Multidisciplinary Journal for Education,
Social and Technological Sciences

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

OPTIMIZATION APPROACHES FOR ROBOT TRAJECTORY PLANNING

Carlos LLOPIS-ALBERT[*], Francisco RUBIO[**], Francisco VALERO[**]

* Departamento de Ingeniería Mecánica y de Materiales. Universitat Politècnica de València, Camí de Vera s/n, Valencia, 46022, Spain. Email: cllopisa@upvnet.upv.es

** Centro de Investigación en Ingeniería Mecánica. Universitat Politècnica de València, Camí de Vera s/n, Valencia, 46022, Spain. Emails: frubio@mcm.upv.es; fvalero@mcm.upv.es

Received: 2017-11-03; Accepted: 2018-02-13

Abstract
The development of optimal trajectory planning algorithms for autonomous robots is a key issue in order to efficiently perform the robot tasks. This problem is hampered by the complex environment regarding the kinematics and dynamics of robots with several arms and/or degrees of freedom (dof), the design of collision-free trajectories and the physical limitations of the robots. This paper presents a review about the existing robot motion planning techniques and discusses their pros and cons regarding completeness, optimality, efficiency, accuracy, smoothness, stability, safety and scalability.

Keywords
Algorithms, Optimal Trajectory, Kinematic and Dynamic constraints, Minimum time, Energy, Obstacle avoidance.

1. Introduction

Trajectory planning is moving a robot between two different configurations over time in order to perform a certain task while fulfilling robot's constraints. A certain configuration entails a set of joint angles of the robot manipulator and the set of all possible joint angles is called the configuration space. The constraints encompass the physical

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

limitations of the robot. They include geometric constraints, which can be expressed in terms of the robot joint angles (i.e., bounds on the joint angles, avoidance of collision with the environment). They also cover kinematics and dynamics constraints that include higher-order time derivatives of the joint angles (i.e., bounds on the joint velocities, accelerations, torques, or motor current inputs).

Furthermore, the task should be performed between the successive configurations in an efficiently and accurately way while optimizing a certain objective, such as minimizing the path traveling distance or execution time, energy consumption (or actuator effort) and jerk or maximizing the smoothness ([1], [2], [3], [4], [5]).

This article reviews the most significant methodologies in trajectory planning of mobile robots with kinematics and dynamics constraints and optimization objectives.

On the one hand, inverse kinematics finds a continuous set of intermediate joint angles of the robot arms between the starting and goal joint angles that allow achieving the desired end-effector position and orientation while avoiding collisions. On the other hand, through a time parameterization the algorithms allow to meet the torque bounds and/or optimizing the execution time or the energy consumption. Finally, a controller takes the inputs and adjusts its outputs by defining a sequential motion law so that the robot can carry out its task. The inputs cover the geometric path, the kinematic and dynamic constraints, while the output are the trajectory of the joints, expressed as a time sequence of position, velocity and accelerations.

2. Trajectory planning algorithms

2.1. Classic approaches

Path planning entails the generation of a geometric path without a time law, while the trajectory planning assigns a time law to the geometric path. Two main categories of trajectory planning algorithms can be distinguished in accordance to the available

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

information, namely off-line and on-line. Off-line robots compute the entire trajectory to the goal before motion begins (i.e., information about obstacles is known in advance), and may lead to globally optimal solutions when the environment is fully known. In this category different issues are analyzed, such as optimality (local and global), complete (a solution will be found if exists), and computational cost and efficiency (allow changes without recomputing or replanning everything).

On-line robots generate the trajectory to the goal incrementally during motion, and lead to locally optimal solutions at best. In this case, the mobile robot obtains the information through sensors while it moves through the environment. In this category the issues raised are completeness (is the robot guaranteed to reach the goal if a solution exists), computational cost and efficiency at each step, and optimality (how far is a solution from the optimal and is it bounded by an upper limit).

There exists a large variety of approaches to trajectory planning. The most important classical techniques are bug-like algorithms, the combinatorial methods, potential field methods and sampling-based methods.

