Research Journal of Applied Sciences 9 (6): 354-360, 2014
ISSN: 1815-932X


# Fast Robot Path Planning with Laplacian Behaviour-Based Control via Four-Point Explicit Decoupled Group SOR

[1]Azali Saudi, [2]Jumat Sulaiman and [1]Mohd Hanafi Ahmad Hijazi
[1]School of Engineering and Information Technology,
[2]School of Science and Technology, University Malaysia Sabah,
Kota Kinabalu, Malaysia

**Abstract:** This study proposed a robot path planning technique that employs Laplacian Behaviour-Based Control (LBBC) for space exploration which relies on the use of Laplace's equation to constrain the generation of the potential function of the configuration space of a mobile point-robot. The LBBC provides the Searching algorithm with the capability to escape from flat region, whilst iteration via Four-point Explicit Decoupled Group SOR (4EDGSOR) provides fast computation for solving the Laplace's equation that represent the potential values of the configuration space.

**Key words:** Robot path planning, Laplace's equation, Laplacian behaviour-based control, four-point explicit decoupled group SOR, configuration space

## INTRODUCTION

One of the most difficult problems in robotics applications is developing robust autonomous motion planning. In order to build a truly autonomous mobile robot, it must have the capability to efficiently and reliably plan a route from start to the goal point without colliding with obstacles in between. Path Planning algorithms attempt to deal with the problem of establishing a medium of communication between initial and final configurations so that the robot can traverse the field safely. Various algorithms exist trying to solve this problem but all have shortcomings. The difficulty is due to the complexity of path planning problem where it increases exponentially with the dimension of the configuration space.

In order to ensure completeness, every point in the configuration space has to be considered in the computation. Many Global Path Planning Methods presuppose a complete representation of the configuration space. Their main drawbacks is that at best they are computationally expensive and often intractable. Potential field and bug approaches are local methods that do not make this assumption but are not complete methods. Thus, produce the occurrence of local minima or loops that will often cause this class of path planners to fail.

This research attempts to solve robot path planning problem by employing local control known as Laplacian Behaviour-Based Control (LBBC) for an efficient exploration of the environment model. This local control of LBBC relies on the temperature distribution in the environment to guide its exploration. Consequently, the temperature distribution in the environment is computed by solving Laplace's equation. The solutions of Laplace's equation also known as harmonic functions can be used to represent temperature values in the configuration space to be used for simulation of path generation. In this research, several experiments were conducted to study the performance of using fast numerical technique via Four-point EDGSOR (4EDGSOR) Iterative Method to generate robot path in several sizes of environment with varying number of obstacles.

## LITERATURE REVIEW

This study is tremendously inspired by the pioneer research carried out by Khatib (1985). His research introduced the use of potential functions for robot path planning. It views every obstacle to be exerting a repelling force on an end effector while the goal exerts an attractive force. Koditschek (1987) using geometrical arguments showed that at least in certain types of domains there exists potential functions which can guide the effector from almost any point to a given point. These potential fields for path planning however, suffer from the spontaneous creation of local minima.

Connolly *et al.* (1990) and Akishita *et al.* (1990) independently developed a global method using solutions

**Corresponding Author:** Azali Saudi, School of Engineering and Information Technology, University Malaysia Sabah, Kota Kinabalu, Malaysia

to Laplace's equations for path planning to generate a smooth, collision-free path. The potential field is computed in a global manner, i.e., over the entire region and the harmonic solutions to Laplace's equation are used to find the path lines for a robot to move from the start point to the goal point. Obstacles are considered as current sources and the goal is considered to be the sink with the lowest assigned potential value. This amounts to using dirichlet boundary conditions. Then, following the current lines, i.e., performing the steepest descent on the potential field, a succession of points with lower potential values leading to the point with least potential (goal) is found out. It is observed by Connolly *et al*. (1990) that this process guarantees a path to the goal without encountering local minima and successfully avoiding any obstacle. Connolly and Grupen (1993) was attracted by the properties of harmonic functions that had been found useful to be utilized for robotic applications.

