Sensors & Transducers, Vol. 177, Issue 8, August 2014, pp. 69-74

Sensors & Transducers

© 2014 by IFSA Publishing, S. L.
http://www.sensorsportal.com

Bidirectional RRT Algorithm for Collision Avoidance Motion Planning of FFSR

*** Huazhong Li, Yongsheng Liang**
Software Department, Shenzhen Institute of Information Technology,
No. 2188 Longxiang Road, Longgang District, Shenzhen, 518029, Guangdong, China
* Tel.: +86-0755-89226179, fax: 0755-89226555
E-mail: lihz@sziit.edu.cn, liangys@sziit.edu.cn

Received: 8 March 2014 /Accepted: 31 July 2014 /Published: 31 August 2014

Abstract: The nonholonomic kinematics characteristic of free-floating space robot (FFSR) in a microgravity environment is a special difficulty in its motion planning. First, kinematics model for FFSR system and state transition equation have been established by applying linear momentum and angular momentum conservation laws followed by FFSR in a microgravity environment. Second, aiming at the nonholonomic characteristic of FFSR, a collision avoidance motion planning algorithm based on bidirectional rapidly-exploring random tree (RRT) has been proposed. This paper focuses on explaining the basic theory of FFSR motion planning, the basic principle for such core algorithms as EXTEND, CONECT and RRT-Connect and the realization of such algorithms. Finally, the correctness of the algorithms proposed has been verified via computer simulation. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Free-floating space robot (FFSR), Rapidly-exploring random tree (RRT), Motion planning, Nonholonomic, Collision avoidance, Kinematics.

1. Introduction

With the great progress of human's exploration on the moon, free-floating space robot (FFSR) has become a research focus in space explorations by every country [1, 2]. FFSR is composed of satellite body and mechanical arm mounted on the body (as shown in Fig. 1); it can fly freely and complete all kinds of operations in space. FFSR mainly fulfils such tasks as invalid satellite capturing, maintaining and redeploying as well as space station building. FFSR system is operated in a space microgravity environment, and it is constrained by linear momentum conservation law and angular momentum conservation law in which the former is a holonomic constraint and the latter is a nonholonomic constraint. This will make FFSR system a nonholonomic dynamics system. It can be reflected as follows: when FFSR moves along different paths from the same configuration, the attitude angles of its satellite body will vary even though the final vectors of joint angles of mechanical arm are the same. Many scholars have made lots of researches on the impact of nonholonomic characteristic of FFSR on its motion control. The most representative achievements include resolved motion rate control method [3] proposed by Y. Umetani and K. Yoshida, virtual manipulator (VM) method [4, 5] proposed by Z. Vafa and S. Dubowsky and bidirectional method (positive solution and inverse solution) based on kinematics and dynamics [6] proposed by Y. Nakamura and R. Mukherjee. However, the existing researches are mainly about motion control of end-effector and attitude angles of satellite body as well as motion planning without obstacles in operating environment, and few researches are involved in collision avoidance motion planning of nonholonomic FFSR.

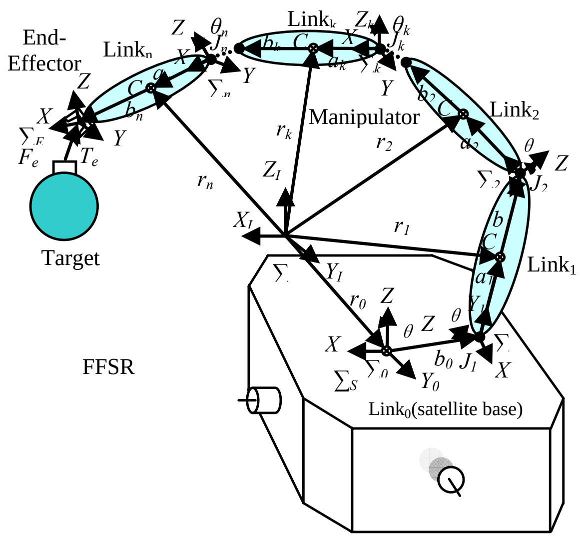

Fig. 1. Kinematics model of free-floating space robot.

