

# Perancangan Laluan Robot menggunakan Fungsi Harmonik melalui Kaedah Lelaran Sapuan Suku

*Robot Path Planning using Harmonic Functions via Quarter Sweep Iterative Method*

Azali Saudi<sup>1</sup> dan Jumat Sulaiman<sup>2</sup>

<sup>1</sup>Knowledge Technology Research Unit, Faculty of Computing and Informatics

<sup>2</sup>Faculty of Science and Natural Resources

Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia

azali@ums.edu.my, jumat@ums.edu.my

## Abstrak

Kertas kajian ini mengemukakan aplikasi fungsi harmonik dalam menyelesaikan masalah perancangan laluan robot. Fungsi harmonik diperolehi dengan menyelesaikan persamaan Laplace. Dalam kajian ini, fungsi-fungsi harmonik bertindak sebagai nilai-nilai potensi pada setiap titik dalam persekitaran robot. Nilai-nilai potensi ini kemudiannya digunakan untuk mencari laluan dari titik mula sehinggalah ke titik destinasi yang telah ditetapkan. Bagi mendapatkan fungsi-fungsi harmonik ini, pendekatan yang paling biasa digunakan adalah kaedah beza terhingga iaitu lelaran Jacobi, Gauss-Seidel dan Successive Over-relaxation (SOR). Namun, kaedah-kaedah lelaran yang lazim ini terlalu perlahan, terutamanya apabila melibatkan persekitaran yang luas. Oleh itu, kajian ini mencadangkan kaedah yang lebih pantas menggunakan gabungan teknik lelaran sapuan suku dengan lelaran SOR. Keputusan ujikaji menunjukkan kaedah-kaedah yang dicadangkan ini telah berjaya mengurangkan masa pengiraan fungsi-fungsi harmonik secara drastik. Ini secara langsung telah meningkatkan prestasi keseluruhan algoritma perancangan laluan, terutamanya aspek masa perlaksanaan.

**Kata kunci** Perancangan laluan robot, fungsi-fungsi Harmonik, kaedah lelaran, sapuan separuh, sapuan suku, persamaan Laplace

## Abstract

This paper presents the application of harmonic functions in solving the robot path planning problem. The harmonic function is obtained by solving the Laplace's equation. In this study, the harmonic functions act as potential values at each point in the robot environment. These potential values are then used to find path from the starting point to the specified target point. In order to obtain these harmonic functions, the most commonly used approaches are the finite difference methods of Jacobi, Gauss-Seidel and Successive Over-relaxation (SOR). However, these standard iterative methods are too slow, especially when it involves large environments. Therefore, this study proposes faster methods that employ the combination of quarter sweep techniques with SOR iteration. The results of experiments show that these proposed methods have successfully reduced the time of computation of harmonic functions drastically. This has directly improved the overall performance of the path planning algorithm, particularly in terms of execution time.

**Keywords** Robot path planning, Harmonic functions, iterative method, half sweep, quarter sweep, Laplace's equation

## **PENGENALAN**

Salah satu isu yang amat sukar dalam bidang robotik adalah masalah perancangan laluan. Antara keperluan utama untuk membina suatu robot automatik yang sebenar adalah kemampuannya untuk merancang laluan secara efisien dari titik permulaan hingga ke titik destinasi yang telah ditentukan dan tanpa perlanggaran dengan sebarang objek atau tersekat oleh halangan semasa melaluinya. Algoritma perancangan laluan cuba menyelesaikan masalah ini supaya robot tersebut dapat bergerak dalam ruang persekitarannya dengan selamat dan lancar. Pelbagai algoritma telah dibangunkan untuk cuba menyelesaikan masalah ini tapi kesemuanya mempunyai kelemahan tertentu. Kesukaran dalam permasalahan ini adalah kerana kadar kekompleksan masalah perancangan laluan yang bertambah secara eksponen dengan saiz atau dimensi ruang persekitaran robot tersebut beroperasi. Bagi memastikan algoritma yang bersifat lengkap, setiap titik dalam ruang persekitaran perlulah diambilkira semasa proses pengiraan. Terdapat pelbagai teknik perancangan laluan yang bersifat global cuba membina model ruang persekitaran yang lengkap. Kelemahan utama teknik sedemikian adalah ianya memerlukan sumber komputer yang terlalu besar dan proses komputeran yang sangat banyak. Pendekatan setempat (atau lokal) pula tidak memerlukan model yang lengkap, tetapi ia mengalami masalah perangkap nilai minimum setempat dan selalunya teknik sebegini akan gagal menjana laluan yang dikehendaki. Secara umumnya, teknik perancangan laluan boleh dibahagikan kepada dua kategori iaitu kaedah setempat yang bertindakbalas serta-merta terhadap data yang diperolehi melalui input daripada sensor dan kaedah global yang memerlukan perancangan dilakukan terlebih dahulu sebelum membuat sebarang tindakan (LaValle, 2011). Beberapa kajian telah dilakukan untuk mencari penyelesaian bagi masalah perancangan laluan (Khatib, 1985; Connolly, Burns, & Weiss, 1990; Karnik, Dasgupta, & Eswaran, 2002; Tran, Nguyen, Vu, & Truong, 2013). Selain itu, pendekatan secara heuristik juga telah digunakan untuk menyelesaikan masalah perancangan laluan robot. Antara kaedah heuristik yang telah digunakan termasuklah Rangkaian Neural, Algoritma Genetik, Pengaturcaraan Genetik, Carian Tabu dan sebagainya.