The bug-like algorithms are among the earliest and simplest sensor-based algorithms with reasonable results [6]. Robot is assumed to be a point in the plane with perfect positioning and with the workspace bounded. They have a contact sensor, which detects the obstacle boundary if it touches it. They are straightforward to implement since entail a movement towards the goal, unless an obstacle is encountered. In that case, they circumnavigate the obstacle until motion toward the goal is again allowable. This is achieved by measuring the distance between any two points.

Combinatorial methods are geometric representation planners, based on the configuration space as the fundamental concept, which are used by most off-line robots.

The geometric representations of the environment may consist of roadmaps or graphs that capture the topology of the free space, generated by different well-known methods such as a Voronoi diagram ([7], [8]); a visibility graph ([9], [10], [11], [12]), a

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

tangent graph [13]; cell decomposition and grid method ([14], [15], [16], [17], [18], [19], [20], [21]); Silhouette [22], and the Subgoal Network [23].

They differ in the way it represents the free space (non-collision space), but all are based on a connected network of path segments that can be traversed from start to goal. The main computational effort in these approaches is the representation of the free space, which includes the mapping of obstacles. Once the roadmap is constructed, the search for the shortest trajectory is carried out by using standard graph search techniques such as Dijkstra's search [24] or A* [25].

These methods have the advantage of providing that the general motion planning problem is NP-complete, but they have the disadvantages of being too slow to be used in practice, especially in high-dimensional problems, and to require an explicit representation of obstacles, which is very complicated to obtain in most practical problems.

Another approach is to overlay a uniform grid over the search space and represent the entire space by an undirected graph [2]. These methods assign high costs to edges that intersect obstacles, which allows to effectively separate between inaccessible nodes and nodes in the free space. The resolution of this method is complete, as all approaches based on a discrete representation of the search space, which implies that at low grid resolutions paths that pass through tight spaces between obstacles can be disregarded. Instead an increase in the graph resolution would lead to high computational effort. As a disadvantage, the number of nodes for the uniform grid representation is much greater than for the roadmap-based algorithms. However, this approach is applicable to problems where obstacles are not clearly defined, such as for mobile robots.

The potential field method constructs a potential field which is high near the obstacles and low at the goal configuration ([1], [26]). The robot is guided towards the goal configuration while avoiding the obstacles by letting its configuration evolve in that potential field. That is, the robot is attracted towards the goal configuration and repulsed from the obstacles. The gradient is a vector which points in the direction that locally

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

maximally increases the artificial potential field and the local variations of the robot reflect the structure of the free space.

This method allows real-time control, but the possibility of getting trapped in a local minimum of the potential field prevents its use in highly cluttered environments.

Sampling-based methods probably are the most widely used methods for trajectory planning because they are efficient and robust algorithms.

Contrary to previous algorithms, the sampling-based planners accepts probabilistic completeness, i.e., the goal may not be reached in a finite time; accepts any solution, not necessarily the optimal; and neglects the explicit geometric representation of the free configuration space in terms of roadmaps or graphs. A roadmap is a graph whose vertices are configurations of free space and connects them by a path entirely contained in the free space. There are two ways of building the roadmap, i.e., by a deterministic or probabilistic approach.

In the Probabilistic Roadmap planner (PRM) instead of following a regular grid, samples are taken at random in free space. Since there is no a priori grid structure, there are several methods for choosing the pairs of vertices to make the connection. This approach works very well for a wide variety of problems ([27], [28]), and it is based on the fact that checking if a single robot configuration is in the free space is less computationally expensive. PRM creates a roadmap in the free space using a coarse sampling to obtain the nodes of the roadmap and a fine one to obtain the roadmap edges (i.e., the free paths between node configurations). Then, planning queries can be answered by connecting the initial and goal configurations to the roadmap. A uniform random distribution ensures the probabilistic completeness of the planner [29]. There are other sampling-based planners depending on the node sampling scheme that may be more effective for single-query planning, such as the Expansive-Spaces Tree planner (EST) [30] and the Rapidly-exploring Random Tree planner (RRT) [24]. There are also methods based on a combination of the previous methods, such as the Sampling-Based Roadmap of Trees (SRT) method, which

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

constructs a roadmap using a PRM and single-query trees. It has been observed that for very difficult path planning problems, single-query planners need to construct large trees in order to find a solution. In some cases, the computational cost of constructing a large tree may be high and it is worthy to use a multiple-query planning.