Apparently, the traditional motion planning algorithms such as grid method [7], octree search method, A* search method, C-Space path search method, Roadmap method and artificial potential field method [8] cannot be directly applied in nonholonomic dynamics system. Therefore, Lavalle et al. proposed rapidly-exploring random tree (RRT) algorithm to solve the problems in motion planning of nonholonomic system [9-11]. RRT algorithm is a single query random search method. In this algorithm, all images of configuration space (C-Space) are not necessarily built, and only C-Space sampling is conducted by certain strategies (such as even random, low discrete degree, low difference and various resolutions). This algorithm can effectively avoid the exponential explosion of calculated amount caused by increased degrees of freedom (DoFs) of the robot in the traditional algorithms. According to the current environmental state, it does not require preprocessing before planning and can adapt to the complex and changing external environments. Therefore, it has been successfully applied in many nonholonomic systems such as motion planning of mobile robot.

A FFSR collision avoidance motion planning algorithm based on bidirectional RRT is proposed in this paper, which can efficiently and reliably solve the problems in collision avoidance motion planning of nonholonomic FFSR system with high DoFs in the complex and static environment.

2. Kinematics Model for FFSR

When moving in a microgravity environment, FFSR system follows linear momentum and angular momentum conservation laws. According to these two laws and through a series of inference, FFSR kinematic model can be obtained as follows [3, 12, 13]:

$$\dot{\mathbf{\Theta}} = \mathbf{G}(\mathbf{\Theta}, \mathbf{\theta}_M)\dot{\mathbf{\theta}}_M , \tag{1}$$

where $\mathbf{\Theta} = [\alpha, \beta, \gamma]^T \in \mathbf{R}^3$ represents the attitude angle of FFSR's satellite body, and $\mathbf{\theta}_M = [\theta_1, \ldots, \theta_n]^T \in \mathbf{R}^n$ represents the vector of joint angle of FFSR's mechanical arm. Formula (1) describes the impact of joint motion of mechanical arm on the satellite body during the motion of FFSR system. $\mathbf{G}(\mathbf{\Theta}, \mathbf{\theta}_M)$ matrix is related to the attitude angle $\mathbf{\Theta}$ of current FFSR's satellite body and the configuration of mechanical arm. It is a kind of complex nonlinear mapping about FFSR configuration $\mathbf{q} = [\mathbf{\Theta}^T, \mathbf{\theta}_M{}^T]^T = [\alpha, \beta, \gamma, \theta_1, \ldots, \theta_n]^T$, which is different from the robot system with pedestals fixed on the ground. Formula (1) is not integral. That is, the attitude angle $\mathbf{\Theta}$ cannot be integrally expressed as a function for the vector of joint angle $\mathbf{\theta}_M$. Therefore, numerical integration must be conducted along the motion paths of FFSR according to formula (1) in order to obtain the attitude angle of FFSR system. This indicates that the attitude angle $\mathbf{\Theta}$ of FFSR system is related to the vector of joint angle $\mathbf{\theta}_M$ upon motion completion and historical paths of each joint motion of FFSR's mechanical arm. The nonholonomic characteristic of FFSR will bring challenges to FFSR collision avoidance motion planning.

3. Bidirectional RRT Motion Planning of FFSR

FFSR motion planning can be defined as: Given start configuration $\mathbf{q}_{start} = [\mathbf{\Theta}^T_{start}, \mathbf{\theta}^T_{Mstart}]^T$ and goal configuration $\mathbf{q}_{goal} = [\mathbf{\Theta}^T_{goal}, \mathbf{\theta}^T_{Mgoal}]^T$, a motion sequence for the vector of joint angle of mechanical arm $\mathbf{\theta}_M = [\theta_1, \ldots, \theta_n]^T \in \mathbf{R}^n$ can make FFSR system move according to the sequence, reach the goal configuration $\mathbf{q}_{goal}$ and avoid the collision with the obstacles in environment.

