Autonomic Management for Mobile Robot Battery Degradation

Martin Doran, Roy Sterritt, George Wilkie

Faculty of Mathematics and Computing
University of Ulster
Jordanstown, N.Ireland
doran-M18@email.ulster.ac.uk, r.sterritt@ulster.ac.uk, fg.wilkie@ulster.ac.uk

Abstract—**the majority of today's mobile robots are very dependent on battery power. Mobile robots can operate untethered for a number of hours but eventually they will need to recharge their batteries in-order to continue to function. While computer processing and sensors have become cheaper and more powerful each year, battery development has progress very little. They are slow to re-charge, inefficient and lagging behind in the general progression of robotic development we see today. However, batteries are relatively cheap and when fully charged, can supply high power output necessary for operating heavy mobile robots. As there are no cheap alternatives to batteries, we need to find efficient ways to manage the power that batteries provide during their operational lifetime. This paper proposes the use of autonomic principles of self-adaption to address the behavioral changes a battery experiences as it gets older. In life, as we get older, we cannot perform tasks in the same way as we did in our youth; these tasks generally take longer to perform and require more of our energy to complete. Batteries also suffer from a form of degradation. As a battery gets older, it loses the ability to retain the same charge capacity it would have when brand new. This paper investigates how we can adapt the current state of a battery charge and cycle count, to the requirements of a mobile robot to perform its tasks.**

Keywords—autonomic, self-adaptive, self-optimizing, degradation

I. INTRODUCTION

Autonomic Computing has drawn many parallels with the complexities and functions of the human body. The term *Autonomic* is derived from the study of human biology. Is describes how the human autonomic nervous system constantly monitors and adjusts without the need of intervention from the individual on a conscious level. With the autonomic system, attributes such as *self-healing*, *self-optimizing* and *self-configuration* are identified. With *Self-Healing* - we can discover and diagnose problems quickly and offer a possible *policy change* to counteract the problem found. *Self-Optimizing* - is useful in making the best use of the resources that are available. There may be a need to shutdown part of a system in-order to re-direct resources to a crucial component. *Self-Configuration* - can dynamically adapt to changing environments were pre-loaded *policies* can be updated to deal with the necessary adjustments needed [1]. Self-Adaptive architecture can be employed which takes the Resource Manager and the Autonomic Manager and combines them to build a system that can be used to allocate resources using autonomic principles [2]. Using autonomic principles, we explore how a mobile robot can adapt itself to the effects of battery degradation. Lead-acid batteries which

contain a lead-calcium grid structure are vulnerable to an aging process, due to the fact they are repeatedly cycled [3]. We will employ a simulated battery configuration based on the actual lead-acid battery contained in our laboratory robot. The research will concentrate on how battery degradation can affect how a mobile robot performs basic tasks, like moving from one location to another. Task management is important, especially if the robot is operating on remote environments. We will investigate the importance of how the battery is maintained and how we can develop an intelligent *charging* strategy. We employ simulations to represent various surface types that the mobile robot may encounter and how battery degradation can influence the decision making on what speed and distances that are achievable. We look at historical data relating to the mobile robots previous tasks. We use this data to evaluate the current state of the simulated battery and what tasks are achievable for the mobile robot.

The structure of this paper is as follows. Section II documents the related work in evaluating battery degradation. Section III documents lead-acid battery characteristics of the battery employed in the simulated Pioneer 3-AT robot. Section IV documents the autonomic architecture. Section V looks at the Autonomic Manager and Resource model. Section VI documents the problem definition. Section VII documents experiments with different battery degradation scenarios. Section VIII documents Battery power Management and Section IX concludes the paper and outlines future work.

II. RELATED WORK

The purpose of the paper is not to investigate the characteristics of lead-acid batteries but to acknowledge that in general, batteries suffer from degradation and how we can adapt a mobile robot to compensate to for this degradation. To manage the *degradation* process, we use the Autonomic principles that were established in [1]. While there has been previous work carried out for mobile robots using self-adaption, self-healing, etc., known as (self-properties) [13][14], there has been little previous work carried out on how mobile robots can implement an Autonomic System to deal with handling the effects of battery degradation.

