

# Towards Autonomous UAV Railway DC Line Recharging: Design and Simulation

Frederik Falk Nyboe<sup>1</sup>, Nicolaj Haarhøj Malle<sup>1</sup>, Gerd vom Bögel<sup>2</sup>, Linda Cousin<sup>2</sup>, Thomas Heckel<sup>3</sup>, Konstantin Troidl<sup>3</sup>, Anders Schack Madsen<sup>1</sup>, Emad Ebeid<sup>1</sup>

**Abstract**—Autonomously recharging UAVs from existing infrastructure has enormous potential for various applications, such as infrastructure inspections, surveillance, and search and rescue. While it is an active area of research, most related work focuses on alternating current (AC) infrastructure while very little work has been done on investigating the potential of recharging UAVs from direct current (DC) infrastructure.

This work proposes a UAV system designed to autonomously recharge from existing DC infrastructure. Two onboard powerline grippers and a motorized cable drum enable the UAV to perform a two-stage landing on railway DC lines where a wire is connected between them through the UAV for recharging. Light-weight electronics designed to be carried by the drone are developed to harvest energy from up to 3kV DC railway lines. The recharge mission is autonomously executed using fully onboard and real-time perception and trajectory planning and tracking algorithms. The potential of the system is shown in lab setting validation, with hardware-in-the-loop simulation, and partly in a real overhead powerline environment, verifying the functionality of the sub-components.

## I. INTRODUCTION

Designing drones that can fly tens of kilometers to inspect a large part of the infrastructure requires a holistic approach that considers the operational environment in the drone system design to cope with a major limitation; the drone energy budget. In this work, we are advancing our previous work in the domain of autonomous inspection and recharging from AC powerlines [1] [2] [3]. We present technology for enabling autonomous recharging from the widespread DC railway network. Currently, the majority of railway inspection operations are done manually and can not cope with the increased number of defects that are associated with the intensive heatwaves that hit Europe in the last few years and compromise railway safety [4].

Our drone design combines the latest sensors and onboard computing technologies, a novel design of a light-weight DC charging circuit, and an advanced and intelligent mechanical system to grasp on two wires (i.e., active and ground wires) individually. The developed technologies have been tested in simulation and in the lab environment to validate the sub-components' functionalities. Fig. 1 shows a conceptual view of the drone connecting two DC powerlines with an onboard cable during a two-stage landing maneuver.

The developed code is supplied as open-source [5]. The contributions of the work are:

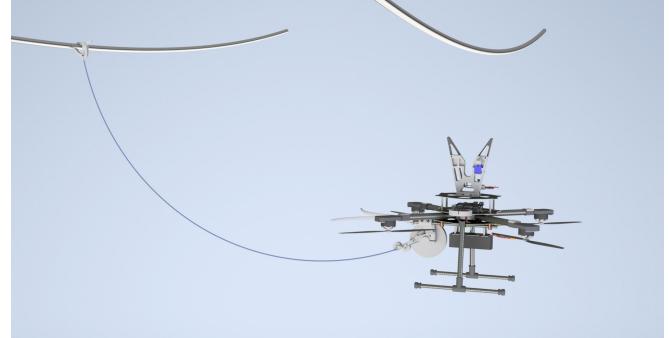


Fig. 1: Conceptual view of the UAV performing a two-stage landing to recharge from the energized railway powerlines.

- Autonomous ROS2-based UAV system for performing two-step landing on lines with onboard processing;
- Light-weight DC charging circuit for harvesting energy from < 3 kV DC railway lines;
- And a novel mechanical design to grasp and connect the two charging terminal points together.

The rest of the paper is structured as follows: Sec. II briefly summarizes related work; Sec. III outlines the applied methodology; Sec. IV describes the mechanical design; Sec. V explains the computational architecture; Sec. VI outlines the charging electronic system; Sec. VII summarizes the system validation efforts; and Sec. VIII concludes the work.

