UAV Based Wireless Charging of Sensor Networks Without Prior Knowledge

Najeeb W. Najeeb[1] and Carrick Detweiler[1]

***Abstract*— Unmanned Aerial Vehicles (UAVs) can charge Wireless Rechargeable Sensor Networks (WRSNs) in remote or hard to access locations. However, the charging efficiency is heavily affected by the distance between the wireless transmitter and receiver. This efficiency impacts the possible power level increase of each charged node. Most charging algorithms require full knowledge of sensor nodes' power levels to identify the nodes to charge. Collecting this power information adds overhead to the network and limits scalability. We propose and implement Charging with Power Transfer Efficiency Compensation (CPTEC), an algorithm that charges a WRSN without the need for *a priori* knowledge of the nodes' power levels. We show that CPTEC compensates for efficiency drops, due to landing alignments, making it practical for real-world power transfer scenarios. Our results show that CPTEC is able to perform with a median at $\approx 72\%$ of the optimal performance of a full knowledge algorithm that assumes maximum power transfer efficiency, while other work drops to $\approx 22\%$. Under constant maximum efficiency CPTEC performs $\approx 90\%$ of the optimal full knowledge case.**

I. Introduction

An Unmanned Aerial Vehicle (UAV) can be used to safely and efficiently charge a Wireless Rechargeable Sensor Networks (WRSN) in remote or dangerous environments using wireless power transfer technology [1]. Most charging algorithms require the full knowledge of sensor nodes' power levels prior to execution. As the size of WRSN increases, it becomes increasingly difficult to collect power information from all the nodes. Another practical limitation of charging using wireless power transfer is power transfer efficiency. This efficiency depends on the alignment and distance between the transmitter and receiver.

In this work, we propose and experimentally evaluate Charging with Power Transfer Efficiency Compensation (CPTEC) a charging algorithm, which overcomes unpredictable efficiency changes when charging a WRSN and operates with no *a priori* power knowledge of the network. CPTEC utilizes probability and probabilistic bounds to eliminate the need for *a priori* knowledge. We implement this algorithm on a UAV, with a wireless power transfer system, and examine both simulation and field results. In this work, we assume the charging path is already planned similar to our previous work [2]. In a wireless power transfer system, using resonant coupling, energy is transferred from a source coil to a receiver coil over the non-conductive medium [3].

*This work was partially supported by the United States Department of Agriculture - National Institute of Food and Agriculture USDA-NIFA 2017-67021-25924, and the National Science Foundation (NSF) 1539070.

[1]NIMBUS Lab, Dept. of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA. {najeeb, carrick}@cse.unl.edu

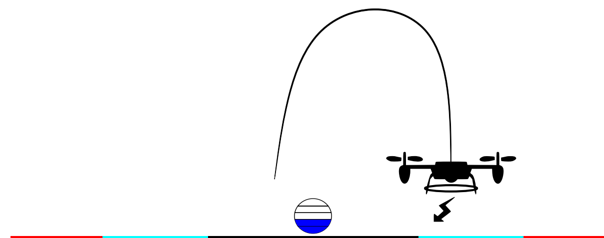

Fig. 1. A UAV landing near a node leading to different actions. Stay Area where it will charge, Retry Area where it will attempt a second landing; Leave Area when it is better to ignore this node and travel to the next.

Typically the coils must be aligned along their axis for best power transfer efficiency. We land a UAV on the sensor nodes and compensate for power transfer efficiency drops due to misalignment. The use of a UAV helps extend the reach and speed of charging while minimizing the WRSN overhead.

A key feature of our algorithm is that it dynamically adapts based on the measured power transfer efficiency in the field. Figure 1 illustrates the impact of landing location on charging efficiency. Assume a UAV is attempting to charge several sensor nodes. The UAV lands near each node and attempts to transfer power. In addition to efficiency, the algorithm must take into account the available power on-board and the sensor node's power level. When the UAV lands, it can initiate a short power transfer to determine the true power transfer efficiency. Based on this it can decide to stay and charge; try another landing to get a higher efficiency; or take-off and not charge that node. We refer to the first case area as the Stay Area, and it is best if the UAV stays where it is and engages charging. The second case is the Retry Area, where the cost of a re-land scenario is less than the cost of low power transfer efficiency at the original landing location. In this case, a reattempt is made in hopes to achieve a better efficiency. The third case is the Leave Area, and it is best if the UAV ignores this node.

The main contributions of this paper are:

- Proposing Charging with Power Transfer Efficiency Compensation (CPTEC), an algorithm with no *a priori* knowledge requirement to extend the life of a WRSN.
- Results showing CPTEC achieves 72% of the theoretical maximum efficiency algorithm with *full* knowledge, and also outperforms other approaches.
- Simulation and experiments validating the implementation of CPTEC with a UAV configured to autonomously perform recharging.

II. Related Work

Node life is one category for determining the life of a WRSN [4]. Node life can be defined as the time until the first sensor runs out of energy [5]; the time when the first cluster head runs out of energy [6]; the time all the sensor nodes run out of energy [7]; the time a percentage of the network runs out of energy [8]. In our work we consider the life of a WRSN as the time until a percentage of the network runs out of energy [9].

Several works introduce mobility and robots to WRSN. While adding cheap mobile sensors was shown to sustain a WRSN [10], the solution did not address power transfer efficiency. Robots are also used for data muling in WRSN and the behaviour is very similar to charging. Data muling can be done both on ground [11] and underwater, also using passive visual techniques for localization [12]. Wireless power transfer can be performed using a ground vehicle as well [13]. When using ground vehicles, the robot travel time cannot be ignored in comparison to the sensor nodes' discharge rates.

