

Hierarchical Markov Decision Based Path Planning for Palletizing Robot

Jiufu Liu*, Zhengqian Wang, Zhe Chen, Zhong Yang, Zhisheng Wang, Chunsheng Liu

College of Automation, Nanjing University of Aeronautics and Astronautics,
Nanjing, 210016, China

*Corresponding author, e-mail:liujiufujs@sina.com

Abstract

On account of the complex application environment and the large number of uncertain conditions for the palletizing robot, we do path-planning for the multiple joints robot by the algorithm based on Hierarchical Markov Decision Process. First, according to the actual working environment, we set the range of the robot's motion and select the conventional movement combination as the basic set of the robot's behaviors. We can get the possible reward of various situations. We divide the state space in accordance with the location information of the obstacle space into a small number of state clusters, sub-level step by step to determine the precise trajectory of palletizing robots. We simulate 3D robot motion trajectory, including barrier-free and spherical obstacle conditions. Finally, experimental verification is carried out, the algorithm has been proved to control the compatible movements of each joint effectively and keep the error within the allowed range. The experiment results meet the requirement well.

Keywords: palletizing robot, path planning, hierarchical markov decision process, state clustering

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Robot path planning is to solve the optimal movement trajectory problem from the initial position to the given target position in the environment of the obstacles. The planned path can be safe, which can make the robot passing all obstacles, can make the robot moving without collision, and ensure the shortest path of obstacle avoidance.

According to the different the states of the environment, path planning can be divided into global path planning and local path planning [1, 2]. Global path planning methods mainly have grid method, visibility graphs and free space method. Local path planning methods mainly have artificial potential field method, fuzzy logic control algorithm, the ant colony algorithm and particle swarm algorithm. Global and local path planning technology are just deal with independent mobile robot path planning problem, and lack of the description of the interaction between each mechanical arm of the multiple joints robot, so the above planning technology are not suitable for palletizing robot path planning.

The real work environment is complex, in which there are the large numbers of uncertain conditions for the palletizing robot. In this paper, our contribution is that based on Markov Decision Process (MDP) [3-5] we introduce state clustering to establish Hierarchical Markov Decision Process (HMDP) model [6, 7] to perform the path planning of the palletizing robot. According to the spatial position of obstacles, we condense palletizing robot's working space for several state clusters, as the state space of HMDP model. Using HMDP algorithm, we can search for the optimal path. After establishing HMDP model, the states clearly reduce, to search for the optimal path is more rapid, the complexity of the algorithm can also be reduced. HMDP algorithm can effectively overcome the problem of large number of uncertain conditions for the palletizing robot.

2. Hierarchical Markov Decision Process

2.1. Markov Decision Process

MDP is the most useful model in planning application based on decision theory, and it is a model describing interaction between Agent and the environment. Agent will put state of the environment as input, and an action for output, and these actions will affect the state of the

environment. Although the influence of Agent's actions to the environmental state with a lot of uncertainty, Agent's perception of the state of the environment is without any uncertainty, that is, it has complete perception.

A Markov Decision Process can be described as a tuple $\langle S, A, P, R \rangle$, where

S is a finite set of states,

A is a finite set of actions,

$P : S \times A \times S \rightarrow [0, 1]$, is the state-transition function, that describes the probability $p(s' | s, a)$ when the system move from state s to s' after the action $a (a \in A)$.

$$p(s' | s, a) = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \quad (1)$$

$R : S \times A \times S \rightarrow \mathbb{R}$, is the reward function, is real bounded function, the expected immediate reward gained when the system move from state s to s' after the action $a (a \in A)$.

In this model, the next state and the expected reward depend only on the previous state and the taken action; even if we consider all previous states, the transition probabilities and the expected reward would remain the same. This is known as the Markov property, which is that the state and reward at time $t + 1$ is dependent only on the state and the action at time t .

2.2. Hierarchical Markov Decision Process

Our method extends the standard MDP framework by adding hierarchical structure. There are two types of MDPs making up a complete hierarchical system [8-10]. They are both derived from a given standard, flat MDP.

The first type, M^n , a tuple $\langle S^n, A^n, T^n, R^n \rangle$ ($0 \leq n \leq N-1$), represents the given MDP at a particular level of abstraction. n indexes the level in the hierarchy, N is the number of levels in the hierarchy. The given flat MDP is M^0 . M^n for $n \geq 1$ is constructed from M^{n-1} by clustering the states in S^{n-1} . Each cluster of states from S^{n-1} becomes a single state in S^n .