In the bidirectional RRT algorithm, the planned FFSR dynamics system can be abstracted to be a control system which has a certain state, can respond to the input and conduct state transition and output it. If the vector of joint angle of mechanical arm $\dot{\mathbf{\theta}}_M = [\dot{\theta}_1, \ldots, \dot{\theta}_n]^T \in \mathbf{R}^n$ is set to express the input of FFSR system, and $\mathbf{q} = [\mathbf{\Theta}^T, \mathbf{\theta}^T_M]^T$ represents the current state of FFSR system, then the state transition equation of FFSR system can be expressed as:

$$\dot{\mathbf{q}} = f(\mathbf{q}, \dot{\mathbf{\theta}}_M), \tag{2}$$

Combining Formula (1), the state equation of FFSR can be expressed as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{\Theta}} \\ \dot{\mathbf{\theta}}_M \end{bmatrix} = \begin{bmatrix} \mathbf{G}(\mathbf{\Theta}, \mathbf{\theta}_M) \\ \mathbf{E} \end{bmatrix} \dot{\mathbf{\theta}}_M = \mathbf{H}(\mathbf{q}) \dot{\mathbf{\theta}}_M , \quad (3)$$

Formula (3) indicates that a linear relation is shown between the state change rate $\dot{\mathbf{q}}$ and the input $\dot{\mathbf{\theta}}_M$ in FFSR system, which is called a symmetric system. That is, if the system starts from the configuration node $\mathbf{q}_{start}$ and reaches the configuration node $\mathbf{q}_{goal}$ along a certain path τ, then the system can start from the configuration $\mathbf{q}_{goal}$ and return to the configuration node $\mathbf{q}_{start}$ [6] along the original path τ. According to this characteristic, FFSR configuration can be defined as: after its state, configuration space $\mathbf{C}$ [14] of FFSR can be divided into two parts, namely feasible free space $\mathbf{C}_{free}$ and forbidden space (obstacle space) $\mathbf{C}_{obs}$. When FFSR state is located in $\mathbf{C}_{obs}$, it indicates that FFSR collides with the obstacles. When it is located in $\mathbf{C}_{free}$, it indicates that FFSR does not collide with the obstacle.

Bidirectional RRT motion planning researched in this paper includes two core algorithms, namely EXTEND and CONECT. RRT-Connect algorithm is made up of EXTEND and CONECT. The basic principles of each algorithm and the realization of pseudo codes are described as follows.

Bidirectional RRT algorithm is the combination of forward RRT search and backward RRT search. A tree grows up from start configurations while other tree grows up from the goal configurations. Opposed to single tree RRT motion planning, it will be more convenient and efficient to use bi-directional RRT for motion planning, because the bi-directional search tree does not only start from the start point but also take the goal point as its root. Two quickly extensible search trees are built respectively, and the existence of feasible paths is determined by whether there is any cross point between the two trees. For bi-directional RRT, its quickly extensibility needs to be guaranteed, and the cross connection of the two spanning trees must be taken into consideration. In addition, the balanced growth of these two spanning trees needs to be guaranteed. Only in this way can the consistence of building time and space for the two trees be ensured.

3.1. EXTEND Algorithm

The entire C-Space is continually explored in RRT algorithm by generating RRT trees. Taking the start configuration $\mathbf{q}_{start}$ of FFSR as a root, a RRT search tree $\mathbf{T}_{start}$ is built. Taking the goal configuration $\mathbf{q}_{goal}$ as a root, anther RRT search tree $\mathbf{T}_{goal}$ is built. In each step of the algorithm, a random configuration $\mathbf{q}_{rand}$ is randomly sampled in $\mathbf{C}_{free}$, and then EXTEND and CONNECT algorithms are applied to extend RRT trees until the two trees are intersected (as shown in Fig. 2).