Various models for battery degradation have been investigated in [4], to determine the *degradation curve* for any lead-acid battery. Data provided from the *degradation curve* at least give the developer the ability to predict when battery degradation will affect the system operations.

With the vast improvements in computer processing and sensors, the need for an improved battery design and reduced degradation effects are in demand more than ever [15].

The Smart Battery System (SBS) [16] offers the ability to adjust the charging profile in response to actual requirements i.e. charging voltage and charging current. It can also monitor various charge states and raise *Alarms* if a damaged battery is detected. Although SBS is aware of charge states, it has no formal processes to deal with battery degradation.

Prognostic-enabled Decision Making (PDM) is a research area that aims to integrate prognostic health data and knowledge of future operations into the decision making when selecting a particular *action* within the System. Health checks include reporting Battery Capacity degradation faults and how various operations, including heavy load capacity and temperature variations [17]. PDM is employed as a fault reporting tool rather than a tool that can react to identifying *degradation* and making adjustments to deal with the discovered fault.

Dynamic Power Management (DPM) dynamically adjusts the power requirements for each component depending on the task being carried out [18]. This type of power management would be useful in making use of what battery power is available: especially if the mobile robot is in a *state* were battery degradation has been detected. If battery degradation is limiting the amount of charge the battery can hold, then any subsequent tasks the robot performs must be scrutinized, so any sensors not required by the task, are powered down.

The Wireless Power Transmission (WPT) [19] offers a method transmitting power wirelessly from one mobile robot to another. The alternative method is to use Coil Energy transfer, were the host robot plugs into the receiving robot [19]. These methods would be useful in remote situations, were battery degradation might affect the ability of the mobile robot returning to base and therefore connecting to the charging station.

III. PIONEER 3-AT ROBOT AND BATTERY PROPERTIES

MRDS (Microsoft Robotics Developer Studio) software framework was used to render a simulated Pioneer 3-AT robot. The MRDS commands used for a real Pioneer Robot can be adapted to run the robot simulation; this includes setting up basic battery parameters [5]. The Pioneer 3-AT uses the YUASA NP Series (NP7.5) battery. The NP7.5 data sheet provided all the necessary battery *capacity* and *charge* ratings [6]. The NP7.5 is rated at 7Ah or 7000mAh. The battery degradation mechanism is unavoidable in lead-acid batteries types. However, the rate of degradation can be managed depending on some known factors [3].

A. Battery Degradation Factors

- Loss of active material from positive plates.
- Loss of capacity due to the physical changes in the active material of positive plates.
- Temperature – elevated temperatures reduce battery life.
- Cycle service – discharge cycles reduce battery life.

For our experimentation, we disregarded temperature changes. We used the Cycle Service to simulate battery degradation, were the number of cycles is related to DOD (Depth of Discharge). DOD is used to describe how deeply the battery is dis-charged. For example: if we say the battery has delivered 30% of its energy, we say the DOD of the battery is 30%. A new battery can be expected to perform at its maximum capacity, were the number of charge cycles is low. If the DOD is kept under 50% then the battery cycle count will increase. Fig.1. shows the DOD characteristics for the battery used in the Pioneer 3-AT mobile robot [6].
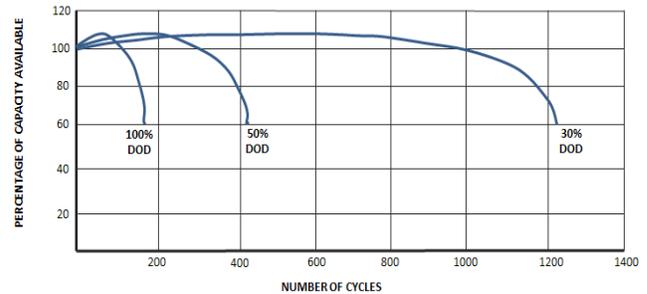

Figure 1. Shows the DOD charateristics for the lead-acid battery used in the Pioneer 3-AT

B. Intelligent Battery Charging

- A lead-acid battery should never be discharged below 80%, otherwise over time the battery will be damaged [10].
- To prevent sulfating and stratification, a lead-acid battery needs an Equalizing Charge. The battery is charged at a higher voltage and this is required every 10 charge cycles [10].
- The Pioneer Robot battery supply should be maintained above 11 VDC (Voltage Direct Current). If it falls below 10 VDC, the battery warning signals.