## II. RELATED WORK

To the best of our knowledge, no previous work on autonomously recharging drones from DC lines exists. However, a mechanical design that may be feasible is presented by Boaz Ben-Moshe [6]. The work's first proposal is to make the drone wide enough to simultaneously land on both wires. The second proposal is a tall stick with a hook at the end that is used to catch the first powerline and hang the drone from it. Then, from the hook, a secondary arm is extended towards the parallel second powerline to create contact. While the paper focuses on 100-250 V AC, the same mechanisms may be used for DC as well. Downsides of the proposed designs include the large physical size of the mechanisms on the drones as well as the need for the powerlines to be relatively parallel and at the same height. Therefore, we propose a solution involving two individual powerline landings to deploy a contact wire and a motorized wire drum to automatically adjust wire slack. Previous work has been done on flying wire drums and Augugliaro et al. [7] [8] show how drones carrying wire drums can be

<sup>1</sup> Drone Infrastructure Inspection and Interaction (DIII), University of Southern Denmark (SDU), [ffn@mmti.sdu.dk](mailto:ffn@mmti.sdu.dk)

<sup>2</sup> Fraunhofer IMS

<sup>3</sup> Fraunhofer IISB

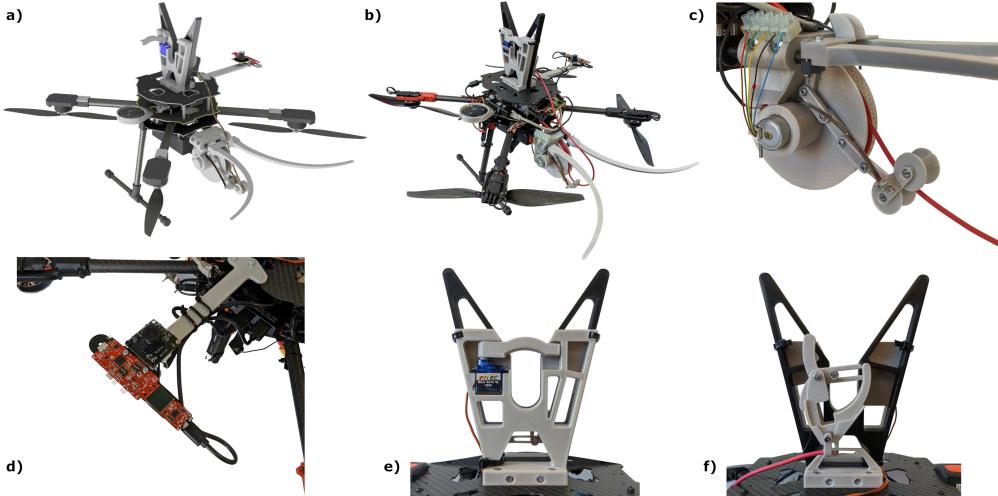


Fig. 2: Overview of the mechanical design. **a)** CAD model of the system. **b)** The real-world drone. **c)** Close-up of the motorized cable drum. The thin ribbon near the four-bar linkage is the flex sensor. **d)** The two main sensors of the system; IWR6843 mmWave radar sensor in red and AR0144 USB camera in black. **e)** The primary gripper. **f)** The secondary gripper.

used to create intricate tensile structures autonomously by weaving the wires in specific patterns. However, these drums are passive and rely on the movement of the drone to unroll wire which simplifies the mechanical problem greatly.

### III. METHODOLOGY

A consideration for inspection drones is to achieve longer flight times in order to operate efficiently. Since increasing the battery capacity will result in additional weight, it is desired to use overhead powerlines as an energy source. It was shown in [2], that the drone can be recharged at AC lines by an inductive harvester. Here the objective is to use DC railway lines for recharging, because DC networks are very widespread in railway supply. While the harvester's electrical and mechanical components add weight to the drone, they allow for significant increase in flight time.

DC power supply for railways is widely used in the field of short-distance trains in many European countries. Typical voltages are 750 V, 1500 V and 3000 V. The relatively low voltages compared to AC power lines allow direct contacting of the line. In this case, the drone first connects electrically to the ground by flying to a suitable point, preferably a ground line, and attaching the secondary gripper. The drone then flies to a high voltage-carrying point and lands there, attaching with the primary gripper. In doing so, it remains connected to the ground-potential line via a cable. The full voltage of the line is now available for the harvester electronics.

### IV. MECHANICAL DESIGN

The mechanical system aids in the capture and grasping of overhead lines and it enables the drone to connect a wire to a powerline. It consists of the motorized cable drum and the powerline capture guide with two grippers. For an overview of the physical system, see Fig. 2.

