Autonomous Vehicles for Cleaning Solar Panels

Nasir K. Memon
Qatar Environment and Energy Research Institute (QEERI), Hamad bin Khalifa University (HBKU), Qatar Foundation
College of Science and Engineering, HBKU
Doha, Qatar
nmemon@qf.org.qa

Abstract: **The impact of soiling or dust is a major challenge in the use of photovoltaic (PV) panels in numerous regions of the world. In this paper we present a technology platform that can be used to deploy autonomous vehicles for cleaning solar panels. A PV power station is modeled using the Virtual Robot Experimentation Platform and Robot Operating System (ROS) is used for the algorithms to control the vehicle. Technical information with regards to mapping, localization, path planning, and obstacle avoidance is presented.**

autonomous vehicles; solar panel cleaning; robot operating system

I. INTRODUCTION

The use of autonomous vehicles within the solar industry can significantly lower the balance of system costs associated with solar. Autonomous vehicles / robots can possibly be used to inspect, clean, track, and install solar panels. The use of such technology can further reduce the cost associated with solar energy. The use of autonomous robots has transformed numerous industries. For example, in the automobile industry, robotic arms have completely changed the assembly line system, resulting in higher manufacturing precision at lower costs. Also, the use of autonomous vehicles is expected to play an important role in transportation, with companies like Google and Uber investing heavily in this field.

Soiling is a great challenge to photovoltaics (PVs) in numerous regions, especially in the MENA region [1]. Soiling reduces the output of PV panels in the range of 5 - 20% per month in most desert areas. Technologies that automate the cleaning process of PV power stations not only increase energy output (kWh) per year, but reduce the operational and maintenance costs associated with solar. Additionally, from a human safety standpoint, there is a potential for electrical shock during manual cleaning of solar farms, as well as health hazards from exposure to harsh outdoor conditions. Automating the solar panel cleaning process would eliminate these concerns.

The focus of this paper is to outline a technical approach for the deployment of autonomous vehicles for cleaning solar panels. A conceptual illustration of the vehicle, which includes dry cleaning is shown in Fig 1. The paper primarily focuses on the technical details of deploying such a vehicle in a PV power station. Key advantages of such an approach is that 1) a single robot/vehicle can be used to clean all panels, 2) limited or no external modifications to an existing solar farm, 3) can possibly be modified for use with solar thermal plants, and 4) limited or no water is needed for cleaning.

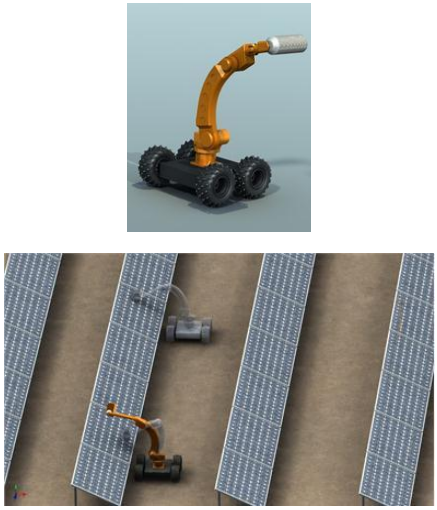
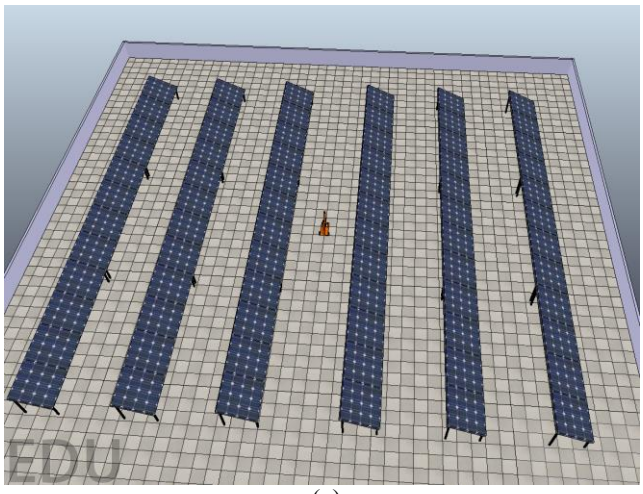