IV. AUTONOMIC ARCHITECTURE

The Autonomic Model provides a means of designing an architecture that will make use of the mobile robots resources, by evaluating *stored knowledge* of the robots previous tasks and relating this data with the current *state* of the battery power. The robot needs to be aware of the amount of resources that are available and how to make use of those resources [7]. The Autonomic Manager (see Fig.2.), is there to orchestrate the monitoring, analyzing, planning and executing of data from sensor inputs. The Autonomic Manager make policy changes based on analysis and will execute those policies accordingly [9]. The Autonomic Manager performs on a 'loop' basis, where it will periodically interrupt the System 'loop', and if required, will make adjustments to the system programming. The Autonomic Manager *feedback loop* is described in the MAPE-K model (see Fig.2.), where the (Knowledge) K, maintains the data from previous sensor (battery) readings and general data from completed tasks [8]. The Knowledge *store* is key in-order to evaluate the current state of the system. Analysis of stored data (knowledge), allows the Autonomic Manager access to how the mobile robot has performed on previous tasks. The Autonomic Manager needs to decide if the robot is still capable of completing those same tasks depending on the battery power

resource that is available. This is vital when it comes to making policy changes were some tasks may need to be abandoned or re-calibrated.
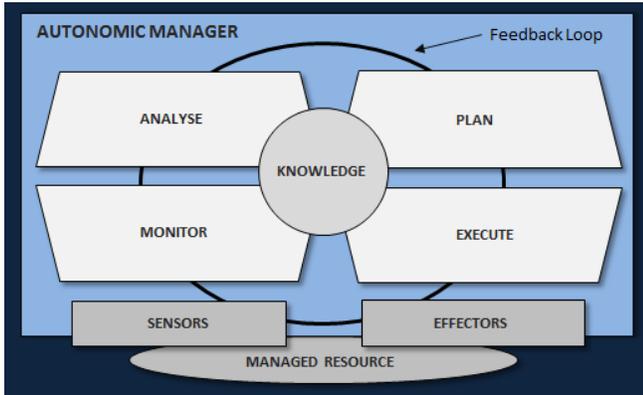

Figure 2.   The Autonomic Manager and MAPE-K feedback loop

V.   AUTONOMIC BATTERY MANAGEMENT

Using the autonomic concepts from Section IV, a model was developed to show how the Autonomic Manager is integrated with resources contained within the mobile robot (see Figure 3.). The Resource Manager orchestrates the inputted tasks from the User Interface and controls the necessary command library required to complete the tasks. The *command* data is then sent to the robot effectors. The robot sensors are used to feedback the data as the robot performs its tasks. The Resource Manager is also responsible for updating the *Knowledge* database with recent completed tasks and battery state. The Battery Monitor will update the current battery charge available and will record the battery Cycle count and DOD value. The Resource Manager will also output any information to the User regarding task monitoring, battery monitoring and policy changes that enforced by the Autonomic Manager.