#### A. Powerline Grasping

To compensate for inaccuracies in the perception system, the drone has a powerline guide to ease the capture of the

line. In Fig. 2 this guide is seen as the black V-shape on top of the drone. On the left side of the powerline guide the primary gripper is located, see Fig. 2 e. The purpose of this gripper is to be able to grasp around a line inside the powerline guide and to support the weight of the entire drone. To close the gripper, a single servo motor rotates a bar across the cable guide opening to trap the powerline. The bar rests against two notches which means that the weight of the drone is carried by the gripper structure and not by the servo motor.

The secondary gripper is located on the other side of the powerline cable guide, see Fig. 2 f. This gripper is smaller and relies completely on the relative motion between the drone and the powerline for its actuation. The line entering the guide drives the gripper's actuation. As the powerline enters deeper into the guide, a lever is pushed down which closes a bar over the cable. A weight inside the lever falls into a notch when the gripper is fully actuated which prevents it from being opened.

The wire from the motorized cable drum is connected to this gripper. When a powerline is caught in the secondary gripper, and the drone pulls away from the powerline, the gripper disconnects from the drone and remains on the powerline. However, the cable remains attached to both gripper and drum, which means that the length of the cable must be constantly adjusted as the drone flies to regulate the cable slack. This is done using the motorized cable drum.

#### B. Motorized Cable Drum

The main purpose of the motorized cable drum is to keep the free-hanging cable in an optimal state; slightly sagged. It consists of a drum holding the cable, a DC motor to turn the drum, a flex sensor to sense the amount of slack in the cable, a four-bar linkage with rollers at the end to link the cable and the flex sensor, propeller guards, and a mounting interface to attach it to the drone, as seen in Fig. 2 c.

The drum holds approximately 6 meters of C1321 cable.

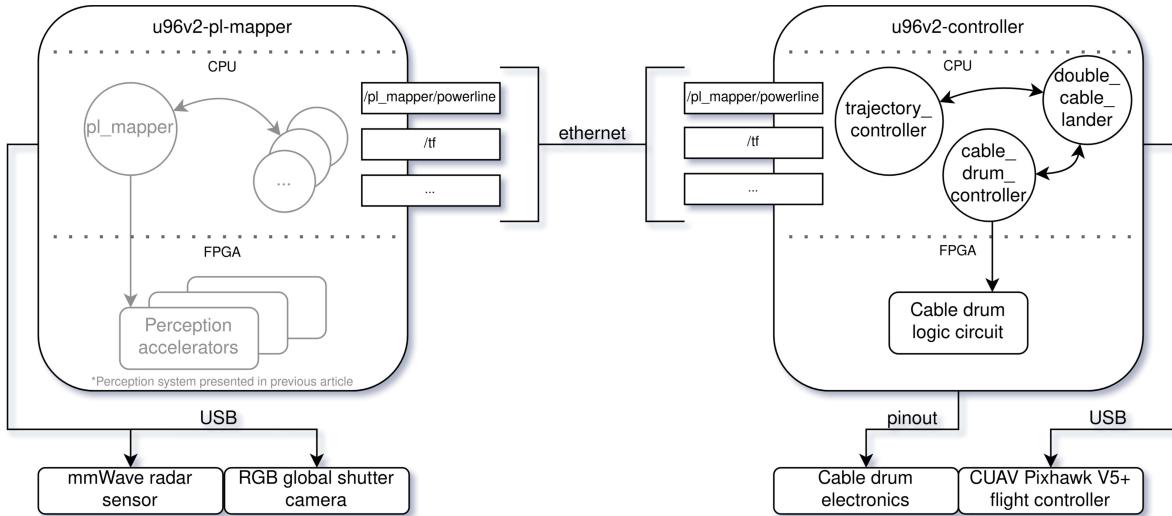


Fig. 3: Diagram of the computational architecture; the perception system (in gray color) was proposed in [3]

The cable is 3.68 mm in diameter with 1.14 mm of insulation for a 5 kV working voltage. As the cable exits the drum, it goes through a set of low-friction rollers connected to an arm to which the flex sensor is also connected. As the cable tautness changes, the arm moves up and down which bends the flex sensor differently. The amount of flex is read by a microcontroller and the onboard computer controls direction and speed of rotation of the drum to compensate for the measured slack. The motor that turns the drum is a FIT0441 12V DC motor from DFRobot with 2.4 kg·cm torque and 159 rpm maximum speed with a built-in motor driver. A 4.5" Spectra Symbol flex sensor measures the cable slack.

