-node expansion [6]. Bidirectional and extended variants improve search efficiency [7], and paths can be extracted by tracing parent links. Nevertheless, RRT-based methods cannot make end-effector trajectory design difficult, and cannot reliably determine feasibility.



(a) Coarse grid

(b) Fine grid

FIGURE 2. Wrong path due to quantization in configuration space

In healthcare environments, where obstacles and human body parts form complex configuration constraints, the inability to determine feasibility becomes a serious safety issue.

3 Perceiving-Acting Cycle

The perceiving-acting cycle is a concept in ecological psychology in which perception and action mutually influence each other to adapt to the environment [8]. Gibson introduced the concept of affordance, defined as the possibilities for action provided by the environment.

The affordance is specified by the attention based on a current posture in the cyclic process of perceiving and acting. In the perception, we take a suitable posture in order to reduce the uncertainty and prediction errors. On the other hand, effectivity is defined as the possibility of realizing action from the viewpoint of action. When we take a specific action, we take a suitable posture beforehand in order to reduce control errors. Furthermore, if the facing environment changes over time, the target state may also change. This means that the target state cannot be decided in advance. In such a dynamic environment, it is important to maintain an external focus in order to perform the intentional task. Here the term of attention is used for short-term perception to recognize a current situation, while the term of focus is used for relatively keeping long-term perception to perform an intentional task.

In manipulator motion planning, posture sampling in configuration space can be seen as virtual action and collision checking as virtual perception. Thus, affordance is represented by non-colliding postures, and effectivity by transitions between postures.

To discuss task feasibility, both affordance and effectivity must be represented, but their regions in configuration space are often highly complex and high-dimensional. Grid search suffers from the curse of dimensionality, and PRM/RRT may require many nodes and edges to capture such structure. Therefore, a flexible topological representation is needed.

Sampling strongly affects how efficiently such a structure is formed. To capture changes in affordance, samples should concentrate near collision boundaries, but uniform sampling is often inefficient and environment-dependent focus regions are hard to define in advance.

Therefore, we propose a sampling method that dynamically defines focus regions in configuration space according to collision information. Samples are first drawn without prior knowledge of the environment, and the sampling policy is then updated to increase the probability near collision boundaries.

Detail of the sampling method is described in Section 3.2.

3.1 Active Resampling GNG for Configuration Space Representation

Growing Neural Gas (GNG) is a type of self-organizing map that can adaptively learn the topological structure of input data distribution by incrementally adding and removing nodes and edges [12].

The proposed method modifies standard GNG to better represent affordance and effectivity in configuration space. The overall algorithmic flow is as follows. Each node v_i in the node set W has a reference vector $\boldsymbol{w}_i \in \mathbb{R}^n$, an accumulated error E_i, a label l_i, and a covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{n\times n}$. Edge e_{ij} connecting nodes v_i and v_j has a weight $a_{ij} \in [0,1]$ and a distribution-based weight $o_{ij} \in [0,1]$. Here, $a_{ij} = a_{ji}$ and $o_{ij} = o_{ji}$ are assumed.

1. Initialization: Generate two initial nodes using sampled vectors $\boldsymbol{x}_1, \boldsymbol{x}_2$ with corresponding labels y_1, y_2. Set them as reference vectors and labels of the nodes. Initialize accumulated errors to zero and covariance matrices as

$$\boldsymbol{\Sigma}_i = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 I \quad (i = 1, 2) \tag{1}$$

Connect the initial nodes with edge strength $a_{12} = 1$ and distribution-based strength $o_{12} = 0$.

2. Input Sampling: Sample an input vector $\boldsymbol{x}$ from the probability density function $p(\boldsymbol{x})$.

3. Winner Selection: Find the first and second winner nodes s_1 and s_2 using Euclidean distance:

$$s_1 = \arg\min_{i\in W} \|\boldsymbol{x} - \boldsymbol{w}_i\| \tag{2}$$

$$s_2 = \arg\min_{i\in W\setminus\{s_1\}} \|\boldsymbol{x} - \boldsymbol{w}_i\| \tag{3}$$

Compute distance vectors:

$$\boldsymbol{d}_{s_1} = \boldsymbol{x} - \boldsymbol{w}_{s_1}, \quad \boldsymbol{d}_{s_2} = \boldsymbol{x} - \boldsymbol{w}_{s_2} \tag{4}$$