The Autonomic Manager will periodically monitor the data supplied by the robot sensors and battery monitor. If performance data is within acceptable thresholds, the Autonomic Manager will not intervene. If the data is outside of expected thresholds, then the Autonomic Manager will run an analysis routine. This *routine* will look at historic data and compare this data with the current reported data. Example: a task has been received which requires the mobile robot to move from destination A to destination B. Using historical data, the Autonomic Manager is able to learn how much battery charge was used for a similar journey. Taking account of the current state of the battery charge, the Autonomic Manager will enforce a policy where the task is either cancelled or it may report what distance can be achieved with the available charge. Using the historical data, the Autonomic Manager can identify data *trends*. For example: if the robot is using increased battery power for the same journey, then it will flag to the User output that the battery is not retaining the same charge after each cycle. This could indicate that the battery is coming to the end of its life in terms of a useful power source. If a battery cannot retain a charge of more than 80%, then it needs to be replaced [3]. Alternatively, the Autonomic Manager could enforce a policy where the DOD rate is changed. Lowering the DOD rate (at the charging station) can extend the batteries life cycle count.
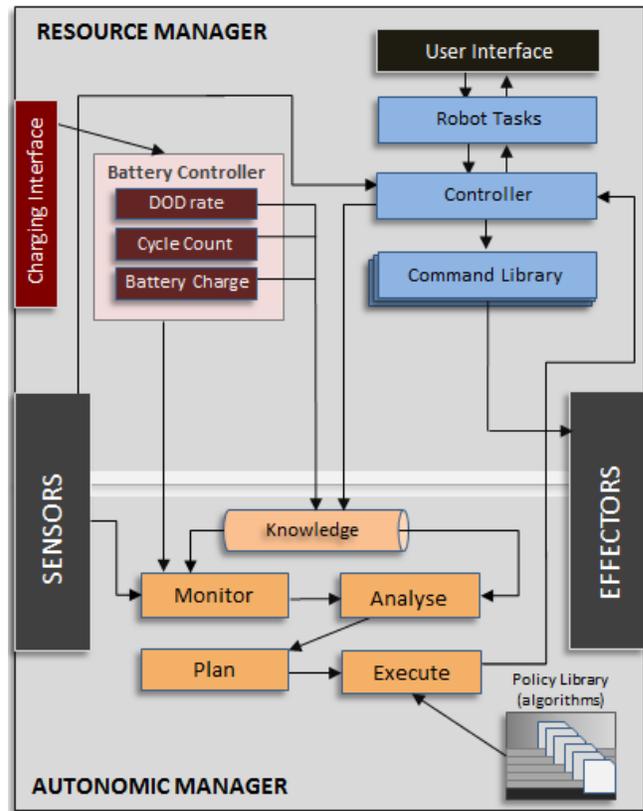

Figure 3.   Autonomic Battery Manager Model.

VI.   PROBLEM DEFINITION

In Section III we discussed the limitations of lead-acid batteries. Our main focus was on battery degradation. How does battery degradation affect the everyday operations of the mobile robot?

When we program a robot to travel from one destination to another (point A to point B), we can use the statistics at the end of the journey to work-out how much battery-charge was required to complete the journey. If the battery in the mobile robot is relatively new, then we can be confident that the statistics for a similar journey will remain constant. However, there are two factors which can impact how the statistics will read as further journeys are made and the battery gets older, DOD charge and Cycle count. In Figure 1, the DOD rate affects the number of cycles the battery can sustain before it becomes unusable. If the percentage capacity falls below 80%, then the battery is discarded [3]. If the DOD rate is set relatively high, i.e. 50%, then after 300 charge cycles, the battery will begin to degrade rapidly. However, the battery is still viable between 300 and 400 charge cycles and will still be used by the mobile robot. However, problems occur because the journey from (A to B), that was taken in the early life of the battery, will now require a greater charge capacity because the battery has begun to degrade.

VII. SIMULATING BATTERY PERFORMANCE

For this experiment, a simulated battery and simulated Pioneer 3-AT robot is used. The simulated Pioneer 3-AT robot is achieved using MRDS (Microsoft Robotics Developer Studio). The Robot Commands used to drive the simulated robot can also be implemented for a physical robot [5]. SPL Editor is used to supply the graphics environment to drive the simulated pioneer robot [11].

Setup Task: the mobile robot travels from point A to point B, which is measured at 200 meters. Table 1 shows the DOD rates at 100%, 50% and 30%. The cycle number represents the number of times the battery has been re-charged.

