IoT Grid Alignment Assistant System for Dynamic Wireless Charging of Electric Vehicles

Ahmad Hassan Kobeissi[1,2], Francesco Bellotti[1], Riccardo Berta[1], Alessandro De Gloria[1]
[1]*University of Genova*
[2]*EDST, Lebanese University*
Genova, Italy
Email:{ahmad.kobeissi, franz, berta, alessandro.degloria}@elios.unige.it

***Abstract*—Several technological solutions are being researched to establish efficient, reliable, and robust wireless vehicular electric charging on a wide scale. We propose an artificial vision system – the Grid Alignment Assistant System (GAAS) - aimed at maximizing the energy flow between the transmitting and receiving goals, by supporting the driver in the task of keeping an alignment between the vehicle and the charging grids in the road. GAAS exploits low cost and open hardware and software components. GAAS supports IoT capabilities as an API cloud service is developed for real-time measurements storage. This paper presents road test results showing that high precision misalignment estimations can be achieved, and presents some challenges imposed by the actual deployment environment.**

Keywords—Electric vehicles, wireless electric charging, image processing and recognition, artificial vision, IoT.

I. INTRODUCTION

Fully electric vehicles (FEV) are gradually starting to supplant internal combustion engine (ICE) vehicles mainly due to their energy efficiency [1]. Unlike hybrid vehicles, FEVs require external charging which is widely acquired by plug-in charging devices [2]. Even though faster plug-in charging methods are being developed [3] and will continue to be, it is still a static mode of charging. Charge-while-driving, also known as dynamic wireless charging, is posing as a promising and valid charging method for FEVs, with ever new powerful solutions being developed [4]. Dynamic charging offers mobility while charging as well as minimizing battery weight leading to more energy saving [5].

The setup for dynamic charging includes transmitter and receiver inductive coils. While the receiver coil is placed at the floor of the vehicle, a series of transmitting coils are placed to form consecutive grids along the road. Those grids could be placed anywhere within the street lane: lane centre, right shifted, or left shifted.

In order to efficiently maintain the power transfer between the grids and the vehicle, the transmitter and receiver coils must be aligned for inductive coupling. Tackling the misalignment problem has been regarded through various different approaches. In [6], a system is developed to deliver an equal amount of energy for all lateral misalignments in the range of ±15 cm, which, according to the authors, improves the expected value of transferred energy by more than 30%. [7] proposes an autonomous coil alignment system (ACAS) using fuzzy steering control based on the voltage difference between two coils installed on the front-left/front-right of the power receiver coil. Similarly, in [8], a coupling sensor measures the coupling level between the road and vehicle coils, while a camera detects the distance of the vehicle from the two lines of the lane.

IoT plays a fundamental role in integrating EVs with smart cities concepts [16]. An act, as simple as uploading sensory data to the cloud for immediate reactions and/or later analysis [17], would optimize the usage of the charging grid.

This paper presents the grid alignment assistant system (GAAS), aimed at maximizing the energy flow in wireless charging. In GAAS we used low-cost hardware and free software to provide a high precision estimation of the offset between the charging grid within the road and the receiver within the car floor. Within FABRIC [9], we displayed to the driver the offset as a pointer across a gauge scale within a Human-Machine Interaction HMI module available in the vehicle's dashboard. In addition, we developed a tailored and light API service for streaming measurements to a local or remote (cloud) storage.

Following this introduction, we present the context of FABRIC project in section II, before we barge into the details of implementation of GAAS in section III. In section IV, we discuss the main results of the experimental development of GAAS. Finally, we conclude with suggested future work in section V.

II. FABRIC PROJECT

The main scientific and technological objective of FABRIC was to conduct a feasibility analysis of on-road charging technologies for long-term electric vehicle range extension. Key wireless charging technologies, trends and relevant R&D activities in the automotive sector were considered, assessing the present and future opportunities for wireless charging, taking into account the needs of EV makers and end users.

