2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Improved Coverage Path Planning Using a Virtual Sensor Footprint: a Case Study on Demining

Sedat Dogru[1] and Lino Marques[1]

Abstract— **Coverage performance in a coverage path planning problem depends both on the path created and on the footprint of the sensor used. The footprint can be increased either by increasing the size of the sensor, or by mounting the sensor on a robotic arm to allow scanning over larger areas as the platform moves, effectively creating a virtual sensor with a larger footprint than the physical sensor's. However, the virtual footprint comes at a cost requiring formulating an optimization problem for the area of interest. In this work, three common strategies to use a metal detector on a platform are discussed, their time and energy performances are formulated and the corresponding optima are found.**

I. Introduction

Buried explosives, whether landmines or unexploded ordinance (UXO), are still widespread and hence pose a huge danger to humans [1], [2]. These explosives are sometimes distributed over large areas, raising the need for optimal cleaning strategies in order to save both time and energy. The explosives can be detected using various sensors like metal detectors, Ground Penetrating Radar (GPR) and odor sensors. These sensors are either carried by human operators or they are mounted on mobile platforms, including robots with varying amounts of autonomy. A crucial step in improving efficiency of search and cleaning is to design trajectories that guarantee both full coverage and optimization of a performance metric for the platform or the operator. Full coverage, with the overall energy consumption as a metric, has already been extensively studied in the literature as the Coverage Path Planning (CPP) problem, and many different methods to find optimum trajectories for both 2D and 3D environments were presented [3], [4]. The methods proposed in the literature try to reduce the overall trajectory length, and in some cases the number of rotations as well, but making sure that the resulting trajectory covers the whole area of interest. Trajectory length is a trivial contributor to the overall energy consumption. Contribution of rotations to the overall energy consumption is variable from platform to platform and it is known to be considerably high especially for skid-steered platforms [5]. Existing CPP approaches generally assume that the sensor or actuator footprint of a robot is fixed and well-known upfront, and use this fixed shape and size to create trajectories optimizing the overall energy consumption. Size of the footprint, although taken to be fixed in the literature, can affect the overall performance considerably. A larger footprint would let covering a given area with fewer passes, reducing both the overall trajectory length and the number of rotations, consequently improving performance.

In landmine detection the footprint is increased using two main approaches. The more direct approach is forming an array out of single sensing units, hence scanning a wider area with a single pass of the array [6]–[9]. The indirect approach is attaching a sensing unit to a robotic arm mounted to a moving platform, and scanning an area as large as the arm allows at each step [10]–[13]. Hence, although the sensing unit's physical footprint is not changed, the virtual footprint created by the motion of the sensor is increased, and as a result the CPP algorithms can be run using a footprint much larger than the physical sensor's footprint. For example, if a robot can scan an area in front twice as wide as the robot with a metal detector, then the effective footprint used in the CPP algorithm would be twice the width of the robot. The first approach requires a more expensive and relatively more complex sensor due to issues like cross-talk and sensor fusion. The second approach relies more on proper control of the sensor with the arm, making it more interesting for and common in the robotics community. It is also possible to fuse the two approaches, i.e. mount an array of sensors on a robotic arm and increase the footprint significantly compared to not only a single sensor unit but also compared to the array [13].

Despite these alternative approaches to increase the footprint in the literature, we are not aware of any study that compares the CPP performance of these different approaches. In this work, we present three sweeping strategies that can be used to increase the virtual footprint of a metal detector. Then we derive a set of equations describing both the time and energy cost of these strategies. Afterwards, the minima of these strategies are calculated for a representative region using time and energy consumption values obtained from a demining robot, eventually highlighting the strength of each strategy.

The first strategy, called Line Sweep Strategy (LSS), is performed by keeping the metal detector array fixed in the front with the long edge lying parallel to the front of the robot (Fig. 1a). This is a very common placement for metal detector arrays [6], [7]. In this configuration, the wide array sweeps along the trajectory of the vehicle as the vehicle moves ahead. If the platform is looking for landmines, not UXOs, the array has to be as wide as the platform to make sure that landmines on which the vehicle may step are

This work was supported by ISR - University of Coimbra, project UID/EEA/00048/2013 funded by FCT - Fundação para a Ciência e a Tecnologia.