As WRSN scale in size, the cost associated with collecting energy information becomes impractical. An efficient energy monitoring protocol was designed to address this problem [14]. However, it requires the use of several mobile charging robot (MCR) units and each requiring full knowledge of the nodes' power and the communication protocols. Full power level knowledge is also required when solving the charging problem as an optimization problem [15], where the solution is to maximise the ratio between the MCR vacation time over cycle time. Even though sensor nodes are low in energy demand, a big portion of their energy budget is consumed by the transceivers [16], a substantial amount of power can be saved by reducing information relay. We address scalability by eliminating the need for *a priori* energy information.

A partial knowledge adaptive approach to increase the life of a WRSN exists [17], where representatives of nodes are used to gain partial knowledge. An MCR adapts its trajectories based on the energy dissipation rate of the representatives. The solution assumes that the representatives reflect the behaviour of nodes in their locality. While this approach requires less information, it still requires *a priori* knowledge of the representatives' power levels.

Two zero knowledge algorithms: centralized charging (CC) and distributed charging (DC), which utilize several MCR units exist [18]. In both CC and DC, the MCR exhaustively visits all the nodes in its designated or negotiated region respectively. The authors reported that both algorithms had low performance when compared to full or partial knowledge counterparts. This we think was due to the exhaustive-search nature of the no-knowledge algorithms. Our approach does not require full network exploration, rendering it more competitive to full knowledge algorithms.

In terms of strongly coupled magnetic resonances, early work [3] demonstrated approximately 40% efficiency. Afterwards, a near constant 70% was demonstrated [19] using an adaptive auto-tuning technique. Inductive power transfer using Class-E amplifiers can provide efficiency up to 95% [20]. While efficiency was increased it always depends on an optimal alignment between the transmitter and receiver coils. This assumption cannot be guaranteed in real-world applications, where precise landings with small UAVs is challenging due to environmental factors such as wind.

III. Problem Formulation

In this section, we provide the formal definition of the no knowledge efficiency compensation wireless power transfer problem. Given a wireless rechargeable sensor network (WRSN) consisting of s sensor nodes $\{n_1, n_2, \ldots, n_s\}$ and a base station, B_S. Without loss of generality, assume all nodes have the same battery capacity, E_S. Each node, n_i, is independent from the others and possesses an energy level e_i, forming a set of energy values N_e,

$$N_e = \{e_1, e_2, \ldots, e_s\}. \tag{1}$$

We define the life of the WRSN as the time when a percentage of the sensor nodes reach zero residual energy. Given a WRSN with fixed size s, the network life is then defined as the time when k nodes reach zero residual energy.

Given N_e there exists a set of nodes Υ, when charged to level L will lead to the maximum WRSN life extension Ψ [2]. Without knowing N_e, charging a set of nodes υ to level l leads to a life extension ψ, $\psi \leq \Psi$. Υ is the optimal set of nodes to charge, while υ is a subset of nodes to charge.

We use a single UAV to charge the WRSN. The UAV uses one battery for both navigation and charging. We assume a single UAV flight to charge the WRSN is shorter than the time between two consecutive discharges of a sensor node. When charging a sensor node, the UAV attempts to align its transmission coil centre to the node's receiver coil centre; we refer to this process as concentric localization. We define the power consumed to perform concentric localization as λ.

While the UAV is charging a sensor node let the power consumed by the UAV be Λ_c. Not all Λ_c reaches the sensor node, due to misalignment and interference. We define η as the power transfer efficiency, that is the percentage of power received by the sensor node while being charged by the UAV. The node received power Λ_n will be

$$\Lambda_n = \eta \Lambda_c. \tag{2}$$

We define the distance between the UAV and the sensor node n after concentric localization as d_λ. The power transfer efficiency at node n, η_n, is a function of the distance d_λ, we define $\varPhi$ as the power efficiency function,

$$\eta_n = \varPhi(d_\lambda). \tag{3}$$

Each node will have a different d_λ causing a different η. Define η_{n_i} as the efficiency charging node n_i given its distance d_{λ_i}, η_υ is the set of η values for each node in υ,

$$\eta_\upsilon = \{\eta_{n_1}, \eta_{n_2}, \ldots \eta_{n_m}\}, m = |\upsilon|. \tag{4}$$

There exists an expected power transfer efficiency $E(\eta)$, based on an expected concentric localization distance $E(d_\lambda)$.

We define C' as the total expected power needed by the UAV to charge the nodes in υ, using expected efficiency,

$$C' = u \times \lambda + \sum_{i=1}^{u} \frac{(l - e_i)}{E(\eta)}, u = |\upsilon|, \tag{5}$$

given actual efficiencies η_υ, the actual power needed, C, is

$$C = u \times \lambda + \sum_{i=1}^{u} \frac{(l - e_i)}{\eta_{n_i}}, \tag{6}$$

We study the case where there is no knowledge of N_e and we want to maximise ψ, using actual efficiency values, η_υ, and $E(\eta)$. The objectives become:

- Finding the charging list υ and target power level l to bring network life increase ψ close to optimal life increase Ψ.
- Maximize ψ to bring it closer to Ψ, after knowing the exact power transfer efficiencies η_υ and updating the charging decisions.

IV. Charging with Power Transfer Efficiency Compensation (CPTEC)

In this section we present our algorithm, Charging with Power Transfer Efficiency Compensation (CPTEC). The algorithm assumes no *a priori* knowledge of sensors' power levels N_e (Equation 1). Figure 2 shows a high-level description of the algorithm flow. The algorithm explores the WRSN by visiting nodes sequentially, and a sub-algorithm, ANLPP, determines the end of exploration. Once exploration is complete, a return and re-visit phase starts. During this phase, the algorithm invokes an sub-algorithm to charge sensor nodes (CHARGE). The charging algorithm determines if visited nodes should be charged, and in turn utilizes another algorithm to compensate for efficiency drops, DETERMINENEWTARGET. Next, we describe in details CPTEC and each sub-algorithm.