In the past, various methods have been proposed for solving Linear System in order to obtain the harmonic functions. The standard methods are Jacobi, Gauss-Seidel and SOR (Sasaki, 1998). Meanwhile, several others have reported the use of harmonic functions in robotics. Silva Jr. *et al*. (2002) used harmonic functions for robot exploration. Kazemi and Mehrandezh (2004) employed harmonic function-based probabilistic maps for their sensor-based robot path planning. Rosell and Iniguez (2005) combine harmonic functions with probabilistic cell decomposition for solving path planning problem. Meanwhile, Daily and Bevly (2008) used analytical solution for arbitrarily shaped obstacles. Garrido *et al*. (2010) had applied finite elements to obtain harmonic functions for robotic motion. More recently, harmonic functions were used for real-time obstacles avoidance in Szulczynski *et al*. (2011).

## LAPLACIAN BEHAVIOUR-BASED CONTROL

Traditional approach robot programming assumes the availability of a complete and accurate model of the robot and its environment, relying on planners to generate actions (Brooks, 1985). Unfortunately, this approach has several disadvantages. One main drawback is that they require huge amounts of computational resources. This drawback is much obvious for an autonomous mobile robot that must carry its own computational resources. Secondly, this approach must be based on highly accurate model thus it requires a number of high-precision sensors which are also often expensive. These sensors however are subject to noisy data. Finally, this sense plan act paradigm is by nature sequential thus it would fail if the world happens to change in between of phases. Furthermore, there is always delay between sensing and act due to longer time required in planning.

As an alternative to the traditional approach, a new paradigm called subsumption architecture also known as behaviour-based control is devised (Arkin, 2001). In this architecture, sensors are dealt with only implicitly in that they initiate behaviours. Each behaviour is simply layers of control systems that all run in parallel. Higher level behaviours have the power to temporarily suppress lower level behaviours. Therefore, a set of priority scheme is used to resolve the dominant behaviour for a given scenario. A more rigorous explanation of behaviour-based approach for controlling robot is presented by Saudi and Hallam (2004).

In this research, inspired by the behaviour-based paradigm approach to robotics control, the Searching algorithm employs Laplacian Behaviour-Based Control (LBBC) for robust space exploration of the configuration space. The LBBC comprises four core behaviours, i.e., keep-forward, follow-wall, avoid-obstacle and find-slope. All these core behaviours make use of the potential values represented by temperature distribution in the configuration space which are computed numerically to provide guidance during search exploration.

**Keep-forward behaviour:** The keep-forward behaviour is a core behaviour that keeps the searching moving forward in the same direction as long as the temperature at current location is higher than the next location. When the searching encounters ascending slope, flat region, obstacles or walls, the keep-forward behaviour stops and other behaviours would take over. The main aim of this behaviour is to guide the searching by following the descending slope until the goal location is found.

**Follow-wall behaviour:** This follow-wall behaviour provides the search with the capability to follow the wall for a specified number of steps. With this behaviour, it will command the searching to keep turning gradually until its direction is parallel with the wall. It provides the searching with the capability of traversing the narrow path and sharp corner. In this implementation, the follow-wall behaviour is executed for every a specified number of steps. After that the searching switches to find-slope behaviour.

**Avoid-obstacle behaviour:** If the searching hits an obstacle or wall, it will trigger the searching to backup and turn 90° to the left or right alternately. By turning alternately to the left and right, it provides the searching with the capability to escape from a difficult position such as sharp corner.

**Find-slope behaviour:** When the find-slope behaviour takes over, it will command the searching to move randomly hoping to encounter a descending slope that consequently triggers keep-forward behaviour. With this behaviour, the searching is capable of moving away from a flat region to continue its descending move towards goal location.

## HARMONIC FUNCTIONS

A harmonic function on a domain $\Omega \subset R^n$ is a function which satisfies Laplace's equation:

$$\nabla^2\phi = \sum_{i=1}^{n} \frac{\partial^2\phi}{\partial x_i^{\ 2}} = 0 \qquad (1)$$

Where:

$x_i$ = The ith cartesian coordinate
n = The dimension

In the case of robot path construction, the boundary of Ω (denoted by ∂Ω) consists of the outer boundary of the workspace and the boundaries of all the obstacles as well as the start point and the goal point in a configuration space representation. The spontaneous creation of a false local minimum inside the region is avoided if Laplace's equation is imposed as a constraint on the functions used as the harmonic functions satisfy the min-max principle. Hence, the only types of critical points which can occur are saddle points. For a path-planning algorithm, an escape from such critical points can be found by performing a search in the neighbourhood of that point. Laplace's equation can be solved numerically. Standard methods are Jacobi and Gauss-Seidel but faster computation can be obtained using Successive-Over-Relaxation (SOR) Iterative Method.