[1]Sedat Dogru and Lino Marques are with the Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal {sedat, lino}@isr.uc.pt

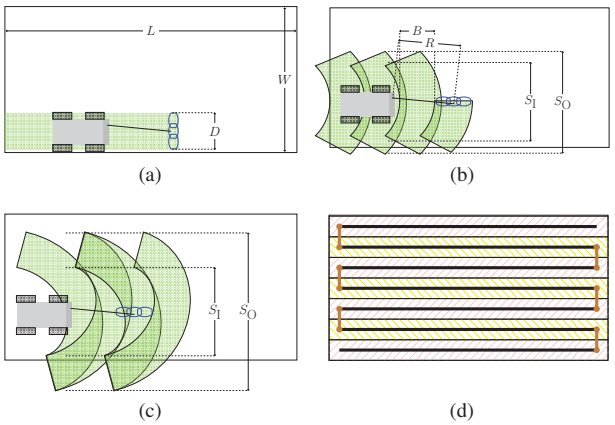


Fig. 1. (a, b, c) Different sweeping strategies implemented with the robot. Each sweeping direction is hatched in a different direction, to highlight overlaps. (a) Line sweep strategy, $\omega_{\text{ARM}} = 0, v_{\text{R}} > 0$ (b) Fixed Circular Sweep strategy $|\omega_{\text{ARM}}| > 0, v_{\text{R}} = 0$ when sweeping, $\omega_{\text{ARM}} = 0, v_{\text{R}} > 0$ between segments, (c) Moving Circular Sweep sweep strategy, $|\omega_{\text{ARM}}| > 0, v_{\text{R}} > 0$ (d) A typical robot trajectory for a CPP on a rectangle. Patches are shown in different colors.

detected in a timely manner.

In the second strategy, which is called Fixed Circular Sweep Strategy (FCSS), the array is fixed on the arm such that the long edge is parallel to the arm (Fig. 1b). In this strategy, the robot stops, sweeps from side to side, then stops the arm, moves sufficiently ahead, and then sweeps again, repeating the process till the end. In the third strategy, which is called Moving Circular Sweep Strategy (MCSS), the array is placed like in FCSS, but in contrast to FCSS, the robot does not stop as the arm sweeps from side to side (Fig. 1c). Both in FCSS and MCSS the width of the area swept in front of the robot can be adjusted by changing the sweep angle.

In order to compare the performance of different sweeping strategies and highlight the advantages and disadvantages of virtually increasing the footprint of the sensor, a rectangular shape with a trivial optimum solution is chosen. This choice is based on the fact that CPP for convex shapes forms the basis solution for more complex shapes, as explained next. The Coverage Path Planning problem is known to be NP-hard in general [14], and therefore the optimal solution is approximated using some heuristic approaches. A common approach to solve the CPP problem is dividing the free space into obstacle free non-overlapping regions using a cell decomposition method like Boustrophedon [15], trapezoidal [16] or Morse decomposition [17]. Then each region is covered with simple back and forth motions. The performance of these decomposition algorithms is improved by choosing a proper sequence to visit the regions, as well as choosing proper orientation for the sweeps [16], [18], [19].

II. Mathematical Models

A. Arena and the Metal Detector Array

For illustrative purposes, it is assumed that the area to be covered is a rectangle measuring $L \times W$, with $L \geq W$. Hence a trivial CPP solution, which is sweeping parallel to the longest edge L, could be used (Fig. 1d). Note that a convex shape is a building block of a general CPP solution, and hence the results are usable for arbitrarily shaped areas. Let N_p be the number of parallel patches the given rectangle can be divided into, assuming that each patch can be fully covered with a single pass of the robot. N_p clearly depends on the effective sweep width of the platform and hence on the sweeping strategy in use. Consequently, a platform has to perform N_p translations along the long edge. Additionally, the platform will perform $(N_p - 1)$ transitions in order to move from one patch to the next. Although it is possible to perform these transitions as rotations of various radius of curvature, in this work the robot performs on the spot rotations. This type of rotations is easily performed by skid-steered robots and require least space. Hence, a robot that needs to sweep the next patch has to stop at the end, perform a 90^o rotation, move to the center of the neighboring patch and perform another 90^o rotation before starting sweeping the new patch.

