CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

Towards improved Validation of Autonomous Systems for Smart Farming

Martijn Rooker
Pablo Horstrand
Aythami Salvador Rodriguez
Sebastian Lopez
Roberto Sarmiento
Jose Lopez
Ray Alejandro Lattarulo
Joshue Manuel Perez Rastelli
Zora Slavik
David Pereira*
Markku Pusenius
Tapio Leppälampi

*CISTER Research Centre
CISTER-TR-180302

2018/04/10

Towards improved Validation of Autonomous Systems for Smart Farming

Martijn Rooker, Pablo Horstrand, Aythami Salvador Rodriguez, Sebastian Lopez, Roberto Sarmiento, Jose Lopez, Ray Alejandro Lattarulo, Joshue Manuel Perez Rastelli, Zora Slavik, David Pereira*, Markku Pusenius, Tapio Leppälampi

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: dmrpe@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

ENABLE-S3 is a use-case driven European researchproject focusing on the implementation and validation ofautonomous cyber-physical systems (CPS) in different applicationdomains. This work describes the efforts done so far in thedevelopment of infrastructure and tools to make improvedvalidation concepts in agriculture, being part of one of the thirteenuse cases included in the project. Aspects related tocommunication, autonomous vehicles, hyperspectral images,image processing, Unmanned Aerial Vehicles (UAVs), andsimulation environments are described. The combination andinteraction of these key technologies give rise to social, economicand environmental implications with enormous benefits,increasing the quality of the crops and reducing efforts duringtheir growth and harvesting.

Towards improved Validation of Autonomous Systems for Smart Farming

Martijn Rooker
TTTech Computertechnik AG
Vienna, Austria
martijn.rooker@tttech.com

Zora Slavik
Forschungszentrum Informatik
Karlsruhe, Germany
slavik@fzi.de

Pablo Horstrand, Aythami Salvador Rodriguez, Sebastian López, Roberto Sarmiento, José López
University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
phorstrand@iuma.ulpgc.es

David Pereira
Instituto Superior de Engenharia do Portoable
Porto, Portugal
dmrpe@isep.ipp.pt

Ray Lattarulo, Joshue Perez Rastelli, José Matute
Tecnalia Research and Innovation
Bilbao, Spain
rayalejandro.lattarulo@tecnalia.com

Markku Pusenius, Tapio Leppälampi
Creanex Oy
Tampere, Finland
markku.pusenius@creanex.fi

***Abstract*—ENABLE-S3 is a use-case driven European research project focusing on the implementation and validation of autonomous cyber-physical systems (CPS) in different application domains. This work describes the efforts done so far in the development of infrastructure and tools to make improved validation concepts in agriculture, being part of one of the thirteen use cases included in the project. Aspects related to communication, autonomous vehicles, hyperspectral images, image processing, Unmanned Aerial Vehicles (UAVs), and simulation environments are described. The combination and interaction of these key technologies give rise to social, economic and environmental implications with enormous benefits, increasing the quality of the crops and reducing efforts during their growth and harvesting.** (*Abstract*)

Keywords—autonomous systems, farming, UAVs, hyperspectral, GPS, simulation, verification, validation

I. Introduction

The ENABLE-S3 project (European Initiative to Enable Validation for Highly Automated Safe and Secure Systems) [1] is a three years research project funded by the EU under the ECSEL-JU. Started in May 2016 with a partnership of 72 industrial and academic institutions of 15 different countries. With a budget of approx. 65M€, it focuses on accelerating highly automated and autonomous systems in six domains: automotive, aerospace, rail, maritime, health care and farming. Each domain encompasses different use-cases, with special emphasis on improving the way in which these systems can be simulated, validated and verified, as a way to reduce their time-to-market.

With respect to the farming use case, the industry has successfully implemented “smart farming” features, focusing on detection of the crop’s needs and problems (e.g. fertilizer, water application and crop spraying according to the needs of individual plants, rather than treating large areas in the same manner). These features have already introduced a high level of automation and have saved millions of tons of fertilizer, pesticides, insecticides etc. According to [2], “there are 10.7 million farm holdings in the EU with an average area of 16 hectares. The average for individual Member States ranges from 1.2 hectares in Malta to 133 hectares in the Czech Republic.”