Pseudo code of EXTEND $(\mathbf{T}, \mathbf{q}_{rand}, \mathbf{q}_{new})$ algorithm is described as follows.

Algorithm EXTEND ($\mathbf{T}$, $\mathbf{q}_{rand}$, $\mathbf{q}_{new}$) {

$\mathbf{q}_{near}$ =NEAREST_NEIGHBOR ($\mathbf{T}$, $\mathbf{q}_{rand}$);

if ($\mathbf{q}_{new}$ =NEW_ CONFIGURATION ($\mathbf{q}_{rand}$, $\mathbf{q}_{near}$, $\dot{\mathbf{\theta}}_{Mnew}$)) {

$\mathbf{T}$ .AddVertex ($\mathbf{q}_{new}$);

$\mathbf{T}$ .AddEdge ($\mathbf{q}_{near}$, $\mathbf{q}_{new}$, $\dot{\mathbf{\theta}}_{Mnew}$);

if($\mathbf{q}_{new} == \mathbf{q}_{rand}$) {

return REACHED;}

else{

return ADVANCED;}}

return TRAPPED;}

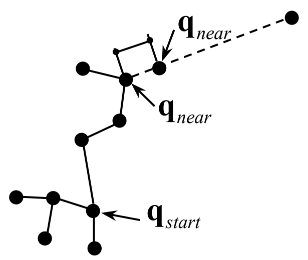

Fig. 2. Diagram for EXTEND algorithm.

3.2. CONNECT Algorithm

As for random sampling configuration $\mathbf{q}_{rand} = [\mathbf{\Theta}^T, \mathbf{\theta}_M^T]^T$, the configuration node $\mathbf{q}_{near}$ that is nearest to $\mathbf{q}_{rand}$ on the RRT tree to be extended is selected in EXTEND algorithm. Starting from $\mathbf{q}_{near}$, the configuration $\mathbf{q}_{new}$ that is nearest to $\mathbf{q}_{rand}$ in all configuration states for FFSR system reached within a tiny time slice Δt is calculated in the algorithm. When Δt is small enough, $\mathbf{q}_{new}$ can be obtained via the following formulas:

$$\dot{\mathbf{\theta}}_M = \mathbf{G}(\mathbf{q}_{rand})^{+} (\mathbf{\theta}_M - \dot{\mathbf{\theta}}_{Mnear}) , \quad (4)$$

$$\mathbf{q}_{new} = \mathbf{q}_{near} + \mathbf{H}(\mathbf{q}_{rand}) \dot{\mathbf{\theta}}_M \Delta t , \quad (5)$$

where $\mathbf{G}(\mathbf{q}_{rand})^{+}$ represents a pseudo-inverse matrix for $\mathbf{G}(\mathbf{q}_{rand})$. After $\mathbf{q}_{new}$ is obtained, a collision detection will be conducted in the algorithm. There may be three circumstances as follows:

If failed, $\mathbf{q}_{new}$ is not in $\mathbf{C}_{free}$;

If reached, $\mathbf{q}_{new}$ has no collision, and its distance from $\mathbf{q}_{rand}$ is less than a certain threshold value $\Delta\|\mathbf{q}\|_{\min}$;

If advanced, $\mathbf{q}_{new}$ without collision is found. A step is advanced towards $\mathbf{q}_{rand}$ based on the original $\mathbf{q}_{near}$.