The state transition function T^n , which defines $p(s_m^n | s_k^n, a^n)$, is constructed by determining, for all states $s_i^{n-1} \in S^{n-1}$ that belong to one cluster and correspond to state s_k^n , the cluster labels of successors s_j^{n-1} that belong to other clusters and correspond to state s_m^n . The probability $p(s_m^n | s_k^n, a^n)$ is estimated by averaging over the corresponding probabilities $p(s_j^{n-1} | s_i^{n-1}, a^{n-1})$. Similarly, the reward function R^n , which defines $r(s_k^n, a^n, s_m^n)$, is constructed by determining, for each state transition from s_k^n to s_m^n , the corresponding $r(s_i^{n-1}, a^{n-1}, s_j^{n-1})$, and averaging over them. As before, the action set A^n is defined as the set of successors s_m^n for each state s_k^n .

The second type of MDPs making up the complete hierarchical system is defined only for $n \geq 1$ and is denoted by $M_{s_k^n, s_m^n}^{n-1}$. $M_{s_k^n, s_m^n}^{n-1}$ is an MDP that represents the lower level ($n-1$) task of navigating from higher level (n) state s_k^n to s_m^n . It is essentially a subset of M^{n-1} , whose states are only those states $s_i^{n-1} \in S^{n-1}$ that correspond to state s_k^n , combined with those state $s_j^{n-1} \in S^{n-1}$ that are successors of states s_i^{n-1} and that correspond to state s_m^n . The states s_j^{n-1} are terminal states.

2.3. Hierarchical Markov Decision Process Model Algorithm

Hierarchical Markov Decision Process model of layered model value iteration algorithm as follows:

Step 1 Initialization. Construct hierarchical MDP model M^n ($0 \leq n \leq N-1$), component elements of M^0 and M^n are given, the target state s_g^0 for M^0 is given, set value function iterative end conditions ε .

Step 2 for $0 < n < N$, the hierarchical MDP model M^n determine the MDP target state s_g^n for M^n .

Step 3 Definite function $Solve(M, n)$, whose value is the value of state function $V(s)$.

while $\delta > \varepsilon$, do

for all $s \in S$, do

$$V_{new} \leftarrow \max_a \sum_{s'} p(s' | s, a) [r(s' | s, a) + V(s')],$$

if $|V_{new} - V(s)| > \delta$, then

$$\delta \leftarrow |V_{new} - V(s)|,$$

$$V(s) \leftarrow V_{new}.$$

Step 4 For $n > 0$, to all states $s \in S$,

(1) If the current state $s = s_g^n$, then

M^{n-1} construct $M_{s_g^n, s_g^{n-1}}^{n-1}$, and

$$V_{s_g^n, s_g^{n-1}}^{n-1} \leftarrow Solve(M_{s_g^n, s_g^{n-1}}^{n-1}, n-1).$$

(2) If the current state $s \neq s_g^n$, then

$$s^* \leftarrow \arg \max_a \sum_{s'} p(s' | s, a) [r(s' | s, a) + V(s')], \text{ else}$$

M^{n-1} construct M_{s, s^*}^{n-1} , and

$$V_{s, s^*}^{n-1} \leftarrow Solve(M_{s, s^*}^{n-1}, n-1).$$

Step 5 $n \leftarrow n - 1$,

if $n = 0$, then iterative end, return the values of the states function V , else
turn to Step 4.

Step 6 By Step 5 solve optimal strategies $\pi^*(s) = \arg V(s)$.

3. Palletizing Robot's Kinematics Model

The mechanical structure of the palletizing robot is simplified as 3D structure diagram shown in Figure 1. The point A is thought as end actuator. There are three connecting rods, the connecting rod 1, the connecting rod 2, the connecting rod 3. The connecting rod 3 rotates around point B up and down, the connecting rod 2 rotates around point C up and down, the connecting rod 1 carries the mechanical arm around the base rotation in horizontal plane.

Each of the robot joints are rotary joints. To solve the robot kinematic analysis problem of we establish the D-H robot coordinate system. The $q_n, \alpha_n, \theta_n, d_n$ ($n=1, 2, 3$) are the robot D-H parameter. The D-H parameter table of palletizing robot coordination system is shown in Table 1.