Figure 1: Autonomous vehicle for solar panel cleaning

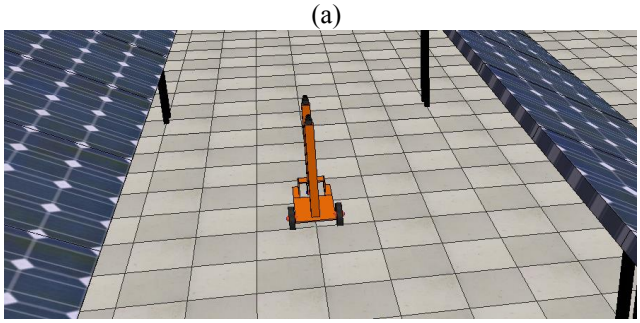
II. TECHNICAL APPROACH

The PV power station is modeled using the Virtual Robot Experimentation Platform (V-REP), which is a versatile simulation framework [2]. Fig. 2a shows the PV power station used for the simulation, which comprises of 6 rows of panels. The dynamics of the vehicle, Fig. 2b, is also modeled within V-REP using the Bullet physics model. Two laser scanners are attached to the robot using the built-in V-REP library for Hokoyo_URG_04LX_UG01. Hence, the laser data obtained in this work, is designed to simulate an actual PV power station.

978-1-5090-5713-9/16/$31.00 ©2016 IEEE



(a)



(b)

Figure 2: (a) V-REP is used to simulate the PV power station with 6 rows (b) vehicle with two laser scanners attached.

Robot Operating System (ROS) is used as the central processing system for autonomously navigating the vehicle around the PV power station. Fig. 3 outlines the different libraries used for navigation, which includes perception, localization, mapping, obstacle avoidance, and path planning. V-REP has plug-in libraries to communicate with ROS topics through publishers and subscribers [2]. V-REP publishes laser data, which ROS uses for localization and mapping. Additionally encoder data is obtained from V-REP for odometry. Finally ROS sends steering and velocity data to V-REP, which controls the robot.

III. MAPPING

Before our vehicle is able to autonomously navigate around the PV power station, it must generate a map of the surrounding, which is also used to localize the vehicle. Using the solar panels as natural landmarks, the robot utilizes simultaneous localization and mapping (SLAM) to generate a map of the environment.

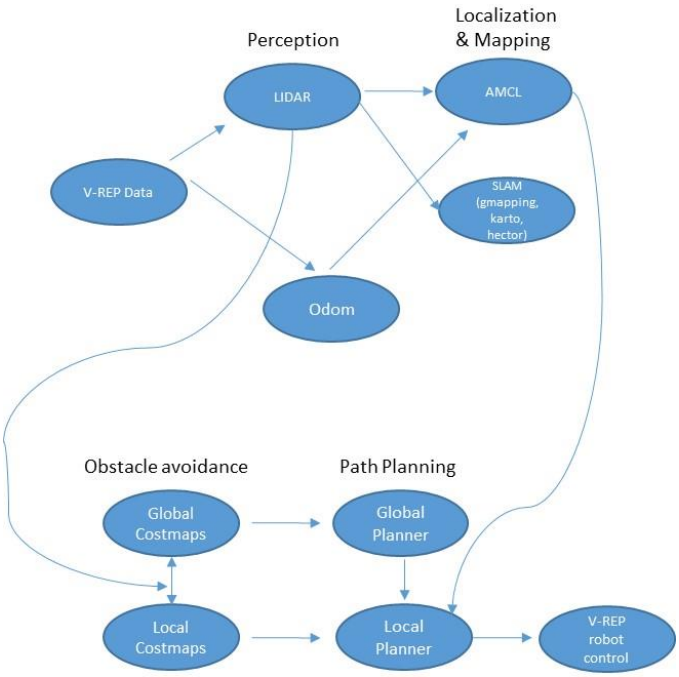


Figure 3: ROS framework used to autonomously navigate the vehicle within the PV power station.

Within ROS there are a number of packages that can enable SLAM mapping of the environment. In our simulation work, the best results are obtained using gMapping, which uses laser range data and odometry. gMapping utilizes Rao-Blackwellized particle filters to generate a map of the surrounding [3,4]. The particles carry information with regards to the position and orientation of the robot. Each new laser scan updates this information, which is then used to generate an occupancy grid. A general overview of the gMapping algorithm is discussed below [3-5]:

1) New laser scan data is received.