TABLE I.   DOD CHARGE RATES FOR PIONEER 3-AT BATTERY

#	Cycle Number	DOD Charge @ 100%	DOD Charge @ 50%	DOD Charge @ 30%
1	0	100	100	100
2	20	104	101	101
3	40	108	103	101
4	80	110	104	102
5	100	101	107	103
6	120	95	107	103
7	140	91	108	103
8	160	80	108	104
9	180	60	109	105
10	200	-	110	106
11	240	-	105	106
12	280	-	102	107
13	320	-	98	107
14	340	-	95	108
15	360	-	91	108
16	380	-	86	109
17	400	-	80	110
18	500	-	-	110
19	600	-	-	110
20	700	-	-	108
21	800	-	-	106
22	900	-	-	102
23	1000	-	-	98
24	1050	-	-	95
25	1100	-	-	90
26	1150	-	-	80
27	1200	-	-	60

The following parameters are defined to create a basic battery simulation involving the DOD values (see Figure 1).

DR – depth of charge rate (this is the percentage of discharge used before the battery is cycled).
DOD – depth of discharge (this is the percentage available in the battery to discharge).

DP – discharge percentage (this is the amount of discharge available at a particular cycle count).
BP – battery percentage (this is the amount of battery charge required to complete a task).
SU – single unit (this unit is calibrated 1% of battery power. This will move the robot a distance of 20 meters).
DISTU – distance unit (this is the number of SU's required given a distance value).

Equation (1) is used to calculate the DISTU value.

$$DISTU = \frac{Distance}{SU} \qquad (1)$$

Using the Cycle Number value from Table I, the DP can be calculated using Equation (2)

$$DP = \frac{DOD * DR}{100} \qquad (2)$$

The percentage of charge required to complete a task is calculated using Equation (3). This percentage is important, as the mobile robot system manager needs to establish if the there is enough charge left in the battery to complete the task.

$$BP = \frac{100 * DISTU}{DP} \qquad (3)$$

Figure 4 shows the percentage charge required to complete the task when the battery is at various stages within its cycle.
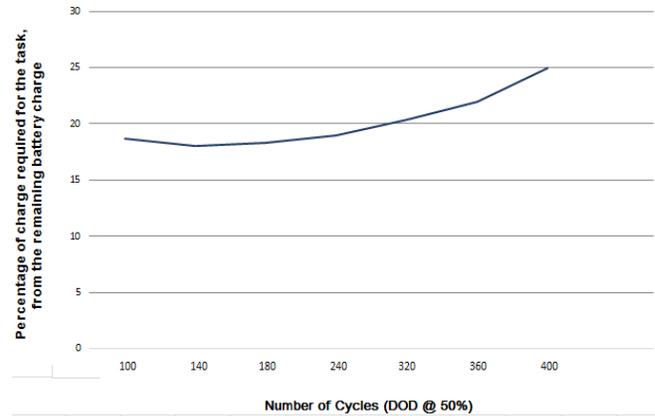

Figure 4.   Shows the percentage charge required to completed as task at various battery cycle stages (DOD set at 50%).

In this example the DOD rate is set at 50%. The robot task distance = 200 meters. In the early stages of battery *life*, the battery charge requirements for the task remain constant. However, after 240 charge cycles, the percentage of charge increases as the battery degradation increases. This would have an impact on tasks that required the robot to travel long distances.

VIII. BATTERY POWER MANAGEMENT

Within the Pioneer 3-AT robot, each component requires a certain level of power-input in order to function. The lead

acid batteries within the 3-AT robot supply the necessary power-input required. Research in [20], shows the relative amount of power required for each of the components. See TABLE II.

TABLE II. COMPONENT POWER/PERCENTAGES LEVELS FOR THE PIONEER 3-AT ROBOT

Component	Power	Percentage
Motion	2.8W ~ 10.6W	12% ~ 44.6%
Sensing (sonar)	0.58W ~ 0.82W	1.9% ~ 5.1%
Microcontroller	4.6W	14.8% ~ 28.8%
Embedded Computer	8W ~ 15W	33.3% ~ 65.3%