4. Mahalanobis Distance Check:

$$D_M^2(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{d}_i^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{d}_i \quad (i = s_1, s_2) \tag{5}$$

If neither node satisfies

$$D_M(\boldsymbol{x}, \boldsymbol{w}_{s_1}) \le \rho \quad \text{and} \quad D_M(\boldsymbol{x}, \boldsymbol{w}_{s_2}) \le \rho \tag{6}$$

insert a new node initialized from the winner. If

$$D_M(\boldsymbol{x}, \boldsymbol{w}_{s_1}) \le \rho \tag{7}$$

update the label:

$$l_{s_1} \leftarrow l_{s_1} + \epsilon_1 \left(1 - \frac{D_M(\boldsymbol{x}, \boldsymbol{w}_{s_1})}{\rho}\right)(y - l_{s_1}) \tag{8}$$

5. Accumulated Error Update:

$$E_{s_1} \leftarrow E_{s_1} + D_M^2(\boldsymbol{x}, \boldsymbol{w}_{s_1}) \tag{9}$$

6. Edge Update: If no edge exists between s_1 and s_2, create one with $a_{s_1 s_2} = 1$ and $o_{s_1 s_2} = 0$. If both winners satisfy

$$D_M(\boldsymbol{x}, \boldsymbol{w}_{s_1}) \le \kappa\rho \quad \text{and} \quad D_M(\boldsymbol{x}, \boldsymbol{w}_{s_2}) \le \kappa\rho \tag{10}$$

set the distribution-based weight $o_{s_1 s_2} = 1$.

7. Edge Decay: For neighbors $j \in \mathcal{N}(s_1) \setminus \{s_2\}$:

$$a_{s_1 j} \leftarrow (1 - \epsilon_2) a_{s_1 j}, \quad o_{s_1 j} \leftarrow (1 - \epsilon_2) o_{s_1 j} \tag{11}$$

Remove edges if $a_{s_1 j} < \theta_a$.

8. Reference and Covariance Update:

$$\boldsymbol{w}_{s_1} \leftarrow \boldsymbol{w}_{s_1} + \epsilon_1 \boldsymbol{d}_{s_1} \tag{12}$$

$$\boldsymbol{w}_j \leftarrow \boldsymbol{w}_j + \epsilon_2 a_{s_1 j}(\boldsymbol{x} - \boldsymbol{w}_j) \tag{13}$$

Update covariance:

$$\boldsymbol{\Sigma}_{s_1} \leftarrow \boldsymbol{\Sigma}_{s_1} + \epsilon_1 (\boldsymbol{d}_{s_1} \boldsymbol{d}_{s_1}^T - \boldsymbol{\Sigma}_{s_1}) \tag{14}$$

9. Node Merging and Insertion (every λ steps):

   a. Merge Selection: Select node i randomly and find the nearest neighbor:

   $$j = \arg\min_{j\in\mathcal{N}(i)} \|\boldsymbol{w}_i - \boldsymbol{w}_j\| \tag{15}$$

   If labels match the most frequent label, merge nodes.

   b. Merge Parameters:

   $$\boldsymbol{w}_m = \frac{\boldsymbol{w}_i + \boldsymbol{w}_j}{2}, \quad E_m = \frac{E_i + E_j}{2} \tag{16}$$

   $$\boldsymbol{\Sigma}_m = \frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2} + \frac{\delta_i \delta_i^T + \delta_j \delta_j^T}{2} \tag{17}$$

   c. Node Insertion:

   $$f = \arg\max_{j\in\mathcal{N}(q)} E_j \tag{18}$$

   $$\boldsymbol{w}_r = \frac{\boldsymbol{w}_q + \boldsymbol{w}_f}{2} \tag{19}$$

   d. Reconnect edges:

   $$a_{qr} = a_{fr} = a_{qf}, \quad o_{qr} = o_{fr} = o_{qf} \tag{20}$$

e. Decay parent errors:

$$E_q \leftarrow (1-\alpha)E_q, \quad E_f \leftarrow (1-\alpha)E_f \tag{21}$$

f. Initialize new node:

$$E_r = \frac{E_q + E_f}{2}, \quad \boldsymbol{\Sigma}_r = \frac{\boldsymbol{\Sigma}_q + \boldsymbol{\Sigma}_f}{2} \tag{22}$$

10. Global Error Decay:

$$E_i \leftarrow (1-\beta)E_i \quad (\forall i \in W) \tag{23}$$

11. Termination: Stop when the maximum number of learning cycles is reached; otherwise return to Step 2.

Using covariance matrices to define node receptive fields enables local shape adaptation. New nodes are created when inputs fall outside the receptive field, introducing ART-like adaptivity [13].