In the framework used in this study, the robot is represented by a point in the environment model or also known as configuration space. The path planning problem is then posed as an obstacle avoidance problem for the point robot from the start point to the goal point in the configuration space which can have either square or rectangular outer boundaries having projections or convolutions inside to act as barriers. Apart from projections of the boundaries some obstacles inside the boundary are also considered. The configuration space is designed in grid or discrete form and the coordinates and function values associated with each node are computed iteratively by applying numerical technique to satisfy Eq. 1. The highest temperature is assigned to the start point whereas the goal point is assigned the lowest. In some cases with Dirichlet conditions, the start point is not assigned any temperature. In this study, Dirichlet boundary conditions are employed thus the results are processed by assigning different temperature values to the boundaries and obstacles and lowest temperature for the goal point. No temperature values are assigned to the start points. In this research, solution to the Laplace's equation were subjected to Dirichlet boundary conditions $\Phi|\delta\Omega = c$, where c is constant.

## FORMULATION OF FOUR POINT-EDGSOR (4EDGSOR) ITERATIVE METHOD

In the literature, Jacobi and Gauss-Seidel (Sasaki, 1998) had been used for solving any linear system. More recently, Daily and Bevly (2008) use analytical solution for arbitrarily shaped obstacles. Others employed Block Iterative Methods mainly on various points of Explicit Group (EG) methods including Evans (1985), Evans and Yousif (1986), Ibrahim and Abdullah (1995), Sulaiman *et al*. (2007) and Hasan *et al*. (2011). They pointed out that the Block Iterative Method is more superior compared to the traditional Point Iterative Methods. In robotics, the previous researches on utilizing block iteration for solving robot path planning via Laplace's equation produce encouraging results (Saudi and Sulaiman, 2010a-c, 2012a-b) although, they were carried out without LBBC. Saudi and Sulaiman (2012c) introduce the use of LBBC for robust robot exploration. Subsequent study (Saudi and Sulaiman, 2012d, 2013, 2014) reported significant performance improvement.

Let us consider the two-dimensional Laplace equation in Eq. 1 defined as:

$$\frac{\partial^2 U}{\partial^2 x} + \frac{\partial^2 U}{\partial^2 y} = 0 \qquad (2)$$

By using the second-order central difference scheme, researchers can simplify the five point second-order standard finite difference approximation equations for problem 2 as generally stated in the following equation:

$$U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j} = 0 \qquad (3)$$

Equation 3 is the standard Gauss-Seidel Iterative Method for solving linear system. To enhance convergence speed, an approach called Successive Over-Relaxation (SOR) Method is added to Eq. 3 as can be shown (Young, 1972):

$$U_{i,j}^{k+1} = \frac{\omega}{4}\left(U_{i-1,j}^{k+1} + U_{i+1,j}^{k} + U_{i,j-1}^{k+1} + U_{i,j+1}^{k}\right) + (1-\omega)U_{i,j}^{k} \qquad (4)$$

Where the optimal value of ω is defined in the range, $1 \le \omega < 2$. In practice, several runs of computer program implementation of Eq. 4 is carried out with different value of ω. The value of ω is considered optimal when the program converges with the less number of iterations. By taking ω = 1, the SOR Iterative Method will represent Gauss-Seidel Method.

Let us consider a block of four node points to form a (4×4) Linear System as shown in Fig. 1 and defined as:

$$\begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 0 & 0 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \\ U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \quad (5)$$

Where:

$S_1 = U_{i-1,j-1}+U_{i-1,j+1}+U_{i+1,j}$

$S_2 = U_{i,j+2}+U_{i+2,j}+U_{i+2,j+2}$

$S_3 = U_{i,j-1}+U_{i+2,j-1}+U_{i+2,j+1}$

$S_4 = U_{i-1,j}+U_{i-1,j+2}+U_{i+1,j+2}$

The Linear System in Eq. 5 can be decomposed independently as two (2×2) matrices. Thus, they can be easily defined as:

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j+1} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \quad (6)$$

And:

$$\begin{bmatrix} U_{i+1,j} \\ U_{i,j+1} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} S_3 \\ S_4 \end{bmatrix} \quad (7)$$