For our next experiment, we investigated how battery degradation would affect how much power (W) would be available to the 'Motion' component - see (Table II). Figure 5 [20], shows the amount of power (W) is required in the 'Motion' component when driving the robot at various speeds.
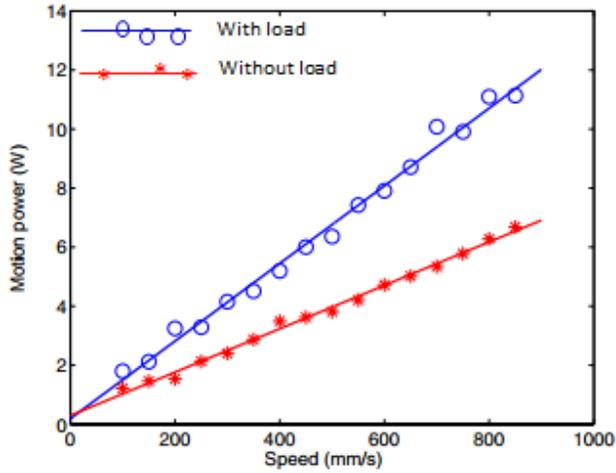

Figure 5. Shows the Power (W) required to use the 'Motion' component when driving the robot at various speeds and different 'loads'[20].

For our next experiment, we investigated how battery degradation would affect how much power (W) would be available to the 'Motion' component during the life-time of our simulated battery (see Section VII). The Autonomic Manager analyses the data from the simulated battery contained within the Resource Manager (See Figure 3.). If analyzed results show that the robot tasks being performed (at a certain battery cycle) require power (W) above the threshold limits of the battery, then Autonomic Manager will make necessary adjustments to the power level available to the 'Motion' component.

A. Initial task set-up

TABLE III. ROBOT TASK 1: SETUP VALUES FOR MOTION COMPONENT

Parameters	Values
RS Robot Speed (mm/s)	600
PR Power required (W)	8W
DIS Distance to travel (M)	5,000

T Time (hr.)	2.31
WHU Watt Hours used	18.51
BC Battery Cycle	0

The pioneer 3-AT robot battery provides 84 watt-hours of power capacity [12], when the battery is at 'cycle' 0 (see Table 1). The normal battery capacity is 100% and the therefore we can say the rating is 7000mAh offering 12 volts (see Section III). We can then use the following equation to calculate the *watt-hour* value for our battery. E = Energy, Q = milliamp hours and V = Voltage.

$$E(wh) = Q(mAh) \; x \; V(v)/1000 \quad (4)$$

Using the values in TABLE III, we can establish the value of the 'Motion' component in terms of watt-hours used.

$$WHU = PR \; x \; T \quad (5)$$

To extend the life of the battery, we employ a DOD rate policy. So for example, if the rate is set to 30%, then we only let the battery charge capacity fall to 70% (see Section VI). The battery at 100% charge gives 84 (watt hour) energy **E**. If we use the 30% DOD policy then we have (**WA**) 25.3wh available for the robot a cycle 0.

Using the **WHU** value from equation (5), we can calculate the percentage of capacity **PC** required for Robot Task 1 – see Table III.

$$PC = \left(\frac{WHU}{WA}\right) * 100 \quad (6)$$

For this experiment, the acceptable threshold value for how much a task uses in its completion is 80%. In the case of Robot Task1, the battery capacity percentage required was 73.16% (see equation 6).

B. Executing a task with battery degradation

Towards the end of a lead-acid battery lifetime, the amount of energy available is reduced. If we run a new task 'Task 2', with the same parameter values in 'Task 1', see Table III, we have to take into account that the (ah) rating is now reduced due to battery degradation. If we run the task at *cycle* 1100, the capacity has fallen from 100% to 90%; we therefore need to re-calculate the 'E' value using equation (4). At cycle 1100, we have now only 6300mAh and therefore our power capacity has been reduced to 75.6 (watt hour). Using the 30% DOD policy, we now have 22.68wh (**WA**) available for the task. If we apply equation (6), then Task 2 we require 81.61% of battery capacity which is above the required threshold of 80%.

C. Making Adjustments to compensate for dedgration

If a task exceeds the battery capacity available, then the Autonomic Manager responds by employing a *compensation policy*. Compensation is achieved by reducing the speed of the

robot and therefore reduced the energy required from the battery. Table IV shows 'Task 3', with a reduced **RS** value.

