

PERANCANGAN JARINGAN SARAF TIRUAN HIERARCHICAL EXTENDED KOHONEN MAP UNTUK WAHANA GERAK MANDIRI PENGHINDAR TABRAKAN

INDRA HARTARTO TAMBUNAN

Lecturer Of Electrical Engineering; Faculty Of Informatics And Electrical Engineering
Del Institute Of Technology, Laguboti, Tobasa, Sumatera Utara
Indra.Tambunan@Del.Ac.Id

ABSTRACT

This paper describes a neural network Hierarchical Extended Kohonen Map design for an obstacle avoidance of autonomous vehicle. Hierarchical Extended Kohonen Map is developed from the principle of Extended Kohonen Map. A grid of competitive layer (sub-net) is added to every cluster of super-net competitive layer of Extended Kohonen Map. This neural network consists of 16 clusters of super-net competitive layer (4x4 neuron) and 10 clusters of sub-net competitive layer. The result shows that the proposed design of Hierarchical Extended Kohonen Map has responded well to vector input test.

Keywords: Neural Network; Kohonen Map; Obstacle Avoidance; Autonomous Vehicle.

PENDAHULUAN

Perkembangan penelitian tentang wahana gerak mandiri menargetkan pengembangan kecerdasan perilaku dan pembentukan sistem yang sesuai dengan kecerdasan perilaku tersebut [1][2]. Metode yang umum dilakukan adalah dengan model-based system dan sensor-based system [3][4]. Dalam implementasinya, wahana gerak mandiri model-based system bergerak berdasarkan model yang sudah ditetapkan melalui program yang dijalankan. Model-based system ini disebut juga sebagai supervised-learning planning motion. Sedangkan dalam sensor-based system, wahana gerak mandiri bergerak berdasarkan input perceptron yang diterima melalui sensor. Metode ini sering disebut sebagai unsupervised-learning.

Salah satu objektif wahana gerak mandiri yang sering digunakan dalam implementasi model-based system adalah menghindari tabrakan (obstacle avoidance). Telah ada beberapa bentuk algoritma yang telah digunakan untuk kasus ini antara lain fuzzy controller [5], genetic algorithm [6], jaringan saraf tiruan (neural network) dengan berbagai pendekatan [7][8][9].

Pengembangan wahana gerak mandiri sensor-based system belum banyak digunakan karena kerumitan perhitungan yang memaksa sistem untuk melakukan proses pembelajaran secara mandiri.

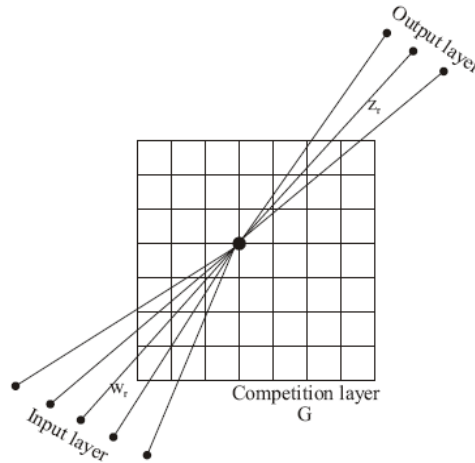
Metode ini telah dikembangkan melalui pengembangan jaringan saraf tiruan (JST) Kohonen Map [10] dan Extend Kohonen Map [11]. Akan tetapi, masih terdapat beberapa kelemahan dari JST ini karena keterbatasan neuron perceptron untuk melakukan analisis perhitungan.

Penelitian ini bertujuan untuk merancang sebuah jaringan saraf tiruan Hierarchical Extended Kohonen Map yang dapat digunakan sebagai kendali untuk wahana gerak mandiri dalam menghindari tabrakan. Hierarchical Extended Kohonen Map merupakan pengembangan Extended Kohonen Map, sehingga diharapkan mampu mengatasi kelemahan yang diperoleh dari metode tersebut.