In more detail, FABRIC addressed the following objectives:

- Collection of end-user requirements and industry demands that determine the potential of success of such technologies in various application sectors [10]

978-1-5386-9585-2/18/$31.00 ©2018 IEEE

- Identification of technology drivers and challenges that impact the implementation of wireless charging technology and the widespread installation of wireless charging infrastructure
- Determination of product and technology development activities by technology developers, EV makers and other key stakeholders
- Proposal of partnerships and collaboration between key stakeholders for the implementation of the technology
- Survey of governmental policies, regulations and public and private funding activities impacting the progress of wireless charging infrastructure
- Evaluation of technology penetration potential for wireless charging in public transportation in addition to the passenger car segment
- Bridging the technological gaps and proposal of a rational solution for both the grid and the road infrastructure.

The vision of FABRIC is the large-scale adoption of pure Electric Vehicles (EVs) in future transportation systems. This wide deployment requires mature EV technology and advanced charging solutions that provide a user experience similar to today's cars.

The main challenge that FABRIC has tackled is “range anxiety” that is caused by the limited range current EVs are suffering from. In the long term, as Fig. 1 shows, electric vehicles might be able to collect energy from the road, in a conductive or contactless mode.

Compared to the current paradigms of larger installed storage capacity, a fast charge or switchable batteries, advanced on-road charging solutions should improve the driving range and battery lifetime of the Full Electric Vehicles (FEV) as well as their energy efficiency and price.

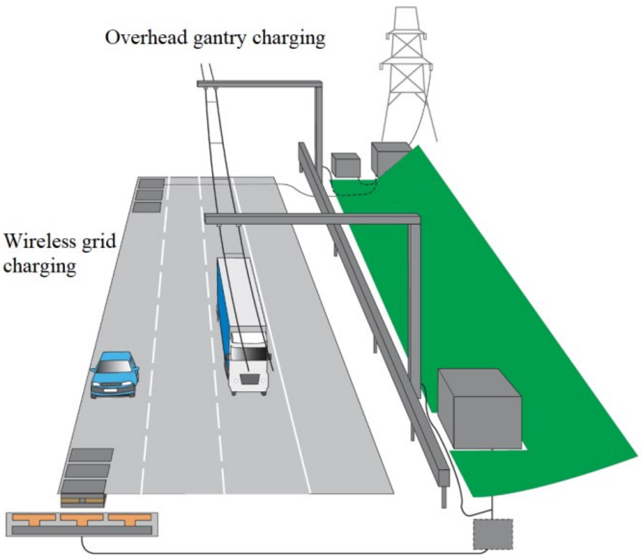

Fig. 1. Environment schema of FABRIC project.

Fabric set up two demonstration installations, with different wireless charging technological solutions, one in Versailles, France, and one at the MotorOasi safe drive track in Susa (TO), Italy. In this paper, we focus on the technology developed for this second test site, managed by the Tecnositaf Fabric's partner, where the project's final review and the event took place in late June 2018. Fig. 2 shows an aerial view of the area, with the green arrows indicating the two charging grid stripes buried in the asphalt.

Fig. 2. TECNOSITAF’s Testing site track in Susa, Italy.

III. GAAS

GAAS is designed to provide grid alignment assistance to drivers. The development was centred on facilitating the (manual) steering of the vehicle to achieve the highest possible outcome from wireless charging. An offset, available in real-time, would be considered when steering to achieve optimal alignment. From the driver’s point of view, this offset is the distance and the direction of the misalignment between the axis of the vehicle and that of the charging grid within the road. GAAS has a user interface which was integrated as part of the Human-Machine Interaction (HMI) module of the FABRIC’s onboard unit (OBU).

A. Specifications

In the project, we aimed at developing a low-cost, energy-efficient, and high-precision system for alignment within a tolerability range of up to +/-20 cm [8]. This range was defined in preliminary studies, based on the physics behind the charging grid [18]. A low-cost system means we did not use cutting-edge technologies in radar and lidar or other expensive sensory hardware. An energy-efficient system dictated that our design would be better to avoid motorized mechanics (electrical motors and shafts) and/or heavy processing (CPU-exhausting functions). Moreover, the possibility of using IoT processing devices as alternatives to PCs would presumably reduce the energy consumption drastically since the power consumption in order of magnitude is significantly lower in PCBs [19].