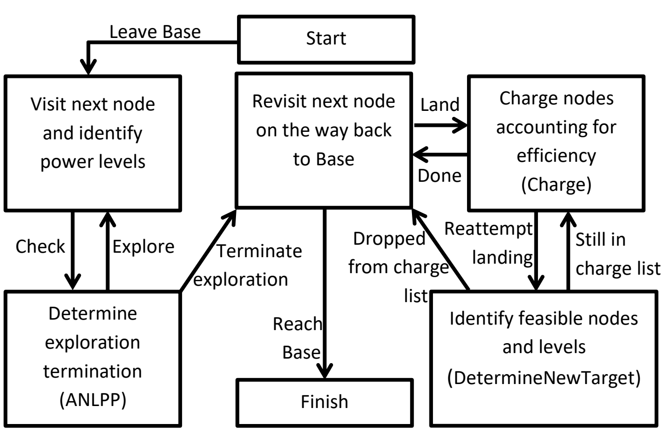

Fig. 2. General overview of CPTEC Mission.

A. CPTEC Algorithm Flow

Our objective is to increase the overall network life, without knowing N_e, and taking into account the impact of power transfer efficiency η. We start with an overall description of the mission to achieve this objective. There are three phases to the mission: 1) Identifying the target nodes; 2) Landing on the target nodes and identifying power transfer efficiency; 3) Determining if charging the nodes is feasible.

Algorithm 1 CPTEC

Require: N ▷ List of node in WRSN
Require: MCR ▷ The traversing UAV
1: **procedure** CPTEC
2: $vNodes \leftarrow empty$ ▷ List of visited nodes
3: $node \leftarrow empty$ ▷ Current node closest to UAV
4: $\upsilon \leftarrow empty$ ▷ List of nodes to charge
5: $l \leftarrow 0$ ▷ Target power level to charge the nodes in υ
6: take-off UAV from B_S
7: $exploreFlag \leftarrow True$ ▷ Flag indicating exploration status
8: **while** $exploreFlag$ **do**
9: visit $node$, the next node in N
10: add $node$ to $vNodes$
11: $power \leftarrow$ Get MCR Power Level
12: $hp \leftarrow$ calcualte power needed to return home
13: $C \leftarrow power - hp$ ▷ Available power for charging
14: $exploreFlag, \upsilon, l \leftarrow$ANLPP($vNodes, C$)
15: **end while**
16: **while** not at B_S **do** ▷ On the way back charge node in υ
17: visit $node$, the next node in N on the way back
18: **if** $node$ in υ **then** ▷ Current node is in charging list
19: $power \leftarrow$ Get MCR Power Level
20: $hp \leftarrow$ calcualte power needed to return home
21: $C \leftarrow power - hp$ ▷ Available power for charging
22: $\upsilon, l \leftarrow$ CHARGE($C, node, \upsilon, l$)
23: **end if**
24: **end while**
25: **end procedure**

Algorithm 1 shows the pseudo code to achieve the objective. The algorithm keeps track of the visited nodes $vNodes$. When the UAV takes off from B_S, $vNodes$ is empty (line 2). The algorithm instructs the UAV to visit the next node and add it to $vNodes$ (line 9). Next, the algorithm reserves power to return back to B_S and the power left is what can be used for charging, C, (lines 11-13). ANLPP, Algorithm 2, then determines if exploration should continue, if so the process is repeated (line 14). Otherwise, the algorithm will instruct the UAV to return to B_S and commence charging while returning. ANLPP also returns the list of nodes to charge υ (line 14). When the UAV is above any node in υ it will invoke CHARGE, Algorithm 3, to charge the node. Available power for charging, C, is computed again and passed to the CHARGE (lines 19-22). C is used for both localizing the UAV over the node and charging the node. CHARGE, Algorithm 3, identifies the feasibility of charging a node and performs the charging. The mission ends once the UAV is back at B_S. Next, we describe ANLPP and CHARGE.

B. All Network Least Possible Probability (ANLPP)

In this subsection, we describe ANLPP, an algorithm that determines the furthest beneficial exploration point of a WRSN in the absence of knowing N_e. It utilizes probability and a probabilistic bound derived from Chernoff bounds. Our previous work shows that ANLPP extends the life of a WRSN on average to 90% of what a full knowledge algorithm can achieve. The detailed results and derived equations can be found in our previous work [2]. However,

Algorithm 2 ANLPP

Require: $vNodes$ ▷ List of visited nodes
Require: C ▷ UAV available power
Require: λ ▷ Localization cost in terms of power
Require: $minTh$ ▷ Low probability threshold for the low power nodes
Require: $maxTh$ ▷ High probability threshold for the low power nodes
1: **procedure** ANLPP($vNodes, C$)
2: $exploreFlag \leftarrow True$ ▷ Flag indicating exploration status
3: $v, C', l \leftarrow$ identify nodes to charge and power needed
4: $powerLeft \leftarrow C - C'$
5: **if** $powerLeft <$ power needed to reach next ndoe **then**
6: $powerToChargeOneNode \leftarrow \lambda + l - min(v)$
7: **if** $C > powerToChargeOneNode$ **then**
8: $P_{lo} \leftarrow findProbLess(min(v))$
9: $P_{hi} \leftarrow findProbLess(max(v))$
10: **if** $P_{lo} \leq minTh$ AND $P_{hi} \leq maxTh$ **then**
11: $exploreFlag \leftarrow False$
12: **end if**
13: **else**
14: $exploreFlag \leftarrow False$
15: **end if**
16: **end if**
17: **return** $exploreFlag, v, l$
18: **end procedure**

ANLPP does not accommodate for varying power transfer efficiency and uses a fixed efficiency assumption.