By adding a weighted parameter ω to Eq. 6 and 7, the implementation of the 4EDGSOR Iterative Method can be shown as:

$$\begin{aligned} U_{i,j}^{k+1} &= \frac{\omega}{15}(4S_1 + S_2) + (1-\omega)U_{i,j}^{k} \\ U_{i+1,j+1}^{k+1} &= \frac{\omega}{15}(S_1 + 4S_2) + (1-\omega)U_{i+1,j+1}^{k} \end{aligned} \quad (8)$$

And:

$$\begin{aligned} U_{i+1,j}^{k+1} &= \frac{\omega}{15}(4S_3 + S_4) + (1-\omega)U_{i+1,j}^{k} \\ U_{i,j+1}^{k+1} &= \frac{\omega}{15}(S_3 + 4S_4) + (1-\omega)U_{i,j+1}^{k} \end{aligned} \quad (9)$$

As shown in Fig. 1, the position of numbers in the solution domain for n = 7 shows that the computational execution starts at number 1 and stops at last number 16. The actual computation can be implemented by using either Eq. 8 or 9. Once executed, this iterative process is only terminated when there are no changes of any node points from one sweep of iteration to the next. In this process, a very high precision of computation is required to reduce the occurrence of flat area in the final solution.

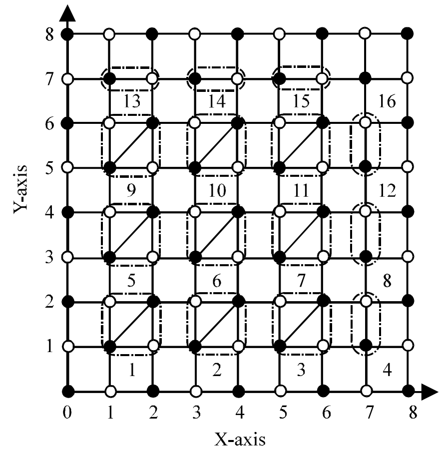


Fig. 1: Illustration of 4EDGSOR iteration for solution domain (n = 7)

## EXPERIMENT

The environment setup consists of three different sizes, namely, 128×128, 256×256 and 512×512. The goal point was set with a fixed and lowest temperature values whereas all three start points were assigned with no particular temperature values. Varying shapes of inner walls and outer boundary walls was placed in the environment with various shapes. In the initial setting, Dirichlet boundary condition was applied where the walls and obstacles were fixed with high temperature values. All other points were set to zero temperature value except goal point which was set to the lowest temperature values.

The computation process was run on Intel Core 2 Duo CPU running at 3 GHz speed with 1 GB of RAM. The iteration process to compute temperature values numerically at all points would continue until the stopping condition is met. When there was no more changes in temperature values the loop is terminated where the difference of calculation values was very small, i.e., $1.0^{-10}$. This very high precision was necessary to avoid flat area, also known as Saddle points, in the solutions thus, would cause the path generation to fail.

Table 1-3 show the number of iterations, maximum error and elapsed time (in m:sec:msec), respectively required to compute all temperature values in the environment for all numerical techniques compared in the experiment.

Table 1: Number of iterations for several iterative methods

| | Size of environment | | |
|---|---|---|---|
| Iterative method | 28×128 | 256×256 | 512×512 |
| GS | 22342 | 81003 | 289040 |
| SOR | 1366 | 5225 | 19236 |
| EGSOR | 1003 | 3898 | 14405 |
| 4EGSOR | 667 | 2696 | 10066 |
| HSSOR | 660 | 2687 | 10052 |
| 4EDGSOR | 489 | 2045 | 7687 |

GS: Gauss-Seidel; SOR: Successive Over-Relaxation; EGSOR: Explicit Group with SOR; 4EGSOR: Four-point EGSOR; HSSOR: Half-Sweep SOR; 4EDGSOR: Four-point Explicit Decoupled Group SOR

Table 2: Maximums error for various iterative methods

| | Size of environment | | |
|---|---|---|---|
| Iterative method | 128×128 | 256×256 | 512×512 |
| GS | $0.9995^{-10}$ | $0.9999^{-10}$ | $0.9999^{-10}$ |
| SOR | $0.9956^{-10}$ | $0.9982^{-10}$ | $0.9997^{-10}$ |
| EGSOR | $0.9838^{-10}$ | $0.9983^{-10}$ | $0.9995^{-10}$ |
| 4EGSOR | $0.9983^{-10}$ | $0.9981^{-10}$ | $0.9987^{-10}$ |
| HSSOR | $0.9770^{-10}$ | $0.9989^{-10}$ | $0.9999^{-10}$ |
| 4EDGSOR | $0.9812^{-10}$ | $0.9926^{-10}$ | $0.9985^{-10}$ |