## **SOROTAN LITERATUR**

Connolly et al. (1990) dan Akishita Kawamura, & Hayashi (1990) telah membangunkan kaedah global yang menggunakan penyelesaian persamaan Laplace untuk menjana laluan robot yang lancar dan selamat. Nilai-nilai potensi dikira secara global, dan penyelesaian Laplace ini yang juga dikenali sebagai fungsi harmonik kemudiannya digunakan untuk mencari laluan robot untuk membolehkannya bergerak dari titik permulaan hinggalah ke titik destinasi. Halangan-halangan bertindak sebagai punca sumber dan titik destinasi pula bertindak sebagai penyerap dan diberikan nilai potensi yang paling rendah. Seterusnya, dengan melaksanakan carian menuruni kecerunan terhadap nilai-nilai potensi sehinggalah ke titik yang mempunyai nilai potensi paling rendah, titik destinasi akan ditemui. Kajian oleh Connolly et al. (1990) mendapati proses ini menjamin laluan ke destinasi tanpa menemui

nilai minimum setempat dan ia dapat mengelak sebarang halangan. Selain itu, kajian oleh Connolly & Grupen (1993) pula mendapati fungsi-fungsi harmonik mempunyai beberapa sifat yang boleh digunakan untuk aplikasi dalam bidang robotik. Terdapat beberapa kajian yang telah dilaporkan menggunakan fungsi harmonik untuk robotik. Antaranya kajian yang telah dijalankan oleh Karnik et al. (2002), Daily & Bevely (2008) dan Garrido, Moreno, Blanco, & Monar (2010). Dalam beberapa kajian yang lepas, terdapat pelbagai kaedah telah dicadangkan untuk menyelesaikan sistem persamaan linear untuk mendapatkan fungsi-fungsi harmonik termasuklah kaedah lalaran Jacobi, Gauss-Seidel dan SOR seperti yang digunakan dalam kajian oleh Sasaki (1998). Kaedah berangka yang lebih pantas menggunakan lalaran sapan separuh (Abdullah, 1991) dan suku (Othman & Abdullah, 2000) juga telah dijalankan. Selain itu, teknik berangka yang menggunakan kaedah lalaran blok juga telah digunakan (Evans, 1985; Sulaiman, Othman, & Hasan, 2007).

## **FUNGSI HARMONIK**

Fungsi harmonik sebenarnya adalah penyelesaian kepada persamaan Laplace (LeVeque, 2007). Kajian oleh Connolly et al. (1990) membuktikan ia mempunyai beberapa ciri yang berguna untuk aplikasi dalam bidang robotik. Fungsi harmonik menawarkan algoritma perancangan laluan yang lengkap kerana ia mengambilkira semua titik dalam ruang persekitaran, dan laluan yang dijana pula adalah lancar. Selain itu, apabila fungsi harmonik diaplikasikan dalam perancangan laluan robot, ia tidak menghasilkan nilai minimum setempat. Fungsi harmonik pada suatu domain  $\Omega \subset R^n$  adalah suatu fungsi yang memenuhi persamaan Laplace,

$$\nabla^2 u = \sum_{i=1}^n \frac{\partial^2 u}{\partial x_i^2} = 0 \tag{1}$$

dengan  $x_i$  adalah koordinat ke- $i$  dan  $n$  adalah dimensi. Dalam masalah perancangan laluan robot, sempadan bagi domain  $\Omega$  terdiri daripada sempadan dalaman dan luaran, sempadan objek-objek halangan, dan titik destinasi. Persamaan Laplace pula digunakan untuk menghadkan penjanaan nilai-nilai potensi dan menghalang penghasilan nilai minimum setempat dalam ruang domain  $\Omega$ .

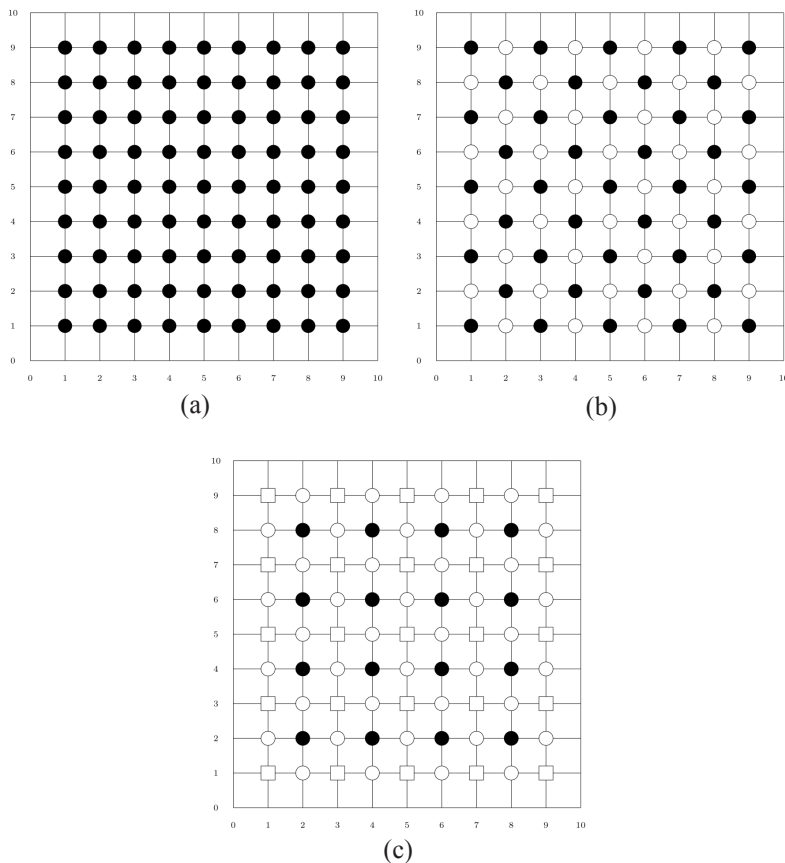
## **RUANG KONFIGURASI**

Dalam kerangka kajian ini, robot merupakan suatu titik dalam ruang konfigurasi yang terdiri daripada sempadan luaran, dinding dalaman serta beberapa objek halangan. Ruang konfigurasi adalah dalam bentuk grid dan nilai potensi untuk setiap titik dikira secara lalaran menggunakan kaedah berangka untuk memenuhi persamaan (1). Dengan menggunakan syarat sempadan Dirichlet, nilai malar yang tidak berubah diberikan pada setiap titik sempadan. Nilai potensi tinggi diberikan kepada titik-titik pada sempadan luaran, dinding dalaman dan objek-objek halangan. Nilai potensi rendah pula diberikan kepada titik destinasi. Tiada sebarang nilai potensi diberikan kepada titik-titik yang lain. Dengan menggunakan analogi pemindahan haba, titik-titik pada sempadan akan bertindak sebagai punca sumber suhu, manakala titik destinasi pula bertindak sebagai penyerap suhu. Setelah

fungsi harmonik diperolehi menggunakan kaedah lelaran, laluan yang diperlukan boleh dijana menggunakan teknik yang dipanggil Carian Menuruni Cerun (CMC) (Connolly et al., 1990; Karnik et al., 2002; Garrido et al., 2010). Dengan menggunakan teknik CMC, suatu carian akan bermula dari sebarang titik permulaan dan kemudian bergerak menuruni cerun pada titik kejiranan bersuhu lebih rendah sehinggalah ke titik yang bersuhu paling rendah iaitu titik destinasi.