In standard GNG, accumulated error is based on squared Euclidean distance, whereas our method uses squared Mahalanobis distance. This increases sensitivity to local distribution shape and promotes insertion in underrepresented regions. Standard GNG is a special case when all covariance matrices are isotropic and identical.

Node integration is inspired by GNG-U [14]. Instead of deleting nodes, redundant neighboring nodes with highly pure labels are merged to preserve learned topology.

3.2 Sampling Method with Focus Regions

In the proposed method, sampling is performed according to the probability density function $p(\boldsymbol{x})$ defined as:

$$p(\boldsymbol{x}) = (1-\xi)p_u(\boldsymbol{x}) + \xi p_f(\boldsymbol{x}) \tag{24}$$

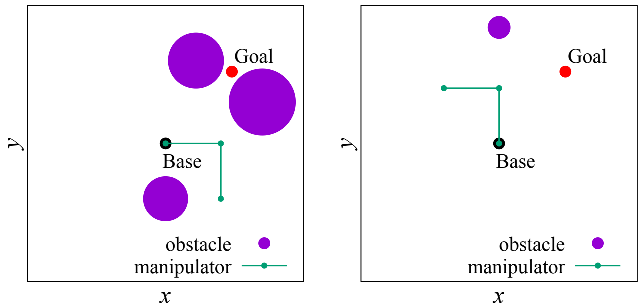
Here, $p_u(\boldsymbol{x})$ is the uniform distribution over the configuration space and $p_f(\boldsymbol{x})$ is the focus-region distribution. The parameter $\xi \in [0,1]$ controls the balance between exploration and exploitation. This idea is related to Fastron [15], which defines focus regions from support vectors. In our method, node set $\mathcal{B}$ is defined as:

$$\mathcal{B} = \{v_i \in W \mid l_i \neq 1.0\} \tag{25}$$

Here, l_i is the label of node v_i, and $p_f(\boldsymbol{x})$ is modeled as a Gaussian mixture over nodes in $\mathcal{B}$. Each component is weighted by the trace of its covariance matrix, and the covariance is scaled by η to avoid excessive concentration.

$$p_f(\boldsymbol{x}) = \sum_{v_i \in \mathcal{B}} \frac{\mathrm{tr}(\boldsymbol{\Sigma}_i)}{\sum_{v_j \in \mathcal{B}} \mathrm{tr}(\boldsymbol{\Sigma}_j)} \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{w}_i, \eta\boldsymbol{\Sigma}_i) \tag{26}$$

Here, $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{w}_i, \eta\boldsymbol{\Sigma}_i)$ is the normal distribution with mean $\boldsymbol{w}_i$ and covariance matrix $\eta\boldsymbol{\Sigma}_i$.



(a) three obstacles **(b)** one obstacle

FIGURE 3. Task-space environments used in the experiment

4 Experimental Results

To verify the proposed method, experiments were conducted using a planar 2-link manipulator with obstacles (Fig. 1). Each link length was 1.0, and joint angles were limited to $[-\pi, \pi]$. As described in Section 3, sampling is treated as a virtual action to obtain collision-related perceptual information. As a collision detection method, we used forward kinematics to determine the positions of the links and check for collisions with circular obstacles. Task-space environments are shown in Fig. 3a and Fig. 3b.

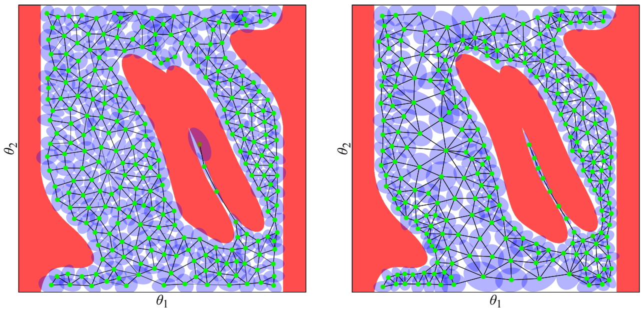
4.1 Graph Structuring using Active Resampling GNG

We compared topological structures generated by active resampling GNG with and without focus regions, using 256 nodes in both cases. The results are shown in Fig. 4.