Algorithm 2 invokes a charging method to determine the list of nodes to charge, v, and the estimated power needed for charging C' (line 3). The charging technique (line 3) is inspired by the Greedy Plus algorithm [13]. In our charging technique, we bring the lowest power node to the power level of the next lowest node. Then we repeat the process by increasing the power of now the two lowest nodes to the power level of the next lowest node and so on. We repeat this process until no power is left or the available power is not enough to bring the nodes to the next power level. The algorithm makes sure it can charge at least a single node (lines 6-7). If there is enough power left after estimating the charging cost the algorithm returns an indicator to continue exploration. If it can charge several nodes without enough power left for exploration it looks into dropping nodes from v (lines 7-12) in favour of exploration.

The algorithm uses probability to drop nodes from its charging list, v, enabling it to further explore the WRSN. By finding $P_r[X \leq Y]$, where Y is substituted with the lowest and highest power nodes in v and compared to a low and high threshold respectively to make an exploration decision (lines 8-12). The probability thresholds are identified based on the WRSN properties and characteristics. ANLPP returns the exploration decision $exploreFlag$, the nodes to charge v, and the target power level l they should reach. Once ANLPP return $False$ after visiting a node, exploration is terminated and the UAV starts its return to B_S and charges the nodes in v using CHARGE. Next, we describe CHARGE.

C. CHARGE

Once the UAV is above a node that is in v, a decision must be made to charge this node or not. ANLPP uses an expected efficiency $E(\eta)$ when identifying the nodes to charge and an estimated charging cost C'. In practice, the actual efficiency may vary from $E(\eta)$, Equation 3, based on landing accuracy and disturbances. CHARGE attempts to compensate for this possible efficiency difference. Given the actual available power for charging, C, the current node under the UAV, $node$, the list of nodes to charge, v, and the target charging level, l, CHARGE (Algorithm 3) identifies the feasibility to charge $node$.

Algorithm 3 Charge based on efficiency

Require: C ▷ Available power to be used for charging
Require: $node$ ▷ Current node to charge
Require: v ▷ List of nodes to charge in descending location order
Require: l ▷ Target power level to charge the nodes in v
Require: λ ▷ Power cost of concentrically localizing UAV with a node
Require: $E(\eta)$ ▷ Expected power transfer efficiency
1: **procedure** CHARGE($C, node, v, l$)
2: $C' \leftarrow$ power needed to charge all nodes in v to l
3: **if** $C < C'$ **then** ▷ Not enough power to charge all nodes in v
4: $v, l \leftarrow$ DETERMINENEWTARGET(C, v, l)
5: **end if**
6: **if** $node$ in v **then** ▷ Current node still in charging list
7: Land on $node$
8: $\eta \leftarrow$ Get efficiency based on landing data
9: $\omega_{stay} \leftarrow (l - node_{power})/\eta + \lambda$
10: $\omega_{reland} \leftarrow (l - node_{power})/E(\eta) + 2\lambda$
11: **if** $\omega_{reland} < \omega_{stay}$ **then** ▷ It is cheaper to re-land
12: take-off
13: $v, l \leftarrow$ CHARGE($C - \lambda, v, l$)
14: **else**
15: Transfer power to $node$ bringing its power up to l
16: **end if**
17: **end if**
18: **return** v, l
19: **end procedure**

Algorithm 3 starts by estimating, C', the power needed to bring the nodes in v to the level l using $E(\eta)$ (line 2). C' also includes the localization cost, λ, associated with charging. If the available power for charging, C, is less than the needed power C', then the given target level l is not achievable and must be reconsidered (lines 3-5). The algorithm invokes DETERMINENEWTARGET that will identify a feasible target and updates the charging list v. Once a charging list is feasible the algorithm charges $node$ if it is still on the charging list v (line 6).

When charging $node$ the UAV is instructed to land on the node, this will cost the UAV λ power (line 7). Once the UAV has landed on the target, or close to it, the actual power transfer efficiency, η, is exactly determined (line 8). The actual cost to charge $node$ is then computed based on η (line 9). Also, the cost of a take-off and landing again over $node$ is estimated (line 10). Based on the approach with lowest cost the algorithm transfers power or reattempt charging (lines 11-16). Algorithm 3 deals with charging a single node, the possibility of updating the charging list v, and the target power level l as needed. Next, we describe DETERMINENEWTARGET the algorithm responsible for the updating of v and l.

D. DETERMINENEWTARGET

When the available power for charging C is not enough to charge all the nodes in v to l then a new target level must be identified. This is the responsibility of DETERMINENEW-TARGET (Algorithm 4). The algorithm addresses this issue recursively (line 9). The algorithm finds the highest power node in v and uses its power level as the new target l (line 5).

