Article ID: 1000 - 8152(2003)02 - 0161 - 07

Connectionist approach for cognitive map learning and navigation based on spatio-temporal experiences

LIU Juan, CAI Zi-xing, TU Chun-ming
(College of Information Science & Engineering, Central South University, Hunan Changsha 410083, China)

Abstract: A connectionist method is proposed for mobile robot, which lacks a priori environmental model and global localization information, to learn goal-directed cognitive map from its own spatio-temporal experiences. Temporal sequence processing network (TSPN), which is constructed at run-time, provides compact representations of history perceptive information, transforms spatial knowledge into cell firing characteristics and retrieves them in later runs to guide the robot. The navigation system integrating TSPN and a reactive safeguard module performs dynamic landmark and heading detection, route learning and collision-free real-time navigation in noisy environments. The simulation and real world experiments demonstrate the effectiveness and flexibility of the system.

Key words: connectionist model; spatio-temporal reasoning; mobile robot; cognitive map; navigation
CLC number: TP183; TP242.6 **Document code**: A

一种基于连接机制和时空经验的认知地图学习与导航方法

刘 娟, 蔡自兴, 涂春鸣
(中南大学 信息科学与工程学院,湖南 长沙 410083)

摘要: 提出了一种连接主义方法,利用移动机器人自身的时空经验,在缺乏全局坐标信息和环境先验模型的情况下,建立面向目标的认知地图.在线形成的时序处理网络(TSPN)可提供简洁的历史感知信息,以神经元激活特性保存空间知识,引导机器人运动.结合 TSPN 和反应式行为模块的导航系统可实现动态的路标及方向检测、路径学习和实时导航功能.仿真和实际实验验证了系统的有效性和适应性.

关键词: 连接机制模型; 时空推理; 移动机器人; 认知地图; 导航

1 Introduction

Purely reactive robots with memoryless intelligence suffer from perceptual aliasing and cyclic behaviors, which is termed as amnesia in biology[1]. So the ability of having and using memory is essential to an autonomous robot. Without a predefined model of the world, spatial memory is necessary in more complex task-environment contexts. It may be presented in different forms. Building a metric map based on odometer and range sensor information seems straightforward since it can easily be used to plan paths and generate detours or shortcuts[2]. But the robot should always be aware of its accurate position and heading relative to a reference point and direction to maintain the spatial consistency, which is difficult due to the integration errors of dead-reckoning process and the absence of necessary sensors. Such a bird's eye world model tends to reflect the worldview of the designer rather than the robot's sensory system. Tolman[3] introduced the concept of 'Cognitive Maps' as a way to interpret experimental findings that in path selection behaviors of rats that used some form of internal spatial representation. Robots may also solve navigation tasks using a cognitive map containing the relationships of places with salient features. How can the cognitive map be constructed?

Grounded on the fact that learning, recognizing and recalling temporal patterns contribute greatly to human intelligence, we conceive that robots may also learn spatial knowledge from the regularity of temporal sequences of sensory and action flows. So the problems need to ad-

Received date: 2002 - 04 - 03 ; Revised date: 2002 - 11 - 04.
Foundation item: supported by the National Natural Science Foundation of China (69974043, 60234030).

dress in order to build such a navigation system are:

1) What is a suitable episodic memory mechanism for the task of navigation?

2) How should the system extract and store meaningful sensory data from noisy real world?

3) How is the temporal memory taken as a cognitive map to generate required actions?

Moreover the mechanism should be computationally cheap so that it is applicable in real time.

In this paper we presents a connectionist architecture—temporal sequence processing network (TSPN) that earns complex temporal motor-sensory flows from the robot's own experiences and retrieves them to perform navigation tasks in a noisy environment. Section 2 gives a description of TSPN. In section 3, we describe the details and relative problems when the network is applied to mobile robot navigation. The experimental results are presented in section 4, which validates our approach on the AmigoBot, a product of Activmedia Inc., in a world with obstacles and colored objects.

2 Network architecture

Due to the incomplete, noisy and imprecise action and perception, learning algorithms for navigation have to compensate for a range of distortions in time and in data. Furthermore, in order to adapt to the ever-changing external and internal environments, robot's knowledge should be updated frequently to expand its capacity, acquire new patterns and preserve old data. Artificial neural networks (ANN) are regarded as one of the most attractive approaches for many reasons such as their connectionist architecture resembling human brain's structure, requiring little knowledge of the task, strong nonlinear mapping ability etc. General problems with most ANN learning methods are that they often require numerous examples and computational work' that the user is unable to do online learning.