## V. COMPUTATIONAL ARCHITECTURE

The computational system built for the application employs a distributed architecture as visualized in Fig. 3. The computational hardware utilized in the application are two instances of the Ultra96-V2 [9]. The Ultra96-V2 features an Multi Processing System-on-Chip (MPSoC) which contains a CPU and FPGA fabric. The images for the Ultra96-V2s are built with the MPSOC4Drones-tool presented previously [10], yielding Ubuntu 20.04-based operating systems (OSs) with ROS2 Foxy and microRTPS interface to an external PX4 flight controller.

Previously, we have proposed a perception system for pose estimation of individual powerlines [3]. The perception system utilizes a mmWave radar sensor and an RGB camera, both pointing upwards, and features a processing scheme based on ROS2 running on the CPU of the Ultra96-V2 with hardware acceleration implemented in the FPGA. For the application proposed in this work, this perception system is running on one Ultra96-V2, denoted `u96v2-pl-mapper` in Fig. 3. The output of the perception system is the poses and uniquely assigned IDs of the detected powerlines.

The autonomy is implemented on a second Ultra96-V2, denoted `u96v2-controller` in Fig. 3 using three nodes; the `trajectory_controller` node exposes a set of ROS2 action servers for performing flight maneuvers such as cable landings and takeoffs; the

`cable_drum_controller` node exposes ROS2 services to set the parameters of the cable drum mechanism; and the `double_cable_lander` exposes a ROS2 action server performing the autonomous double cable landing. Additionally, the `u96v2-controller` communicates with PX4 and publishes transforms to the `/tf` topic.

Utilizing the inherent networking features of ROS2, a clean interface between the `u96v2-pl-mapper` and the `u96v2-controller` is achieved using an ethernet connection, thus allowing nodes to subscribe to topics across the two Ultra96-V2s.

### A. Trajectory Planning and Tracking

When approaching and landing on a line from a position beneath the line, the drone must fly upwards until the line enters the designated guiding mechanism, as previously explained. In order to do this in a precise fashion, a linear model predictive control (MPC) formulation is utilized, at each timestep  $t$  solving the discrete optimal control problem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{u}} \sum_{k=0}^{N-1} & (||\mathbf{x}_{t+k} - \mathbf{r}||_Q^2 + ||\mathbf{u}_{t+k}||_R^2) \quad \text{s.t.} \\ & \mathbf{x}_t = \mathbf{x}(t) \\ & \mathbf{x}_{i+1} = \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}_i \\ & \mathbf{x}_{\min} \leq \mathbf{x}_i \leq \mathbf{x}_{\max} \\ & \mathbf{u}_{\min} \leq \mathbf{u}_i \leq \mathbf{u}_{\max} \quad \forall i, \end{aligned} \quad (1)$$

where  $\mathbf{x}_i = [x_i \ y_i \ z_i \ \dot{x}_i \ \dot{y}_i \ \dot{z}_i]^T$  and  $\mathbf{u}_i = [\ddot{x}_i \ \ddot{y}_i \ \ddot{z}_i]^T$  are the translational state of the drone and the translational acceleration, considered in the formulation as the control input,  $\mathbf{r} = [r_x \ r_y \ r_z \ 0 \ 0 \ 0]^T$  is the state reference point with  $r_x$ ,  $r_y$ , and  $r_z$  being the desired position of the drone,

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \Delta t \cdot \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}_{(3 \times 3)} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0.5 \cdot (\Delta t)^2 \cdot \mathbf{I} \\ \mathbf{0} \end{bmatrix}_{(3 \times 3)}$$

are the state transition matrices originating from the laws of translational motion,  $\mathbf{I}$  is the identity matrix,  $\mathbf{0}$  is the

zero matrix,  $\mathbf{x}_{\min}$ ,  $\mathbf{x}_{\max}$ ,  $\mathbf{u}_{\min}$ , and  $\mathbf{u}_{\max}$  are the state and control bounds,  $\mathbf{Q}$  and  $\mathbf{R}$  are the weighting matrices,  $\Delta t$  is the timestep period, and  $N$  is the prediction horizon.