Algorithm 4 Determine next charging power target level

Require: C ▷ Available power to be used for charging
Require: υ ▷ List of nodes to charge in descending location order
Require: λ ▷ Power cost of concentrically localizing MCR with a node
Require: $E(\eta)$ ▷ Expected landing efficiency
1: **procedure** DETERMINENEWTARGET(C, υ, l)
2: **if** υ is empty **then**
3: **return** $\upsilon, zero$
4: **end if**
5: $l \leftarrow$ power of maximum power node in υ ▷ New target
6: remove maximum power node from υ
7: $C' \leftarrow$ power needed to charge all nodes in υ to l using $E(\eta)$
8: **if** $C' > C$ **then** ▷ Not enough power, reduce levels more
9: $\upsilon, l \leftarrow$ DETERMINENEWTARGET($C, \upsilon,$)
10: **end if**
11: **return** υ, l
12: **end procedure**

Then this highest power node is drops from υ (line 6). Since υ is now a smaller list the power needed to charge υ is also reduced. It also decreases the target power, l, reducing the power needed for each individual node in υ. After that, it estimates the power needed to charge all the nodes in the new υ to the new l, C' (line 7). If C' is still greater than C then the algorithm recursively calls itself, further reducing l and the size of υ. Otherwise, it will return the current υ and l as a feasible solution (line 11). The algorithm has two recursion breaking conditions; the first is when υ is empty (line 2-3); the second is when charging υ to l is feasible.

DETERMINENEWTARGET uses expected efficiency $E(\eta)$ in its computations. While the algorithms' proposed list, υ, is theoretically possible and most probable, it may still fail if the next landing results in a very low efficiency case. In that case, the whole process is repeated and DETERMINE-NEWTARGET will make a new decision. Next, we describe the system we used to conduct experiments.

V. EXPERIMENTS SETUP

In this section, we describe the hardware and software components used in our system. We also identify and explain the models and external resources used in our experiments.

A. System Details

When conducting our power transfer experiments we used an Ascending Technologies Firefly quadrotor UAV to carry the transmitting coils and power system. This UAV has a 600g payload capacity. An Odroid XU-4 computer was mounted on the vehicle to run the algorithm and conduct autonomous landings. The Odroid XU-4 is a single board computer. The Odroid is equipped with an eight-core ARM processor, and two gigabytes of memory, and both an eMMC and a micro-SD flash storage. The Odroid is running an Ubuntu 16.04 operating system. The autonomous code is written in C++ running under Robot Operating System (ROS) middleware. The vehicle is controlled using a Proportional Integral Derivative (PID) controller. The control system uses position reference points to control and navigate the vehicle through the WRSN. When generating a WRSN we use real-world parameters from previous works [2] [21].

As defined in Equation 3 an efficiency function must be identified. We use the WiBotic Adaptive Wireless Charging

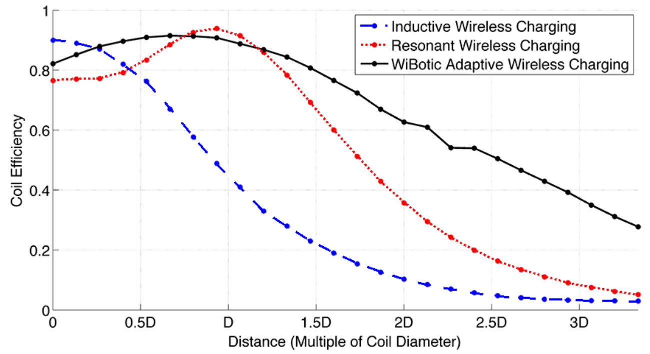

Fig. 3. WiBotic Distance vs Efficiency plot[1].

System[1]. Figure 3 shows the efficiency of the WiBotic and other systems. The power transfer efficiency is a function of the distance between the centre of the power transmitting coil and the power receiving coil. The transmission and receiving coils are located on the UAV and sensor node respectively. The diameter of our coil is $45cm$.

Figure 3 shows that the WiBotic system does not peak as high as a resonant wireless charging system. But the WiBotic system has a larger range of operation. Both resonant and inductive power charging reach $\approx 5\%$ efficiency at distance 3.5 times the coil diameter. On the other hand, the WiBotic system provides $\approx 24\%$ efficiency at 3.5 times the coil diameter. This enables the utilization of wireless power transfer with a higher error in coils' alignment.

B. Landing Experimental Setup

While GPS can be used to instruct the navigation of the UAV, it fails to accurately position the UAV at a specific location. In order to achieve high power transfer efficiency, a sub-meter accuracy is needed to locate a sensor node. The 2017 official GPS Performance Analysis Report for the Federal Aviation Administration reported the R95 average errors is in the three-meter range [22]. The vehicle manufacturer also reports a drift of $\pm 2m$. We compensate for this GPS error by using visual localization.

We implemented visual localization using both colour segmentation and shape detection. The camera we used is an mvBlueFox-MLC model manufactured by Matrix Vision. The camera is mounted underneath the vehicle, connected to the Odroid via USB. The camera provides 8-bit red green blue (RGB) images. Because hue saturation value (HSV) is closer to human perception of colours [23], the image colours are mapped to an HSV colour space. After computing HSV values for each pixel, the image is thresholded to identify areas of colour interest. Thresholds were set up based on actual colour readings of printed markers placed over the target locations. A thresholded image is shown in Figure 4, middle. We approach the known coordinates of a sensor using GPS, then search the area to identify the target. The target is identified by a centre and a radius, that is passed

[1]Wibotic wireless power, radio frequency safety, CoMotion Labs, 4545 Roosevelt Way NE, Suite 400, Seattle, WA, 98105-4721 (2017) http://www.wibotic.com/wireless-power/

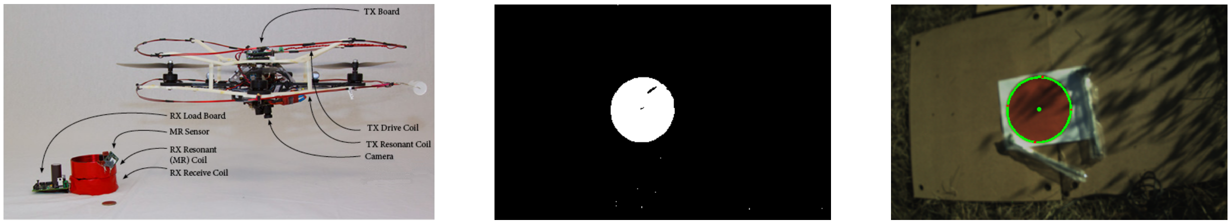

Fig. 4. Left: The vehicle used for wireless power transfer. An image going through the visual localization method. Middle: image after conversion and thresholds. Right: the projected centre and radios back on original image

on to the landing control code. An imposed target over the original image can be seen in Figure 4, right.