Metal detector coils may have different shapes, like double D, ellipse or circular. Additionally, metal detector arrays may have different shapes depending on the placement of the coils. Since in this work the metal detector is used in a sweeping motion, the effective cross-section of the array is more important compared to the exact shape, and hence the sensor footprint is characterized by a D-long line segment that passes through the center of the array and that is perpendicular to the sweeping direction (Fig. 1a).

An important parameter that has to be taken into account when comparing the performance of different sweeping strategies is the maximum speed the metal detector can be moved across the surface. Metal detectors have a finite response time, and some are equipped with low pass filters to improve resilience to noise. Both of these factors limit the maximum speed the metal detector can be moved over the surface without misdetections. Note that, this limit, denoted as $v_{\text{MD}}^{\max}$ in this work, where v_{MD} stands for the speed of the metal detector with respect to ground, may restrict the maximum speed of the platform with respect to ground, v_R, as well.

B. Performance Metrics

During a demining mission a robot will be spending time on sweeping with the arm (t_{sweep}), moving along patches ($t_{\text{translation}}$) and moving from patch to patch (t_{corner}). Both sweeping and translation of the platform along the patch may coincide depending on the sweeping strategy. Hence, with $s = 0$ indicating coincidence (MCSS and LSS), and $s = 1$ no coincidence (FCSS), the total time spent during a demining mission is given by

$$t_{\text{coverage}} = s t_{\text{sweep}} + t_{\text{translation}} + t_{\text{corner}} \quad (1)$$

Similarly a robot will be consuming energy for sweep with the arm (E_{sweep}), motion along the patches ($E_{\text{translation}}$) and transition from patch to patch (E_{corner}). The robot will also be spending energy on its sensors and on-board computer, which would be on during the whole mission (E_{system}). Although

$E_{\text{translation}}$ and E_{corner} are consumed by the same components, namely the wheel motors, E_{corner} contains rotations, which cause significantly higher consumption, hence it is better to treat the components separately. In contrast to the time, E_{sweep} exists whenever the arm is used actively (FCSS, MCSS). Hence the energy consumed during a mission can be calculated as

$$E_{\text{coverage}} = E_{\text{sweep}} + E_{\text{translation}} + E_{\text{corner}} + E_{\text{system}} \quad (2)$$

The individual components of energy can generally be calculated integrating the power consumption of the individual components along the whole trajectory. However, since all the work is done at constant speed the power is constant and therefore the energy can be calculated simply using the product of time spent and power as $E_x = \Delta t_x P_x$, with x standing for the consumer. Note that since transitions on the corners are divided into 90^{o} rotations and straight translations, to calculate the consumed energy on transitions, t_{corner} has to be divided into two components, $t_{\text{crotations}}$ and $t_{\text{ctranslations}}$.

C. Sweep Strategies

1) Line Sweep Strategy: In this strategy the metal detector array is fixed in the front with the long edge parallel to the front of the robot, and hence as the robot moves forward an area of width D is scanned (Fig. 1a), allowing to scan the whole $L \times W$ area by dividing into $N_p = \lceil W/D \rceil$ parallel patches, with $\lceil\ \rceil$ standing for the ceiling function. Since the robot has to travel along all the patches the total distance covered by the robot will be $N_p L$. Additionally, the robot will need to perform $(N_p - 1)$ patch to patch transitions. The transitions will consist of $2(N_p - 1)$ many 90^{o} rotations, and $(N_p - 1)$ many D long translations. During the mission, the robot's speed along the patch is limited by $v_{\text{R}} < v_{\text{MD}}^{\text{MAX}}$. During the transitions between the patches however, the robot is free to choose the most optimum rotational and translational speeds. The time needed can be calculated as the ratio of the distance covered and the corresponding speeds. Hence the individual components of equation 1 for minimum time needed to cover the $L \times W$ rectangle by the robot using line sweep strategy are given by

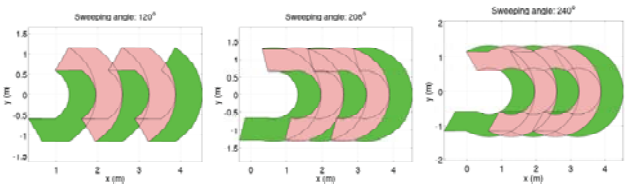
$$t_{\text{translation}} = N_p \frac{L}{v_{\text{MD}}^{\text{MAX}}} \quad (3)$$

$$t_{\text{sweep}} = N_p \frac{L}{v_{\text{MD}}^{\text{MAX}}} \quad (4)$$

$$t_{\text{corner}} = (N_p - 1)\frac{\pi}{\omega_R} + (N_p - 1)\frac{D}{v_R} \quad (5)$$

with $s = 0$ in equation 1.