B. System Architecture

To meet the above-mentioned objectives, we implemented a prototype using a set of low-cost and low-power consumption hardware and free software tools for accomplishing a high precision grid alignment assistant system (Fig. 3).

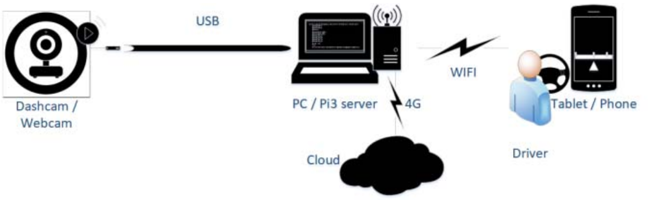

Fig. 3. Block diagram representing GAAS.

For hardware, we used a video camera (Logitech c920), a high definition (1920x1080) webcam to capture 30 frames per second of the road ahead as shown in Fig. 4. Mounted on the top-middle of the windshield using a suction cup, the camera is connected to a server-node device: the Raspberry Pi 3 [11], or a regular PC (Fig. 3), both of which run python on Windows or Linux OS. The Pi is very suitable for size, cost, and energy-saving. However, in the development phase, as working with the Pi 3 module required a monitor, mouse, and keyboard for monitoring, tracing, and editing the code, we used an ordinary laptop instead for mobility reasons. For the final system release, the laptop could be seamlessly replaced with the low-cost Pi 3 platform.

Our laptop choice was a Samsung Chronos 770Z5E with 3rd gen Intel Core i7, 8 GB of main memory RAM, and an AMD Radeon HD 8870M graphics card with 2 GB of dedicated RAM.

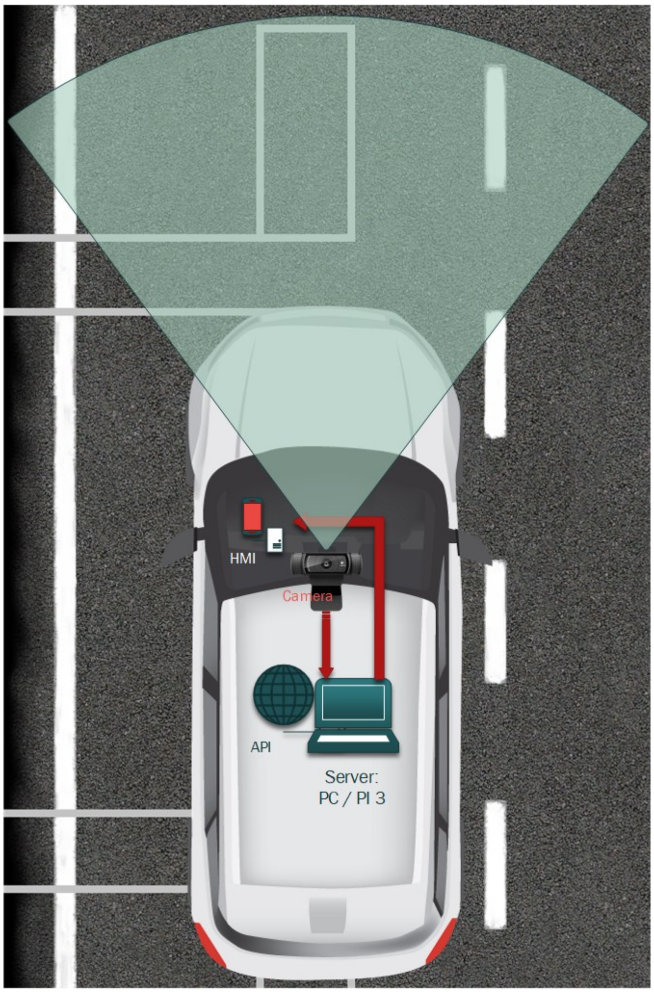

Fig. 4. GAAS deployment schema.