Temporal sequence processing network (TSPN) specializes in dealing with complex sequences, different from ANNs used in approximation or mapping. It is a non-symmetrical recurrent network with two kinds of neurons — original and abstract units, as shown in Fig.1. Original units represent symbols that constitute sequences while an abstract unit may represent a sequence.

2.1 Learning and retrieval processes

The learning process is forming a network while the retrieval process is using stimulus and other information to trigger and recall remaining part of a sequence.

The activity of a neuron i at time t is denoted by $y_i(t)$, where $i = 1,\cdots,N$. If the neuron fires in time t, then $y_i(t) = 1$ or -1. Self-connection tp_{ii} causes the decay of excitation level of unit i at each time step by the same factor, which enables associations between units reflecting time delay. If unit i is not activated or inhibited at time t,

$$y_i(t) = y_i(t-1) \cdot tp_{ii}. \tag{1}$$

It is more compact to use the self-decay mechanism than an external time counter, which will be discussed in section 3. The activation will be reset to 0 if it drops below a predefined value.

During the learning phase, when unit i is activated, other units with non-zero activities are all regarded as pre-synaptic neurons that influence its activation. The connections are created and updated by the rule:

$$tp_{ij}(t) = \begin{cases} y_j(t), & tp_{ij} = 0 \text{ and } y_j(t) \neq 0, \\ tp_{ij} + \alpha(y_j(t) - tp_{ij}), & \text{else}. \end{cases} \tag{2}$$

In Fig.1, the white disks represent the original units and abstract units respectively. For clarity, not all of possible connections are drawn. Dash lines refer to those connections that transfer to the abstract units. Although the connections in one direction have been transferred, those in the other direction are intact.

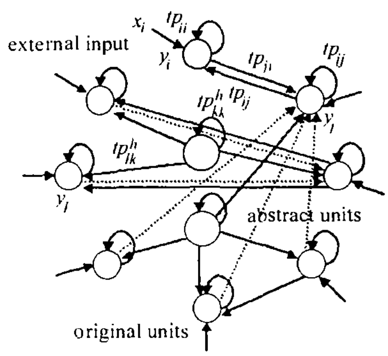


Fig.1 Schema of TSPN

Connections between unit i and j, tp_{ji} and tp_{ij}, may be different. They reflect when the activations of pre-synaptic units have the greatest impact on the post-synaptic units. During the retrieval phase, once input stimuli of a unit cause its post-synaptic potential $V_i(t)$

to rise above a threshold θ, the unit is activated. Then it is combined with other activated units to fire next units. A wave of activity is formed and propagates stably. The firing chain is the sequence stored in TSPN.

$$y_i(t) = F(V_i(t) - \theta), \tag{3}$$

where

$$F(x) = \begin{cases} 0, & x \leqslant 0 \\ 1, & x > 0 \end{cases}, \quad tp_{ij} \in [-1, 1],$$

$$V_i(t) = x_i(t) + c \sum_{\substack{j \neq i \\ tp_{ij} \neq 0}} G(tp_{ij}, y_j(t-1)),$$

$$G(a, b) = \operatorname{sgn}(ab) \cdot \min\{\operatorname{abs}(a/b), \operatorname{abs}(b/a)\}. \tag{4}$$

$x_i(t)$ is the external input signal and c is a normalizing factor. $G(a, b)$ is used to measure the proximity. If a, b have different signs, it means an obvious inconsistency and the negative product reduces the potential of unit i.

In complex sequences, a later occurrence of a symbol may overwrite its earlier occurrence stored in the unit as its activation, which dramatically decreases depth n (i.e. the ability to record n repetitions of the same symbol in a sequence) of the network. In order to deal with the overwriting problem, a unit is expanded to a network[4]. It has multiple terminals to hold different occurrences of a symbol, with multiple connections to other units (see Fig.2(a)). The terminals will be added dynamically during the learning stage. Suppose unit i has p terminals, the activation of qth terminal is represented by y_i^q holding the qth occurrence of unit i in a sequence and its connections with other units are tp_{ji^q} or $tp_{i^q j}$.