2) Fixed Circular Sweep Strategy: In the second strategy the long edge of the array is parallel to the arm, hence as the arm sweeps an α wide arc a D-thick ring segment is swept at each step (Fig. 1b). In this mode, the robot starts stationary and the arm sweeps from $\frac{-\alpha}{2}$ to $\frac{\alpha}{2}$, where α is the sweep angle; then the arm stops and the robot moves ahead by B. Afterwards the robot stops and the arm sweeps from $\frac{\alpha}{2}$ to $-\frac{\alpha}{2}$. This sequence is repeated till the end of the line. Then the robot moves to a neighboring patch.



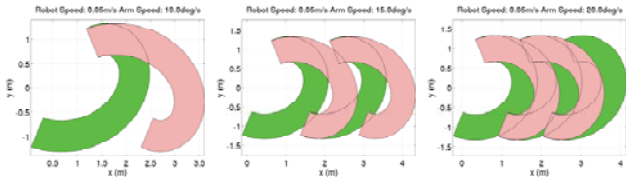
(a) (b) (c)

Fig. 2. Different sweeping angles showing different intersection points between the successive scans for the FCSS.

In Fig. 1b it can be seen that, the area swept by the arm as the robot goes ahead, forms a concave shape with many holes on the edges. In order to ensure full coverage, the robot has to re-sweep this area, which is lying between S_{O} and S_{I}, which respectively represent the length of the maximum width and the minimum width covered as the robot moves along a line. Both, S_{O} and S_{I} depend on α, and S_{I}, the effective sweep width, directly determines the optimality of the final trajectory. S_{O} can be directly calculated as the width of the segment of the outer circle, and it is given by $S_{\text{O}} = 2(R + \frac{D}{2}) \sin \frac{\alpha}{2}$ for sweep angles up to 180^{o}, after which it is fixed to $(2R + D)$. Value of S_{I} similarly depends on the sweep angle, where for smaller values of α it can be calculated from the intersection of the line boundary of the second sweep with the outer circle of the first sweep (Fig. 2a). For larger α values, at which the end of the second sweep starts to pass over the first sweep (at $\alpha \geq \alpha_c$), S_{I} becomes fixed and can be calculated using the intersection of the outer circles of the first and the second sweep (Fig. 2b). Hence, S_{I} can be written as

$$S_{\text{I}} = 2 \begin{cases} \sin\frac{\alpha}{2} \times \\ \left\{ -D\cos\frac{\alpha}{2} + \sqrt{\left(R + \frac{D}{2}\right)^2 - B^2 \sin^2\frac{\alpha}{2}} \right\}, & \text{if } \alpha \leq \alpha_c \\ \sqrt{\left(R + \frac{D}{2}\right)^2 - \left(\frac{B}{2}\right)^2}, & \text{otherwise} \end{cases} \quad (6)$$

When using the fixed circular sweep strategy, the number of patches should be calculated using the effective sweep width, hence the number of patches will be $N_p = \lceil \frac{W}{S_{\text{I}}} \rceil$. The robot has to separately spend time to sweep with the metal detector, and to move itself from one measurement point to the next. This would require sweeping $N_s = \lceil \frac{L}{B} \rceil$ many times along a patch, and covering a distance of $N_p L$. Note that, since W is not an integer multiple of S_{I} in general, W will consist of N_p patches that are as wide as S_{I}, and a final patch that is smaller than S_{I}. Hence, in order to further improve the performance, this final patch can be swept with a smaller sweep angle, called α_{leftover}. In this strategy sweep speed of the arm, ω_{arm}, is restricted by $\omega_{\text{arm}} \leq \frac{v_{\text{MD}}^{\text{MAX}}}{R + D/2}$. However, the robot is free to choose v_R as it wishes, though it is restricted by the dynamics of the robot. As a result the times needed for the individual tasks to cover the $L \times W$



(a) (b) (c)