2) Laser scan data is matched with previous scan data.

3) Sampling particles $\{x_t^{(i)}\}$ is determined from the previous scan $\{x_{t-1}^{(i)}\}$ and from the following proposed distribution $\pi(x_t|z_{1:t}, u_{0:t})$, where z and u represent the laser and odometry data, respectively.

4) Particle weights are assigned based on the following formula:

$$w(i) = p(x_t^{(i)}|z_{1:t}, u_{0:t}) / \pi(x_t^{(i)}|z_{1:t}, u_{0:t}) \quad (1)$$

5) Particles the have a low importance weight are replaced with particles that have a higher weight. This step is important as we only have a finite number of particles.

6) Finally a map estimate is obtained from the data.

Figure 4 shows the SLAM map generated from the PV power station simulation, where the black lines represent occupied spaces. A fairly accurate map is obtained without the need of any external landmarks. Other techniques such as KartoSLAM [6], a graph-based approach, and HectorSLAM [6] were also investigated. However in both cases a high quality map was not generated and manual loop closure was necessary.

Figure 4: gMapping used to obtain a map of the PV power station.

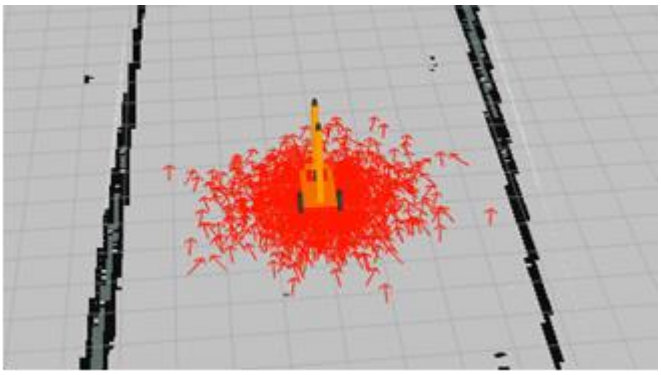
IV. LOCALIZATION

Once a map of the environment is generated, the robot must localize itself with respect to the map. Adaptive Monte-Carlo localization (AMCL) is used for this purpose. AMCL uses particle filters to estimate the robot pose, where KLD sampling is used to reduce the number of particles. The number of particles at each step is governed by the following equation [7]

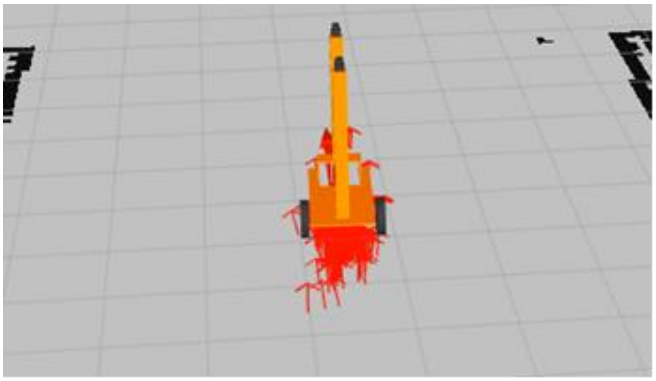
$$\eta = (\chi^2_{k-1,1-\delta}) / (2\varepsilon) \qquad (2)$$

where ε is the threshold that limits the number of particles based on the distance from the maximum likelihood estimate of the position and $\chi^2_{k-1,1-\delta}$, represents a chi-square distribution.

Fig. 5 shows the results of the AMCL performed on the vehicle, while it is driving through the PV power station. The red arrows represent the posterior of the robot based on the particles. Fig. 5a is the AMCL distribution shown at initialization, where a wider distribution is clearly visible, however as the robot navigates, AMCL accurately localizes the robot and a smaller distribution of poses based on the particle occurs, Fig 5b.