C. Methodology & Development

The python server node processes the frames one by one, estimating the current position of the centre of the vehicle w.r.t the centre of the grid. This offset estimation is transmitted wirelessly to the HMI app placed on the vehicle's dashboard, running on an ASUS ZenPad S 8.0, and displayed for the driver to consider while steering for a better alignment. At the same time, the offset and charging measurements provided by the electronic sensors within the charging module onboard the car are uploaded to get timestamped on a local and/or remote cloud through a custom-made API cloud service. Fig. 3 shows a block diagram representation of GAAS, while a top-view with the embedded system drawn out is previewed in Fig.4.

The core functions of the system run on the server and are written in python. We used Jupyter Notebook [12], previously known as IPython Notebook, an interactive computing utility for code development using Python 3.7.0 [13] to easily visualize, track, and trace the running code. Within the notebook, a sequence of image processing functions runs iteratively with the goal of estimating the offset between the vehicle's centre and grid's centre. The vehicle's centre is set as the frame centre on the x-axis. To ensure that the centre of the frame is representative of the centre of the vehicle, we attached a visible thin strip on the bottom of the windshield exactly at the vehicle's centre. Accordingly, within the camera feed, we drew a similar line on the equator of the frame (Fig. 5's and Fig. 6's pink line in the bottom of the left-side images) so that upon mounting the camera both lines must be overlapping.

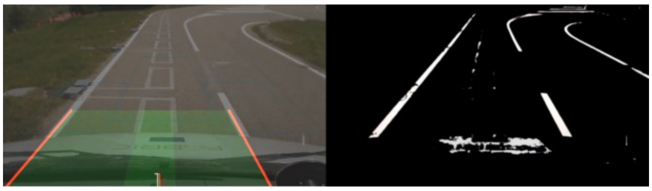

Fig. 5. Intermediate estimation of grid and lane centres (left) and gaussian blur mask (right) on the straight left-side lane of the track.

Fig. 6. Intermediate estimations (left) and median blur mask (right) on the curvy right-side lane of the track.

Three steps formulate each frame's processing: object recognition, borderlines averaging, and middle point estimation. The objects we set to recognize are the grid and the lane. The grid is the prime priority, we switch to lane borders recognition in case the grid fails to be recognized.

Grid recognition could fail for many reasons including, but not limited to, sun glare, colour blending between the grid and the surrounding pavement, and the change of the grid's placement within the lane. Should that occur, we try to estimate the grid's centre based on lane recognition information. Moreover, since it is unknown beforehand whether a grid could be recognized or not, we implemented a parallel image processing pipeline on two copies of each frame one for grid recognition and another for lane recognition.

For each object, we set specific colour masks and region masks. Two colour masks, or filters, were defined for the grids - one is light grey and the other dark grey - corresponding to the two types of grid installations on the Susa test site track (Fig. 2). We tried to replace the two masks with a single grey mask, but it introduced a lot of noise whilst processing, that could not be filtered because of the similar grey colour of the road pavement. As for the lane lines, we considered the two internationally standardized lane colours: white and yellow. The colour mask is not a single colour code, but rather an interval of colour codes that covers the shades of a specific colour.

Next, the algorithm starts a parallel processing pipeline where low-pass filters are applied to the two instances of the original frame. In the first one, we apply a median-blur which was more suitable for grid detection as shown in the right-side image of Fig. 5, while in the second one, a Gaussian filter seemed more suitable for lane lines detection as shown in the right-side image of Fig. 6. For either filter, the kernel size was set at 5x5, as the best trade-off between accuracy and time performance, as we will see. After that, both instances undergo edge detection using the canny algorithm that shapes out the contour of the grid and of the lane lines. Then, region masks selection is performed. The grid region mask is centred, while lane lines mask is composed of two separate regions at the right and left sides of the frame. Fig. 7 shows the regions in blue poly-lines, the centre one is dedicated to the grid detection region, and the outer two symmetrical regions are dedicated to lane borders detection regions.

After the frame instances are cropped by the region masks, they become ready for Hough lines estimation, in which a transform function is performed to produce every possible straight line within specific slope and bias parameters. These parameters are adjusted to fit the physical perception of the dimensions of the road. A concentration of Hough lines would form in each region, signifying the recognition of grid or lane borders. A clustering algorithm identifies each concentration, and then a median line is computed for each cluster. Thus, each couple of near-symmetric lines are associated together as an object (grid or lane) representation.