Fig. 3. Different overlap patterns for the MCSS depending on the speed of the platform and the arm.

rectangle using fixed circular sweep strategy are

$$t_{\text{translation}} = N_p \frac{L}{v_R} \quad (7)$$

$$t_{\text{sweep}} = (N_p - 1) N_s \frac{\alpha}{\omega_{\text{arm}}} + N_s \frac{\alpha_{\text{leftover}}}{\omega_{\text{arm}}} \quad (8)$$

$$t_{\text{corner}} = (N_p - 1) \frac{\pi}{\omega_R} + (N_p - 1) \frac{S_{\text{I}}}{v_R} \quad (9)$$

with $s = 1$ in equation 1.

In order to find time-optimal or energy-optimal parameters equation 1 or 2 has to be minimized with respect α. Increasing S_{I} will decrease the number of patches N_p, hence will contribute to save time and energy. However, increasing S_{I} requires increasing the sweep angle α, which would increase the time needed to sweep.

3) Moving Circular Sweep Strategy: With the moving circular sweep strategy, the robot keeps moving at constant speed as the arm sweeps from side to side. A clear advantage of this strategy over the previous one is removing the need for accelerations to start and stop the platform. Accelerations and decelerations may introduce significant energy cost particularly for bigger platforms. Additionally accurate control and trajectory following becomes more difficult with frequent high speed starts and stops.

The moving circular sweep strategy, similar to the fixed circular sweep strategy, has an effective sweep width S_{I}. In this strategy, depending on the speed of the platform and speed of the arm, consecutive sweeps may intersect at different points. The outer arc of the first sweep may cut the inner arc of the second sweep (Fig. 3a) or the border end line of the second sweep (Fig. 3b). However, in both cases the inner arc of the second sweep should always stay inside the outer arc of the first sweep till the first intersection, else gaps in the sweep pattern may form (Fig. 3b). Unfortunately S_{I} can not be calculated in open form for this strategy. However, the parametric equations of the curves given below can be used to numerically calculate S_{I}.

$$x_i(t) = \left(R - \frac{D}{2}\right) \cos\left(\frac{s\alpha}{2} - s\omega t\right) + x_0 + kvT \quad (10)$$

$$y_i(t) = \left(R - \frac{D}{2}\right) \sin\left(\frac{s\alpha}{2} - s\omega t\right) \quad (11)$$

$$x_o(t) = \left(R + \frac{D}{2}\right) \cos\left(\frac{s\alpha}{2} - s\omega t\right) + x_0 + kvT \quad (12)$$

$$y_o(t) = \left(R + \frac{D}{2}\right) \sin\left(\frac{s\alpha}{2} - s\omega t\right) \quad (13)$$

$(k-1)T < t < kT$, with $s = +1$ if k is even, $s = -1$ if k is odd. In these equations T stands for the time needed to complete a sweep from side to side ($T = \frac{\alpha}{\omega_{\text{arm}}}$), and k is the index of the current sweep. $(x_i(t), y_i(t))$ and $(x_i(t), y_i(t))$ stand respectively for the inner and outer arcs of the sweeps.

Calculation of S_{I} is a three stage process. The first stage is finding the intersections of the outer arc of the first sweep with the inner arc of the second sweep using $(x_o(t), y_o(t)) = (x_i(t'), y_i(t'))$ and consecutive k values, like 0 and 1. If real number solutions are found, depending on their position with respect to the x-axis (above or below), they are taken either as valid positive effective width values ($S_{\text{I}} = 2y_o(t_{\text{intersection}})$), or as negative invalid values. This can be better explained using Fig. 3a. Since in Fig. 3a the first sweep is from $-\frac{\alpha}{2}$ to $\frac{\alpha}{2}$, an intersection with the second sweep above the x-axis will create a negative S_{I}. This is not a desired intersection, because it means at least half of the front of the robot is not covered by the arm, introducing a dangerous situation for the robot. If no real number solution is found (i.e. the outer arc of the first sweep and the inner arc of the second sweep do not intersect), the next step is checking whether the straight line segment corresponding to the edge of the sweep and given by $y \cot \frac{\alpha}{2} + x_0 + (k+1)vT = x$ intersects the outer circle of the first sweep, and in case not, the intersection of the outer circles of both sweeps are searched, and this would give the largest S_{I} possible.