The MPC was designed in MATLAB using the MPC Toolbox [11]. In MATLAB simulation, the desired convergence properties of the controller was obtained by empirical tuning of the weights. Additionally, decent run-time performance was achieved with  $N = 20$  and  $\Delta t = 0.2$  s. Using MATLAB Coder [12], stand-alone C++ code was generated from the MPC formulation.

The ROS2 node `trajectory_controller` was implemented featuring a state machine callback running with a period of  $\Delta t$ . The state machine would monitor the flight mode from the microRTPS topics from PX4 and change its state accordingly.

The node exposes three ROS2 action servers to trigger its functionalities; `FlyToPosition`, `CableLanding`, and `CableTakeoff`. Each action callback communicates to the state machine, and the state machine rejects or accepts the action requests depending on its state.

All three actions call the MPC function for obtaining the next trajectory set-point. For the `FlyToPosition` action, the target position is received along with the action goal request, whereas with the `CableLanding` action only the ID of the target line is received. Throughout the cable landing procedure, the target position is updated based on the most recent pose estimate obtained from the perception system.

### B. Cable Drum Mechanism

The cable drum mechanism comprises external electronics, logic circuitry implemented in the FPGA of the `u96v2-controller`, Fig. 3, and a ROS2 node controlling the behavior of the mechanism through a memory interface to the logic circuit while exposing a set of ROS2 services to access the functionality.

The flex sensor is read via the internal ADC of an Adafruit Pro Trinket [13] which then exposes the 4-bit readings to the pinout of the `u96v2-controller` in parallel. Additional electronics map between the logic levels of the Ultra96-V2, the Pro Trinket, and the PWM input of the DC motor.

The logic circuit reads from a mode-select register; in manual mode, the PWM duty cycle and desired roll direction is read directly from a register within the FPGA and asserted on the output, whereas in reference following-mode, the duty cycle and roll direction is determined automatically based on the flex sensor readings. This is done using a P-controller which tracks a reference flex value. The reference and controller gain is read from an internal register. Finally in off-mode, the output PWM signal is always zero.

The control registers are exposed to the CPU using Advanced eXtensible Interface (AXI) along with a register containing the most recent flex sensor reading. A ROS2 node `cable_drum_controller`, Fig. 3, exposes ROS2 services for setting the cable drum mechanism P-controller gain, flex reference, and mode. Additionally, it exposes a ROS2 action server `DrumManualRoll` for rolling the cable drum in or out with a given PWM duty cycle for a set

amount of time. Finally, the node continuously monitors the flex value and publishes it as a ROS2 topic. Underneath, the node reads and writes to the designated addresses in memory through a userspace I/O device file which maps to the registers exposed with AXI.

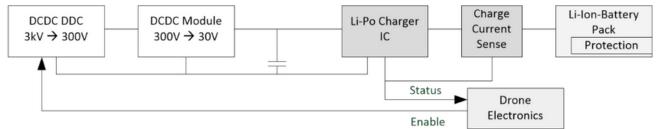


Fig. 4: Overview of the HV-DC/DC drone charging system

### C. Mission Autonomy

Handling the actual procedure of landing on the two lines while operating the cable drum, a ROS2 node `double_cable_lander`, Fig. 3, is implemented, exposing a ROS2 action server for performing the full mission. In the action request, the IDs of the first and second line are passed. The node then operates the following procedure with calls to actions and services of the two other nodes:

- 1) `FlyToPosition` with target under first line;
- 2) `CableLanding` with the first line as target line;
- 3) `DrumManualRoll` to initiate rolling out the cable drum, and `CableTakeoff` with a target position underneath the cable;
- 4) Set the cable drum mode to reference tracking through a service call to `cable_drum_controller`;
- 5) `FlyToPosition` with target under second line;
- 6) `CableLanding` with the second line as target line.

After execution of the action sequence, the procedure is successful. The procedure is aborted if any action fails.

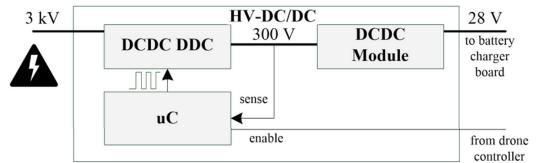


Fig. 5: Simplified block diagram of the HV-DC/DC-converter

## VI. DC CHARGING SYSTEM

In direct down conversion (DDC), the voltage is transformed into a low voltage, as in a switched-mode power supply. It was decided to develop a two-stage converter. The first stage converts the voltage from 3 kV to 300 V and the second stage converts from 300 V to the required voltage of the charge controller of 30 V. This makes it easy to adapt the solution to the common railway DC voltages in the first stage and use a market-available component as the second stage. Fig. 4 shows the complete chain of the harvester electronics from the high voltage connection, the two stage down conversion, the charger circuit and the battery pack.