The notation q_n is the length of the connecting rod n , α_n is the axis angle of the connecting rod n and the connecting rod $n+1$, θ_n is the rotation angle between the connecting rod $n-1$ and the connecting rod n , and d_n is the relative position distance between the common perpendicular lines of the joint $n-1$ and the joint n .

Robot kinematics model is obtained based on D-H coordination transformation method. In the palletizing robot structure diagram, connecting rods establishes the rear coordination system based on D-H coordination transformation. The two adjacent connecting rods coordination system transformation relationship can be used by matrix T_n^{n-1} , in which $n-1$ denotes the connecting rod $n-1$ coordination system; n denotes the connecting rod n coordination system.

$$T_n^{n-1} = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cdot \cos \alpha_n & \sin \theta_n \cdot \sin \alpha_n & q_n \cdot \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cdot \cos \alpha_n & -\cos \theta_n \cdot \sin \alpha_n & q_n \cdot \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

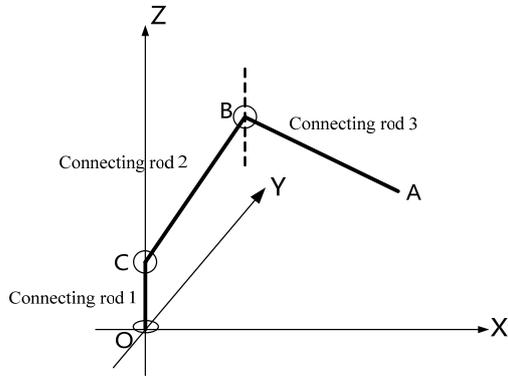


Figure 1. Palletizing Robot Structure Diagram

Table 1. Palletizing Robot D-H Parameter Table

connecting rod n	q_n	α_n	d_n	θ_n
1	0	$\frac{\pi}{2}$	l_1	θ_1
2	l_2	0	0	θ_2
3	l_3	0	0	θ_3

Multiply coordinate transformation matrix of end stem can be expressed as Equation (3).

$$T_3^0 = T_1^0 T_2^1 T_3^2 = T_1^0(\theta_1) T_2^1(\theta_2) T_3^2(\theta_3) \quad (3)$$

By Equation (4), T_1^0 , T_2^1 , T_3^2 can be written out.

$$T_1^0 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_2 \cdot \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & l_2 \cdot \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_3^2 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_3 \cdot \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & l_3 \cdot \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Uniting equation (3-2), (3-3), (3-4), (3-5), we can attain the transformation matrix T_3^0 of palletizing robot end position relative to the base coordinate system.

$$T_3^0 = \begin{bmatrix} c1c2c3 - c1s2s3 & -c1s2c3 - c1c2s3 & s1 & l_2c1c2 + l_3c1c2c3 - l_3c1s2s3 \\ s1c2c3 - s1s2s3 & -s1s2c3 - s1c2s3 & -c1 & l_2s1c2 + l_3s1c2c3 - l_3s1s2s3 \\ s2c3 + c2s3 & c2c3 - s2s3 & 0 & l_1 + l_2s2 + l_3s2c3 + l_3c2s3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

By Equation (7), we can attain end actuator's space position relative to the base coordinate system.

$$\begin{cases} p_x = l_2 c_1 c_2 + l_3 c_1 c_2 c_3 - l_3 c_1 s_2 s_3 \\ p_y = l_2 s_1 c_2 + l_3 s_1 c_2 c_3 - l_3 s_1 s_2 s_3 \\ p_z = l_1 + l_2 s_2 + l_3 s_2 c_3 + l_3 c_2 s_3 \end{cases} \quad (8)$$

Among Equation (7), (8), $s_i = \sin \theta_i, c_i = \cos \theta_i (i = 1, 2, 3)$.

4. Palletizing Robot's Motion Trajectory Simulation

In the actual control system, mechanical arm of palletizing robot is driven by the servo system, the controller can accurately control servo system's rotation angle and the angular velocity. To palletizing robot, the external environment of the state, the position and the size of the obstacles in the controller are known, namely the system state is completely visible. The movement planning system of palletizing robot as shown in Figure 2.