JARINGAN SARAF TIRUAN EXTENDED KOHONEN MAP (EKM)

Jaringan saraf tiruan Extended Kohonen Map merupakan pengembangan dari SOFM (Kohonen neural networks). Dari sudut pandang arsitekturnya, EKM merupakan SOFM yang diperluas dengan menambahkan untuk setiap neuron r pada layer kompetisi G sebuah referensi vektor (bobot) output z_r . Nilai z_r ini digunakan untuk menyimpan nilai output dari neuron r (Gbr. 1).

Pengembangan ini memungkinkan proses pembelajaran pada jaringan menggunakan metode baik supervised (model-based system) maupun unsupervised learning (sensor-based system).



Gbr. 1 Arsitektur EKM

Proses pembelajaran pada EKM ini dilakukan sebagai berikut: ketika input vektor x diterima oleh jaringan melalui perceptron, pada competitive layer G terjadi kompetisi antar neuron r yang melibatkan perhitungan jarak antara input vektor x dengan referensi vektor (bobot) input w_r seperti ditunjukkan pada (1).

$$d(x, w_r) = \|x - w_r\|^2 \tag{1}$$

Neuron s dengan referensi vektor input w_s , memiliki jarak paling dekat dengan vektor input x (yaitu dengan jarak d yang paling kecil). Sehingga neuron s merupakan pemenang kompetisi di competitive layer G dengan bobot seperti pada (2).

$$s = \arg \min_r d(x, w_r) \tag{2}$$

Referensi vektor output z_s yang bersesuaian dengan neuron s ditetapkan sebagai output dari jaringan untuk menanggapi input vektor x . Proses perubahan bobot terjadi pada neuron s dan neuron tetangganya pada competitive layer G sesuai dengan (3).

$$w_s^{new} = w_s^{old} + \alpha(t)[x - w_s^{old}] \tag{3}$$

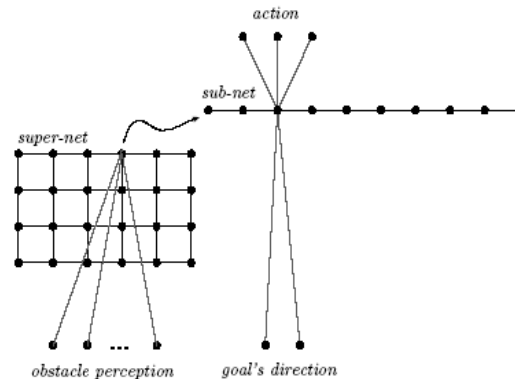
dengan $\alpha(t)$ adalah learning rate dari grid jaringan saraf tiruan.

JARINGAN SARAF TIRUAN HIERARCHICAL EXTENDED KOHONEN MAP

HEKM merupakan pengembangan dari EKM dengan menambah satu grid (sub-net grid H) pada super-net grid (grid G) untuk setiap cluster grid. Penambahan grid pada jaringan berupa flat 2 dimensi merupakan cara agar, seperti halnya pada super-net grid, perhitungan pada sub-net grid juga dapat dilakukan. Sehingga untuk setiap perhitungan goal direction yang berbeda tidak perlu dilakukan pengulangan. Dengan proses ini, diharapkan pendistribusian data hasil adaptasi dari SOFM dapat dilakukan secara sekuensial.

Jadi dengan penambahan suatu sub-net grid H pada setiap neuron pada grid G , maka akan ada 2 perbedaan yang terjadi antara lain yakni jaringan akan memiliki 2 buah learning rate dan 2 buah neighborhood parameters. Karena

untuk setiap grid akan memiliki 1 buah learning rate dan 1 buah neighborhood parameters. Proses pembelajaran pada HEKM akan mengikuti proses keseluruhan dari proses pembelajaran yang dilakukan pada EKM.