## KONSEP LELARAN SAPUAN SEPARUH DAN SUKU

Oleh kerana sistem linear untuk menyelesaikan masalah dalam persamaan (1) akan menghasilkan matriks yang besar dan jarang (kebanyakan nilai adalah sifar), kaedah lelaran adalah teknik yang paling sesuai (Faires & Burden, 2012; Kress, 1998). Namun, salah satu kelemahan kaedah lelaran adalah kadar penumpuan yang perlahan (Saad & van der Horst, 2000). Oleh itu dalam kajian ini, pendekatan yang mampu mengurangkan kekompleksan pengiraan iaitu lelaran sapuan separuh dan suku akan digunakan untuk meningkatkan prestasi kaedah lelaran. Dengan menggunakan kaedah lelaran untuk mengira nilai-nilai potensi, pertimbangkan suatu ruang domain  $\Omega$  dijadikan dalam bentuk



**Rajah 1** Titik-titik yang dipertimbangkan untuk sapuan (a) penuh, (b) separuh, dan (c) suku.

grid bersaiz  $N \times N$  dengan jarak  $h$  antara nod-nod adalah sama, seperti yang ditunjukkan dalam Rajah 1 (a). Dalam kajian yang lepas (Connolly et al., 1990; Karnik et al., 2002; Sasaki, 1998), kesemua nod yang berjenis  $\bullet$  diambil kira semasa proses lelaran. Proses yang menggunakan kaedah lelaran lazim ini dipanggil kaedah lelaran sapuan penuh. Bagi kaedah lelaran sapuan separuh dan suku pula, ruang domain  $\Omega$  dibahagikan seperti yang ditunjukkan masing-masing dalam Rajah 1 (b) dan Rajah 1 (c). Aplikasi lelaran sapuan separuh dan suku hanya mempertimbangkan nod yang berjenis  $\bullet$  semasa proses lelaran sehinggalah kriteria penumpuan dicapai. Kemudian, pengiraan untuk baki titik-titik  $\circ$  dan  $\square$  dibuat menggunakan teknik langsung (Abdullah, 1991; Othman & Abdullah, 2000). Oleh itu, kekompleksan pengiraan akan dikurangkan sekitar 50% bagi lelaran sapuan separuh dan 75% bagi lelaran sapuan suku.

## KAEDAH BERANGKA UNTUK MENYELESAIKAN PERSAMAAN LAPLACE

Pertimbangkan suatu ruang konfigurasi 2-dimensi, model matematik bagi nilai-nilai potensi dalam ruang konfigurasi tersebut boleh dinyatakan oleh persamaan Laplace 2-dimensi

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{2}$$

Persamaan Laplace boleh diselesaikan dengan dua cara iaitu secara analitik (Daily & Bevely, 2008) atau berangka (Connolly et al., 1990). Dalam kajian ini, penyelesaian persamaan Laplace akan diperolehi melalui kaedah berangka menggunakan kaedah beza terhingga. Dengan kaedah ini, persamaan Laplace (2) ditukar kepada set persamaan serentak atau sistem linear. Apabila sistem linear ini ditulis dalam bentuk matriks, majoriti elemen adalah sifar. Dalam masalah perancangan laluan robot, sistem linear yang dihasilkan biasanya sangat besar dan jarang. Ini memerlukan kapasiti memori yang besar. Oleh itu, cara penyelesaian yang lebih cekap menggunakan kaedah lelaran adalah lebih sesuai (Connolly et al., 1990). Kaedah lelaran juga tidak memerlukan storan untuk menyimpan matriks yang besar. Bagi menyelesaikan masalah ini dengan kaedah beza terhingga, persamaan (2) boleh dianggarkan pada titik  $(x_p, y_j)$  dengan beberapa cara. Anggaran yang paling mudah bagi  $\nabla^2 u$  ialah

$$\nabla^2 u_{ij} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2} \tag{3}$$

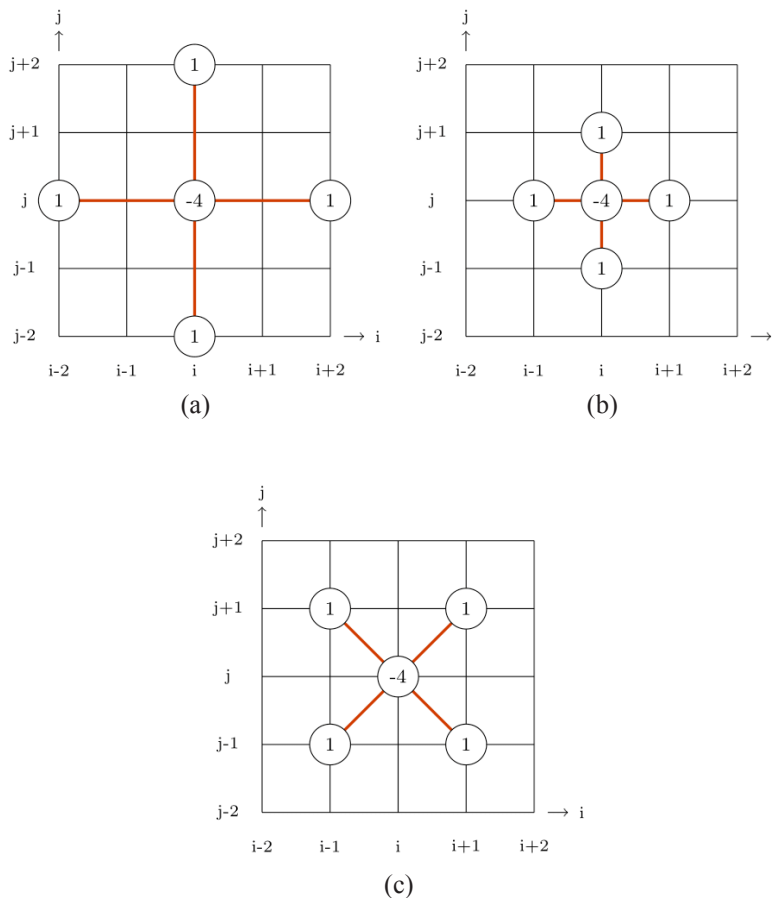
Di sini,  $h =$  pada arah  $x$  dan  $y$  dengan  $x_i = ih, y_j = jh; (i, j = 0, 1, 2, \dots, n)$  dan  $u_{i,j} = u(x_p, y_j)$ . Satu lagi jenis anggaran adalah berasaskan putaran paksi  $x$ - $y$  sebanyak  $45^\circ$  (Dahlquist & Bjorck, 1974; Yousif & Evans, 1986). Ini menghasilkan anggaran 5-titik putaran dan ditulis seperti berikut:

$$\nabla^2 u_{ij} = \frac{u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} - 4u_{i,j}}{2h^2} \tag{4}$$