Sampling-based methods are able to deal with robots with many degrees of freedom and constraints. For instance, kinematic and dynamic constraints, energy and stability constraints, closed-loop kinematics, visibility and constraints, and reconfigurable robots.

As a summary, the main disadvantages of classic approaches that make them inefficient in practice are that they entail a high computational cost to determine a feasible collision-free path in high dimensions; tend to get locked in local optimal solution; lead to non-deterministic polynomial time hard problems (NP-hard) for trajectory planning of mobile robots with multiple obstacles [22]; and the solution is quite complicated when the environment is dynamic and complex [31]. These drawbacks prevent their use in complex environments.

2.2. Heuristic approaches

To solve the aforementioned drawbacks of classic approaches the heuristic and metaheuristic approaches have been developed. They encompass methods such as Probabilistic Roadmaps (PR); Rapidly-exploring Random Trees (RRT); Ant Colony Optimization (ACO), that relies on the foraging behavior of ants for finding the shortest path to the food source ([32], [33], [34]); Simulated Annealing (SA), which is a heuristic random search approach that resembles the cooling process of molten metals through annealing ([35], [36]); Neural Network [37]; Genetic Algorithms (GA), which are based on the mechanics of natural genetics and selection ([38], [39], [40]); Particle Swarm Optimization (PSO), which are inspired by social behavior of bird flocking or fish schooling and are easier to implement than GA and with a fewer parameters to be adjusted ([41], [42], [43], [44], [45], [46]); Stigmergy, which is a mechanism of indirect

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

coordination, through the environment, between agents or actions [47]; Wavelet, which is based on wave-like oscillation theory [48]; Fuzzy Logic, which is a form of many-valued logic where the truth values of variables may be any real number between 0 and 1 [49]; and Tabu Search, which is a local-search method used for mathematical optimization [36].

Heuristic algorithms do not ensure to find a solution, but when they do it is performed much faster than deterministic methods.

2.3. Decision-making process techniques for trajectory planning

There are several approaches for tackling the problem of predicting the trajectory of a moving object when its exact geometric description and information about its environment is not available. In such cases, the information about the environment derives from measurements provided by a set of imperfect noisy sensors. Therefore, the trajectory planning is carried out under uncertainty, which needs to be modelled.

This uncertainty has an effect on the predictability about the current and future states (in either discrete or continuous state spaces and continuous time) of the robot and its environment. Those states are based on the initial conditions, sensors, and the memory of formerly applied actions. Therefore, trajectory planning methods under uncertainty cover problems such as localization, map building, pursuit-evasion and manipulation [2].

Some methods are able to account for the uncertainty and the decision-making process in a greater or lesser extent. For instance, the worst-case, expected-case or probabilistic models, game theory analyses (with players with conflicting goals) and more complex techniques such as sequential decision making (which is a sequence of basic decision-making problems), control theory and artificial intelligence.

Probabilistic estimation methods rely on probability density function (PDF) of the robot location over the space instead of a deterministic location, which allows dealing with uncertainties. The aim is to keep a position PDF over all possible robot poses. An efficient

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

example is the Kalman filter, which provides a recursive method for estimating the state of a noisy dynamical systems [50]. This is carried out by means of Bayesian inference and estimating a joint probability distribution over the variables for each timeframe. Its output is a Gaussian probability density function (PDF) of likely robot positions instead of a single position estimate, with the mean and covariance of the error covariance matrix a distribution.