TABLE IV.   ROBOT TASK 3: VALUES FOR COMPENSATION POLICY

Parameters	Values
RS Robot Speed (mm/s)	500
PR Power required (W)	6.5W
DIS Distance to travel (M)	5,000
T Time (hr.)	2.77
WHU Watt Hours used	18.00
BC Battery Cycle	1100

If we apply equation (5) to using the values in Table IV, then Task 3 we require 79.37% of the battery capacity which is below the 80% threshold. By reducing the speed of the robot, we are able to complete the task; however, using this *compensation policy*, we have increased the time it will take for the task to complete. If we compare Task 1 (**T**) to Task 3 (**T**), then we have increased the journey time by 16.61%.

D.   Task Input and Analysis Algorithm

Table V shows the algorithm for inputting the 'task' values and then analyzing the battery capacity required to complete the task.

TABLE V.   (ALGORITHM 1) – TASK INPUT AND ANALYSIS

```
1:      //Get the DOD charge rating
2:      DOD = selectedRatingValueInput
3:      //Get the current Battery Cycle Value
4:      batteryCycleCount = GetCurrentBatteryCycleCount()
5:      //Get the DOD cycle charge percentage value
6:      batteryCycleValue = GetCycleValue(BatteryCycleCount, DOD)
7:      //Get the upper cycle count value for DOD rating
8:      upperCycleValue = GetUpperCycleValue(DOD)
9:      //Check that the battery has not reached its cycle end-of-life
10:     if ( batteryCycleCount >upperCycleValue) then
11:       battery_expired = true
12:      else
13:       battery_expired = false
14:     end if
15:     // If battery has expired, then output message to User
16:     if (!battery_expired) then
17:      // Enter task parameters
18:      RS = RobotSpeedInput()//set the motor power value
19:      PR = PowerRequiredInput(watts)
20:      DIS = DistanceToTravelInput()
21:      // Calculate travel time for task
22:      T = DIS / (RS/1000)
23:      // Calculate watt hours used for the task
24:      WHU  = T * PR
25:      // Energy available from battery @ cycle count
26:      Q(mAh) = 7ah * batteryCycleValue
27:      E = Q(mAh) * VoltageInput/1000
28:      // Using DOD rate, calculate the working capacity
29:      WA = DOD * E/100
30:      // Calculate the percentage capacity required by the robot task
31:      PC = WHU/WA*100
32:      // Check that the percentage of battery capacity for the robot
33:      // does not exceed the Threshold value
34:      if (PC >80%) then
35:        thresholdExceeded = true
36:       end if
37:     end if
```

```
38:      // If the Threshold value is exceed, then engaged the
39:      // Adjustment algorithm
```

E.   Compensation algorithm for battery degradation

If the battery capacity required exceeds the 'threshold value' i.e. 80%, then we engage the *compensation* algorithm (see Table VI); reducing the power to the motor will reduce the robot speed and therefore require less battery capacity.

TABLE VI.   (ALGORITHM 2) – COMPENSATION ALGORITHM

```
1:      //Adjust the speed of the robot to compensate for the
2:      //robot exceeding the Threshold value
3:      if (thresholdExceeded) then
4:       motor_power_scale_factor = robotSpeedInput
5:       //set the motor power value
6:       Drive.SetDrivePowerRequest request = new
         drive.SetDrivePowerRequest()
7:       Request.LeftWheelPower = (double)OnMove.Left *
         motor_power_scale_factor
8:       RequestRightWheelPower = (double)OnMoveRight*
         motor_power_scale_factor
9:      end if
10:     //Run the task again with the new speed adjustment
11:     //If the Threshold value is still exceeded, then adjust the
12:     //robot motor speed further until PC value is under 80%
```

IX.   CONCLUSION AND FUTURE WORK

The purpose of this research paper is to identify how battery degradation can affect the performance of a mobile robot. When battery degradation is detected, the battery loses the ability to hold the same percentage of charge as it was manufactured for. Battery degradation is an unavoidable process and therefore we have to adapt our policies to handle this type of disability.