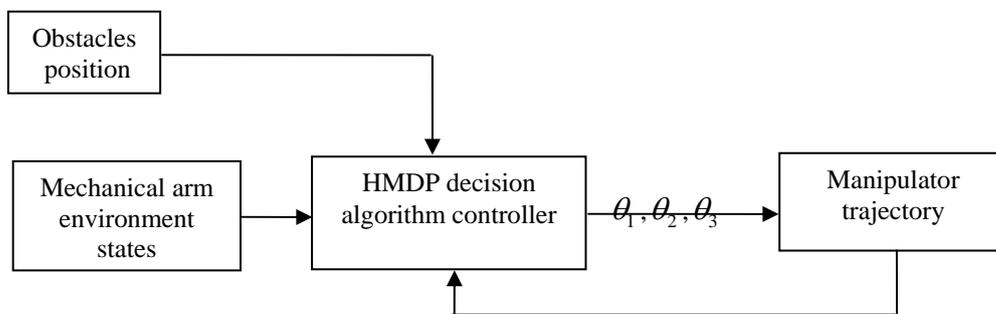


Figure 2. Palletizing Robot Motion Planning System

4.1. Basic State Set

Based on the designed mechanical structure of palletizing robot, the range change of joint variables θ_1, θ_2 and θ_3 is sure, such as $0 < \theta_1 < \frac{2}{3}\pi, \frac{\pi}{6} < \theta_2 < \frac{5}{6}\pi, \frac{7}{6}\pi < \theta_3 < \frac{11}{6}\pi$.

According to the control requirements of the palletizing robot and the drive system needs, we choose movement step of 1° to discrete its working space.

4.2. Basic Action Set

The connecting rod 1, 2 and 3 all have three movements. The connecting rod 1 around the base can do clockwise and counterclockwise rotation, or not move. Connecting rod 2 and 3 around the node C and B do clockwise and counterclockwise rotation, or not move. The combination actions of a palletizing robot have 27 kinds. In the movement process of palletizing robot from the starting position to the target position, only a fraction of the combination actions is effective. As shown in Figure 1, when on the path an obstacle is detected, the connecting rod 1 does counterclockwise movement, and the connecting rod 2 or 3 does counterclockwise movement to evade the obstacle, when the obstacle is passed by, the connecting rod 2 or 3 does clockwise movement to come back the original trajectory. All the effective combination actions in 7 groups are shown in Table 2.

4.3. State Transition Function

By probability theory, the sum probability of all palletizing robot actions is 1, namely, $\sum_t a'_1 \otimes a'_2 \otimes a'_3 = 1$. a_1, a_2, a_3 are the actions of connecting rod 1, 2 and 3, $a_1 \otimes a_2 \otimes a_3$ is the

combination action of the connecting rod 1, 2 and 3. The connecting rod 1, 2, 3 action sets and the probability of state transition is shown in Table 2.

4.4. Reward Function

For a certain behavior combination, the reward function is shown in the following.

$$R = \begin{cases} 10, & \text{close to the target position} \\ 6, & \text{close to the obstacle} \\ -20, & \text{hit the obstacle} \end{cases} \quad (9)$$

Table 2. The Movements of Connection Rod 1, 2, and 3 and the State Transition Probability

Serial number	Connection rod 1	Connection rod 2	Connection rod 3	probability
1	counterclockwise	counterclockwise	counterclockwise	0.06
2	counterclockwise	counterclockwise	no move	0.15
3	counterclockwise	no move	counterclockwise	0.25
4	counterclockwise	no move	no move	0.01
5	counterclockwise	clockwise	clockwise	0.03
6	counterclockwise	clockwise	no move	0.15
7	counterclockwise	no move	clockwise	0.35

4.5. Trajectory Simulation

We set the initial position $\theta_1=0$, $\theta_2=\frac{5}{6}\pi$, $\theta_3=\frac{7}{6}\pi$, the target position $\theta_1=\frac{2}{3}\pi$, $\theta_2=\frac{\pi}{6}$, $\theta_3=\frac{11}{6}\pi$. The starting position is $x=1100\text{mm}$, $y=0$, $z=750\text{mm}$, and the target position is $x=-1850\text{mm}$, $y=3200\text{mm}$, $z=750\text{mm}$.