C. CPTEC Experiments Setup

In our field experiments, we generate a WRSN of size 150 nodes. Nodes are then visited to identify their power levels. Once the algorithm identifies the furthest point of exploration the set of nodes to charge, υ, is identified. Finally, the UAV returns back to B_S, and on the way back it charges the nodes in υ. The size of υ, in our field experiment, was found to be 11 nodes. To ease experiment setup and evaluation, instead of having 11 different nodes, the UAV landed on the same node 11 times. Between landings, the UAV travelled the distance between two nodes. We evaluate two charging algorithms: ANLPP with static landing, and CPTEC. First, we consider ANLPP with static landing, where the UAV lands on a node and identifies a charging efficiency then uses this charging efficiency to charge the node. The second approach uses CPTEC, where efficiency compensation is conducted. When CPTEC is invoked the UAV may charge a node, attempt re-landing or skip a node.

D. CPTEC Simulation Setup

In our simulations, we evaluate the performance of four approaches to charging with lack of *a priori* power knowledge. We assess the performance of each algorithm to an optimal full knowledge algorithm with maximum power transfer efficiency additional details on the optimal algorithm can be found in our previous work [2]. The four algorithms are Naive, Min, ANLPP, and CPTEC. All algorithms explore the WRSN and identify nodes to charge then return back to the base station and charge nodes on the way back. Naive computes the amount of power needed to charge the nodes it visits. Once it runs out of power to charge more nodes it terminates exploration and returns. Min will keep exploring until it identifies a minimum power node in the list of nodes it should charge. Once it finds a minimum power node, based on a predefined threshold, it terminates exploration and returns to base. ANLPP does not compensate for efficiency drops, while CPTEC does.

VI. Experiment Results

In this section, we show the results of field and simulation experiments. We start with landing characterization experiments, then conduct field experiments. We finally, use the data from real landings to perform additional simulations.

A. Landings

While CPTEC does not require the knowledge of sensor nodes' power levels, N_e, the algorithm does need the expected power transfer efficiency $E(\eta)$. Since power transfer efficiency, η, is a function of landing distance, λ, Equation 3. In order to identify $E(\eta)$, we need to find $E(d_\lambda)$. Landings are characterized to determine $E(d_\lambda)$.

Our first set of experiments characterize landing distances. We then use the landing distances to compose a landing probability distribution function (PDF). Using the PDF we can determine the expected landing distance $E(d_\lambda)$. Figure 5 shows the landing PDF gained from attempting 60 landings in winds varying between $0.2 - 4.5$ m/s and measuring the distance away from the target. The figure shows a peak close to the $1D$, a coil diameter distance $\approx 45cm$. The resulting mean from all landings is $81.8cm$, this is $E(d_\lambda)$.

By applying Equation 3, we use $E(d_\lambda)$ to determine the expected power transfer efficiency $E(\eta)$. We use Figure 3 as Φ. We find that $E(\eta)$ is 69.08%. With the expected efficiency, $E(\eta)$, identified we proceed with the experiments.

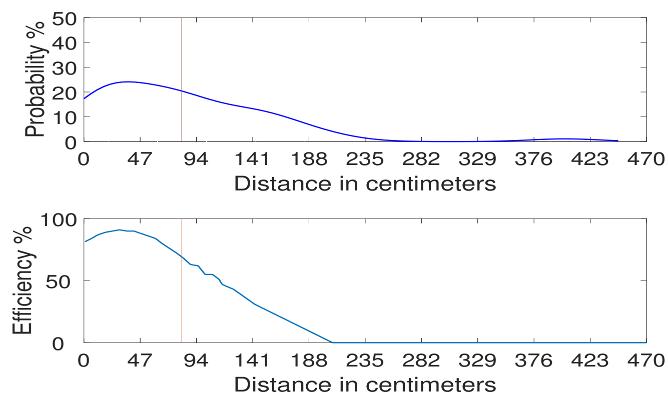

Fig. 5. Top: Landing probability distribution. Bottom: Landing power transfer efficiency. The red line indicates the mean value.

B. CPTEC Results

Figure 6, shows each node in υ with its power level before and after charging. A node is represented with a bar that is split in two, each part represents the behaviour under each charging algorithm. The nodes are ordered by their distance from B_S, where a sensor node with lower ID is closer to B_S. At the top of each node the action taken by the algorithm can be seen, where C indicates the node is charged at first landing, L means the algorithm decided to leave this node.

R indicates the algorithm decided to re-land, and the node is either charged or not based on the second landing.