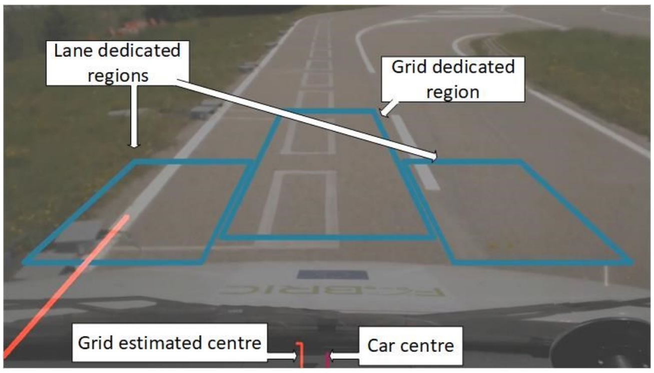

Fig. 7. Region masks outline.

As mentioned before, the priority in image processing is for grid detection. If a grid was recognized in its own region, the algorithm defines the two lines representing the right and left sides of the grid according to a validity check including the parameters: line slope and bias. Then, a middle line can be generated, the centre of which is our estimation for the grid centre coordinates. The offset can now be computed as the difference between the estimated grid centre and the frame middle. In the other pipeline, the offset is the lane centre, which we use to compute the estimated grid centre after settling the deficit between both.

The offset that is computed at this stage by either pipeline is a pixel metric and needs to be converted to centimetres. The conversion relies on the pre-set grid size in cm, that we measured to be 55 cm in the testing track in Susa, lane size which, 250 cm, and displacement of the grid within the lane. Based on that, a function returns a pixel-to-cm-ratio when it is fed with the grid sizes both in pixels (the difference between grid/lane estimated sides) and cm. This ratio is updated for every grid or lane recognition, then used to convert the offset from pixel to cm metric unit.

Once the offset is computed, from a grid or lane based estimation, it is sent over WIFI to the tablet device where an Android app displays the offset as a pointer on a linear gauge. At the top right corner of the screen, a button would take the user to a setting page, as shown in Fig.8. The settings form includes an IP address for the server, a port number, and the grid size value needed to adjust the gauge scale representation of the grid.

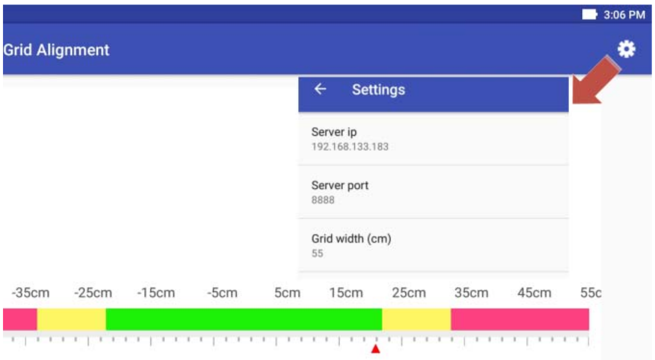

Fig. 8. GAAS's linear gauge and settings in the Android App.

A POST HTTP request is also sent to a private online server with the grid misalignment estimation, its direction, and the vehicle charging state (received via CAN bus) as payloads. A server is dedicated to handling the requests and storing the data into a document-based MongoDB. The API service, constructed in node.js, serves, in addition to the POST, a GET command to provide all measurements, and another GET with query support to select specific measurements. A local server instance is used when an internet connection is not available and uploads all the temporarily unsent measurements as soon as a connection becomes available. In this way, the online database has all the data for monitoring, analysis, and sharing.

IV. Experimental Deployment

Upon comparing the estimated misalignment measurements provided within GAAS with manual misalignment measurements, we recorded varying differences. The error difference was low to 0-3 cm when setting a larger kernel size for filters and increasing the detection threshold of lines. The drawback was an increase in latency to more than 500 MS. While when we lowered the detection threshold of lines and decreased the kernel size of filters the latency was eliminated but the estimation of offset grew inconsistent which was reflected as instability in the gauge pointer. Furthermore, the error difference increased to 5-8 cm. Thus, it was a trade-off between challenge between accuracy and latency. Nevertheless, GAAS suffered some other challenges during testing. High-speed driving caused vibrations on semi-rugged roads, which obstructed the recognition process for small durations. Sun glare, both direct and reflected by the road, introduced noise into the frames, which we dealt with as well as possible using software low-pass filters although hardware anti-glare and polarizer lenses could help reduce the solar luminosity. Rainy and snowy conditions were no challenges yet snow covering the sides of the track, as shown in Fig.9, was troublesome for recognition of white lane lines.