(1) Trajectory simulation with no obstacle

Without placed obstacles, palletizing robot grabs objects and moves according to the above HMDP algorithm. The biggest reward simulation trajectory is shown in Figure 3 below. In Figure 3, the starting position is $x=1100\text{mm}$, $y=0$, $z=750\text{mm}$, the simulation arrival position is $x=-1850\text{mm}$, $y=3200\text{mm}$, $z=750\text{mm}$. The error between the simulation arrival position and the target position is 0 mm, which is in the allowable error scope. The object can be putted to the correct position.

(2) Trajectory simulation with obstacles

With a spherical obstacle, palletizing robot grabs objects, whose position is (0, 320mm, 1300mm) and radius is 400mm, and moves. Due to the speed of palletizing robot is fast, in order to completely avoid obstacles, robot can only moves in more than 50mm outside from spherical obstacles, namely, the avoided spherical obstacles is bigger than the actual radius 50mm. The simulation trajectory is shown in Figure 4 below. In Figure 4, the starting position is $x=1100\text{mm}$, $y=0$, $z=750\text{mm}$, the simulation arrival position is $x=-1850\text{mm}$, $y=3198\text{mm}$, $z=752\text{mm}$. The error between the simulation arrival position and the given target position is 3 mm, which is in the allowable error scope. the object can still be putted to the correct position.

4.6. Algorithm Performance Comparison between MDP and HMDP

Table 3 lists comparison of the combination states, planning time and path cost using MDP model algorithm and HMDP model algorithm, from which based on HMDP model the path planning can make planning time more short, path cost less. With no obstacles using MDP model algorithm, combination states are 3, planning time is 64ms and path cost is 7500. With no obstacles using HMDP model algorithm, combination states are 3, planning time is 64ms and path cost is 7500, too. With no obstacles using MDP model algorithm and HMDP model algorithm, the computing effectiveness of the combination states, planning time and path cost is the same. With an obstacle using MDP model algorithm, combination states are 5, planning time is 92ms and path cost is 8943. With an obstacle using HMDP model algorithm, combination states are 4, planning time is 65ms and path cost is 7515. With an obstacle comparing MDP model algorithm and HMDP model algorithm, the combination states reduce 20.0%, planning time reduces 29.3% and path cost reduces 16.0%.

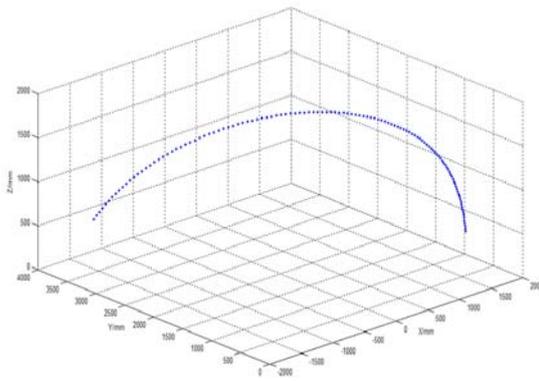


Figure 3. Trajectory Simulation with no Obstacle

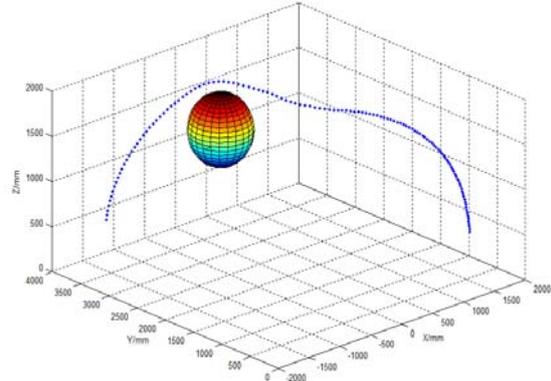


Figure 4. Trajectory Simulation with Obstacles

Table 3. Comparison between MDP and HMDP

The simulation experiment		Combination states	Combination states reduction percentage	Planning time (ms)	Planning time reduction percentage	Path cost	Path cost reduction percentage
No additional obstacles	MDP	3	-	64	-	7500	-
	HMDP	3	0	64	0	7500	0
Additional obstacles	MDP	5	-	92	-	8943	-
	HMDP	4	20.0%	65	29.3%	7515	16.0%

5. Experiment and Discussions

The controller of palletizing robot is made up of ARM chip and peripheral expansion interface circuit. The pulse signals are produced to drive servo system with mechanical drive joint operation. We choose Fuji 1.5KW AC servo motor, GYG152CC2-T2E and the relevant servo driver RYC152C3-VVT2. The experimental settings of palletizing robot are that the initial position is (1100mm, 0,750mm) and the target position is (-1850mm, 3200mm, 750mm). The center of the spherical obstacle is (0, 320mm, 1300mm), the radius of the spherical obstacle is 400mm. Through HMDP algorithm, the two path trajectories are generated with no obstacle and with an obstacle, respectively. The error is calculated between the target locations and actual position. With no obstacles the error range is below 3mm, with an obstacle it is not more than 6mm, which can satisfy the real requirements of palletizing robot operation. The experiment results are shown in Table 4.