The aim is to define new V&V methodologies, based on existing ones and a scenario-based approach to test and validate automated CPS at reasonable costs. There are three main pillars in which the farming use case is sustained: sensing, processing and acting. The first one is devoted to the monitoring of the crop growth in order to quantify biochemical and biophysical attributes. In parallel, this research also deals with other critical actions such as feasible and robust communication systems between vehicles, a system that detects persons and/or animals in front of the harvester in order to avoid fatal accidents, and a simulation environment to accelerate the system testing and validation of the different introduced technologies. Summing up, the goal is to transfer systems, sensors and results from a much more mature technology, such as in the automotive domain, and adapt it to the agriculture sector, where possible. Doing that, important differences, such as lack of driving signals, need to be addressed as well as the introduction of other sensor types.

For a realistic V&V methodology, recorded data from real scenes was used, where available, e.g. camera or radar perception. Virtual environments are provided by tools like CarMaker®[3], VTD [4] or Dynacar [5]. Connecting the virtual tools requires appropriate modelling and dependable (real-time) simulation of sensors, controllers, actuators, data fusion, human interaction, weather conditions, etc. Co-simulation is used to integrate multiple simulation tools to foster modularity, simulation scalability, tool independence and cross-domain interoperability. Many co-simulation tools are already available on the market [6][7], however, real-time co-simulation is not established yet. Finally, runtime validation is required as real-world scenarios and continuously changing environment can leave a gap at deploy time of the system. These validations are sometimes also called observers, which can be generated from formal specifications in order to monitor desired conditions of the system. Observers can detect internal conditions, they are typically used to assert that the components are used in the defined context. An example are control strategies that are based on predictive dynamic models, which are used to estimate the effect of different actions on the future evolution of the environmental situations [8].

In this paper, we first explain the different scenarios of the harvesting process. Section III introduces the different Systems under Test (SUT). Section IV describes the first test systems (TSY) developed for the various SUTs consisting of combined simulation tools, sensor and communication systems, and finally, section V deals with preliminary validation and verification of the system. This work finalizes with some conclusions and future work.

II. Scenario-Based Validation and Verification

The basic approach of the validation contains scenarios, which represent a domain/use-case specific typical or critical combination of actions/events and settings of both the entity to be validated as well as the respective environment. The validation technology developed within this use case is applied to three different scenarios that together form the whole harvesting process including the three pillars.

A. *Scenario 1: Inspection Flight*

During the inspection flight, a drone equipped with a hyperspectral sensor identifies the status of the crops on the field. The drone flies autonomously over a specific area in order to determine the quality of the crops and the areas to be harvested. After scanning the whole field, the drone returns to its starting point and communicates the gathered information to the harvester. This can be accelerated by transmitting the data during the flight time with the use of some electronics included in the drone (mainly a GPU).

B. *Scenario 2: Automated Harvesting*

The main part of the demonstration is in the second scenario that tackles the concept of automated harvesting. The harvester receives the processed data from the drone (from scenario 1) and builds a trajectory based on all the information (size of field, waypoints, maturity levels and health of the crops, etc.). The vehicle drives autonomously over the field and performs the "harvesting" process, covering the identified harvest area. The drone, equipped with a hyperspectral camera and a thermal sensor, accompanies the harvester some meters ahead of the vehicle observing the area in front of it and automatically scouts for animals or persons in reduced visibility conditions. In addition, a radar sensor on the harvester supports the object detection in adverse environment conditions.

C. *Scenario 3: Grain Collection*

The final scenario is the grain collection. After a certain amount of time, the harvester needs to be emptied. Instead of driving back to the farm, a second autonomous vehicle, the tractor (with simulated trailer), moves autonomously towards the harvester and performs a parallel driving maneuver next to it, so that the harvester is able to unload its tank to the tractor. During this process, the harvester continues its task of harvesting. After finalizing the unloading, the tractor returns to the farm and the harvester continues on its track. During this maneuver, the drone is all the time in front of both vehicles, supervising the scenario.