Figure 6 shows the results of running CPTEC in the field. We compare the behaviour of CPTEC and ANLPP with static landing. It can be seen that ANLPP with static landings clearly failed to charge all the nodes in υ. ANLPP with static landing charged the fifth node (ID 7), in υ on its way back, even though it had a near zero efficiency. Charging the fifth node (ID 7) drained all the available charging power and prevented it from charging the other nodes. While CPTEC decides to drop the fifth node, as well as the ninth node (ID 3) and charged the rest of the nodes. CPTEC also determined that re-landing was a better option three times. One of the re-lands resulted in a better efficiency enabling the charge of node 6. The other two re-landings resulted in worse efficiency causing the abandonment of nodes one and two. CPTEC was able to charge nine out of 11 nodes, while ANLPP with static landing was only able to charge five nodes. This shows the ability of CPTEC to overcome and compensate for low-efficiency power transfer cases.

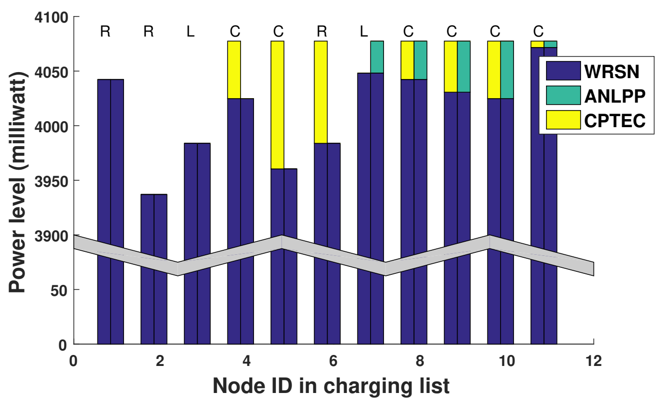

Fig. 6. Field results of charging nodes in υ using CPTEC & ANLPP static landing, L indicates Leave Area, R is Retry Area, C is Stay Area.

C. CPTEC Simulation Results

We generate one million random WRSN of size 150 nodes each. For each network, we run an optimal full knowledge charging algorithm that determines the maximum possible network life increase. We run each of the four algorithms on the same WRSN and record the performance of each algorithm. Our first set of experiments assume an optimal landing on each sensor node producing a maximum power transfer efficiency. We show simulation results in Figure 7, left. Figure 7, left, shows that both ANLPP & CPTEC are able to perform with a median of 90% of what a full knowledge optimal algorithm can achieve. The Naive algorithm fails since it runs out of power to explore in favour of charging the few nodes it encounters in the beginning. Min algorithm does perform a bit better, it makes sure to charge at least a single low power node. But it wastes power charging nodes that are not part of the optimal set of nodes to charge. CPTEC and ANLPP perform identically when the power transfer efficiency is the best possible.

We repeat the simulation experiments but use simulated landings. We use the landing PDF from Figure 5. Using Equation 3 efficiency η is determined for each charging. Each algorithm's performance is shown in Figure 7, right.

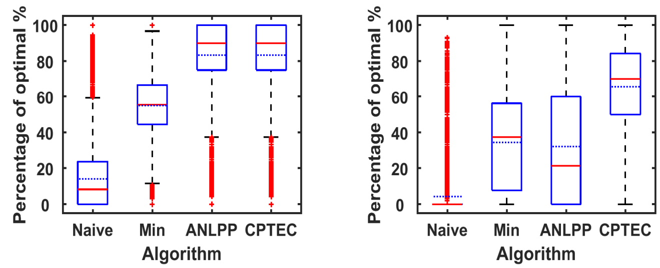

Fig. 7. Left: Performance of the four charging algorithms using maximum efficiency for each landing. Right: Performance of the four charging algorithms when landings produce different efficiencies. Red line represents the median and the blue dotted line represents the mean.

Figure 7, right, shows the impact of varying efficiency on each algorithm, compared to the left figure, as well as the algorithms compared to each other. Each algorithm's individual performance drops. While ANLPP drops the most in comparison, CPTEC not only maintains performing higher than all the other algorithms, 72%, it also suffers the least drop in overall performance.

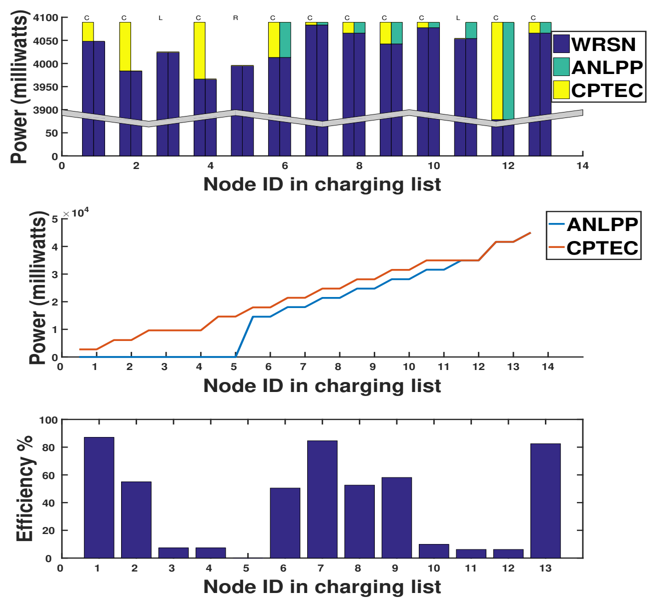

Fig. 8. Top: A simulate WRSN being charged by UAV. Middle: Charging energy left on-board UAV over each node. Bottom: Power transfer efficiency at each node.

Finally, we examine a single WRSN charging simulation. Figure 8 demonstrates both CPTEC and ANLPP with static landing charging a WRSN. CPTEC drops the third node it encounters (node 11) due to low efficiency and available charging power level. When CPTEC encounters node 5 it finds nearly zero efficiency that leads it to discard this node. On the other hand, ANLPP with static landing commences charging in spite of near zero charging efficiency, and this leads to losing all the UAV charging power. Figure 8 shows that charging node 5 leads to near zero power for ANLPP while ignoring this node only costs CPTEC the landing cost. CPTEC still dropped another node due to not having enough power left to charge all the remaining nodes. While CPTEC was still not able to charge all the nodes in υ, it significantly outperformed the static land and charge approach. Sacrificing some nodes proved to be very beneficial.