Manakala, apabila saiz langkau yang dipertimbangkan adalah  $2h$ , ia akan menghasilkan anggaran berikut (Othman & Abdullah, 2000):

$$\nabla^2 u_{ij} = \frac{u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2} - 4u_{i,j}}{4h^2} \quad (5)$$

Sebenarnya, anggaran 5-titik dalam persamaan (3), (4) dan (5) adalah masing-masing untuk kes sapan penuh, separuh (Abdullah, 1991) dan suku (Othman & Abdullah, 2000). Rajah 2 menunjukkan sebahagian daripada grid pengiraan anggaran 5-titik pada titik  $(i,j)$  untuk lelaran sapan (a) penuh, (b) separuh (Abdullah, 1991) dan (c) suku (Othman & Abdullah, 2000).



**Rajah 2** Sebahagian daripada grid pengiraan menggunakan anggaran 5-titik pada titik  $(i, j)$  untuk kes sapan (a) penuh, (b) separuh dan (c) suku.

Berdasarkan anggaran (3), (4) dan (5), persamaan anggaran bagi menyelesaikan masalah (2) untuk kes sapan penuh, separuh dan suku boleh ditulis masing-masing seperti berikut:

$$u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} \tag{6}$$

$$u_{i+1,j+1} + u_{i-1,j-1} + u_{i+1,j-1} + u_{i-1,j+1} - 4u_{i,j} \tag{7}$$

$$u_{i-2,j} + u_{i+2,j} + u_{i,j-2} + u_{i,j+2} - 4u_{i,j} \tag{8}$$

Apabila persamaan-persamaan (6), (7) atau (8) digunakan untuk menganggar titik-titik dalam domain  $\Omega$  bagi menyelesaikan persamaan (2), suatu sistem linear yang besar dan jarang akan dihasilkan. Sistem linear ini boleh dinyatakan dalam bentuk persamaan matriks sebagai

$$Au = b \tag{9}$$

Di sini, matriks  $A$  dan vektor  $b$  diketahui, manakala vector  $u$  pula tidak diketahui. Memandangkan sistem linear dalam persamaan (9) adalah besar dan jarang, kaedah lelaran adalah sangat sesuai.

Apabila kaedah lelaran digunakan untuk menyelesaikan persamaan (9) dengan skema lelaran Gauss-Seidel, skema lelaran bagi sapuan penuh, separuh dan suku bagi persamaan-persamaan beza terhingga (6), (7) dan (8) boleh dibina dan dinyatakan masing-masing seperti berikut:

$$u_{i,j}^{(k+1)} = \frac{1}{4} \left[ u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} \right] \tag{10}$$

$$u_{i,j}^{(k+1)} = \frac{1}{4} \left[ u_{i-1,j-1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} + u_{i-1,j+1}^{(k)} + u_{i+1,j+1}^{(k)} \right] \tag{11}$$

$$u_{i,j}^{(k+1)} = \frac{1}{4} \left[ u_{i-2,j}^{(k+1)} + u_{i+2,j}^{(k)} + u_{i,j-2}^{(k+1)} + u_{i,j+2}^{(k)} \right] \tag{12}$$

Selain itu, Young (1950) telah memperkenalkan penggunaan parameter  $\omega$  melalui kaedah lelaran Successive Over-relaxation (SOR) yang didapati boleh mempercepatkan kadar penumpuan pengiraan secara drastik. Dengan memasukkan pemberat  $\omega$  ke dalam persamaan-persamaan (10), (11) dan (12), skema lelaran SOR bagi sapuan penuh, separuh dan suku boleh dinyatakan masing-masing seperti berikut:

$$u_{i,j}^{(k+1)} = \frac{\omega}{4} \left[ u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} \right] + (1-\omega)u_{i,j}^{(k)} \tag{13}$$

$$u_{i,j}^{(k+1)} = \frac{\omega}{4} \left[ u_{i-1,j-1}^{(k+1)} + u_{i+1,j-1}^{(k+1)} + u_{i-1,j+1}^{(k)} + u_{i+1,j+1}^{(k)} \right] + (1-\omega)u_{i,j}^{(k)} \tag{14}$$

$$u_{i,j}^{(k+1)} = \frac{\omega}{4} \left[ u_{i-2,j}^{(k+1)} + u_{i+2,j}^{(k)} + u_{i,j-2}^{(k+1)} + u_{i,j+2}^{(k)} \right] + (1-\omega)u_{i,j}^{(k)} \tag{15}$$

Secara umumnya, nilai optimum  $\omega$  adalah dalam julat  $1 \leq \omega < 2$  [22]. Perhatikan jika  $\omega = 1$ , skema lelaran SOR ini akan membentuk semula skema lelaran Gauss-Seidel. Dalam kajian ini, teknik sapuan penuh, separuh dan suku akan digabungkan dengan lelaran SOR untuk menyelesaikan persamaan (2). Kaedah lelaran asal SOR yang menggunakan

lelaran sapuan penuh dipanggil juga sebagai kaedah lelaran Sapuan-Penuh SOR (SPSOR). Manakala, gabungan teknik sapuan separuh dan suku dengan SOR pula, masing-masing dipanggil sebagai kaedah lelaran Sapuan-Separuh SOR (SSSOR) dan Sapuan-Suku SOR (SKSOR). Dalam kajian sebelum ini, hanya kaedah SPSOR (Connolly et al., 1990; Karnik et al., 2002) dan SSSOR (Saudi & Sulaiman, 2016) yang pernah digunakan untuk aplikasi pencarian laluan robot. Kedua-dua kaedah SPSOR dan SSSOR serta kaedah baru SKSOR akan digunakan untuk menyelesaikan sistem linear yang dinyatakan dalam persamaan (9). Seterusnya, pertimbangkan matriks  $A$  dalam persamaan (9) dipecahkan kepada berikut:

$$A = D - L - T \quad (16)$$

Di sini,  $D$  adalah matriks blok pepenjuru,  $L$  adalah matriks segitiga bawah, dan  $T$  pula adalah matriks segitiga atas. Dengan menggunakan pecahan dalam persamaan (16), skema umum bagi kaedah SOR dalam bentuk matriks boleh dinyatakan seperti berikut (Young, 1950):

$$u^{(k+1)} = (D - \omega L)^{-1} [(1 - \omega)D + \omega T] u^{(k)} + \omega b \quad (17)$$