Another approach for tracking mobile robots under dynamic environments is by means of the Markov process or Markov decision process (MDP), which also make use a probabilistic framework for dealing with decision making in situations where outcomes are partly random and partly under the control of a decision maker It has the advantage of generating an optimal path, but has the disadvantage of limiting the robot to choose from a finite set of action. This lead to a non-smooth path. However, Fuzzy Markov decision processes (FDMPs) are able to generate smooth trajectories using a fuzzy inference system [51].

Bayesian methods uses the same iterative prediction-update process than in the Kalman filter, but they do not rely on its restrictive assumptions [52]. The pros are that they can use nonlinear models for both trajectory planning and sensing and an arbitrary distribution instead of a Gaussian. However, this may lead to higher computational cost compared to Kalman filters.

2.4. Mathematical programming

The methods based on mathematical programming deal with obstacle avoidance by means of a set of inequalities on the configuration parameters. Then the motion planning is posed as a mathematical optimization problem that finds a curve between the start and goal configurations minimizing or maximizing a certain objective function, such as minimizing the path traveling distance or execution time, energy consumption (or actuator effort) and jerk or maximizing the smoothness (e.g., [53], [54], [55]). This leads to a

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

complex non-linear optimization problem with many inequality and differential constraints, which needs be solved by a numerical method. Furthermore, multi-objective optimization problems have been developed in the literature through Pareto optimal solutions ([56], [57]).

3. Comparison of approaches

The pros and cons of the different approaches are presented in Table 1. They cover aspects such as completeness (if the path exists, the path and the trajectory are found), optimality (the plan obtained is optimal regarding some parameter, not trapped in a local minimum), efficiency (computational cost of the algorithm, i.e., if it can change world and queries without recomputing everything or replannig from scratch), accuracy (high precision path tracking and control even at high speed), smoothness (i.e., chattering avoidance), stability (dynamically-stable motion planning), safety (for the robot, its environment and humans), scalability (the problem scales well when increasing configuration space dimensions), and execution time (lower times are desirable). However, the optimization approaches for robot trajectory planning are in continuous developing (e.g., [59], [60], [61]).

Table 1. Approaches comparison.

Approach	Pros	Cons
Potential fields	Real-time, good scalability	Not complete, not efficient world and queries updates, path not optimal (local minimum), potential field forces must be set
Cell decomposition	Complete, robust	High computational cost, and high execution time
Visibility graph	Complete and yields minimum length paths, optimal	High computational cost, and high execution time, bad dof scalability, not efficient world and queries updates
Voronoi diagram	Complete and generates roadmap with maximum	Possibly inefficient paths, time, bad *dof* scalability, path not optimal

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

	distance, efficient world and queries updates	
Heuristic approaches	Low execution time, parallel search	Not complete, possibility of providing smooth paths
Exact cell decomposition	Complete	High computational cost, high execution time
Approximate cell decomposition	Robust and useful when only a coarse representation of workspace is available	Not complete
Bug (bug 1 and bug 2)	Complete, easy implementation, parameters easy to adjust	Long paths, high execution time
A*	Complete, optimal grid	Not efficient, bad *dof* scalability, not efficient world and queries updates
Rapidly exploring random tree (RRT)	Complete, semi-efficient world and queries updates, good scalability	Path not optimal
Probabilistic roadmaps (PRMs)	Complete, semi-efficient queries updates, optimal graph, good scalability	Not efficient world updates

4. Conclusions

This paper provides a review about optimal trajectory planning algorithms for autonomous robots. They cover a wide range of aspects such as the kinematics and dynamics of robots, the achievement of collision-free trajectories and the consideration of the physical limitations of the robots. The different motion planning techniques are discussed, and their advantages and disadvantages presented. As a consequence of these pros and cons, diverse solutions can be used for the wide variety of robot's applications.

References

[1] J.C. Latombe, Robot motion planning. Kluwer Academic Publisher, Boston, 1991.

[2] S.M. LaValle, Planning Algorithms. Published by Cambridge University Press, 842p., 2006.