In (ii) and (iii), $\mathbf{q}_{new}$ is added to the current RRT tree by EXTEND algorithm. It can be seen that the current RRT tree will externally extend one configuration node when EXTEND algorithm is conducted each time. To accelerate the algorithm speed, CONNECT algorithm is applied. After one configuration sampling node $\mathbf{q}_{rand}$ is given, CONNECT algorithm keeps calling EXTEND algorithm to guide RRT tree to grow towards $\mathbf{q}_{rand}$ until FFSR configurations cannot advance further. The pseudo codes of this algorithm are described as follows:

```
Algorithm CONNECT (T, q_rand, q_new){
do {
    Result=EXTEND(T, q_rand, q_new);
}while(Result!= ADVANCED);
return Result;}
```

3.3. RRT-Connect Algorithm

EXTEND and CONNECT algorithms are used in RRT algorithm, and one of them keeps extending $\mathbf{T}_{start}$ and $\mathbf{T}_{goal}$ until the distance between the configuration $\mathbf{q}_{goal}$ and a certain leaf node is less than the error value. According to the different selections of EXTEND and CONNECT algorithms, RRT algorithm will have various forms. After experiment, RRT-Connect algorithm is more suitable for motion planning of FFSR system [12] (the schematic diagram is shown in Fig. 3):

```
RRT-Connect (q_start, q_goal) {
T_start.init(q_start);
T_goal.init(q_goal);
for ( k=1; k<MAX_sample;k++) {
   q_rand =RANDOM_CONFIGURATION ( );
if(CONNECT(T_start, q_rand, q_new)!=TRAPPED) {
if(CONNECT(T_goal, q_rand, q_new)==ADVANCED)
{
            return Solution(T_start, T_goal);}
  SWAP(T_start, T_goal);}}
return  FAILURE ; }
```

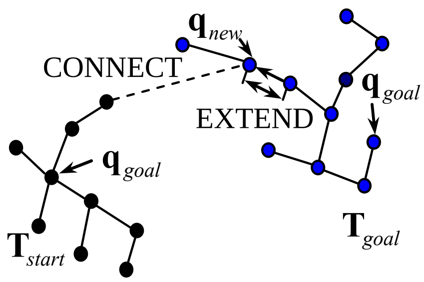

Fig. 3. Diagram for RRT-connect algorithm.

In the algorithm, a randomly sampled configuration node $\mathbf{q}_{rand}$ is firstly generated in each iteration, and then a RRT tree is alternatively selected from $\mathbf{T}_{start}$ and $\mathbf{T}_{goal}$ to execute EXTEND algorithm. Thus, a new configuration node $\mathbf{q}_{new}$ is obtained, and then another RRT tree has CONNECT extension for the new configuration node until the two trees meet each other.

4. Computer Simulation

To verify the effectiveness of bidirectional RRT motion planning algorithm, this algorithm has been designed and realized, and three-dimensional visualization simulation is conducted in this paper. During simulation, a parallel collision detection algorithm based on hybrid bounding box is adopted to judge whether FFSR collides with the obstacles [15-20]. All the simulation experiments were conducted in a computer whose configurations are as follows: Intel Pentium Dual Core G640 Processor (dominant frequency: 2.8 GHz), 4 GB RAM, 3 MB three-level buffer. The compilation tool is VS2008 C++.

The simulation system framework is built by typical MCV design mode in this paper (as shown in Fig. 4).

Some of the key functions for bidirectional RRT motion planning software are described as follows:

i) CBBRrt: It is derived from CMotionPlanner, realizing the sample-based bidirectional RRT planner proposed in this paper and adopting extension and improvement strategy.

ii) CSpace target can be used to set the complex system configurations, robot's DoFs and collision detection strategy etc.

iii) CRobot: It defines the robot target, which can load the model data from XML files.

iv) CObstacle: It defines the obstacle. Call createBox ( ) or load vi files from XML, create obstacle model.

v) CSpace: It defines the C-Space of motion planning, including CRobot, CRobotNodeSet and CDManager.

vi) CSpaceSampled: It is derived from CSpace, indicating sampling-based C-Space. CRobotNodeSet: It appoints the size and boundary of robot's node set in CSpace.

vii) CDManager: It is used to detect whether CRobot collides with the obstacle of CObstacle, realizing the collision detection algorithm proposed in this paper.