Gbr. 2. Arsitektur HEKM

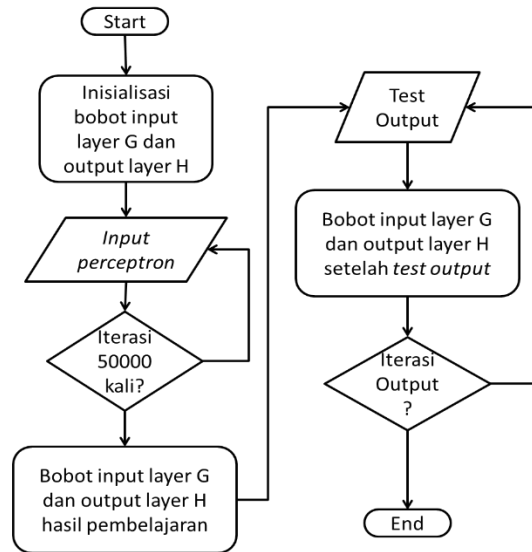
Obstacle perception merupakan input layer dari super-net grid G, sedangkan output layer dari super-net grid menjadi persepsi arah tujuan (goal's direction) dari input layer sub-net grid. Output layer dari sub-net grid H merepresentasikan output action dari wahana gerak mandiri.

PERANCANGAN JST HEKM

Pengembangan JST Extended Kohonen Map (EKM) menjadi Hierarchical Extended Kohonen Map (HEKM) dilakukan dengan penambahan grid pada neuron super-net (grid G). Komposisi jaringan HEKM yang dibentuk adalah:

- Satu super-net (grid G) dengan komposisi 4x4 neuron dengan maksimal cluster yang terbentuk pada super-net sebanyak 16 cluster.
- 8 sub-net (grid H) dengan komposisi 1x10 neuron untuk setiap sub-net, sehingga jumlah maksimal cluster yang terbentuk pada sub-net adalah $8 \times 1 \times 10 = 80$ cluster.
- Dengan penambahan grid maka pada HEKM ini terdapat 9 competitive layer (1 competitive layer grid G dan 8 competitive layer grid H)
- Jaringan saraf tiruan HEKM memiliki 2 learning rate (super-net dan sub-net) dan 2 buah neighborhood parameter (super-net dan sub-net).

Dalam simulasi program yang dilakukan inisialisasi dari learning rate untuk super-net dan sub-net ditetapkan dengan nilai yang sama yakni 0,28. Diagram alir dari simulasi perancangan jaringan saraf tiruan HEKM dibuat di Gbr. 3.