III. Systems Under Test

For the validation process within the ENABLE-S3 project, the concept of System under Test (SUT) and Test Systems (TSY, see section IV) was established. The TSY will be used to perform improved testing on the SUTs. This section describes the SUTs within the farming use case.

A. *Drone*

One of the key elements of the demonstrator in this use-case is an Unmanned Aerial Vehicle (UAV), or drone. For the work within the use case, a Dji hexacopter Matrice 600 drone [9] is applied. Although this drone exceeds the needs of an application like the one presented in this work, the aim behind its application is to incorporate different sensors (hyperspectral, RGB camera, thermal sensor) in order to evaluate their performance and to make use of fusion techniques to extract better information.

In order to optimize the flight of the drone and deal with the limitations imposed by its battery restrictions, an application has been developed including a Graphical User Interface (GUI). Initially, two coordinates are fixed in a map and the application adds two others in order to delimit a rectangle. Additional waypoints are then introduced automatically in the borders of the rectangle so that the drone explores the whole area in a zig-zag fashion. Moreover, the application determines the speed and height of flight according to the battery level in order to obtain a good spatial resolution but at the same time being able to sweep the whole rectangle. Fig 1 shows a screenshot of the application with some input variables that are introduced by the user.

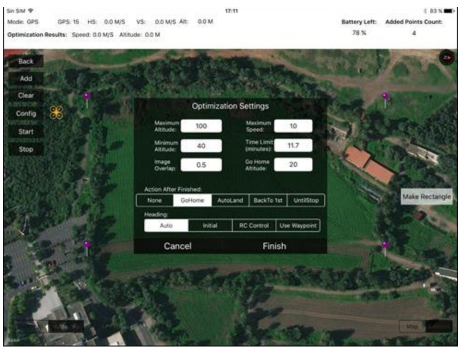

Fig 1: Screenshot of the application developed for the optimization of the flight of the drone

This application is part of a validation tool that permits simulations of the flight of the drone before doing real tests in the field, reducing time and effort.

A Specim FX10 hyperspectral camera [10] is included as payload with the drone. Most of the applications in precision agriculture make use of multispectral cameras, which are much cheaper and easy to use. However, the FX10 hyperspectral camera allows us to obtain a vast amount of data that gives a rich variety of information in order to make correct decisions in the future such as determining the most valuable wavelengths depending on the type of crops.

Fig 2: On the left, the Dji Matrice 600 in the laboratories with the RONIN-MX gimbal and the hyperspectral Specim FX10. On the right, a detail of the NVIDIA Jetson TK1 GPI on top of the camera.

Hyperspectral imaging is a powerful technique combining imaging and spectroscopy to survey a scene, extracting detailed information related to its chemical composition based on their reflected light in different narrow wavelengths or bands invisible to the human eye [11][12]. Every material has its own spectral signature, when compared with well-known data can be clearly identified. Similarly, the spectral signature of a portion of the crop gives information related to its maturity level and health as well as the humidity of the soil in the surroundings [13].

B. *Autonomous Farming Vehicles*

The platforms used for validation of the farming use case are two Renault Twizy (Fig 3), one for the harvester and one for the tractor. Even if these platforms are of different magnitudes in comparison with real tractors or harvesters, they are useful for the V&V of the algorithm developed as proof of concept.

Fig 3: Vehicles and sensors

The vehicles are instrumented and capable of working fully automatically. They are equipped with a Differential Global Position System (DGPS) obtaining precisely the position in the field. This technology uses the GPS and a base that is in a fix position. Additionally, this system has the incorporation of an Inertial Measurement Unit (IMU) to record inertial variables in the vehicle used for validation. Additionally, the vehicle has a four-layer LiDAR system that permits braking the system in emergencies, such as frontal obstacles. A stepper motor controls the steering wheel automatically. The brake system has a linear actuator and the throttle system is controlled with a parallel connection to the vehicle ECU. It receives the information from an onboard embedded computer that runs the automated driving software. This SW is based on MATLAB/Simulink and it integrates the six main blocks that conforms most of the automated driving architectures [14], i.e. acquisition, perception, communication, decision, control and actuation. The acquisition module has the task of collecting the data coming from vehicle sensors and odometry. The perception block uses the information of the acquisition to generate a complete description of the vehicle for the decision making and control. The communication module sends and receives information of the other participants in the scenario. The decision module is the one that generates the smooth trajectories that the tractor and the harvester will be tracking. The control generates corrections of position and speed (lateral control and longitudinal control) controlling the steering wheel, throttle and brake. The architecture is depicted in Fig 4 and explained in [14].