In this strategy both the speed of the robot and the speed of the arm are constrained by $v_{\text{MD}}^{\text{MAX}}$, because the robot has to move as the arm is sweeping. The constraint can be written as $|\vec{\omega}_{\text{arm}} \times \vec{R} + \vec{v}_R| \leq v_{\text{MD}}^{\text{MAX}}$ in vector form. Note that although the norm $|\vec{\omega}_{\text{arm}} \times \vec{R}|$ is fixed, the heading of the arm speed vector is changing, and hence v_R's maximum value depends on the sweep angle.

$t_{\text{translation}}$, t_{sweep} and t_{corner} will have the same formulas as the continuous sweep strategy, however since sweep and translation are coincident $s = 0$ in equation 1.

III. Experimental Setup

For this work a customized Husky A200 from Clearpath was used (Fig. 4). The robot is equipped with a computer with an i7 processor, and various other components like stereo cameras, a tilt unit for a lidar, a lidar, a dual antenna Global Navigation System and an outdoor Wi-Fi antenna. The robot also has a 4 degrees of freedom (DoF) robotic arm with a three coil pulse induction metal detector from Vallon (Vallon VMP3). The first two DoF of the arm correspond to the horizontal (yaw) and vertical (pitch) axes, and they are independently controlled by electric motors. Control in the yaw axis allows sweeping with the metal detector over the surface. Control in the vertical axis allows keeping a fixed distance between the metal detector and the surface. The last two DoF are manually controlled and they are used to position the metal detector with respect to the arm. The metal detector array used in this work has a low pass filter and hence it's maximum speed $v_{\text{MD}}^{\text{MAX}}$ was fixed to 0.25 m/s. The array has a length of 0.65 m. The length of the robotic arm is 1.05 m.

Fig. 4. The robot used for this work.

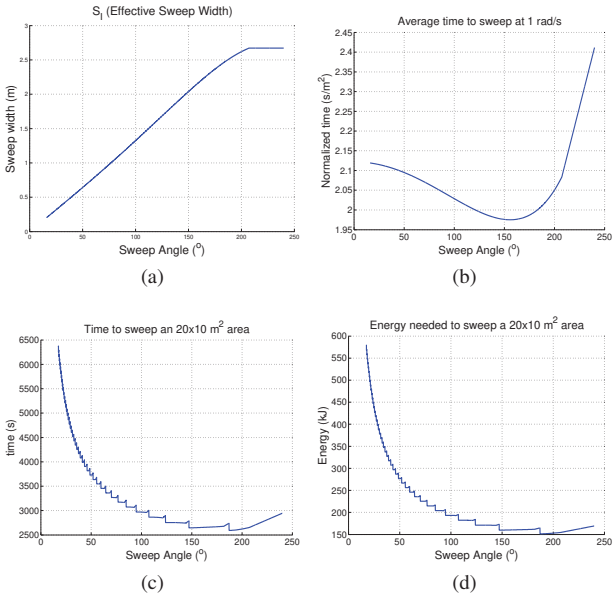


Fig. 5. Sweeping performance of the fixed continuous sweep strategy (FCSS) (a) The effective width (b) Normalized sweep speed (c) Time needed to sweep the testing area (d) Energy needed to sweep the testing area

In order to asses the performance of the three studied strategies the time needed and the energy consumed to cover a 10 m x 20 m region was calculated using typical operating velocities as well as energy consumption values measured on the field using Husky. The speed of the robot was kept at $v_{\text{MD}}^{\text{MAX}}$ for the LSS strategy, and the arm was kept fixed. For the FCSS strategy the linear speed of the arm was fixed at $v_{\text{MD}}^{\text{MAX}}$ and the translational speed of the robot as it moved between the sweeps was also kept at $v_{\text{MD}}^{\text{MAX}}$ to keep the robot stable during start and stop. The rotational speed of the robot on the corners were fixed to 0.4 rad/s, and the translational speed on the corners was also fixed to $v_{\text{MD}}^{\text{MAX}}$.