Gbr. 3. Diagram alir dari simulasi JST HEKM

Bobot setiap input layer dan output layer untuk setiap grid jaringan ditetapkan random dengan nilai antara 0 dan 1. Hasil inisialisasi dari bobot untuk input layer grid G dan grid H terlihat di Gbr. 4 dan Gbr. 5. Perubahan terhadap bobot tersebut akan terlihat setelah dilakukan iterasi hingga 50000 kali. Dalam simulasi ini ditetapkan 5 komponen input layer G sebagai representasi input dari sensor dan 3 komponen sebagai hasil referensi output layer grid G. Kemudian 3 komponen input layer H sebagai representasi input goal direction dan 2 komponen output layer sebagai representasi output gerak motor kiri dan kanan.

```

C:\Program Files\Microsoft Visual Studio\Work>main

Nilai w_G :
0.9820 0.4795 0.6494 0.7832 0.3487
0.6813 0.2667 0.5914 0.8052 0.0699
0.2847 0.5904 0.6513 0.3117 0.6284
0.9231 0.7572 0.2777 0.4675 0.9091

Nilai z_G
0.8472 0.4076 0.7133
0.2887 0.4975 0.9650
0.8452 0.1499 0.8981
0.5285 0.4835 0.6334

0.7802 0.4735 0.6633 0.4406 0.2168
0.8821 0.3177 0.6354 0.6224 0.7233
0.5225 0.2428 0.1389 0.0440 0.1748
0.6254 0.4036 0.6224 0.4885 0.4545

0.8072 0.3307 0.7153
0.9481 0.5395 0.3477
0.3656 0.2997 0.8082
0.0769 0.7902 0.5085

0.9800 0.4585 0.8821 0.8661 0.2507
0.5664 0.2807 0.6294 0.5944 0.5854
0.2857 0.2258 0.0679 0.0859 0.3626
0.8691 0.4935 0.1159 0.7882 0.8581

0.0879 0.6154 0.6723
0.0420 0.4166 0.3497
0.4156 0.2008 0.4525
0.7403 0.9041 0.7992

0.4126 0.8252 0.4106 0.9411 0.0030
0.3966 0.1249 0.5584 0.7283 0.7502
0.1888 0.2747 0.9560 0.2997 0.7922
0.4875 0.4206 0.7712 0.0480 0.8821

0.6863 0.6853 0.0390
0.9351 0.1469 0.1159
0.9970 0.6513 0.9261
0.0430 0.0679 0.4426

Nilai alpha_G :
0.2800 0.2800 0.2800 0.2800
0.2800 0.2800 0.2800 0.2800
0.2800 0.2800 0.2800 0.2800
0.2800 0.2800 0.2800 0.2800

Nilai indeks alpha_G
3 3 3 3
3 3 3 3
3 3 3 3
3 3 3 3

Nilai R_G :
3 3 3
3 3 3
3 3 3
3 3 3

Nilai indeks RG :
0 0 0
0 0 0
0 0 0
0 0 0

Maximum perhitungan : 0.
Enter file name :
hekml.txt
    
```

Gbr. 4. Inisialisasi bobot awal jaringan G

```

=====Hasil Inialisasi H - 0:=====
Nilai w_H :                               Nilai z_H :
0.9820 0.4795 0.6494                       0.7832 0.3487
0.8472 0.4076 0.7133                       0.6813 0.2667
0.5914 0.8052 0.0699                       0.2887 0.4975
0.9650 0.2847 0.5904                       0.6513 0.3117
0.6284 0.8452 0.1499                       0.8981 0.9231
0.7572 0.2777 0.4675                       0.9091 0.5285
0.4835 0.6334 0.7802                       0.4735 0.6633
0.4406 0.2168 0.8072                       0.3307 0.7153
0.8821 0.3177 0.6354                       0.6224 0.7233
0.9481 0.5395 0.3477                       0.5225 0.2428

alpha_H :
0.2800 0.2800 0.2800 0.2800 0.2800 0.2800 0.2800 0.2800 0.2800

Nilai R_H :
3      3      3      3      3      3      3      3      3

Nilai indeks alpha_H :
0      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0

Nilai indeks RH :
0      0      0      0      0      0      0      0      0

Maximum perhitungan : 0.

Clustering Result:

Neuron[0][0]:
Neuron[0][1]:
Neuron[0][2]:
Neuron[0][3]:
Neuron[0][4]:
Neuron[0][5]: 0 1 2 3 4 5 6 7
Neuron[0][6]:
Neuron[0][7]:
Neuron[0][8]:
Neuron[0][9]:

Mapping ResultH:
. . . . . $ . . . .
    
```