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

[3] S.H. Tang, W. Khaksar, N.B. Ismail, and M.K.A. Ariffi, A Review on Robot Motion Planning Approaches. Pertanika Journal of Science & Technology 20(1), (2012) 15–29.

[4] P. Raja, and S. Pugazhenthi, Optimal path planning of mobile robots: A review. International Journal of Physical Sciences 7(9) (2012) 1314-1320. DOI: 10.5897/IJPS11.1745.

[5] Z. Shiller, Off-Line and On-Line Trajectory Planning. In Carbone, G., and Gomez-Bravo, F. (eds.), Motion and Operation Planning of Robotic Systems, Mechanisms and Machine Science 29 (2015), Springer, Cham. DOI: 10.1007/978-3-319-14705-5_2.

[6] V. Lumelsky and A. Stepanov, Path planning strategies for point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica 2 (1987) 403–430.

[7] C.O. Dunlaing, and C.K. Yap, A retraction method for planning the motion of a disc. Journal of Algorithms 6 (1985) 104-111.

[8] T. Osama and R.J. Schilling, Motion Planning in a Plane Using Generalized Voronoi Diagram. IEEE Transactions on Robotics and Automation 5(2) (1989) 143-150.

[9] T. Lozano-Perez, and M.A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles. Communications ACM, 22(10) (1979) 560–570.

[10] T. Asano, L. Guibas, J. Hershberger, and H. Imai, Visibility-polygon search and Euclidean shortest paths. Paper presented at the Foundations of Computer Science, 26th Annual Symposium on, 21-23 October. Automation. ETFA '06. IEEE Conference on. (1985).

[11] L. Li, T. Ye, M. Tan, and X. Chen, Present state and future development of mobile robot technology research. Robot 24(5) (2002) 475-480.

[12] R. Siegwart, and I.R. Nourbakhsh, Introduction to autonomous mobile robot. Massachusetts Institute of Technology press, Cambridge, U.S.A (2004).

[13] Y.H. Liu, and S. Arimoto, Path planning using a tangent graph for mobile robots among polygonal and curved obstacles. International Journal of Robotics Research 11(4) (1992) 376–382.

[14] T. Lozano-Perez, Spatial planning: A configuration approach. IEEE Transactions on Computers C-32(3) (1983) 108-120.

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

[15] R.A. Brooks, and T. Lozano-Perez, A subdivision algorithm in configuration space for find path with rotation. Proceedings of eighth International Joint Conference on Artificial Intelligence, Karlsruhe, Germany, pp. 799-806 (1983).

[16] J.M. Keil, and J.R. Sack, Minimum decomposition of polygonal objects. Machine Intelligence and Pattern Recognition 2 (1985) 197-216.

[17] D.J. Zhu, and J.C. Latombe, New heuristic algorithms for efficient hierarchical path planning. Technical report STAN-CS-89-1279, Computer Science Department, Stanford University, USA (1989).

[18] D.W. Payton, J.K. Rosenblatt, and D.M. Keirsey, Grid-based mapping for autonomous mobile robot. Robotics and Autonomous Systems 11(1) (1993) 13-21.

[19] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, Anytime dynamic A*: An anytime, replanning algorithm. Proceedings of the International Conference on Automated Planning and Scheduling, pp. 262-271 (2005).

[20] O. Hachour, The proposed genetic FPGA implementation for path planning of autonomous mobile robot. Circuits, Systems and Signal Processing 2(2) (2008) 151-167.

[21] O. Hachour, Path planning of autonomous mobile robot. International journal of systems applications, engineering & development 4(2) (2008) 178-190.

[22] J. Canny, and J. Reif, New lower bound techniques for robot motion planning problems. Proceedings of IEEE Symposium on the Foundations of Computer Science, Los Angeles, California, 49-60(1987).

[23] B. Faverjon, and P. Tournassoud, A local approach for path planning of manipulators with a high number of degrees of freedom. In Proceeding of IEEE International Conference on Robotics and Automation, 1152-1159 (1987).