(a)

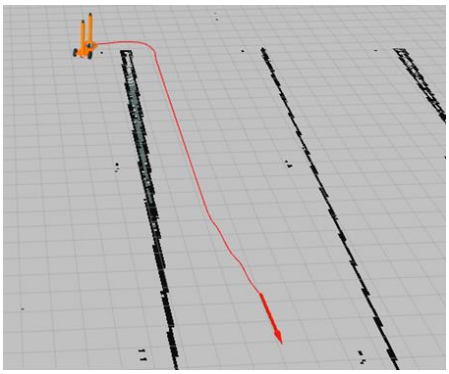


(b)

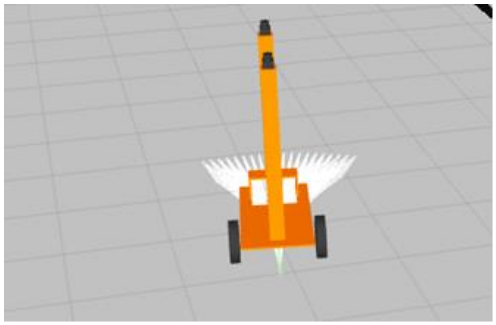
Figure 5: AMCL performed to localize the vehicle (a) AMCL performed at initialization, (b) after the robot drives within PV power station.

V. PATH PLANNING

Once a goal position is sent to the robot, two different planners are used for navigation. For the global path, Dijkstra algorithm is used and is shown in Fig. 6a. For the local path Dynamic Window Approach (DWA) [8] is used, which is responsible for sending velocity commands and ensuring no collisions occurs. Based on the kinematics of the robot, DWA determines various possible paths that the vehicle can take and choses an optimal path based on a number of parameters. These parameters include 1) following the global path, 2) distance from an obstacle, 3) progress towards goal, and 4) forward velocity of the robot. In Fig. 6b the white lines represent the different paths being analyzed by the DWA algorithm.



(a)

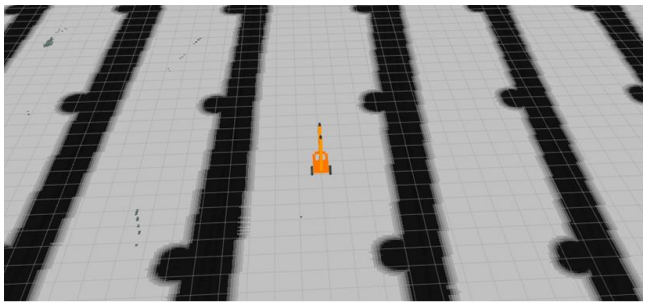


(b)

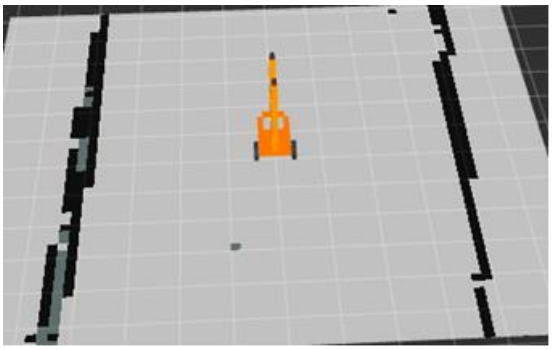
Figure 6: (a) Global path planner based on Dijkstra algorithm, (b) Local planner using DWA

VI. OBSTACLE AVIODANCE

Obstacles within the ROS navigation package are represented using costmap [9], which is an occupancy grid of the map. For a given costmap, each cell can have 255 different values to indicate if the cell is cell is occupied. Black cells indicate an obstacle is present. A global and local costmap are used for the global and local planner, respectively. Fig. 7a, is the global costmap of the entire environment, while in Fig. 7b, a much smaller costmap is used to represent the area immediately surrounding the vehicle. The local costmap in Fig. 7b, dynamically updates to capture the area around the robot. For the costmap an inflation radius can be set around an obstacle to ensure the robot maintains a safe distance from the obstacle.



(a)

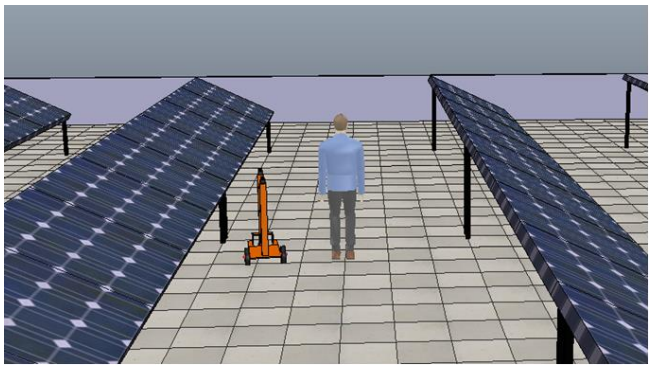


(b)