Table 4. Obstacles or No Robot's Position and Error

Serial number	Existence of obstacles	Initial position (x, y, z)/mm	Target position(x, y, z) /mm	Actual position(x, y, z) /mm	Error /mm	Whether meet requirement or not
1	no	(1100,0,750)	(-1850,3200,750)	(-1849,3200,751)	1.4	meet
2	no	(1100,0,750)	(-1850,3200,750)	(-1851,3202,749)	2.4	meet
3	yes	(1100,0,750)	(-1850,3200,750)	(-1853,3200,746)	5	meet
4	yes	(1100,0,750)	(-1850,3200,750)	(-1854,3198,754)	6	meet

6. Conclusion

(1) To adopt the standard Markov Decision Process to perform the palletizing robot path planning problem can appear the explosion problem of state space that influences on the planning efficiency. We introduce Hierarchical Markov Decision Process algorithm, which can greatly increase the efficiency and the speed of the palletizing robot path planning.

(2) The kinematics model of palletizing robot is established based on the D-H method. We can reach the space position of end actuators through the coordination transformation matrix of various connecting rods.

(3) According to the obstacles space information, the state space of palletizing robot is divided into state cluster. We roughly plan the path of end actuators, and then gradually finely plan the movement trajectory of end actuators in layers.

(4) In order to validate the ability to avoid obstacles of the palletizing robot, we should increase the number of obstacles in the movement path, or change the shape of the obstacles in the experiment environment to further test the effect of the Hierarchical Markov Decision Process algorithm.

Acknowledgements

This research has been partially supported by the National Natural Science Funds of China (60674100) and Nanjing University of Aeronautics and Astronautics Scientific Research Funds (NS2010069).

References

- [1] Kaelbling L, Littman M, Cassandra A. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*. 1998; 101(1): 99–134.
- [2] Hong Y, Wang H, Bian X. Global Path Planning for AUV Based on Hierarchical Markov Decision Process. *Journal of System Simulation*. 2008; 20(9): 2361-2363.
- [3] Hong Y, Fang J. Hierarchical Markov decision processes based path planning for UAV in three-dimensional environment. *Journal of Beijing University of Aeronautics and Astronautics*. 2009; 35(1): 100-103.
- [4] Liu C, Yang G, WU H. Three Dimensional Path Generation Method of Flying Robot for Overhead Powerline Inspection. *Journal of Tongji University (Natural Science)*. 2010; 38(12): 1822-1827.
- [5] Bram Bakker, Zoran Zivkovic, Ben Krose. Hierarchical Dynamic Programming for Robot Path Planning. *IEEE International Conference on Intelligent Robots and Systems*. 2005: 3720-3725.
- [6] Milos Hauskrecht, Nicolas Meuleau, Craig Boutilier. Hierarchical Solution of Markov Decision Processes using Macro-actions. *Uncertainty in Artificial Intelligence*. 1998; 109(9): 220-229.
- [7] Jennifer B, Leslie PK, Tomas LP. *Approximate Hierarchical Solution of Large Markov Decision Processes using Macro-actions*. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence. 1998: 220-229.
- [8] Zoran Zivkovic, Bram Bakker, Ben Krose. Hierarchical Map Building Using Visual Landmarks and Geometric Constraints. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005: 7-12.
- [9] Chiddarwar SS, Babu NR. Multi-agent System for Off-line Coordinated Motion Planning of Multiple Industrial Robots. *International Journal of Advanced Robotic Systems*. 2011; 8(1): 102-112.
- [10] Benoit Morisset, Malik Ghallab. Learning how to combine sensory-motor functions into a robust behavior. *Artificial Intelligence*. 2008; 172(4): 392-412.