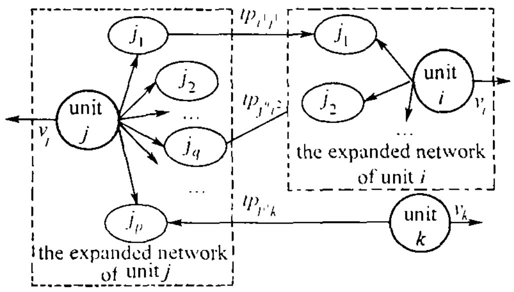


Fig. 2(a) The expanded network of units

For example, sequence S_1 is as follows and "$\xrightarrow{m}$" denotes the time delay m:

$$A \xrightarrow{2} B \xrightarrow{5} A \xrightarrow{7} C \xrightarrow{3} A \xrightarrow{6} C \xrightarrow{9} D \xrightarrow{4} B \xrightarrow{3} A.$$

We assume tp_{ii} to be 0.9 here. The learned network structure is shown in Fig.2(b). The figures on connections are values of tp_{ij}. Since A, B, C occur more than once in S_1, their expanded networks respectively have 4, 2,2 terminals. This is a simplified example to demonstrate how the network works. In fact more than two units can be co-activated in applications. Their activations all leave traces in connections. In Fig.2(b) and Fig.2(c), light gray cells with letters are original units, cells with 'H' is an abstract unit.

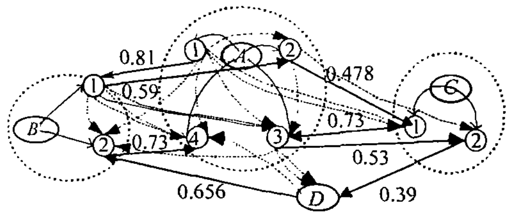


Fig. 2 (b) Demonstrations of the network for S_1

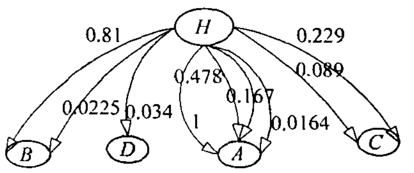


Fig. 2 (c) Demonstrations of the network for S_1

2.2 Abstract units

Obviously patterns stored in the above network are unstable because if a new sequence is presented, the connections may be changed to represent the new pattern, forgetting old relationships. We have to introduce something to retain what the network has learned so that the lifelong learning is feasible. The abstract unit in TSPN organizes and stores the experiences for the lifetime of the individual. The connections between original units and their terminals serve as a fast, temporary storage created immediately when the experiences come in, just like the short-term memory. Memories stored in the connections between abstract units and original units may be taken as long-term memory.

An abstract unit is created when a new pattern shows up. It collects the connections relative to the last sequence and arranges them according to the order in which units occur in the sequence. The algorithm to create an abstract unit and connections is as follows:

1) Create an abstract unit H with activity 1,

$$y_H(1) = 1.$$

2) Connect H and the first activated unit j with weight 1.

3) Decrease its activation by

$$y_H(t) = tp_{HH} \times y_H(t-1).$$

4) Retrieve the sequence from j.

If the end of the sequence is reached, end.

Else

If there is no unit activated in this time step, go to 3).

Else if unit k is activated,

connect H and k with weight $y_H(t)$;

5) $j = k$, go to 3).

Note that $tp_{HH} = tp_{ii}$. The hidden unit representing S_1 is the green unit H in Fig.2(c). When the memory of a sequence is transmitted to an abstract unit, the connections and terminals relative to the sequence will be deleted. There is no upper bound for the number of abstract units or the terminals of units.

3 TSPN for robot learning and control

As a mobile robot travels through an environment, acting selection should depend not only on the latest input, but also on the history perception and action data. Otherwise it might suffer from 'perceptual aliasing'. Information about the past can be made available through the incorporation of an input buffer that stores a number of the latest inputs. But it is very hard to find an appropriate size for the buffer. To sort out inputs with useful information is also a formidable task. Actually the changes of inputs carry most of the important environmental features for localization and task performance. Assume the robot's translation velocity and rotation velocity are uniform during the learning phase. The duration of an action can reflect the distance or the change of its heading. So temporal sequences can be regarded as another kind of spatial representation. A system based on TSPN capable of change detection and sequence recalling is designed to guide the robot.