Di sini,  $\omega$  adalah parameter pemberat dan  $u^{(k+1)}$  pula adalah vektor tidak diketahui. Dengan menentukan nilai matriks  $D$ ,  $L$  dan  $T$  seperti yang dinyatakan dalam persamaan (9), maka algoritma bagi kaedah lelaran SPSOR, SSSOR dan SKSOR boleh dinyatakan masing-masing seperti dalam Algoritma 1, 2 dan 3.

#### Algoritma 1: Kaedah Lelaran Sapuan-Penuh SOR (SPSOR)

$u \leftarrow$  SetRuangKonfigurasi( sempadan, destinasi)  
 Tentukan nilai  $\omega$   
 Kira semua titik bukan sempadan menggunakan persamaan (13)  
 $iteration \leftarrow 0$   
 repeat  

$$u_{i,j}^{(k+1)} = \frac{\omega}{4} [u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)}] + (1 - \omega)u_{i,j}^{(k)}$$
  
 $iteration \leftarrow iteration + 1$   
 until  $\|u^{(k+1)} - u^{(k)}\| \leq \epsilon$

#### Algoritma 2: Kaedah Lelaran Sapuan-Separuh SOR (SSSOR)

$u \leftarrow$  SetRuangKonfigurasi( sempadan, destinasi)  
 Tentukan nilai  $\omega$   
 Kira semua titik hitam • bukan sempadan menggunakan persamaan (14)  
 $iteration \leftarrow 0$   
 repeat  

$$u_{i,j}^{(k+1)} = \frac{\omega}{4} [u_{i-2,j}^{(k+1)} + u_{i+2,j}^{(k)} + u_{i,j-2}^{(k+1)} + u_{i,j+2}^{(k)}] + (1 - \omega)u_{i,j}^{(k)}$$
  
 until  $\|u^{(k+1)} - u^{(k)}\| \leq \epsilon$



Kira kesemua baki titik putih  $\circ$  bukan sempadan dengan kaedah langsung:  
**foreach** titik putih  $\circ$  **do**

Guna persamaan (10):  $\nabla^2 u_{ij} = \frac{u_{i-2,j} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} - 4u_{i,j}}{2h^2}$   
 end

---

Sebelum proses pengiraan bermula, nilai-nilai potensi pada titik-titik sempadan perlu disetkan kepada nilai tinggi. Titik destinasi pula disetkan kepada nilai yang paling rendah. Kesemua nilai-nilai potensi pada titik-titik sempadan dan destinasi ini adalah malar dan tidak berubah semasa proses pengiraan. Kemudian nilai-nilai potensi pada titik-titik yang terlibat akan dikira berulang kali dalam suatu gelung sehinggalah ralat menumpu memenuhi kriteria yang telah ditetapkan iaitu

$$\|u^{(k+1)} - u^{(k)}\| \leq \varepsilon \quad (18)$$

Bagi kaedah SPSOR, semua titik dalam ruang domain  $\Omega$  diambilkira semasa pengiraan. Seperti yang dinyatakan dalam seksyen 5, bagi kaedah SSSOR pula, hanya titik-titik hitam  $\bullet$  diambilkira semasa pengiraan. Selepas penumpuan dicapai, baki titik-titik putih  $\circ$  dikira menggunakan kaedah langsung yang dinyatakan dalam persamaan (10).

---

### Algoritma 3: Kaedah Lelaran Sapuan-Suku SOR (SKSOR)

---

$u \leftarrow \text{SetRuangKonfigurasi}(\text{sempadan}, \text{destinasi})$   
 Tentukan nilai  $\omega$   
 Kira semua titik  $\bullet$  bukan sempadan menggunakan persamaan (15)  
 $iteration \leftarrow 0$   
 repeat  
      $u_{i,j}^{(k+1)} = \frac{\omega}{4} [u_{i-2,j}^{(k+1)} + u_{i+2,j}^{(k)} + u_{i,j-2}^{(k+1)} + u_{i,j+2}^{(k)}] + (1 - \omega)u_{i,j}^{(k)}$   
      $iteration \leftarrow iteration + 1$   
 until  $\|u^{(k+1)} - u^{(k)}\| \leq \varepsilon$

Kira kesemua baki titik  $\square$  dan  $\circ$  bukan sempadan dengan kaedah langsung:  
**foreach** titik segiempat  $\square$  **do**

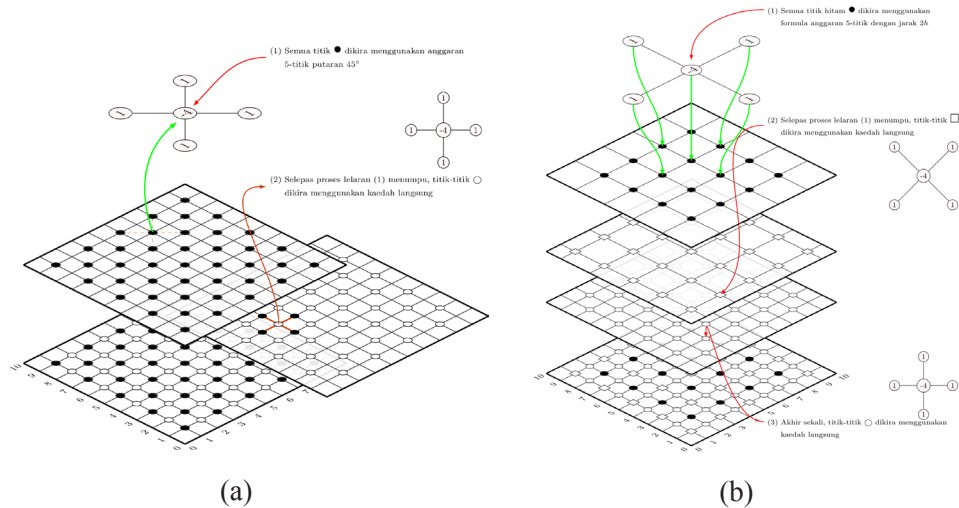
Guna persamaan (11):  $\nabla^2 u_{ij} = \frac{u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2} - 4u_{i,j}}{4h^2}$   
 end