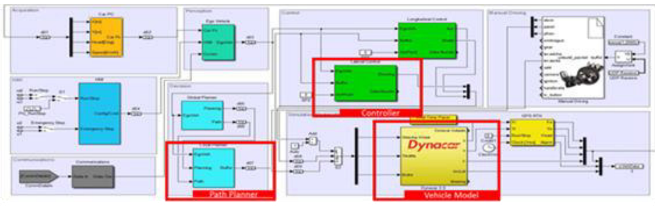

Fig 4: Harvester/Tractor control architecture (MATLAB/Simulink software)

IV. Test Systems

To perform V&V on the SUTs, different test systems (TSY) are developed, coming from different areas, like automotive, aerospace, communication technology, sensor technology, etc. Additionally to the SUTs (that are also used as test system for validation), there are other TSY applied which are described in more detail in the following subsections. Within the project, and accordingly the farming use case, one of the approaches for better validation is the inclusion of simulation tools. Within this use case, a combination of simulation tools/models are merged to improve the validation for autonomous farming systems. The SUTs already provide simulation tools to test and validate their behavior. Additionally, an environment simulation and weather models are introduced in the use case, and are described in the following section. Fig 5 shows how the different simulations are connected and what kind of data is exchanged.

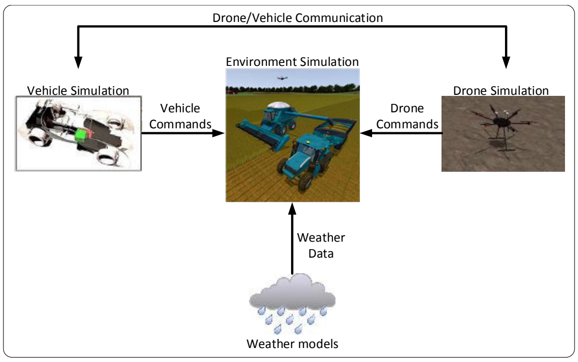

Fig 5: Test System Setup

A. *Simulation Environment*

The farming machine simulator AgSim imitates the machine operations in a virtual farming environment in real time, which is an added value compared to the virtual environments mentioned in section I. The simulated environment consists of a farming field and co-simulation of a harvester, a tractor with trailer and the drone. The simulation is supported by a 3D visual system, which visualizes simulated testing events and makes it easy to verify that the simulation runs as expected and fulfils the machine operations under test (Fig 6). The state of the simulation can be saved and then loaded as the initial state of a testing scenario. The saved scenario may contain description of variable situation and events. These are, for example, stationary or moving objects, obstacles not known by the control systems or animals running on the field. The simulated models of the machines have interfaces for external control algorithms running in Python or Matlab/Simulink. The simulation environment supports also the hardware in the loop testing setup. The I/Os of the simulated models can be connected to I/Os of real embedded control modules either using I/O boards or adapters for serial communication such as CAN or Ethernet. The python interface allows recording of the control system I/O signals and requesting sensor data during the simulation (Fig 6).

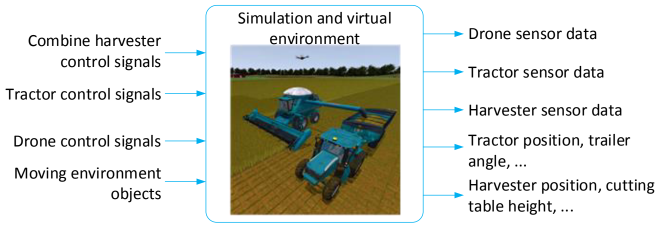

Fig 6: Farming simulation environment

The control systems of the autonomous machines must be aware of the surrounding environments. It is measured with various sensors (e.g. radar, LIDAR or camera). The simulation environment produces data for virtual sensors, measures from which actual sensor signals may be simulated. For example, the visual system generates a visible objects view, where for each object of interest a unique color is given (see Fig 7). For visible objects in the sensor view, the color identifies them and the simulation models provides object lists with attributes: a) type (material) of the objects, b) size of the objects in visibility view and c) location and velocity of the objects relative to the sensor. Based on the object list, one can simulate virtual sensor signals to be fed as an input signal to the control system.