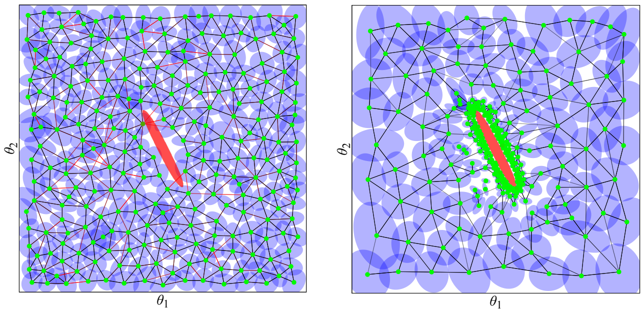
Without focus regions (Fig. 4a), nodes are uniformly distributed over the configuration space, failing to represent narrow passages accurately. In contrast, with focus regions (Fig. 4b), narrow passages are captured as flattened ellipses formed by nodes with anisotropic covariance matrices. Similar behavior is observed in the single-obstacle case. In Fig. 4c, red edges cross configuration obstacles, indicating that standard GNG edges, which ignore local distributions, fail to preserve the topology of the configuration space.

4.2 Motion Planning using Constructed Topological Structure

Using the constructed topological structures, motion planning was performed for reaching tasks. The start posture and target position were set as shown in Fig. 3a. Paths were searched using the A* algorithm from the start node to the



(a) three obstacles without focus **(b)** three obstacles with focus



(c) one obstacle without focus **(d)** one obstacle with focus

FIGURE 4. Topological structures using active resampling GNG

reachable node closest to the target. The A* cost function is defined as

$$f(n) = g(n) + h(n) \tag{27}$$

$$g(n) = \sum_{i=1}^{k} \left(c_i^{risk} \|\boldsymbol{\mu}_{n_i} - \boldsymbol{\mu}_{n_{i-1}}\| \right) \tag{28}$$

$$c_i^{risk} = 1 + \frac{\gamma}{h_i^{\partial} + 0.1} \tag{29}$$

$$h(n) = \|\boldsymbol{\mu}_n - \boldsymbol{p}_{target}\| \tag{30}$$

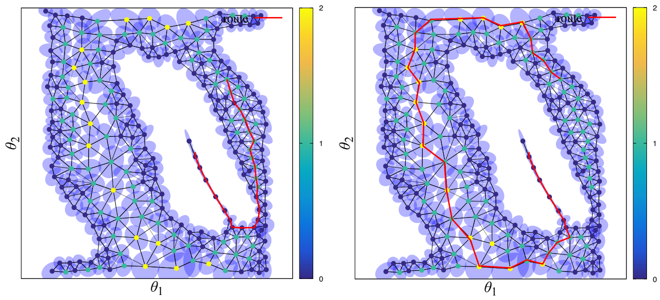
Here, c_i^{risk} penalizes nodes near obstacles, where γ controls the risk weight and h_i^{∂} denotes the hop count to the nearest boundary node set $\mathcal{B}$ defined in Section 3.2.

Each node represents a local distribution described by a covariance matrix. Using the Jacobian matrix, the local distribution is approximately transformed into task space:

$$\boldsymbol{\Sigma}_i^{task} = \boldsymbol{J}_i \boldsymbol{\Sigma}_i \boldsymbol{J}_i^T \tag{31}$$

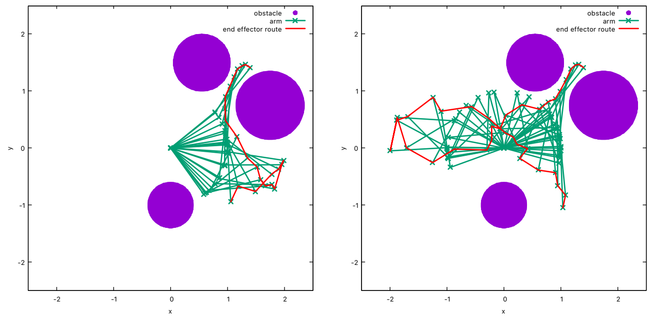
This corresponds to an extension of manipulability ellipsoids [16] considering local distribution shape. Reachable nodes are then identified using the Mahalanobis distance:

$$(\boldsymbol{p}_{target} - \boldsymbol{\mu}_i)^T (\boldsymbol{\Sigma}_i^{task})^{-1} (\boldsymbol{p}_{target} - \boldsymbol{\mu}_i) \leq \kappa^2 \rho^2 \tag{32}$$



(a) $\gamma = 0.0001$ **(b)** $\gamma = 10$

FIGURE 5. Paths in configuration space with different prudence



(a) $\gamma = 0.0001$ **(b)** $\gamma = 10$

FIGURE 6. Motion in task space with different prudence

Because this transformation assumes small displacements, reachability is evaluated in two stages: candidate nodes are first selected using the Mahalanobis distance, and then validated using incremental motion based on the Jacobian pseudo-inverse.