Table 3: The performance of various iterative methods in seconds

| | Size of environment | | |
|---|---|---|---|
| Iterative method | 128×128 | 256×256 | 512×512 |
| GS | 0:15:875 | 3:59:797 | 65:6:297 |
| SOR | 0:00:859 | 0:15:390 | 4:23:498 |
| EGSOR | 0:00:750 | 0:12:375 | 3:16:859 |
| 4EGSOR | 0:00:500 | 0:08:219 | 2:12:906 |
| HSSOR | 0:00:219 | 0:03:109 | 0:45:641 |
| 4EDGSOR | 0:00:203 | 0:03:016 | 0:40:594 |

Figure 2 shows the performance of several iterative methods in varying sizes of environment. Clearly, 4EDGSOR Iterative Method proved to be very fast compared to the previous methods.

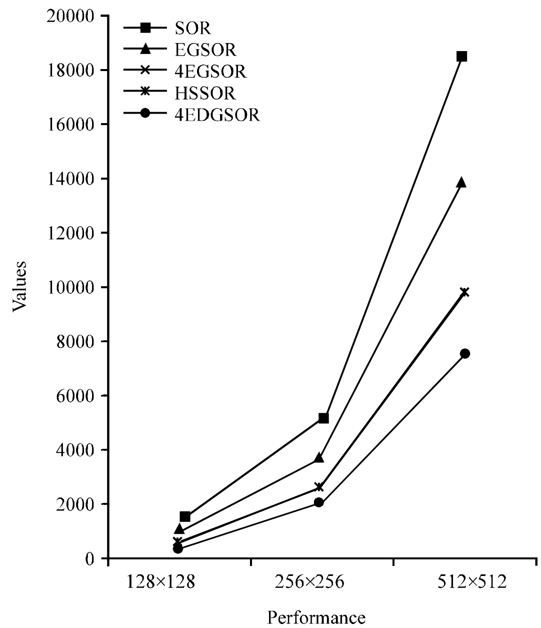


Fig. 2: Number of iterations against various iterative methods for different sizes of environment

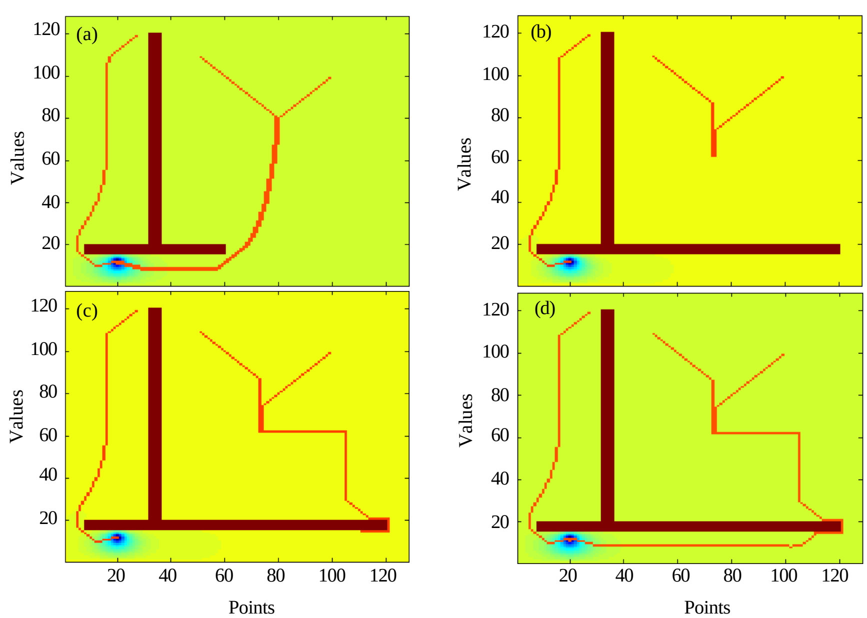


Fig. 3: a) Path is successfully generated in a simple and sparse environment from three starting points to a goal point; b) The path generation process failed to reach the goal point when the length of horizontal wall is extended twice to the right; c) With LBBC, the algorithm switches to find-slope behaviour and then follow-wall behaviour to escape from flat region and d) The LBBC algorithm switches back to keep-forward behaviour to find the goal point