### A. Circuit Design

A buck converter was selected as the architecture of the converter after considering the characteristics and simulating commonly used circuits. Here, both the energy efficiency and the weight of the circuit determined the selection, since

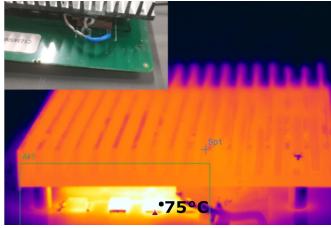


Fig. 6: Thermal characterization of the buck converter

relatively small coils and capacitors can be used. Fig. 10 shows the schematic of the basic circuit. The simulation results showed coverage of a wide operating range up to the specified output power of 150 W and a reasonably low output voltage ripple of less than 1 V. To ensure

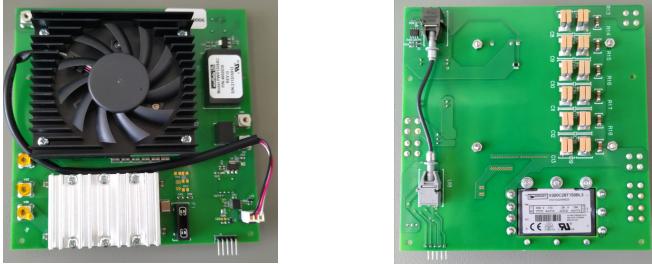


Fig. 7: Assembled PCB front and back side

sufficient dielectric strength, reasonable size and weight and to distribute the resulting power dissipation, both the shunt diode and the inductor were realized from a series connection of several similar components. A central component is the switch, which is typically cascaded for high voltages. For this operating case, a suitable switch could be found that can be used as a single element. The switch has a dielectric strength of 4.5 kV. In order to be able to use the HV-DC/DC



Fig. 8: **Left:** Primary gripper in open position. **Right:** Primary gripper in closed position supporting the drone.

controller with a variable load (here the battery charge controller), a feedback loop with a load-dependent control had to be added to the design, Fig. 5. This was realized with a microcontroller, which regulates the switching pulses depending on the generated output voltage (input voltage of the next stage; the DC/DC module). If the voltage rises above a threshold, the microcontroller reduces the output power and therefore, the output voltage is limited. The voltage threshold has been selected with respect to the upper input voltage limit of the DC/DC module and the timing of the microcontroller.

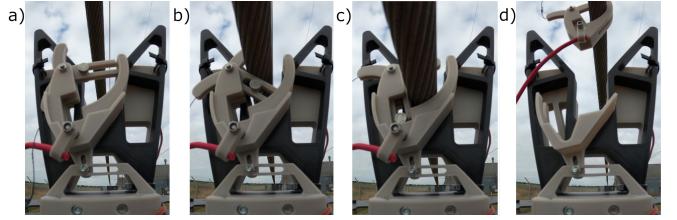


Fig. 9: **a-d)** Actuation and latching of secondary gripper.

### B. Thermal Design

Before designing the layout, a test circuit was implemented to investigate the heat generation and to design an optimal heat sink. For this purpose, the components were connected in a direct wiring and put into operation in a high-voltage laboratory. This served both to test the function and to investigate the heating. Fig. 6 shows the setup of the test circuitry of the thermal imaging for the assembled PCB.

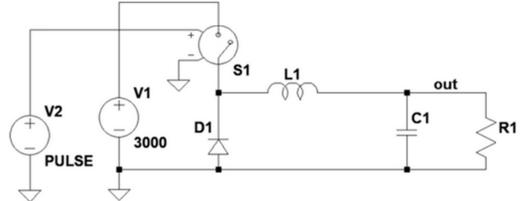


Fig. 10: Schematic of buck converter for simulation

### C. PCB Realization

The realized circuit of buck converter together with the feedback control loop and the second stage, an off the shelf DC/DC converter (300 V to 28 V) is shown in Fig. 7. For galvanic isolation the controller is controlled by an optical fiber and the switch as well as inductor of the buck converter are covered by a heat sink.