VII. Conclusion and Future Work

In this work, we presented Charging with Power Transfer Efficiency Compensation (CPTEC) a WRSN charging algorithm. CPTEC operates in the absence of *a priori* sensor nodes' power level knowledge and compensates for power transfer efficiency loss, due to real landings. We showed that our solution can increase the life of a WRSN on average to 72% of a theoretical optimal solution that assumes full knowledge and a constant maximum efficiency. CPTEC also achieved 90% of optimal under constant efficiency, similar to our previous work ANLPP [2]. On the other hand, ANLPP drops to almost 22% with varying efficiency.

We used an off the shelf UAV and a commercially available wireless power transfer system with real-world operation parameters [21]. We used visual and GPS data to control the descent of the UAV on the sensor. We conducted simulations to demonstrate the improvement in performance with and without efficiency drops. Our results showed a promising and practical solution to increase the life of large-scale WRSN.

The next steps are to incorporate a larger vehicle, create a test bed of sensors, and investigate the impact of no power information delegation alone on the life of a WRSN. In addition, we plan to investigate the impact of changing network parameters, and conduct a competitive analyses.

References

[1] B. Griffin and C. Detweiler, "Resonant wireless power transfer to ground sensors from a uav," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 2660–2665.

[2] N. W. Najeeb and C. Detweiler, "Extending wireless rechargeable sensor network life without full knowledge," *Sensors*, vol. 17, no. 7, 2017.

[3] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljačić, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, no. 5834, pp. 83–86, 2007.

[4] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 5:1–5:39, Feb. 2009.

[5] I. Kang and R. Poovendran, "Maximizing static network lifetime of wireless broadcast ad hoc networks," in *Communications. ICC '03. IEEE International Conference on*, vol. 3, May 2003, pp. 2256–2261.

[6] S. Soro and W. B. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *19th IEEE International Parallel and Distributed Processing Symposium*, April 2005, p. 8.

[7] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA. New York, NY, USA: ACM, 2002, pp. 32–41.

[8] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3, 2001, pp. 1388–1397 vol.3.

[9] M. Najimi, A. Ebrahimzadeh, S. M. H. Andargoli, and A. Fallahi, "Lifetime maximization in cognitive sensor networks based on the node selection," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2376–2383, July 2014.

[10] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *2003 IEEE International Conference on Robotics and Automation (IROS) (Cat. No.03CH37422)*, vol. 1, Sept 2003, pp. 19–24 vol.1.

[11] R. Wang, M. Veloso, and S. Seshan, "Active sensing data collection with autonomous mobile robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2583–2588.

[12] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle," in *Proceedings 2006 IEEE International Conference on Robotics and Automation. ICRA 2006.*, May 2006, pp. 2091–2098.

[13] Y. Peng, Z. Li, W. Zhang, and D. Qiao, "Prolonging sensor network lifetime through wireless charging," in *2010 31st IEEE Real-Time Systems Symposium*, Nov 2010, pp. 129–139.

[14] C. Wang, J. Li, F. Ye, and Y. Yang, "Multi-vehicle coordination for wireless energy replenishment in sensor networks," in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, May 2013, pp. 1101–1111.

[15] L. Xie, Y. Shi, Y. T. Hou, and H. D. Sherali, "Making sensor networks immortal: An energy-renewal approach with wireless power transfer," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1748–1761, Dec 2012.

[16] E. Gelenbe, D. Gesbert, D. Gunduz, H. Külah, and E. Uysal-Biyikoğlu, "Energy harvesting communication networks: Optimization and demonstration (the e-crops project)," in *2013 24th Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*, Sept 2013, pp. 1–6.

[17] C. M. Angelopoulos, S. Nikoletseas, and T. P. Raptis, "Wireless energy transfer in sensor networks with adaptive, limited knowledge protocols," *Computer Networks*, vol. 70, pp. 113 – 141, 2014.

[18] A. Madhja, S. Nikoletseas, and T. P. Raptis, "Distributed wireless power transfer in sensor networks with multiple mobile chargers," *Computer Networks*, vol. 80, pp. 89 – 108, 2015.

[19] A. P. Sample, D. T. Meyer, and J. R. Smith, "Analysis, experimental results, and range adaptation of magnetically coupled resonators for wireless power transfer," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 544–554, Feb 2011.

[20] M. Pinuela, D. C. Yates, S. Lucyszyn, and P. D. Mitcheson, "Maximizing dc-to-load efficiency for inductive power transfer," *IEEE Transactions on Power Electronics*, vol. 28, no. 5, pp. 2437–2447, May 2013.

[21] C. Detweiler, M. Eiskamp, B. Griffin, J. Johnson, J. Leng, A. Mittleider, and E. Basha, *Unmanned Aerial Vehicle-Based Wireless Charging of Sensor Networks*. Cham: Springer International Publishing, 2016, pp. 433–464.

[22] W. J. H. T. Center, "Global positioning system (gps) standard positioning service (sps) performance analysis report," http://www.nstb.tc.faa.gov/reports/PAN96_0117.pdf, Atlantic City, NJ: Federal Aviation Administration, 2017 (accessed September 1, 2017).

[23] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the hsv color space for image retrieval," in *Proceedings. International Conference on Image Processing*, vol. 2, 2002, pp. II–589–II–592.