3.1 System setup

AmigoBot, the test-bed robot of our experiments is equipped with a color camera and eight sonar sensors. Its task is to explore an environment, construct and maintain a network that contains the relations of actions and their spatio-temporal contexts incrementally. The robot that sets out from random position learns, rather than programmed or trained, to follow routes that may lead it to particular spots which have salient features and are potential goals of navigation tasks. Routes are motion sequences stored in TSPN represented by the temporal characteristics of cell firing. A particular envionmental context may trigger a firing chain in TSPN and the output is used as routing signal to guide the robot to reach the other end of the path. The robot may not execute every step as it has learned since the learned action flow may be not applicable in present situation.

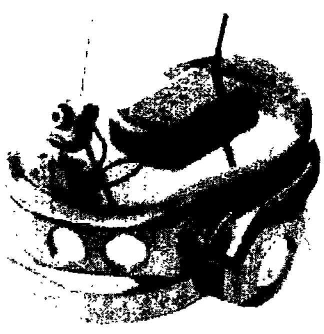
3.2 Context extraction

Three kinds of context information should be extracted from the sensory flow: sonar range readings, landmarks and motor signals. To reduce the sensor space complexity and to learn the structural feature of the environment, the qualitative properties of the data are used rather than their exact values.

Robot control cycle is 100 ms. Due to the noisy and imprecise sensor readings, recording all of the data in every cycle is unnecessary. Those numerous and jumbled data may conceal the regularities in action and perception. The data recording and learning frequency is 1 Hz so as to integrate the information in past 10 control cycles.

Fig.3 illustrates the organization of 8 sonar sensors on the robot. They are divided into 4 groups: front, left, right and back sensor, which are represented by 4 units in TSPN. To prevent oscillation at the boundary of the two classes, a hysteresis of a few centimeters is used to classify inputs in terms of 'near' and 'far'. Fig. 4 shows an example of the mapping between right sonar readings and the activation of right sonar unit. In Fig.4, (light line, [0,300]) and the activation of right sonar unit (dark line, [-1,1]). Two dashed lines denote the rising and falling treshold of the hysteresis classifier. $S_{f+} = 1500$, $S_{n-} = 800$. The activation value

$$y_i^S(t) = \begin{cases} -1, & \text{input} > S_{f+} \text{ and } y_i^S(t-1) > 0, \\ y_i^S(t-1) \cdot tp_{ii}, & \text{else}, \\ 1, & \text{input} < S_{n-} \text{ and } y_i^S(t-1) < 0. \end{cases} \tag{5}$$



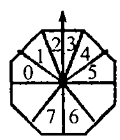


Fig. 3 The AmigoBot and the organization of sonar sensors

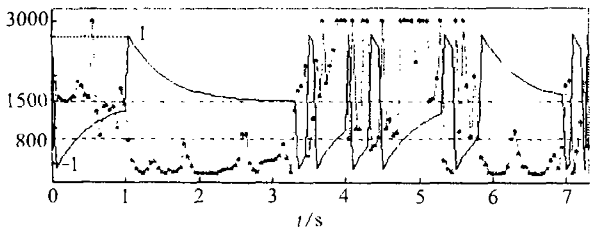


Fig. 4 The relations of right sonar readings

The decay mechanism records sensor experiences so that the robot can incorporate both the past and present sensor inputs and comprehend the nature of its environment. The lower the activity, the fewer the fluctuations of sensor data. Therefore a snapshot of unit activations contains past sensory information.

The vision system processes images with 640 × 480 pixels at 10 Hz. It is used to discover salient objects and obstacles. An object with a distinct color is taken as a landmark. Combining sonar units' activities and the vision signal, observations of an object from different viewpoints can be discriminated. we uses a location unit to stand for the discovery of a landmark from a particular direction. So there may be several location cells for the same landmark. The activity of location cell $y_i^L(t)$ is:

$$y_i^L(t) = \begin{cases} 0, & \text{no object detected} \\ & \text{or } y_k^L(t-1) = 1 \ (k \neq i), \\ 1, & \text{the object detected and } y_i^L(t-1) = 0 \\ & \text{and } y_j^s(t) w_{ij} > 0 \text{ and } \min\left(\dfrac{y_j^s}{w_{ij}}, \dfrac{w_{ij}}{y_j^s}\right) > \delta, \\ y_i^L(t-1) tp_{ii}, & y_i^L(t-1) = 0 \text{ and } y_k^L(t-1) = 0 \ (k \neq i). \end{cases} \tag{6}$$

The location cells inhibit each other, but a connection ε_{ik} is generated between the successive activated cells to indicate a known routs.

$$\varepsilon_{ik} = y_i^L(t-1) y_k^L(t), \tag{7}$$

in which t denotes the processing cycle when location cell k is activated. ε_{ik} implies the length of the path thanks to the self-decay of $y_i^L(t)$.