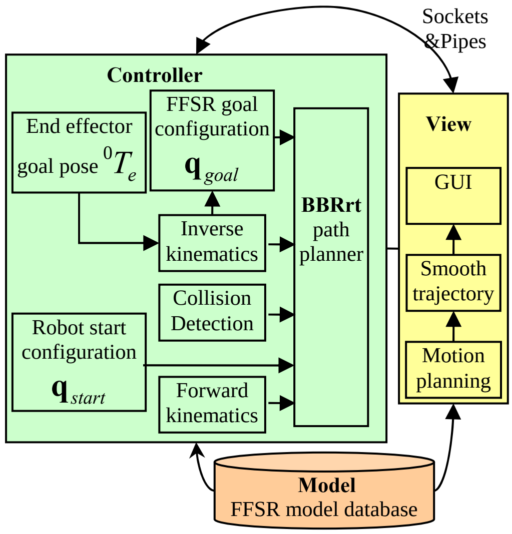

Fig. 4. Simulation system framework.

Fig. 5 is three-dimensional scene of bidirectional RRT motion planning of FFSR.

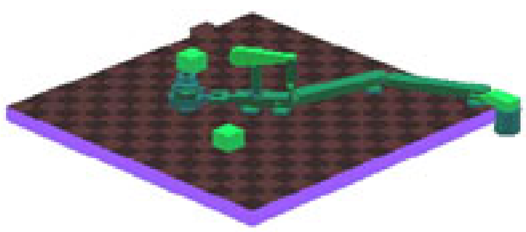

Fig. 5. Three-dimensional scene of bidirectional RRT motion planning of FFSR.

6. Conclusions

Bidirectional RRT algorithm is quite suitable for the motion planning of nonholonomic dynamics system. A large number of simulation experiments and researches show that the bidirectional RRT algorithm proposed in this paper is applicable to FFSR collision avoidance motion planning. Due to the special nonholonomic characteristic of FFSR, the calculation of the configuration $\mathbf{q}_{new}$ node that is nearest to $\mathbf{q}_{rand}$ is dependent on the pseudo-inverse matrix of $\mathbf{G}(\mathbf{q}_{rand})$, and it will be affected by the singularity of dynamics during motion planning. Since the motion of FFSR's mechanical arm joints will interfere with the configurations of satellite body, and the sampling time interval needs to be small enough. This has increased the sampling quantity of configuration space and reduced the planning speed. In addition, a lot of kinematics value calculations are required for the configuration nodes extended by bidirectional RRT tree, which has increased the calculated amount of motion planning. How to effectively reduce the sampling configuration nodes and improve the planning efficiency and the theory and argument of probabilistic completeness is the key point for next step.

Acknowledgements

This work was supported in part by a grant from Harbin Institute of Technology Robotics and System National Key Laboratory (SKLRS-2012-MS-06), Shenzhen Science and Technology Program (JC201006020820A and JCYJ20120615101640639), Team of Scientific and Technological Innovation of Shenzhen Institute of Information Technology (CXTD2-002), Natural Science Foundation of Guangdong Province, China (S2013010013779 and S2011040000672), Guangdong Vocational Education Information Technology Fund (XXJS-2013-1019).

References

[1]. F. Aghili, Coordination control of a free-flying manipulator and its base attitude to capture and detumble a non-cooperative satellite, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2365-2372.

[2]. Satoko Abiko and Gerd Hirzinger, Computational efficient algorithms for operational space formulation of branching arms on a space robot, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Acropolis Convention Center Nice, France, September, 2008, pp. 3312-3317.

[3]. Y. Umetani, K. Yoshida, Resolved motion rate control of space manipulators with generalized Jacobian matrix, *IEEE Transactions on Robotics and Automation*, Vol. 5, Issue 3, 1989, pp. 303-314.

[4]. Z. Vafa, S. Dubowsky, The kinematics and dynamics of space manipulators: the virtual manipulator approach, *International Journal of Robotics Research*, Vol. 9, Issue 4, 1990, pp. 3–21.