Planned trajectories in configuration space are shown in Fig. 5a, and the corresponding end-effector trajectories in task space are shown in Fig. 6a. These results demonstrate that the proposed method generates obstacle-avoiding trajectories by utilizing the constructed topology.

When the prudence parameter γ is small (0.0001), trajectories pass close to obstacles. In contrast, larger values ($\gamma = 10$) produce safer paths with larger clearance, as shown in Fig. 5b. This indicates that γ controls the trade-off between path optimality and safety.

Overall, the results demonstrate that the proposed method enables motion planning by explicitly utilizing reachability (effectivity) represented in configuration space.

5. Conclusions

In this paper, we proposed a method to represent the topological structure of configuration space for manipulators to perform motion planning. To represent motion feasibility, the method captures affordance and effectivity in configuration space through a perceiving-action cycle. Experimental results using active resampling GNG showed that the proposed method efficiently captured key topological features such as narrow passages. Rejection of unsafe edges based on local distribution information improved motion planning accuracy. Furthermore, motion planning experiments demonstrated that the prudence parameter controls the trade-off between path optimality and safety margin. These results indicate that the proposed method enables feasibility-aware motion planning for healthcare robots, with adjustable behavior based on task requirements. Future work will extend the method to high-dimensional manipulators and more complex environments.

Acknowledgements

This work is partially supported by Japan Science and Technology Agency (JST), Moonshot R&D, under grant number JPMJMS2034.

References

[1] N. Baba and N. Kubota, "Collision avoidance planning of a robot manipulator by using genetic algorithm. a consideration for the problem in which moving obstacles and/or several robots are included in the workspace," in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, 1994, pp. 714–719 vol.2.

[2] N. Kubota, T. Arakawa, and T. Fukuda, "Hierarchical trajectory planning of redundant manipulators with structured intelligence," *Advanced Robotics*, vol. 12, no. 3, pp. 209–225, 1997.

[3] T. Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 224–238, 1987.

[4] B. Faverjon, "Obstacle avoidance using an octree in the configuration space of a manipulator," in *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, vol. 1, 1984, pp. 504–512.

[5] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[6] S. LaValle, "Rapidly-exploring random trees : A new tool for path planning," *Computer Science Dept. Oct.*, vol. 98, no. 11, 1998.

[7] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. 2, 2000, pp. 995–1001.

[8] J. J. Gibson, *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979.

[9] N. Kubota, "Perception-based robotics based on perceiving-acting cycle with modular neural networks," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 1, 2002, pp. 477–482 vol.1.

[10] F. Ardilla, M. Yani, A. A. Saputra, W. H. Chin, and N. Kubota, "Topological mapping based on perceiving-acting cycle in sharing cognitive environments for robot partners," *Communications in Information and Systems*, vol. 22, no. 4, pp. 431–458, 2022.

[11] R. Geraerts and M. Overmars, *A Comparative Study of Probabilistic Roadmap Planners*. Springer Berlin Heidelberg, 11 2003, vol. 7, pp. 43–58.

[12] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. MIT Press, 1994, pp. 625–632.

[13] P. Carpenter, M. Just, and P. Shell, "What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test," *Psychological Review*, vol. 94, no. 3, pp. 404–431, 1987.

[14] B. Fritzke, "Be busy and unique — or be history—the utility criterion for removing units in self-organizing networks," in *KI-99: Advances in Artificial Intelligence*, W. Burgard, A. B. Cremers, and T. Cristaller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 207–218.

[15] N. Das, N. Gupta, and M. C. Yip, "Fastron: An online learning-based model and active learning strategy for proxy collision detection," in *Conference on Robot Learning*, 2017.

[16] T. Yoshikawa, "Manipulability and redundancy control of robotic mechanisms," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 1004–1009.