**foreach** titik putih  $\circ$  **do**

Guna persamaan (10):  $\nabla^2 u_{ij} = \frac{u_{i-2,j} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i+1,j+1} - 4u_{i,j}}{2h^2}$   
 end

---

Rajah 3 (a) menunjukkan ilustrasi pengiraan titik-titik yang terlibat menggunakan kaedah lalaran SSSOR. Bagi kaedah SKSOR pula, selepas pengiraan kesemua titik-titik  $\bullet$  memenuhi syarat menumpu yang telah ditetapkan, baki titik-titik putih  $\square$  dan  $\circ$  dikira menggunakan kaedah langsung yang dinyatakan masing-masing dalam persamaan (11) dan (10). Rajah 3 (b) menunjukkan ilustrasi proses pengiraan kesemua titik-titik yang terlibat menggunakan kaedah lalaran SKSOR.



**Rajah 3:** Ilustrasi proses pengiraan titik-titik yang terlibat menggunakan kaedah (a) SSSOR dan (b) SKSOR.

## ALGORITMA PERANCANGAN LALUAN

Algoritma perancangan laluan bermula dengan memasukkan peta ruang konfigurasi ke dalam memori. Peta ruang konfigurasi terdiri daripada sempadan dinding, objek halangan dan ruang kosong. Selain itu, titik destinasi perlu ditandakan pada peta konfigurasi sebelum proses pengiraan bermula. Titik mula pula tidak perlu dinyatakan. Selepas itu, nilai-nilai dalam  $u$  diberikan nilai awal. Titik-titik yang terletak pada sempadan atau objek halangan diberikan nilai yang tinggi, manakala titik destinasi pula diberikan nilai yang paling rendah. Titik-titik pada ruang kosong tidak diberikan sebarang nilai. Semasa proses pengiraan sedang berjalan, hanya titik-titik pada ruang kosong yang diambil kira. Titik-titik yang lain pula diabaikan dan nilainya tidak berubah. Selepas itu, nilai parameter pemberat  $\omega$  disetkan kepada nilai optimum iaitu antara  $1 \leq \omega < 2$ . Beberapa eksperimen perlu dilakukan untuk mendapatkan nilai  $\omega$  yang optimum. Kemudian proses lalaran bagi mengira nilai-nilai pada titik-titik ruang kosong akan dilakukan berulang kali dalam suatu gelung sehingga syarat penumpuan dipenuhi. Lalaran ditamatkan apabila perbezaan mutlak antara nilai  $u^{(k+1)}$  (nilai-nilai pada lalaran terkini) dan  $u^{(k)}$  (nilai-nilai pada lalaran sebelumnya) adalah lebih kecil daripada ralat penumpuan yang ditetapkan iaitu  $1.0^{-15}$ , seperti yang dinyatakan dalam persamaan (18). Kejituan yang sangat tinggi ini diperlukan bagi mengelak kewujudan kawasan yang rata, iaitu taburan nilai-nilai potensi yang sama

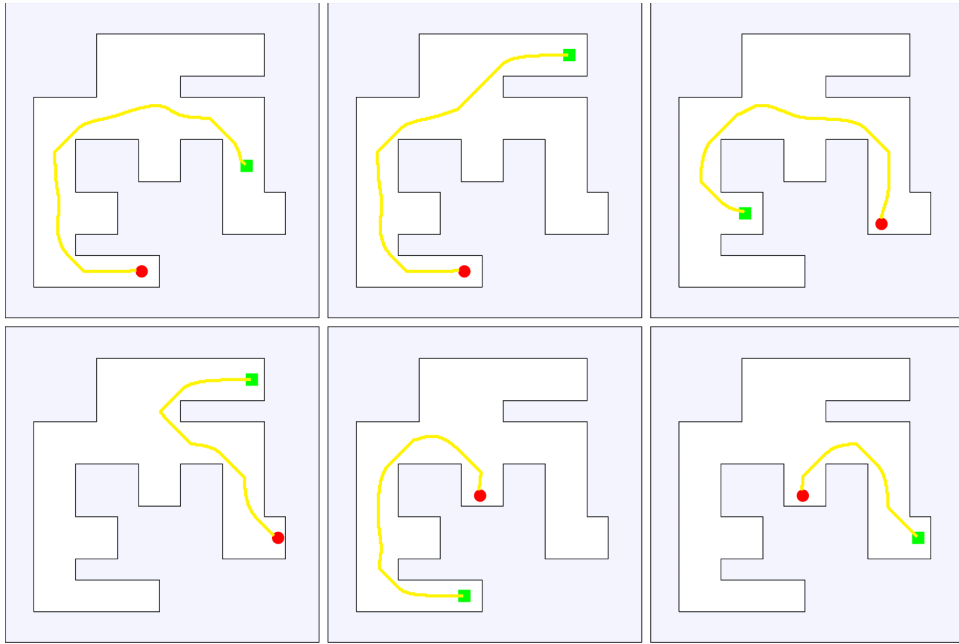
kerana ia akan menyebabkan penjanaaan laluan gagal. Dalam kajian ini, terdapat tiga kaedah lelaran yang akan dipertimbangkan iaitu kaedah SPSOR, SSSOR dan SKSOR. Algoritma ketiga-tiga kaedah lelaran ini dinyatakan masing-masing dalam Algoritma 1, 2 dan 3. Setelah nilai-nilai harmonik  $u$  pada ruang konfigurasi diperolehi melalui kaedah lelaran, proses penjanaaan laluan boleh dilakukan dengan menggunakan teknik CMC. Secara ringkasnya, bermula dari sebarang titik mula, carian teknik CMC bergerak ke titik kejiranan yang lebih rendah nilainya. Proses carian ini berulang sehinggalah ia sampai ke titik destinasi yang mempunyai nilai yang paling rendah. Setelah tiba ke destinasi, laluan yang telah dilalui direkodkan. Selain itu, bilangan lelaran yang diperlukan untuk memenuhi syarat penumpuan serta masa pelaksanaan proses lelaran turut direkodkan.