Once the temperature values were obtained, the Searching algorithm would make use of them to guide its exploration. In the previous research (Saudi and Sulaiman, 2010a-c, 2012a-b), the path can be generated successfully even without LBBC, if the environment space was simple and sparse in which the gradient from start points to goal point are smooth (Fig. 3a). However, the Searching algorithm failed to reach the goal point when the horizontal wall was extended as reported by Saudi and Sulaiman (2012c, d, 2013, 2014). In Fig. 3b, only one path was successfully generated whereas the other two start points got stuck in the flat region. Whereas, in Fig. 3c by employing LBBC, the Searching algorithm would be able to escape from flat region and continue its exploration by utilizing find-slope behaviour until it detects a wall in which the algorithm switches to follow-wall behaviour. Finally, in Fig. 3d, the LBBC algorithm switches back to keep-forward behaviour until it reaches the goal point.

## CONCLUSION

The experiment demonstrates that complete search offered by numerical technique is indeed very attractive and feasible for solving difficult robot path planning problem. This is mainly due to the availability of advanced techniques recently as well as the availability of fast machine nowadays. Figure 2 shows that 4EDGSOR is very much faster than the earlier iterative methods. Table 3 shows that 4EDGSOR is slightly faster than HSSOR whereas SOR, EGSOR and 4EGSOR take >4, 3 and 2 min, respectively. The benchmark Gauss-Seidel is very slow by taking >1 h to compute all nodes.

In the future research, researchers would consider faster numerical implementation for solving robot path planning including quarter-sweep iteration as discussed by Muthuvalu and Sulaiman (2011a, b) and Fauzi and Sulaiman (2012) for computing the solution of Laplace's equation.

## REFERENCES

Akishita, S., S. Kawamura and K. Hayashi, 1990. New navigation function utilizing hydrodynamic potential for mobile robot. Proceedings of the IEEE International Workshop on Intelligent Motion Control, Volume 2, August 20-22, 1990, Istanbul, Turkey, pp: 413-417.

Arkin, R.C., 2001. Combining deliberation, reactivity and motivation in the context of a behavior-based robot architecture. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, July 29-August 1, 2001, Banff, Aberta, Canada, pp: 290-295.

Brooks, R.A., 1985. A robust layered control system for a mobile robot. IEEE J. Robot. Automat., 2: 14-23.

Connolly, C.I. and R.A. Grupen, 1993. The applications of harmonic functions to robotics. J. Robotic Syst., 10: 931-946.

Connolly, C.I., J.B. Burns and R. Weiss, 1990. Path planning using Laplace's equation. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 3, May 13-18, 1990, Cincinnati, OH., pp: 2102-2106.

Daily, R. and D.M. Bevly, 2008. Harmonic potential field path planning for high speed vehicles. Proceeding of American Control Conference, June 11-13, 2008, Seattle, WA., pp: 4609-4614.

Evans, D.J. and W.S. Yousif, 1986. Explicit group iterative methods for solving elliptic partial differential equations in 3-space dimensions. Int. J. Comput. Math., 18: 323-340.

Evans, D.J., 1985. Group explicit iterative methods for solving large linear systems. Int. J. Comput. Math., 17: 81-108.

Fauzi, N.I.M. and J. Sulaiman, 2012. Quarter-sweep modified SOR iterative algorithm and cubic spline basis for the solution of second order two-point boundary value problems. J. Applied Sci., 12: 1817-1824.

Garrido, S., L. Moreno, D. Blanco and F.M. Monar, 2010. Robotic motion using harmonic functions and finite elements. J. Intell. Robotic Syst., 59: 57-73.

Hasan, M.K., J. Sulaiman, S.A.A. Karim and M. Othman, 2011. Development of some numerical methods applying complexity reduction approach for solving scientific problem. J. Applied Sci., 11: 1255-1260.

Ibrahim, A. and A.R. Abdullah, 1995. Solving The two dimensional diffusion equation by the four point Explicit Decoupled Group (EDG) iterative method. Int. J. Comput. Mathe., 58: 253-256.

Kazemi, M. and M. Mehrandezh, 2004. Robotic navigation using harmonic function-based probabilistic roadmaps. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 5, April 26-May 1, 2004, New Orleans, LA, USA., pp: 4765-4770.