Test cases can be written in Python. An application is developed to run the test cases fully automatically or assisted by an operator. In a virtual farming environment, events can be determined and the conditions modified for the wanted test case allowing highly variable test scenarios.

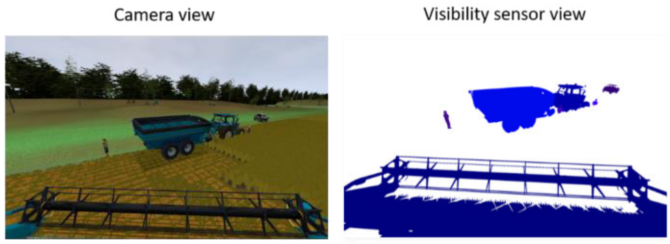

Fig 7: Simulated virtual sensor signals

B. *Weather and Environment Models for Stimuli Generation*

Weather and environment conditions occur in real testing conditions in a way that is hard to reproduce or predict, while they can have critical effects on sensor systems. In addition, the characteristics of the conditions are subject to strong variations. As autonomous driving systems such as an autonomous farming machine depend on a reliable and well-known sensor behavior, these effects need to be considered in V&V. Farming specific environment conditions lead to additional challenges for sensors that were developed for automotive road applications. The goal is to meet these challenges with a virtual stimuli generation for radar sensors in off-road or farming scenarios. As attenuation and clutter resulting from weather, particles or various off-road surfaces cannot be tackled in the same way as clutter in automotive applications, it is crucial to apply the sensor stimulation already before entering the data processing chain and at an early signal processing stage. Fig 8 shows an overview schematic of a radar signal acquisition and processing chain with an additional filter stage for weather and environment stimulation (WES). The RF frontend contains the transmit and receive antennas. In the pre-processing block, the receive signal $s_r(t)$ is mixed down into the baseband and compared to the transmit signal. The resulting low-frequency beat signal $x_b(t)$ is then sampled so that the discrete signal $x_b[k]$ can enter the digital signal processing unit. From the resulting data, high-level information can finally be derived. The signal stimulation approach targets the sampled beat signal $x_b[k]$. The parameterized filter stage WES is applied to the sampled beat signal and results in the condition dependent beat signal $x_{b,c}[k]$. The impact of the effects of the weather and environment conditions can be observed in the subsequent signal and data processing. Using a development platform for radar systems, the input of the signal processing unit is available, which is normally not true for current automotive radar systems on the market.

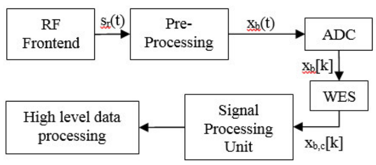

Fig 8: Block diagram of radar processing chain with application of WES

We chose a data-driven approach to develop a model for the stimulus generator using recorded data sets that cover the most important and critical scenarios in the farming use case. From the collected data, characteristic signal features are extracted for various weather and environment conditions with machine learning techniques. Test drives show that the characteristic ground and particle clutter in off-road applications plays an important role for object detection in the farming use case. For data collection, we use INRAS' Radarbook with their 77 GHz RF frontend. Additional sensors, such as a GPS sensor, a temperature and humidity sensor and a camera are used to observe the location, time and environment to support annotation and labeling. A Raspberry Pi configures and controls the radar via Wi-Fi connection. The complete measurement system was mounted in a weatherproof case allowing a flexible and autarkic operation.