IV. Results & Discussion

Due to its power to change the number of rotations and overall path length, the most important parameter of a sweeping strategy is the effective sweep width, S_{I}. As was shown in the previous section S_{I} depends on the sweep angle, and the graph of this dependence for FCSS is shown in Fig. 5a. S_I, can be seen to be increasing almost linearly up to 208^{o}, after which it is constant. Increasing the sweep angle however does not come for free and it has a penalty in terms of coverage time. In Fig. 5b it can be seen that for a single sweep with the arm from side to side, the least time needed to cover 1 m^2 area happens at 155^{o}. When scanning the much larger arena studied in this work, the optimum time to area ratio occurs at a bigger sweep angle (Fig. 5c). A similar effect could be seen for the energy in Fig. 5d. With increasing sweep angle the total energy needed to sweep a region decreases. Note that both Figs. 5c and 5d show jumps at some sweep angles due to the ceiling function used to more accurately represent the times and energies.

MCSS's performance depends not only on the sweep angle, but also the speed of the robot. This can clearly be seen in Fig. 6. In Fig. 6a it can be seen that the effective sweep width is small especially when the robot is faster, which is due to the fact that high speeds leave gaps in the sweep pattern as was shown in Figs. 3a and 3b. This shows itself as very low efficiency, both in time and energy (Fig. 6b, 6c, 6d). Optimum time and energy consumption values appear around bigger sweep angles, and speeds.

Performance of LSS depends only on the arena and it does not have any parameters to be adjusted. Hence the performance is fixed for a given arena, and it is given in the first rows of tables I and II.

For comparison purposes the optimum time and energy performance of FCSS and MCSS are given on tables I and II. The line sweep strategy can be seen to be the fastest strategy, despite its need for considerably more time to perform the bigger number of rotations compared to the other strategies. The other two strategies save on rotation time but have to spend considerable time sweeping the terrain with the arm, almost doubling the overall mission time.

The minimum energy consumption values given in table II reveal that if the energy spent only by the arm and the wheels are taken into account (ignoring the system), both FCSS and MCSS surpass LSS with a clear margin. In such a case FCSS, MCSS and LSS would be consuming only 39 kJ, 60 kJ and 100 kJ respectively, showing that FCSS consumes only 39% of LSS to cover the arena in this work. Despite being less efficient than FCSS, MCSS is also clearly much better than LSS when only arm and wheel motor consumptions are taken into account. However, the metal detector array used in this work is a slow device. Therefore, the longer it takes for a mission to complete, the more apparent the contribution of the system to the overall energy consumption becomes. As can be seen in table II, in terms of net mission energy FCSS is better than LSS only by 11 kJ, corresponding to just a 7% improvement. With the system's energy consumption

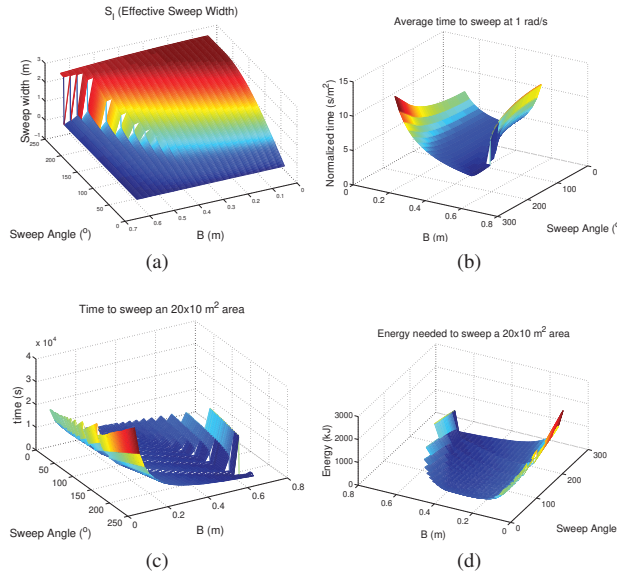


Fig. 6. Sweeping performance of the moving circular sweep strategy (MCSS) (a) The effective width (b) Normalized sweep speed (c) Time needed to sweep a testing area (d) Energy needed to sweep a testing area

TABLE I

BEST TIME PERFORMANCE OF EACH SWEEPING STRATEGY

Strategy	Arm Sweep Time (s)	Platform Translation Time (s)	Platform Corner Time (s)	Net Mission Time (s)	$\alpha(^{\circ})$
LSS	0	1280	157	1437	N.A.
FCSS	2226	312	54	2592	187.2
MCSS	2832	2832	54	2886	229

included, MCSS becomes the worst performer with 184.30kJ

V. CONCLUSION

In this work time and energy performance of three different sweeping strategies using a metal detector mounted on a robotic arm were studied for a representative shape. It was shown that LSS is clearly the fastest sweeping strategy, and MCSS is the slowest. It was also shown that, FCSS and MCSS, thanks to their considerably increased effective sweep width, are able to cover the same area spending much less energy on the wheel and arm motors. However, due to the low limit on the maximum speed of the metal detector, energy consumption of the on-board computer and sensors exceeds, in some cases considerably, the power spent on the motors.