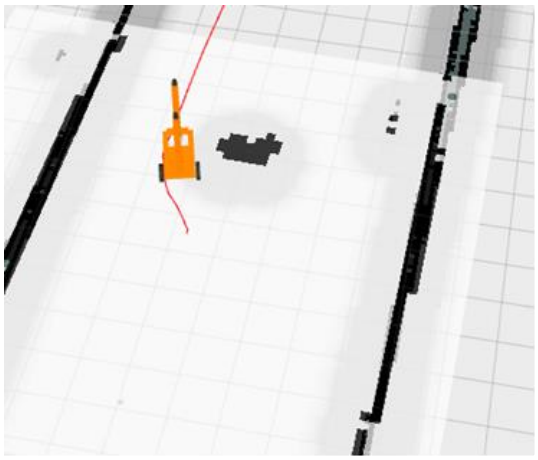
Figure 7: (a) Global costmap, (b) Local costmap

Various obstacles can be simulated and inserted into the PV power station using V-REP. An example of the obstacle avoidance approach used by the vehicle is shown in Fig 8. A person is inserted into the V-REP model, Fig. 8a, and the robot successfully avoids the obstacle and navigates around the person. Fig. 8b represents the perspective from ROS, where the dark region in the middle of the map represents the person. The laser scanner detects the person and updates the costmap with this information. Note a lighter shaded region is also visible around the person to ensure the local planner maintains a safe distance from the obstacle. In Fig. 8b, we also observe the global path, red line, successfully obtains a path that avoids the person.

One limitation of the current obstacle avoidance algorithm is that it does not track dynamic fast-moving obstacles, which would be a problem if we had numerous robots / vehicles around the PV power station. However, for the cleaning purposes we do not anticipate the use of many vehicles and additional controls can be built to avoid collisions between vehicles.



(a)



(b)

Figure 8: Obstacle avoidance of behavior of the robot (a) V-REP model with a person inserted, (b) ROS perspective showing the robot avoiding the obstacle.

VII. CONCLUSION

This work presents a general approach for deploying autonomous vehicles within a PV power station, which can be used for cleaning solar panels. V-REP is used to model the PV power station along with the vehicle and 2 laser scanners. Using the PV panels as a natural landmark the vehicle is able to successfully generate a map of the environment using gMapping. Localization is performed using AMCL. Additional ROS libraries are used for path planning and obstacle avoidance, where the vehicle is able to navigate around the PV power station. Future work will focus on attaching a robotic arm to the vehicle along with the actual simulation of removing dust particles from the panels.

REFERENCES

[1] Aïssa, B., Isaifan, R. J., Madhavan, V. E., & Abdallah, A. A. (2016). Structural and physical properties of the dust particles in Qatar and their influence on the PV panel performance. Scientific Reports, 6.

[2] Rohmer, E., Singh, S. P., & Freese, M. (2013, November). V-REP: A versatile and scalable robot simulation framework. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 1321-1326). IEEE.

[3] Grisettiyz, G., Stachniss, C., & Burgard, W. (2005, April). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation (pp. 2432-2437). IEEE.

[4] Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE transactions on Robotics, 23(1), 34-46.

[5] Uslu, E., Çakmak, F., Balcılar, M., Akıncı, A., Amasyalı, M. F., & Yavuz, S. (2015, September). Implementation of frontier-based exploration algorithm for an autonomous robot. In Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on (pp. 1-7). IEEE

[6] Santos, J. M., Portugal, D., & Rocha, R. P. (2013, October). An evaluation of 2D SLAM techniques available in robot operating system. In 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) (pp. 1-6). IEEE

[7] Rosa, S., Russo, L. O., & Bona, B. (2014, September). Towards a ROS-based autonomous cloud robotics platform for data center monitoring. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA) (pp. 1-8). IEEE.

[8] Zhang, J., Lyu, Y., Roppel, T., Patton, J., & Senthilkumar, C. P. (2016, March). Mobile robot for retail inventory using RFID. In 2016 IEEE International Conference on Industrial Technology (ICIT) (pp. 101-106). IEEE.

[9] Lu, D. V., Hershberger, D., & Smart, W. D. (2014, September). Layered costmaps for context-sensitive navigation. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 709-715). IEEE.