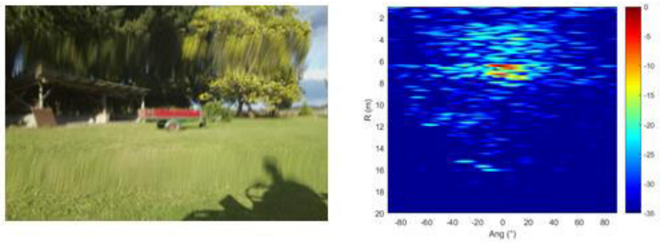

Fig 9: Off-road measurement of traler with a) Raspberry Pi camera picture, b) Range-Angle -Map from radar sensor

Fig 9 represents an example of the measurement results in dry off-road conditions, where the measurement system was mounted on a turf tractor. The camera picture on the left side is blurred due to the vibrations of the tractor. The range-azimuth-map resulting from the radar on the right side depicts the location of various power levels. The power levels are normalized to the maximum power level of the receive signal. A threshold cuts off all normalized power levels below -35 dB. The trailer can be spotted clearly at 6 m. The scattered power levels result from ground clutter and multipath reflections.

C. *Testing and Verifying Real-Time Properties*

In highly automated and autonomous systems such as the one described in this paper, the real-time properties of the various components/sub-systems are relevant to ensure full system safety. Real-time analysis and associated verification methods have been under development over several decades and allow to mathematically define and analyze scheduling and sharing of computational resources in applications and communication as well as their interaction with the external environment. From these analyses, more realistic tests can be derived and used in the global Test System and use them to feed other Test Systems that need to address real-time concerns.

Whenever possible, we will apply Runtime Verification (RV) over identified real-time properties with the aim of producing rigorous specifications in a formal language (traditionally, a temporal logic) and derive monitors from those specifications, in a correct-by-construction way. By using RV, we can complement other types of off-line testing or verifications by relying on the generated monitors when testing and stronger verification methods are seen as insufficient in terms of the overall system or individual component coverage.

RTMLD3Synth [16][17] is such a framework for specifying real-time properties based on time durations (it was already used to verify at runtime a prototype auto-pilot as reported in [15]). The framework takes as input specifications considering order and durations of events, and produces the corresponding software monitors (currently in C++ and OCaml) than can, afterwards, be coupled to the monitored application. This coupling requires an instrumentation of monitored application in order for it to capture the events of interest for the specification from which the monitors have been derived. In our developments, we will investigate how RMTLD3Synth can be integrated with more real-time analysis of intra-vehicle communications to ensure the required safety levels, and guarantee that the generated monitors will not affect the overall real-time requirements of the applications being monitored and verified during execution time. The generated monitors can be used within/connected to virtual testing components, and if working according to what they were specified to verify, they can be deployed into the final system without needs of modification, and contribute to its verification and validation.

V. First Validation Concepts

A. *Vehicle Motion*

One of the objective of the current approach is to validate the correctness and optimality of covering the cropping area with the vehicle. This area is a square form where the square corners will be given by scouting the zone with the drone that finds the patch that is ready to be cropped. After this, a trajectory that covers most of this area will be generated considering, as major factor, the physical limitations of the vehicle. The maximum curvature is the principal factor that is verified in the trajectory limitations. At low speeds, a kinematic linear bicycle model is used. The equation that relates the maximum curvature with the description of the vehicle is:

$$\tan(\alpha_{max}) = \frac{L}{R_{min}}$$

where α_{max} is the maximum rotation angle for the frontal wheel, L is the battle of the vehicle (distance between axis) and R_{min} is the mimum curvature radio that the vehicle can do over the maximum rotation of the frontal wheel. With the maximum radius of the curvature K_{max}, this is translated to

$$K_{max} = \frac{L}{tan(\alpha_{max})}$$

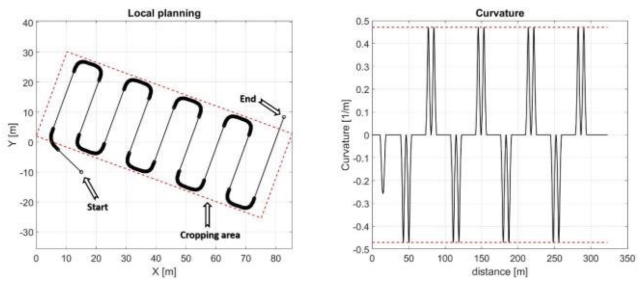

Fig 10: First approach to cover the cropping area

Fig 10 on its left shows a first approach to cover the area using parametric curves, specifically Bezier, and the way how it cover the maximum possible area without overlapping. On the right is the verification that the generated curves have a curvature under the maximum theoretical curvature.