## VII. EXPERIMENTAL RESULTS

Tests were conducted to validate the implemented components of the system using both simulation and lab testing.

### A. Gripper Powerline Grasping

The primary gripper has been tested by simulating a powerline cable landing. During this maneuver the drone is lifted such that the powerline enters the powerline cable guide. Once in, the gripper's servo motor is actuated to close the gripper which then holds the drone, see Fig. 8. The weight of the drone is 3220 g. A takeoff from the powerline is then simulated by lifting the drone, opening the gripper with the servo motor, and moving the drone away from the powerline. On the left the gripper is in an open state ready to grasp the powerline, and on the right the gripper is in a closed state while hanging on the powerline. The powerline cable used for this test is 20mm in diameter.

The secondary gripper was also tested on a powerline. The drone is lifted to let a line enter the guide, and while doing so the powerline actuates and latches the gripping mechanism. A simulated disengagement from the powerline cable moves the drone away from the powerline while the secondary gripper and the drum cable is still connected to it. Fig. 9 shows this test.



Fig. 11: **Left:** The drum unrolls more cable to increase slack. **Right:** The drum rolls the cable in to reduce the slack.

### B. Automatic Cable Slack Adjustment

The automatic cable slack adjustment functionality was validated by manually manipulating the cable to simulate when the drone is in flight. Pulling the cable more taut raises the linkage and roller assembly and thereby lessens the bend in the flex sensor - and vice versa. Through calibration, a suitable flex reference is found. Fig. 11 on the left shows a situation where wire is fed to combat tautness, and on the right the drum rolls wire back in to combat excessive slack.

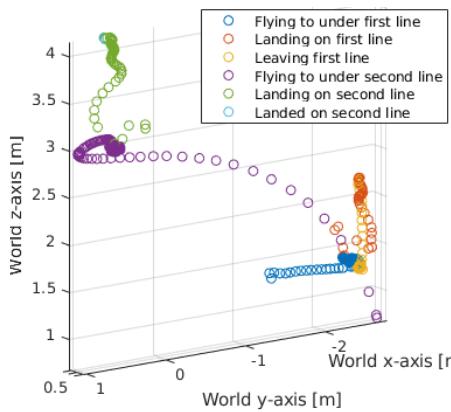


Fig. 12: Trajectory of the drone while performing the double cable landing during HITL simulation

### C. Autonomy Validation

The system autonomy is validated using hardware-in-the-loop (HITL) simulation. Through ethernet, the computational system of Fig. 3 is connected to a desktop running Gazebo simulation. A Gazebo world containing a powerline setup is utilized for the validation. Running the full computational system but sensors and flight controller replaced with corresponding parts in the simulation, the `double_cable_lander` node is instructed to perform the full autonomous mission on two cables. The logged trajectory of the drone is shown in Fig. 12.

From the flown trajectory, it is clearly seen that the node progresses through the states and that the drone is landed on the lines from beneath in an upwards direction. It is thus validated that the proposed processing implementations can run on the hardware platform.

### D. HV-DC/DC Converter Switch-On Test

The switch-on behavior of the first and second stages combine with a static and dynamic load at the output was successfully tested. Fig. 13 shows the response with a dynamic load. The 3 kV are present at the input when the conversion is initiated by a 5 V trigger signal. It can be

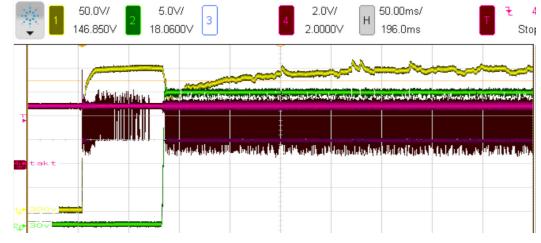


Fig. 13: Test results with HV-DC/DC system and the charger board as dynamic load

observed that the output voltage of the first stage (yellow, 1) is limited a threshold. The second stage output voltage (green, 2) has some delay during power up. The pulses (magenta/black, 4) stop when the threshold is reached and are continued when the voltage drops below a second threshold.