[24] J.J. Kuffner, and S.M. LaValle, RRT-connect: An efficient approach to single-query path planning. In IEEE International Conference on Robotics and Automation, 995–1001 (2000).

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

[25] Q.-C. Pham, and S. Caron, Y. Nakamura, Kinodynamic planning in the configuration space via velocity interval propagation, in Robotics: Science and System (2013).

[26] O. Khatib, Real time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research 5(1) (1986) 90-98.

[27] L.E. Kavraki, P. Švestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(4) (1996) 566–580.

[28] M. Overmars, and P. Svestka, A probabilistic learning approach to motion planning. In K. Goldberg, D. Halperin, J. C. Latombe, and R. Wilson (eds.). Algorithmic Foundations of Robotics (WAFR), 19–37, A. K. Peters, Ltd (1995).

[29] L.E. Kavraki, J.C. Latombe, R. Motwani, and P. Raghavan, Randomized query processing in robot path planning. Journal of Computer and System Sciences 57(1) (1998) 50–60.

[30] D. Hsu, Randomized Single-Query Motion Planning In Expansive Spaces. PhD Thesis, Department of Computer Science, Stanford University (2000).

[31] K. Sugihara, and J. Smith, Genetic algorithms for adaptive motion planning of an autonomous mobile robot. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, California, pp. 138-143 (1997).

[32] M.M. Mohamad, N.K. Taylor, and M.W. Dunnigan, Articulated Robot Motion Planning Using Ant Colony Optimisation. Paper presented at the Intelligent Systems, 2006 3rd International IEEE Conference on (2006).

[33] T. Guan-Zheng, H.E. Huan, and A. Sloman. Global optimal path planning for mobile robot based on improved Dijkstra algorithm and ant system algorithm. Journal of Central South University of Technology 13(1) (2006) 80-86.

[34] M.A.P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, and P. Melin, Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost evaluation. Applied Soft Computing 9 (2009) 1102-1110.

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

[35] K. Manousakis, T. McAuley, R. Morera, and J. Baras, Using multi-objective domain optimization for routing in hierarchical networks. Proceedings of the International Conference on Wireless Networks, Communication and Mobile Computing, 1460-1465 (2005).

[36] E. Masehian, and M.R. Amin-Naseri, Composite models for mobile robot offline path planning, Mobile robots: Perception and navigation. In Tech Publisher, Croatia (2007).

[37] A. Zhu, and S.X. Yang, A Neural Network Approach to Dynamic Task Assignment of Multirobots. IEEE Transaction on Neural Networks 17(5) (2006) 1278-1287.

[38] J.H. Holland, Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, USA (1975).

[39] D.E. Goldberg, Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Delhi, India (2000).

[40] Z. Qingfu, S. Jianyong, X. Gaoxi, and E. Tsang, Evolutionary Algorithms Refining a Heuristic: A Hybrid Method for Shared-Path Protections in WDM Networks Under SRLG Constraints. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 37(1) (2007) 51-61.

[41] J. Kennedy, and R.C. Eberhart, Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1942-1948 (1995).

[42] Y.Q. Qin, D.B. Sun, N. Li, and Y.G. Cen, Path Planning for mobile robot using the particle swarm optimization with mutation operator. Proceedings of the IEEE International Conference on Machine Learning and Cybernetics, Shanghai, 2473-2478 (2004).

[43] M. Saska, M. Macas, L. Preucil, and L. Lhotska, Robot Path Planning using Particle Swarm Optimization of Ferguson Splines. Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on. Prague, Czech Republic, 20-22 September. DOI: 10.1109/ETFA.2006.355416

[44] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, Dynamically-stable motion planning for humanoid robots. Autonomous Robots 12(1) (2002) 105-118.

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

[45] A.Z. Nasrollahy, H.H.S. Javadi, Using particle swarm optimization for robot path planning in dynamic environments with moving obstacles and target. Third European Symposium on computer modeling and simulation, Athens, Greece, 60-65 (2009).