## **EKSPERIMEN DAN KEPUTUSANNYA**

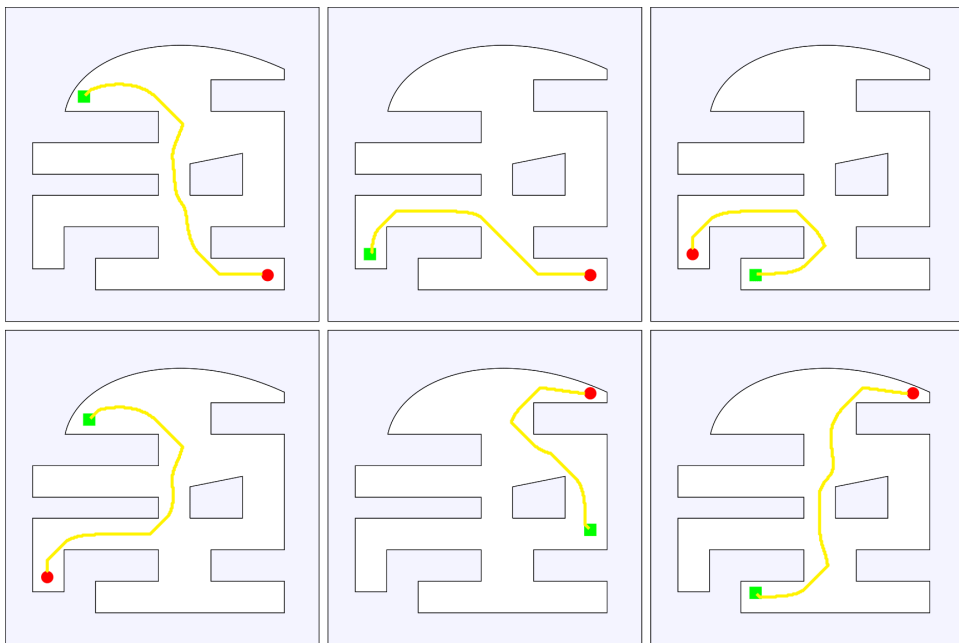
Eksperimen dalam kajian ini telah dijalankan pada ruang konfigurasi dalam bentuk grid yang pelbagai saiz iaitu  $300 \times 300$ ,  $600 \times 600$ ,  $900 \times 900$ ,  $1200 \times 1200$ ,  $1500 \times 1500$  dan  $1800 \times 1800$ . Terdapat dua ruang konfigurasi yang telah digunakan iaitu kes 1 dan kes 2. Komputer yang digunakan mempunyai kelajuan 3GHz dengan muatan RAM sebanyak 1GB. Rajah 5 menunjukkan laluan yang telah berjaya dijana menggunakan algoritma Perancangan laluan dengan kaedah-kaedah lelaran yang dicadangkan. Titik mula bertanda kotak hijau dan titik destinasi bertanda bulatan merah diletakkan pada beberapa lokasi yang berbeza. Kesemua eksperimen yang dijalankan telah berjaya menjana laluan dari titik mula hingga ke titik akhir. Jadual 1 pula menunjukkan bilangan lelaran dan masa pelaksanaan bagi kesemua kaedah lelaran yang telah dipertimbangkan. Kaedah SKSOR adalah yang paling pantas, diikuti oleh kaedah SSSOR dan kaedah SPSOR. Jadual 2 menunjukkan kadar peratus pengurangan bilangan lelaran dan masa pelaksanaan bagi kaedah lelaran SSSOR dan SKSOR berbanding kaedah lelaran lazim SPSOR. Didapati, kaedah SSSOR telah mengurangkan bilangan lelaran sebanyak 48% hingga 49%, manakala kaedah SKSOR pula sebanyak 73% hingga 78%. Daripada segi masa pelaksanaan pula, kedua-dua kaedah SSSOR dan SKSOR adalah lebih pantas berbanding kaedah lazim SPSOR dengan mengurangkan masa pelaksanaan masing-masing sebanyak 48% hingga 52% dan 92% hingga 95%.

## **KESIMPULAN**

Sebagai kesimpulan, eksperimen yang telah dijalankan menunjukkan bahawa kaedah lelaran yang dicadangkan menawarkan teknik pengiraan yang lebih pantas. Ini secara langsung telah meningkatkan prestasi keseluruhan algoritma perancangan laluan robot. Jadual 1 menunjukkan kaedah lelaran yang menggunakan sapuan suku iaitu kaedah SKSOR adalah yang paling pantas diikuti oleh kaedah SSSOR yang menggunakan sapuan separuh. Kaedah SSSOR dan SKSOR telah mengurangkan kekompleksan pengiraan masing-masing sebanyak 50% dan 75% berbanding kaedah SPSOR. Akan datang, teknik berangka yang lebih pantas seperti lelaran sapuan octo (Akhir, Othman, Suleiman & Sulaiman, 2012) dan kaedah lelaran blok (Abdullah, 1991; Othma & Abdullah, 2000; Evans, 1985; Sulaiman et al., 2007) akan dipertimbangkan.



(a)



(b)

**Rajah 5** Laluan yang dijana untuk dua persekitaran iaitu (a) kes 1 dan (b) kes 2.

**Jadual 1** Bilangan lelaran,  $k$  dan masa pelaksanaan,  $t$  (saat).

	Kaedah	N											
		300		600		900		1200		1500		1800	
		$k$	$t$	$k$	$t$	$k$	$t$	$k$	$t$	$k$	$t$	$k$	$t$
Kes 1	SPSOR	2804	5.64	10686	99.78	23218	553.55	40322	1720.28	61840	4190.00	87686	8605.36
	SSSOR	1426	1.52	5473	25.44	11915	147.75	20698	471.20	31780	1137.26	45065	2328.05
	SKSOR	709	0.26	2804	6.19	6135	33.11	10665	120.70	16373	306.44	23218	628.84
Kes 2	SPSOR	2189	4.59	8331	79.06	18130	446.23	31473	1398.06	48258	3415.03	68431	7017.78
	SSSOR	1109	1.19	4260	21.52	9299	121.83	16153	383.69	24786	927.31	35158	1902.27
	SKSOR	542	0.22	2189	5.03	4781	27.08	8331	99.86	12764	249.02	18130	512.30

**Jadual 2** Peratus pengurangan bilangan lelaran dan masa pelaksanaan kaedah SSSOR dan SKSOR berbanding kaedah lazim SPSOR.