Gbr. 5. Inialisasi bobot awal jaringan H

HASIL SIMULASI

Pengujian hasil rancangan JST ini dilakukan dengan memberikan input file yang berisi kumpulan vektor input yang terdiri dari 8 buah komponen (.txt). Lima komponen pertama merupakan referensi input G dari sensor dan 3 komponen terakhir merupakan persepsi output G. Setelah itu, jaringan akan mengalami iterasi belajar sehingga diperoleh hasil bobot baru dan hasil clustering baru, terlihat di Gbr. 6.

```

=====Hasil Learning G=====
Nilai w_G :                               Nilai z_G
1.0000 1.0000 0.0000 0.0000 0.0000       1.0000 0.0000 0.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000

1.0000 1.0000 0.0000 0.0000 0.0000       1.0000 0.0000 0.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000

1.0000 1.0000 0.0000 0.0000 0.0000       1.0000 0.0000 0.0000
1.0000 1.0000 0.0000 0.0000 0.0000       1.0000 0.0000 0.0000
0.9339 0.0005 0.4595 0.2324 0.6793         0.9339 0.4595 0.6793
1.0000 1.0000 0.0000 0.0000 0.0000       1.0000 0.0000 0.0000

0.9504 0.0317 0.0321 0.0779 0.3009         0.9504 0.0321 0.3009
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000
0.0000 1.0000 1.0000 1.0000 1.0000       0.0000 1.0000 1.0000

Nilai alpha_G :                           Nilai indeks alpha_G
0.0000 0.0000 0.0374 0.0000               0      0      0
0.0024 0.0000 0.0000 0.0000               0      0      0
0.0000 0.1169 0.0000 0.0000               0      1      0
0.0000 0.0000 0.0000 0.0000               0      0      0

Nilai R_G :                               Nilai indeks RG :
0      0      0                               3582 3581 1733 2504
0      0      0                               1983 2261 3711 2131
0      1      0                               2620 1165 6959 4032
0      0      0                               4187 3416 2949 3186

Maximum perhitungan : 50000.
    
```

Gbr. 6. Bobot hasil pembelajaran (learning) jaringan G

Hasil output G akan dikonversi menjadi bilangan integer sebagai penunjuk membuka text file sebanyak 8 file yang berisikan goal direction dari H serta output grid H. hasil dari proses pembelajaran jaringan H ditunjukkan pada Gbr. 7.

```

=====Hasil Learning H=====
Hasil learning H - 0:
Nilai w_H :          Nilai z_H :
0.0000 0.5399 0.0000 0.5399 1.0000
0.4518 0.0000 1.0000 1.0000 0.5482
0.0000 1.0000 1.0000 0.0000 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000
1.0000 1.0000 0.0000 1.0000 1.0000
1.0000 1.0000 0.0000 1.0000 1.0000
1.0000 1.0000 0.0000 1.0000 1.0000
1.0000 0.0000 0.0000 0.0000 1.0000
0.9789 0.0011 0.0000 0.0011 1.0000

alpha_H :
0.0000 0.0000 0.0000 0.1057 0.0000 0.1036 0.0000 0.0981 0.0000 0.0906

Nilai R_H :
0      0      0      1      0      1      0      1      0      0

Nilai indeks alpha_H :
3690 3852 2548 1245 2378 1260 2319 1299 2556 1353
Nilai indeks RH :
3690 3852 2548 1245 2378 1260 2319 1299 2556 1353

Maximum perhitungan : 22500.

Clustering Result:
Neuron[0][0]: 0 3
Neuron[0][1]: 1 6
Neuron[0][2]: 7
Neuron[0][3]: 4
Neuron[0][5]: 1
Neuron[0][6]: 5
Neuron[0][7]: 2
Neuron[0][8]: 2
Neuron[0][9]: 1

Mapping ResultH:
$ $ $ . $ . $ . $ .

```

Gbr. 7. Bobot hasil pembelajaran (learning) jaringan H

Pengujian akan dilanjutkan melakukan uji jaringan (network test) dengan memasukkan nilai input sensor dan goal direction secara bersamaan sebagai input jaringan. Hasil yang diperoleh ditunjukkan dalam Gbr. 8.

```

Quit?
Test Networks :
Data input vektor (8digit):
0 0 1 0 1 0 1 0

G :
idxWinGx : 2, idxWinGy : 2
output network G : 1 0 1

H :
idxWinH : 6
output network H : 0 1

Quit?
Test Networks :
Data input vektor (8digit):
1 1 1 0 1 0 1 0

G :
idxWinGx : 2, idxWinGy : 2
output network G : 1 0 1

H :
idxWinH : 6
output network H : 0 1

Quit?
Test Networks :
Data input vektor (8digit):
0 0 0 1 0 1 1 1

G :
idxWinGx : 3, idxWinGy : 0
output network G : 1 0 0

H :
idxWinH : 2
output network H : 1 0

Quit?
Test Networks :
Data input vektor (8digit):
0 0 1 1 1 0 1 1

G :
idxWinGx : 1, idxWinGy : 2
output network G : 0 1 1

H :
idxWinH : 2
output network H : 0 1