[5]. Z. Vafa, S. Dubowsky, On the dynamics of space manipulator using the virtual manipulator with

application to path planning, *Journal of the Astronautical Science*, Vol. 38, Issue 4, 1990, pp. 441–472.

[6]. Y. Nakamura, R. Mukherjee, Nonholonomic path planning of space robots via a bidirectional approach, *IEEE Transactions on Robotics and Automation*, Vol. 7, Issue 4, 1991, pp. 500–514.

[7]. Tae-Kyeong Lee, Sang-Hoon Baek, Young-Ho Choi, Se-Young Oh, Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation, *Robotics and Autonomous Systems*, Vol. 59, Issue 10, 2011, pp. 801-812.

[8]. Y. Hwang and N. Ahuja, Gross motion planning – a survey, *ACM Computing Surveys*, Vol. 24, Issue 3, 1992, pp. 219-291.

[9]. S. M. LaValle and J. J. Kuffner, Randomized kinodynamic planning, *International Journal of Robotics Research*, Vol. 20, Issue 5, 2001, pp. 378–400.

[10]. S. M. LaValle, Planning algorithms, *Cambridge University Press*, 2006.

[11]. S. M. LaValle and J. J. Kuffner, Rapidly-exploring random trees: Progress and prospects, Algorithmic and Computational Robotics: New Directions. A. K. Peters, *Wellesley*, MA, 2001, pp.293–308.

[12]. Li Hua-Zhong, Liang Yong-Sheng, Hong Bing-Rong, Modeling and simulation of continual motion trajectory control for space robot, *Robot*, Vol. 30, Issue 5, 2008, pp. 410-415.

[13]. Li Hua-Zhong, On motion control with minimum attitude disturbance for redundant space robots, *Computer Engineering and Applications*, Vol. 45, Issue 19, 2009, pp. 12-16.

[14]. J. J. Kuffner, and S. M. LaValle, RRT-connect: An efficient approach to single-query path planning, in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000, Vol. 2. pp. 995-1001.

[15]. S. Gottschalk, M. C. Lin and D. Manocha, OBBTree: A hierarchical structure for rapid interference detection, in *Proceedings of SIGGRAPH'96*, 1996, pp. 171-180.

[16]. Anwesha Mukherjee and Debashis De, DAS: An Intelligent Three Dimensional Cost Effective Movement Prediction of Active Users in Heterogeneous Mobile Network, *Journal of Computational Intelligence and Electronic Systems*, Vol. 2, Issue 1, 2012, pp. 31-47.

[17]. Rajeev Ranjan and Shirshu Varma, Collision-free time synchronization for multi-hop wireless sensor networks, *Journal of Computational Intelligence and Electronic Systems*, Vol. 2, Issue 1, 2012, pp. 200-206.

[18]. Li Huazhong, Liang Yongsheng, Wang Meini, Dan Tangren, Design and implementation of improved RRT algorithm for collision free motion planning of high-dimensional robot in complex environment, in *Proceedings of the 2nd International Conference on Computer Science and Network Technology (ICCSNT'2012)*, Changchun, China, 2012, pp. 1391-1397.

[19]. Li Huazhong, Liang Yongsheng, Dan Tangren, Zheng Hongying, Wu Xianfeng, Collision-free motion planning algorithm based on RRT for robot, *Journal of Shenzhen Institute of Information Technology*, Vol. 10, Issue 3, 2012, pp. 20-27.

[20]. Chuan Hong, Khaled Benkrid, Xabier Iturbe, and Hanaa Hussain, Efficient run-time system support for high performance reliable reconfigurable systems, *Journal of Computational Intelligence and Electronic Systems*, 2012, Vol. 2, Issue 1, pp. 213-219.

2014 Copyright ©, International Frequency Sensor Association (IFSA) Publishing, S. L. All rights reserved.
(http://www.sensorsportal.com)