	Kaedah	Bilangan lelaran %	Masa pelaksanaan %
Kes 1	SSSOR	48.61 - 49.14	72.61 - 74.50
	SKSOR	73.52 - 74.71	92.69 - 95.39
Kes 2	SSSOR	48.68 - 49.34	72.56 - 74.07
	SKSOR	73.51 - 75.24	92.70 - 95.21

## RUJUKAN

- Abdullah, A. R. (1991). The four point Explicit Decoupled Group (EDG) method: A fast Poisson solver. *International Journal of Computer Mathematics*, 38(1-2), 61-70.
- Akhir, M. K. M., Othman, M., Suleiman, M., and Sulaiman, J. 2012. A new Octo-sweep iterative method for solving two-dimensional elliptic equations. *International Journal of Applied Mathematics and Statistics*, 30(6): 60-71.
- Akishita, S., Kawamura, S., and Hayashi, K. 1990. Laplace potential for moving obstacle avoidance and approach of a mobile robot. In *Japan-USA Symposium on Flexible Automation, A Pacific Rim Conference*, July 9-13, 1990, Kyoto, Japan, pp. 139-142.
- Connolly, C. I., & Grupen, R. A. (1993). The applications of harmonic functions to robotics. *Journal of Field Robotics*, 10(7), 931-946.
- Connolly, C. I., Burns, J. B., & Weiss, R. (1990, May). Path planning using Laplace's equation. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on* (pp. 2102-2106). IEEE.

- Connolly, C. I., Burns, J., and Weiss, R. 1990. Path planning using Laplace's equation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 13-18, 1990, Cincinnati, USA, pp. 2102-2106.
- Dahlquist, G. and Bjorck, A. 1974. *Numerical Methods*. New Jersey: Prentice Hall.
- Daily, R., & Bevely, D. M. (2008, June). Harmonic potential field path planning for high speed vehicles. In *American Control Conference, 2008* (pp. 4609-4614). IEEE.
- Evans, D. J. (1985). Group explicit iterative methods for solving large linear systems. *International Journal of Computer Mathematics*, 17(1), 81-108.
- Faires, J. and Burden, R. 2012. *Numerical Methods*, 4th Edition. Boston: Cengage Learning.
- Garrido, S., Moreno, L., Blanco, D., & Monar, F. M. (2010). Robotic motion using harmonic functions and finite elements. *Journal of Intelligent and Robotic Systems*, 59(1), 57-73.
- Karnik, M., Dasgupta, B., & Eswaran, V. (2002). A comparative study of Dirichlet and Neumann conditions for path planning through harmonic functions. *Computational Science—ICCS 2002*, 442-451.
- Karnik, M., Dasgupta, B., and Eswaran, V. 2002. A comparative study of Dirichlet and Neumann conditions for path planning through harmonic functions. In P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra. (eds.). *Lecture Notes in Computer Science 2330: Computational Science - ICCS 2002*, pp. 442-451. Berlin Heidelberg: Springer-Verlag.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1), 90-98.
- Khatib, O. 1985. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Mar 25-28, 1985, St. Louis, USA, vol. 2, pp. 500-505.
- Kress, R. 1998. *Numerical Analysis*. New York: Springer-Verlag.
- LaValle, S. M. 2011. Motion planning. *IEEE Robotics and Automation Magazine*, 18(1): 79-89.
- LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Society for Industrial and Applied Mathematics.
- Othman, M., & Abdullah, A. R. (2000). An efficient four points modified explicit group poisson solver. *International Journal of Computer Mathematics*, 76(2), 203-217.
- Saad, Y., & Van Der Vorst, H. A. (2000). Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1), 1-33.
- Sasaki, S. (1998, September). A practical computational technique for mobile robot navigation. In *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on* (Vol. 2, pp. 1323-1327). IEEE.
- Sasaki, S. 1998. A practical computational technique for mobile robot navigation. In *Proceedings of the IEEE International Conference on Control Applications*, Sep 4, 1998, Trieste, Italy, vol. 2, pp. 1323-1327.
- Saudi, A., & Sulaiman, J. (2016). Path Planning Simulation using Harmonic Potential Fields through Four Point-EDGSOR Method via 9-Point Laplacian. *Jurnal Teknologi*, 78(8-2), 12-24.
- Sulaiman, J., Othman, M., & Hasan, M. K. (2007, August). Red-black EDGSOR iterative method using triangle element approximation for 2D Poisson equations. In *International Conference on Computational Science and Its Applications* (pp. 298-308). Springer, Berlin, Heidelberg.
- Sulaiman, J., Othman, M., and Hasan, M. K. 2007. Red-Black EDGSOR iterative method using triangle element approximation for 2D Poisson equations. In O. Gervasi and M. L. Gavrilova. (eds.). *Lecture Notes in Computer Science 4707: Computational Science and Its Applications - ICCSA 2007*, pp. 298-308. Berlin Heidelberg: Springer-Verlag.
- Tran, N., Nguyen, D.-T., Vu, D.-L., and Truong, N.-V. 2013. University of Information Technology, Vietnam National University-Ho Chi Minh City, Vietnam. In *Proceedings of the IEEE International Conference on Control, Automation and Information Sciences*, Nov 25-28, 2013, Ho Chi Minh, Vietnam, pp. 317-321.

- Young, D. (1954). Iterative methods for solving partial difference equations of elliptic type. *Transactions of the American Mathematical Society*, 76(1), 92-111.
- Young, D. M. 1950. Iterative Methods for Solving Partial Difference Equations of Elliptic Type. PhD Thesis. Harvard University.
- Yousif, W. S., & Evans, D. J. (1986). Explicit group over-relaxation methods for solving elliptic partial differential equations. *Mathematics and Computers in Simulation*, 28 (6), 453-466.