In this paper we employed autonomic principles to manage how the mobile robot can adapt to changes in available battery capacity. Using a simulated battery, we showed how using different DOD values, we could increase the lifetime of the battery and therefore prolong the degradation effects. We also investigated how adjusting robot functions like 'motor speed', can reduce the demands on the battery capacity required,

Using the simulated battery, we also discovered how degradation affects mobile robot tasks. When the charge capacity available to the robot is reduced, the greater the percentage 'charge' is required to complete tasks.

In future work, we would like to expand the battery simulation to take account for temperature differences; battery degradation is affected by variations in changing temperature levels.

REFERENCES

[1] D. M. Chess, A. Segal, I. Whalley, and S. R. White, "An architectural blueprint for autonomic computing," IBM Corporation, 2004.

[2] S. Jha, M. Parashar, O. Rana, "Self-Adaptive Architectures for Autonomic Computational Science" in Self-Organizing Architectures, Springer, pp. 177-197, 2010.

[3] PowerThruu: Lead Acid Battery working – lifetime study. [Online]. Available from: http://www.power-thru.com

[4] D. C. C. Freitas, M. B. Ketzer, M. R. A. Morais and A. M. N. Lima, "Lifetime estimation technique for lead-acid batteries," *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, 2016, pp. 2076-2081.

[5] Microsoft. Microsoft Robotics Developer Studio. [Online]. Available from: http://www.microsoft.com/robotics/2017/02/04

[6] Yuasa Batteries − NP serires NP7.5-12 data sheet. [Online]. Available from: http://www.yuasabatteries.com

[7] A. Hernando, R. Sanz, R. Calinescu, "A Model-Based Approach to the Autonomic Management of Mobile Robot Resources", in proc. Int. Conf. on Adaptive and Self-Adaptive Systems and Applications, Lisbon, Portugal, 2010.

[8] P. Arcaini, E. Riccobene and P. Scandurra, "Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation," *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, Florence, 2015, pp. 13-23.

[9] J. O. Kephart and D. M. Chess. The vision of autonomic computing. IEEE Computer, 36(1):41–50, 2003.

[10] Progreesive Dynamcis Inc. [Online]. Available from: http://www.progressivedyn.com/battery_basics.html/2017/05/20

[11] Stem Education (Microsoft Developer Studio). [Online]. Available from: http://www.helloapps.com/index.html/ [Accessed 23 May 2017]

[12] Adept Mobile Robots. Pioneer 3 Operations Manual, Version 6, 2010.

[13] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," Computer, vol. 36, no. 1, pp. 41–50, 2003.

[14] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," IBM Syst. J., vol. 42, no. 1, pp. 5–18, 2003.

[15] American Scientific: Could Battery Advances Mean Better Robots?. [Online]. Available from: https://www.scientificamerican.com/article/robot-battery-technology-life-spa/ [Accessed 2 June 2017]

[16] Lucas, Nestor, Cosmin Codrea, Thomas Hirth, Javier Gutierrez and Falko Dressler. "RoBM2: Measurement of Battery Capacity in Mobile Robot Systems." (2005).

[17] Sweet, A., Gorospe, G., Daigle, M., Celaya, J. R., Bal-aban, E., Roychoudhury, I., and Narasimhan, S.,"Demonstration of prognostics-enabled decision making algorithms on a hardware mobile robot test platform". In Annual Conf. of the Prognostics and HealthManagement Society, pp. 142–150, 2014

[18] Deshmukh, A., Vargas, P.A., Aylett, R., Brown, K.:Towards Socially Constrained Power Management for Long-Term Operation of Mobile Robots. In Towards Autono-mous Robotic Systems, (2010).

[19] Jae-O Kim and Chanwoo Moon, "A Vision-Based Wireless Charging System for Robot Trophallaxis", International Journal of Advanced Robotic Systems, Vol 12, Issue 12, 2015

[20] Y. Mei, Y. Lu, Y. C. Hu and C. S. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques", ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005., Seattle, WA, 2005, pp. 492-49