[46] D.W. Gong, J.H. Zhang, and Y. Zhang, Multi-objective particle swarm optimization for robot path planning in environment with danger sources. Journal of Computers 6(8) (2011) 1554-1561.

[47] R.R. Cazangi, F.J. Von Zuben, and M.F. Figueiredo, Evolutionary Stigmergy in Multipurpose Navigation Systems. Evolutionary Computation, 2006. CEC 2006. IEEE Congress on. Vancouver, BC, Canada, 16-21 July. (2006). DOI: 10.1109/CEC.2006.1688332.

[48] D.K. Pai, and L.M. Reissell, Multiresolution rough terrain motion planning. IEEE Transaction on Robotic and Automation, 14(1) (1998) 19- 33.

[49] Q.R. Zhang, and G.C. Gu, Path planning based on improved binary particle swarm optimization algorithm. Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics, Chendu, China, 462-466 (2008).

[50] T.L. Lee, and C.J. Wu, Fuzzy Motion Planning of Mobile Robots in Unknown Environments. Journal of Intelligent and Robotic Systems 37(2) (2003) 177-191. doi:10.1023/A:1024145608826.

[51] P.S. Maybeck, The Kalman Filter: An Introduction to Concepts. In: Cox I.J., Wilfong G.T. (eds) Autonomous Robot Vehicles. Springer, New York, NY (1990).

[52] M. Fakoor, A. Kosari, and M. Jafarzadeh, Humanoid robot path planning with fuzzy Markov decision processes. Journal of Applied Research and Technology 14(5) (2016) 300-310. doi:10.1016/j.jart.2016.06.006.

[53] H. Choset, K.M. Lynch, and S. Hutchinson, Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press. ISBN: 9780262033275. (2005).

[54] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, Survey of Robot 3D Path Planning Algorithms. Journal of Control Science and Engineering (2016) pp. 22. DOI: 10.1155/2016/7426913.

MUSE

Multidisciplinary Journal for Education, Social and Technological Sciences

https://doi.org/10.4995/muse.2018.9867
ISSN: 2341-2593

[55] F. Rubio, C. Llopis-Albert, F. Valero, and J.L. Suñer, Assembly line productivity assessment by comparing optimization-simulation algorithms of trajectory planning for industrial robots. Mathematical Problems in Engineering (2015). DOI: 10.1155/2015/931048.

[56] F. Rubio, C. Llopis-Albert, and F. Valero, J.L. Suñer, Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. Robotics and Autonomous Systems 86 (2016) 106–112. DOI: 10.1016/j.robot.2016.09.008.

[57] C. Llopis-Albert, F. Rubio, and F. Valero. Improving productivity using a multi-objective optimization of robotic trajectory planning. Journal of Business Research 68(7) (2015) 1429-1431. DOI: 10.1016/j.jbusres.2015.01.027.

[58] C. Llopis-Albert, and D. Palacios-Marqués, Applied Mathematical Problems in Engineering. Multidisciplinary Journal for Education, Social and Technological Sciences 3(2) (2016) 1-14. DOI: 10.4995/muse.2016.6679.

[59] W. Xing, J. Zhang, W. Lu, and P. Bao, An Improved Potential Field Based Method for Crowd Simulation. International Journal of Software Engineering and Knowledge Engineering 25(03) (2015) 427-451. DOI: 10.1142/S021819401540015X.

[60] J. Sobecki, Comparison of Selected Swarm Intelligence Algorithms in Student Courses Recommendation Application. International Journal of Software Engineering and Knowledge Engineering 24(1) (2014) 91-109. DOI: 10.1142/S0218194014500041.

[61] A. Saeed, S.H.A. Hamid, and A.A. Sani, Cost and Effectiveness of Search-Based Techniques for Model-Based Testing: An Empirical Analysis. International Journal of Software Engineering and Knowledge Engineering 27(04) (2017) 601-622. DOI: 10.1142/S021819401750022X.