Fig. 9. The snow-surrounded track in Susa, Italy.

Another challenge was the case of misrecognition occurring when a grid recognition failed as well as failure in lane recognition. The lane recognition would fail when the dashed lines in the centre of the road (that is a border of a lane) had too wide a gap between its coloured dashes that the algorithm fails to provide a continuous recognition. To tackle this situation, we developed a function based on the recognition of a single line. The function implements an online algorithm that trains an estimated distance between the two lane-borders to the provisional grid centre using the previous 10 frames with true recognition of both borderlines. The online algorithm is a recursive procedure in which we compute the average misalignment to serve as a provisional offset in special cases. If the frame loses sight of one borderline (typically, the central dashed one), the function uses the average of the trained estimated distance to provide the grid's centre position. Fig. 7 shows a case when the algorithm estimates the grid's centre using one lane line as an anchor point.

As for the IoT side, the size of each POST request payload is 54 Bytes. Since we process 30 frames per seconds, we upload 1.62 KB every second. That accumulates to a 5.832 MB per hour, which is considered light for storage and bandwidth. We ran live tests for several kilometres during which we made recordings for in-lab testing.

V. Conclusion and Future Work

The GAAS system was designed to aid the alignment of vehicle and grid to maximize power transfer for wireless charging on the go. Driven by low-cost hardware and free software, GAAS provided high precision estimations of misalignment, which are passed to an in-vehicle HMI for assisting the driver in re-alignment along with light payloads of IoT cloud storage for monitoring and analysis all in real time. However, the precision had its costs in latency of computations and stuttering of transmissions. The literature contains several a system that deals with the issue of misalignment in wireless charging of FEVs. To the best of our knowledge, these systems neither use our methodology and approach nor suffer the challenges we faced. The IoT data that is recorded to that cloud could be of utmost importance for learning about the charging procedure with respect to the vehicle and street conditions.

As we have discussed earlier, our system depends on geographic knowledge of the charging area, such as grid width, lane width, and grid displacement. This information could be recorded and made available in a dedicated "e-road" layer of a local dynamic map [14]. Further developments could also include the integration of GAAS with a collaborative road mobility application [15]. In addition, the electronics involved in this prototype needs refactoring to be ready for integration with the on-board vehicle ECU for a centralized control and better power management.

Acknowledgement

We would like to thank the FABRIC coordinator, Professor Angelos Amditis and all the colleagues that allowed a successful performance of the project.

This work was supported in part by the EU, under the Feasibility Analysis and Development of on-road charging solutions for future electric vehicles (FABRIC) integrated project (FP7-SST-2013-RTD-1 605405).

References

[1] P. E. Ross, "What will the transportation miracle be? [Blueprints for a Miracle]," in IEEE Spectrum, vol. 55, no. 6, pp. 46-47, June 2018. doi: 10.1109/MSPEC.2018.8362227.

[2] T. Zhao and Z. Ding, "Distributed Initialization-Free Cost-Optimal Charging Control of Plug-In Electric Vehicles for Demand Management," in IEEE Transactions on Industrial Informatics, vol. 13, no. 6, pp. 2791-2801, Dec. 2017. doi: 10.1109/TII.2017.2685422.

[3] D. Sbordone, I. Bertini, B. Di Pietra, M.C. Falvo, A. Genovese, and L. Martirano, "EV fast charging stations and energy storage technologies: A real implementation in the smart micro grid paradigm," Electric Power Systems Research, Volume 120, 2015, Pages 96-108, doi: 10.1016/j.epsr.2014.07.033.