TABLE II

BEST ENERGY PERFORMANCE OF EACH SWEEPING STRATEGY

Strategy	Arm Sweep Energy (kJ)	Platform Transl. Energy (kJ)	Platform Corner Energy (kJ)	System Energy (kJ)	Net Mission Energy (kJ)	$\alpha(^{\circ})$
LSS	0	51.20	49.07	62.07	162.36	N.A.
FCSS	16.03	12.48	10.92	112.00	151.43	187.2
MCSS	20.39	28.32	10.93	124.66	184.30	229

REFERENCES

[1] International Campaign to Ban Landmines Cluster Munition Coalition (ICBL-CMC), *Landmine Monitor 2017*, 2017.

[2] M. Sato and Y. Kadoya, "UXO clearance operation in Laos," in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIII*, vol. 10628. International Society for Optics and Photonics, 2018, p. 106280J.

[3] H. Choset, "Coverage for robotics a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[4] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[5] S. Dogru and L. Marques, "Power characterization of a skid-steered mobile field robot with an application to headland turn optimization," *Journal of Intelligent & Robotic Systems*, Jan 2018.

[6] J. Marble, I. McMichael, and D. Reidy, "Estimating object depth using a vertical gradient metal detector," in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIII*, vol. 6953. International Society for Optics and Photonics, 2008, p. 695313.

[7] K. C. Ho and P. D. Gader, "On the estimation of target depth using the single transmit multiple receive metal detector array," in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*, vol. 8357. International Society for Optics and Photonics, 2012, p. 835709.

[8] L. Nuzzo, G. Alli, R. Guidi, N. Cortesi, A. Sarri, and G. Manacorda, "A new densely-sampled ground penetrating radar array for landmine detection," in *Proceedings of the 15th International Conference on Ground Penetrating Radar*, June 2014, pp. 969–974.

[9] G. De Cubber, H. Balta, and C. Lietart, "Teodor: A semi-autonomous search and rescue and demining robot," in *Applied Mechanics and Materials*, vol. 658, 2014, pp. 599–605.

[10] P. G. de Santos, J. Cobano, E. Garcia, J. Estremera, and M. Armada, "A six-legged robot-based system for humanitarian demining missions," *Mechatronics*, vol. 17, no. 8, pp. 417–430, 2007.

[11] E. F. Fukushima, M. Freese, T. Matsuzawa, T. Aibara, and S. Hirose, "Humanitarian demining robot Gryphon-current status and an objective evaluation," *International journal on smart sensing and intelligent systems*, vol. 1, no. 3, pp. 735–753, 2008.

[12] H. Montes, L. Mena, R. Fernández, J. Sarria, and M. Armada, "Inspection platform for applications in humanitarian demining," in *18th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines. Assistive Robotics*, Sept 2015, pp. 446–453.

[13] S. Dogru and L. Marques, "Shape reconstruction using a mobile robot for demining and UXO classification," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, May 2017.

[14] E. Arkin, S. Fekete, and J. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geometry: Theory and Applications*, vol. 17, no. 1-2, pp. 25–50, 2000.

[15] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[16] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *Journal of Field Robotics*, vol. 26, no. 8, pp. 651–668, 2009.

[17] E. Acar, H. Choset, A. Rizzi, P. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.

[18] P. A. Jimenez, B. Shirinzadeh, A. Nicholson, and G. Alici, "Optimal area covering using genetic algorithms," in *Advanced Intelligent Mechatronics, 2007 IEEE/ASME International Conference on*. IEEE, 2007, pp. 1–5.

[19] S. Dogru and L. Marques, "Energy efficient coverage path planning for autonomous mobile robots on 3D terrain," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, April 2015, pp. 118–123.