We use several motor primitives, i.e. going straight, turning left and turning right, to code mobile robot's action repertoire as they are sufficient for the robot to reach almost any place in a plane. The turning action in 100 ms only causes about 2° heading difference. So a less than 10° heading change means a go-straight behavior and the turning actions mingled in this cycle can be taken as slight heading adjustments since the robot does not intend to turn to another direction. Once a motion unit is activated, all other units including location and sonar units with non-zero excitation level are connected with this unit. The weight tp_{ij} represents how long action or perception j lasts before action i is activated. Its activity will decrease by tp_{ii} in every processing cycle as long as no other motion cell is activated. Otherwise a new motion cell is generated. The ensemble of connections forms the firing context of the motion cell.

3.3 Learning and recalling of routes

The navigation system has two working modes: exploration mode and navigation mode. In the exploration mode, TSPN is constructed and updated to learn spatial and temporal relations of perceptions and actions. When a goal is designated, the robot applies the knowledge stored in TSPN to find its way, which may be concatenated paths, from various starting positions.

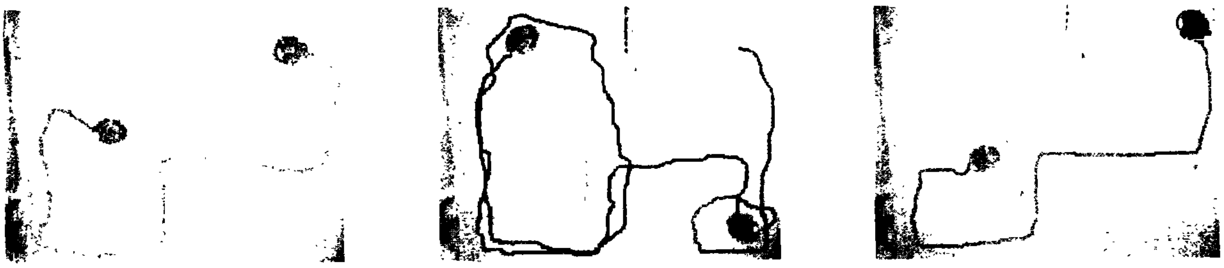
When a location cell is fired, the network constructing process begins. Motion cells generated during the travel from one location to another cluster to constitute a subnet corresponding to a route. The firing of the terminal location cell inhibits all motion cells in the same subnet. Fig.6(a) shows a learnt network.

In the navigation mode, a landmark is given as the goal. Reaching it from any direction is allowable. Before a learned sequence is stimulated to function, the robot takes reactive behaviors. Once it reaches a location that can be linked with the goal by one or several routes, the guidance signal begins to function. Units in the subnet fire in a precise temporal sequence, retrieving the learnt motion chain. The activation of a motion cell needs cooperative efforts of location cells, sonar cells and other pre-synaptic motion cells.

4 Experimental results

The environment of the physical experiment shown in Fig.5 consists of a 3.5 m × 3.1 m pen containing some of wooden blocks and plastic bottles as obstacles. A varicolored ball and a green ball are used as landmarks.

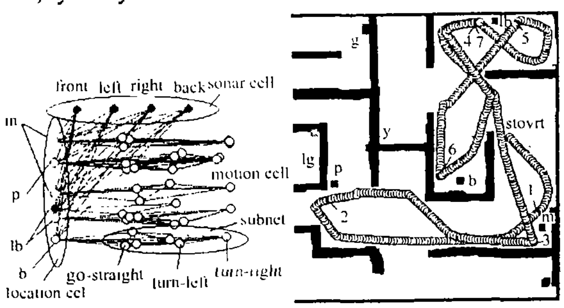
A reactive safeguard module is combined with TSPN to avoid collision. Initially the robot freely explored the environment using its reactive controller. We placed some indications such as black strips and blocks in the places where the robot may be trapped. The robot could perceive them by the vision system and would not go further. The robot memorized what it had done and perceived in TSPN, the utilized the knowledge to perform navigation tasks in an environment without indications, where the innate controller was not competent. Without global localization information, it learned a cognitive map of the environment. The experimental result in Fig.5 shows that the robot can select the right path and tune its heading timely with the right angle.