[4] R. Ruffo, V. Cirimele, M. Diana, M. Khalilian, A. L. Ganga and P. Guglielmi, "Sensorless Control of the Charging Process of a Dynamic Inductive Power Transfer System With an Interleaved Nine-Phase Boost Converter," in IEEE Transactions on Industrial Electronics, vol. 65, no. 10, pp. 7630-7639, Oct. 2018.

[5] S. Jeong, Y. J. Jang and D. Kum, "Economic Analysis of the Dynamic Charging Electric Vehicle," in IEEE Transactions on Power Electronics, vol. 30, no. 11, pp. 6368-6377, Nov. 2015. doi: 10.1109/TPEL.2015.2424712.

[6] R. Tavakoli and Z. Pantic, "Analysis, Design and Demonstration of a 25-kW Dynamic Wireless Charging System for Roadway Electric Vehicles," in IEEE Journal of Emerging and Selected Topics in Power Electronics. doi: 10.1109/JESTPE.2017.2761763.

[7] K. Hwang, J. Park, D. Kim, H. H. Park, J. H. Kwon, S. I. Kwak, and S. Ahn, "Autonomous Coil Alignment System Using Fuzzy Steering Control for Electric Vehicles with Dynamic Wireless Charging," Mathematical Problems in Engineering, vol. 2015, Article ID 205285, 14 pages, 2015. doi: 10.1155/2015/205285.

[8] V. Cirimele, O. Smiai, P. Guglielmi, F. Bellotti, R. Berta, and A. De Gloria, "Maximizing power transfer for dynamic wireless charging electric vehicles," 2017Lecture Notes in Electrical Engineering, Volume 429, 2017, Pages 59-65, International Conference on

Applications in Electronics Pervading Industry, Environment and Society, APPLEPIES 2016; Rome; DOI: 10.1007/978-3-319-55071-8_8, ISBN: 978-331955070-1.

[9] A. Amditis, G. Karaseitanidis, I. Damousis, P. Guglielmi and V. Cirimele, "Dynamic wireless charging for more efficient FEVS: The fabric project concept," MedPower 2014, Athens, 2014, pp. 1-6..

[10] O. Smiai, F. Bellotti, A. De Gloria, R. Berta, A. Amditis, Y. Damousis, and A. Winder, "Information and communication technology research opportunities in dynamic charging for electric vehicle," in Proceedings - 18th Euromicro Conference on Digital System Design, DSD 2015, August 2015.

[11] Raspberry Pi 2 Model B. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[12] Jupyter Notebook. https://jupyter.org/

[13] Python 3.7. https://www.python.org/downloads/release/python-370/

[14] K. Nizomjon and R. D. Oh, "Middleware structure using Local Dynamic Map within its environment," 2017 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, 2017, pp. 1-5. doi: 10.1109/ICISCT.2017.8188564.

[15] F. Bellotti, S. Kopetzki, R. Berta, P. K. Paranthaman, G. R. Dange, P. Lytrivis, A. J. Amditis, M. Raffero, E. Aittoniemi, R. Basso, I. Radusch, and A. De Gloria, "TEAM applications for Collaborative Road Mobility," in IEEE Transactions on Industrial Informatics. in press.

[16] M. Aloqaily, I. A. Ridhawi, B. Kantraci and H. T. Mouftah, "Vehicle as a resource for continuous service availability in smart cities," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, 2017, pp. 1-6. doi: 10.1109/PIMRC.2017.8292752.

[17] S. Khan, D. Paul, P. Momtahan and M. Aloqaily, "Artificial intelligence framework for smart city microgrids: State of the art, challenges, and opportunities," 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, 2018, pp. 283-288. doi: 10.1109/FMEC.2018.8364080.

[18] V. Cirimele, M. Diana, F. Freschi and M. Mitolo, "Inductive power transfer for automotive applications state-of-the-art and future trends," in IEEE Transactions on Industry Applications. doi: 10.1109/TIA.2018.2836098.

[19] M. Chiandone and G. Sulligoi, "Energy control in all-electric ship: State of the art and IoT perspectives," 2017 AEIT International Annual Conference, Cagliari, 2017, pp. 1-4. doi: 10.23919/AEIT.2017.8240546.