Khatib, O., 1985. Real time obstacle avoidance for manipulators and mobile robots. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 2, March 25-28, 1985, Philadelphia, pp: 500-505.

Koditschek, D.E., 1987. Exact robot navigation by means of potential functions: Some topological considerations. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 4, March 1987, Raleigh, NC., pp: 1-6.

Muthuvalu, M.S. and J. Sulaiman, 2011a. Numerical solutions of first kind linear fredholm integral equations by using quarter-sweep SOR with piecewise linear collocation method QSSOR with piecewise linear collocation method for fredholm equations. Aust. J. Basic Applied Sci., 5: 18-24.

Muthuvalu, M.S. and J. Sulaiman, 2011b. Numerical solution of second kind linear fredholm integral equations using QSGS iterative method with high-order newton-cotes quadrature schemes. Malaysian J. Math. Sci., 5: 85-100.

Rosell, J. and P. Iniguez, 2005. Path planning using harmonic functions and probabilistic cell decomposition. Proceedings of the IEEE International Conference on Robotics and Automation, April 18-22, 2005, Barcelona, Spain, pp: 1803-1808.

Sasaki, S., 1998. A practical computational technique for mobile robot navigation. Proceedings of the IEEE International Conference on Control Applications, Volume 2, September 1-4, 1998, Trieste, pp: 1323-1327.

Saudi, A. and B. Hallam, 2004. A tale of two behaviour-based robots mindstorms vs technic. Proceedings of the IEEE Region 10 Conference, Volume 4, November 21-24, 2004, Chiang Mai, Thailand, pp: 479-482.

Saudi, A. and J. Sulaiman, 2010a. Block iterative method using nine-point laplacian for robot path planning. Eur. J. Scient. Res., 43: 204-211.

Saudi, A. and J. Sulaiman, 2010b. Robot path planning based on four point-EGSOR iterative method. Proceedings of the IEEE Conference on Robotics Automation and Mechatronics, June 28-30, 2010, Singapore, pp: 476-481.

Saudi, A. and J. Sulaiman, 2010c. Numerical technique for robot path planning using four point-eg iterative method. Proceedings of the 2010 International Symposium on Information Technology, June 15-17, 2010, Kuala Lumpur, Malaysia, pp: 831-836.

Saudi, A. and J. Sulaiman, 2012a. Robot path planning via EGSOR iterative method using nine-point laplacian. Proceedings of the 3rd International Conference on Intelligent Systems, Modelling and Simulation, February 8-10, 2012, Kota Kinabalu, Malaysia, pp: 61-66.

Saudi, A. and J. Sulaiman, 2012b. Laplacian Behaviour-Based Control (LBBC) for the path planning of mobile robot via four point-EGSOR. Int. J. Comput. Inform. Technol., 1: 81-87.

Saudi, A. and J. Sulaiman, 2012c. Robot path planning using four point-explicit group via nine-point laplacian (4EG9L) iterative method. J. Procedia Eng., 41: 182-188.

Saudi, A. and J. Sulaiman, 2012d. Laplacian behaviour-based control for robot path planning using Full-Sweep Successive Over-Relaxation via nine-point Laplacian (FSSOR9L). Int. J. Applied Sci. Technol., 2: 255-261.

Saudi, A. and J. Sulaiman, 2013. Indoor path planning for mobile robot using LBBC-EG. Int. J. Imaging Robotics, 11: 37-45.

Saudi, A. and J. Sulaiman, 2014. Hybrid path planning for indoor robot with laplacian Behaviour-based control via four Point-explicit group. Int. J. Imaging Robotics, 12: 12-21.

Silva Jr. E.P., P.M. Engel, M. Trevisan and M.A.P. Idiart, 2002. Exploration method using harmonic functions. Robotics Autonomous Syst., 40: 25-42.

Sulaiman, J., M. Othman and M.K. Hasan, 2007. Red-black EDGSOR iterative method using triangle element approximation for 2D poisson equations. Proceedings of the International Conference on Computational Science and Its Applications, August 26-29, 2007, Kuala Lumpur, Malaysia, pp: 298-308.

Szulczynski, P., D. Pazderski and K. Kozlowski, 2011. Real-time obstacle avoidance using harmonic potential functions. J. Autom. Mobile Robotics Intell. Syst., 5: 59-66.

Young, D.M., 1972. Second-degree iterative methods for the solution of large linear systems. J. Approximation Theory, 5: 137-148.