(a) The trajectory of learning phase (b) The trajectory of the robot moving reactively, without memory (c) The trajectory of the robot guided by TSPN

Fig. 5 The physical experiment

Figure 6(b) shows the simulated environment and the exploration process. The generated network in Fig.6(a) indicates that the robot can distinguish various headings using history sensor data. There are two location cells for light blue landmark and two for magenta landmark. When the robot approached light blue landmark for the third time, it inferred from the environmental context that it was at the same location where it saw the landmark for the first time. So new location cell was not generated, the fourth location cell was activated instead. During the exploration mode, the number of cells will increase fast at the early stage and tend to be stable later. In Fig.6(a), the gray level of a cell indicates its activity from 0 for white cells to 1 for black cells. For clarity, connections between sonar cells and motion are not presented. In Fig.6(b), characters indicatethe color of a landmark. b — blue, lb — light blue, g — green, lg — light green, m — magneta, p — purple, r — red, y — yellow.



(a) (b)

Fig. 6 The structure (a) of TSPN learnt during the exploration in a world shown in (b)

5 Conclusions

The approach has some common ground with DRAMA proposed by Billard and Hayes[5], but DRAMA (dynamical recurrent associative memory architecture) is not used to provide navigation instructions and it has difficulties in processing complex sequences. Different from some map learning and navigation system[6~8], TSPN is used to solve navigation problems where the robot has no idea about the global coordinates and direction, but can identify some special places, as human beings do in their daily life. It processes past sensory data based on change detecting and stores them in a connectionist way, other than a grid-based map. As a computationally inexpensive and concise method to encode the history of motor and sensor readings, TSPN provides necessary contexts for distinguishing similar perceptions. It is an incremental network with hierarchical structure, which enables learning throughout the entire lifetime to cope with dynamic environments. Experiments in simulation and real world have shown that the model is a fast and robust architecture for learning and navigation.

References:

[1] MUKERJEE A, DATTATRAYA A. Reactive robots and amnesics: a comparative study in memoryless behavior [J]. *IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 1999,29(2):216 - 226.

[2] SALICHS M A, MORENO L. Navigation of mobile robots: open questions [J]. *Robotica*, 2000,18:227 - 234.

[3] TOLMAN E C. Congnitive maps in rats and men [J]. *Psychological Review*, 1948,55(4):189 - 208.

[4] WANG D L, ARBIB M A. Timing and chunking in processing temporal order [J]. *IEEE Trans on Systems, Man, and Cybernetics*, 1993,23(4):993 - 1009.

[5] BILLARD A, HAYES G D. A connectionist architecture for control and learning in autonomous robots [J]. *Adaptive Behavior*, 1999,7(1):35 - 63.

[6] MATARIĆ M. Integration of representation into goal-driven behavior-based robots [J]. *IEEE Trans on Robotics and Automation*, 1992, 8(3):304 - 312.

[7] MALLOT H A. BÜLTHOFF B, SCHÖLKOPF B, et al. View-based cognitive map learning by an autonomous robot [A]. *Proc of ICANN'95* [C]. Nanterre: EC2,& Cie, 1995,2:381 - 386.

[8] TANI J, NOLFI S. Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor flow [J]. *Neural Networks*, 1999,12:1131 - 1141.

作者简介:

刘 娟 (1975 —),女,现为中南大学信息科学与工程学院博士生,研究方向为智能机器人,计算智能,机器视觉等,E-mail: ljcic@263.net;

蔡自兴 (1938 —),男,1962 年毕业于西安交通大学工业电气化与自动化专业,现为中南大学教授,博士生导师,出版著作、教材 18 部,在国内外发表论文 300 余篇,目前研究方向为智能控制,人工智能,智能机器人等;

涂春鸣 (1976 —),男,现为中南大学信息科学与工程学院博士生,研究方向为智能控制,电力系统自动化,柔性交流输电系统.

Main contents of the next issue