```

Gbr. 8. Test hasil simulasi JST HEKM

Bila dimasukkan input = 0 0 1 0 1 0 1 0 ==> sensor ke-2 dan sensor ke-4 ada halangan dan goal direction ada di arah depan. Hasil network adalah 0 1 ==> motor kanan bergerak maka wahana belok kiri. Seharusnya untuk output G dihasilkan 0 1 1, akan tetapi hasil output G 1 0 1. Sedangkan pergerakan motor sudah mengambil langkah yang benar. Begitu juga dengan pengujian input network 0 0 1 1 1 0 1 1. Output G adalah 0 1 1 dan dengan goal direction 0 1 1 (sebelah barat daya wahana), aksi wahana adalah 0 1 atau wahana akan berbelok ke kiri.

KESIMPULAN

Rancangan jaringan saraf tiruan Hierarchical Extended Kohonen Map berhasil disimulasikan dan memberikan hasil yang baik. Jaringan yang dibentuk dapat melakukan aksi berdasarkan model-based system dan juga sensor-based system. Jaringan saraf tiruan HEKM yang berhasil dibangun memiliki komposisi 4x4 grid jaringan super-net dan 1x10 grid jaringan sub-net. Perubahan learning rate dan neighborhood parameter pada akhir iterasi, menunjukkan bahwa di dalam jaringan yang dibangun, setiap neuron mampu berkompetisi dengan neuron yang lain dalam satu grid.

DAFTAR PUSTAKA

- [1] Pfeifer.R, Scheier.C. Understanding Intelligence. MIT Press, Cambridge, MA, 1999.
- [2] Lichtensteiger.L, Salomon.R. The evolution of an artificial compound eye by using adaptive hardware. In: Processing of the 2000 congress on evolutionary computation, La Jolla Marriott, San Diego, CA, USA, 1144-1151, 16-19 July 2000.
- [3] Nehmzow, Ulrich. Mobile Robotics: A Practical Introduction. Springer-Verland London Limited. 2000.
- [4] Russell, Stuart. Artificial Intelligence: A Modern Approach. Prentice-Hall. 1995.
- [5] C. Ye, N.H.C. Yung, D. Wang. "A Fuzzy Controller With Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance," IEEE Transaction on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 33, No. 1, 2003.
- [6] X. Du, H.H. Chen, W.K. Gu, "Neural network and genetic algorithm based global path planning in a static environment" Journal of Zhejiang University SCIENCE, Vol 6A(6):549-554, 2005.
- [7] Z. Mao, T.C. Hsia, "Obstacle avoidance inverse kinematics solution of redundant robots by neural networks," Robotica, Cambridge University Press Vol. 15, 1997 .
- [8] B.Q. Huang, G.Y. Cao, M. Guo, "Reinforcement learning NN to the problem of autonomous mobile robot obstacle avoidance," Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.
- [9] S. X. Yang, C. Luo, "A Neural Network Approach to Complete Coverage Path Planning" IEEE Transaction on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 34, No. 1, 2004.
- [10] Guthikonda, Shyah M. Kohonen Self-Organizing Maps. Wittenberg University. December 2005.
- [11] Hendro, Tsani. Perancangan dan Implementasi Wahana Gerak Mandiri yang Adaptif Menggunakan Jaringan Saraf Tiruan Extended Kohonen Map (EKM). Bandung. 2002.