### E. DC Energy Harvester Test

The results achieved in the test of the entire energy harvester chain, Fig. 14, are shown in Tab. I. These results show that the requirements are met. The circuit operates at the full high voltage and regulates input and load changes so that the charging current is within the intended range. Likewise, the harvester ramp-up and -down functions, which were tested by switching the ramp-up circuit on and off.

$V_{in}$	3.0 kV	Full input voltage were applied
$I_{in}$	0.066 A	Input current from high voltage source
$V_{out} (V_{bat})$	22.3 V	Output voltage of the charger at set current
$I_{out}$	6.0 A	Set charging current

TABLE I: HV-DC/DC energy harvester test results

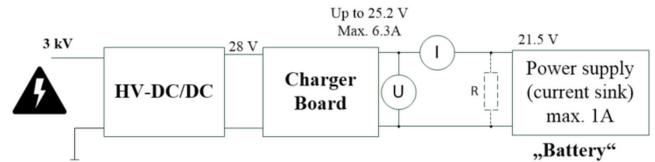


Fig. 14: Energy harvester laboratory test setup

## VIII. CONCLUSION

In this work, we have presented technologies for enabling drones to autonomously attach to and recharge from existing DC railway infrastructure. This includes a DC down-conversion circuit, an autonomous cable drum mechanism, and a computational processing architecture. We have validated in lab test and simulation that the implementations show potential for real-world applications.

In future work, we will perform physical outdoor testing at a real railway DC setup as well as improve the efficiency of the implemented system.

## ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 861111, Drones4Safety.

## REFERENCES

- [1] N. Iversen, A. Kramberger, O. B. Schofield, and E. Ebeid, "Pneumatic-mechanical systems in uavs: Autonomous power line sensor unit deployment," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 548–554.
- [2] N. Iversen, O. Schofield, L. Cousin, N. Ayoub, G. Vom Bögel, and E. Ebeid, "Design, integration and implementation of an intelligent and self-recharging drone system for autonomous power line inspection," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Dec. 2021, pp. 4168–4175.
- [3] N. H. Malle, F. F. Nyboe, and E. Ebeid. Onboard powerline perception system for uavs using mmwave radar and fpga-accelerated vision (pre-print). Pre-print, submitted to IEEE Access, under review. [Online]. Available: [https://github.com/DIII-SDU-Group/Onboard-Powerline-Perception-System/blob/main/IEEE\\_Access\\_2022\\_\\_Powerline\\_Perception\\_System.pdf](https://github.com/DIII-SDU-Group/Onboard-Powerline-Perception-System/blob/main/IEEE_Access_2022__Powerline_Perception_System.pdf)
- [4] theconversation. Why uk railways can't deal with heatwaves – and what might help. Visited on 14/09/2022. [Online]. Available: <https://theconversation.com/why-uk-railways-cant-deal-with-heatwaves-and-what-might-help-187311>
- [5] DIII. Github for ros2 package source code. Visited on 15/09/2022. [Online]. Available: <https://github.com/DIII-SDU-Group/CableLandingController-ROS2-pkg>
- [6] B. Ben-Moshe, "Power line charging mechanism for drones," *Drones*, vol. 5, 2021. [Online]. Available: <https://www.mdpi.com/2504-446X/5/4/108>
- [7] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D'Andrea, "Building tensile structures with flying machines," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3487–3492.
- [8] F. Augugliaro, E. Zarfati, A. Mirjan, and R. D'Andrea, "Knot-tying with flying machines for aerial construction," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5917–5922.
- [9] Avnet. Ultra96-v2 avnet. Visited on 14/09/2022. [Online]. Available: <https://www.avnet.com/wps/portal/us/products/new-product-introductions/npi/aes-ultra96-v2/>
- [10] F. F. Nyboe, N. H. Malle, and E. Ebeid, "Mpsoc4drones: An open framework for ros2, px4, and fpga integration," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022.
- [11] MathWorks. Matlab mpc toolbox product page. Visited on 14/09/2022. [Online]. Available: <https://se.mathworks.com/products/model-predictive-control.html>
- [12] ———. Matlab coder product page. Visited on 14/09/2022. [Online]. Available: <https://se.mathworks.com/products/matlab-coder.html>
- [13] Adafruit. Adafruit pro trinket product page. Visited on 14/09/2022. [Online]. Available: <https://www.adafruit.com/product/2000>