B. *Vehicle-Drone Communication*

In first tests, the drone was able to follow the harvester while scouting the area in front of it in order to detect anomalies. A first version of the tracking system was based on image processing. However, as dust is one of the problems when dealing with agricultural environments, this type of solutions are not feasible. A new concept has been implemented using a Dji Phantom 4. We incorporated a commercial GPS into the harvester (in our case we used a remote control car as shown in Fig 11, but the final prototype will include the Renault Twizy with its own differential GPS system), and while driving, the GPS position will be sent to the base station at 100 Hz, which will be the one in charge of sending this position to the drone (using Lightbridge) in order to position the drone in the same coordinates as the harvester but some meter above.

Fig 11: Prototype for the autonomous GPS tracking system, with the drone following a remote control vehicle.

At this phase, one of the problems encountered relates to the communication channel. Harvesting fields are normally located in remote areas, and a point-to-point communication system is a good alternative. Hence, we first used LORA (Long Range Wide Area) technology 0 to send GPS coordinates from the vehicle to the drone, allowing long communication distances with low power consumption. Different tests were performed to verify the attenuation in the quality of the signal as vehicle and base station are separated from each other. Fig 12 shows the results obtained for a maximum separation of 1.4 km (which corresponds to the maximum of the curve in blue) and the quality of the signal (in orange). It has to be mentioned that all the values obtained for the quality of the signal are above -100dBm which is an acceptable signal intensity here.

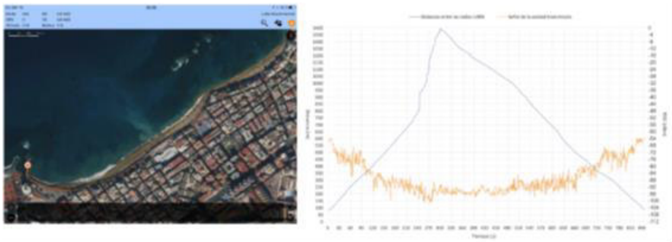

Fig 12: Verification of the signal integrity transmitted from the vehicle to the base station using LORA

Accuracy was another parameter to be tested. For this purpose, we used a vehicle driving over the lines in a football field, and obtained results for the separation distance between the GPS and the base station, as well as attenuation of the signal with the distance, but also the accuracy between the drone and the vehicle to be tracked, which are represented on the left side of Fig 13, with the drone in red color and the vehicle in yellow. As it can be seen, both lines match, which means that the drone is following the vehicle in its vertical.

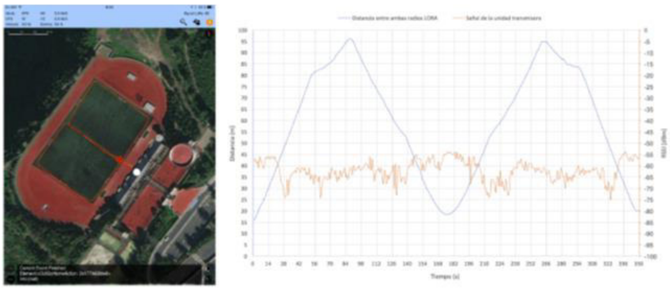

Fig 13: Accuracy of the autonomous GPS tracking system

Although these preliminary experiments show good results in terms of communication quality and accuracy, in the final prototype the differential GPS included in the Renault Twizy will substitute the commercial GPS while LORA will be substituted by a 4G device in order to establish a direct communication channel between the harvester and the drone.

VI. Conclusion and Future Work

The agriculture domain offers a great number of challenges but at the same time an enormous amount of opportunities to develop new technological ideas that could be shared with other domains. The ENABLE-S3 project encourages a transfer of ideas between the automotive and the farming world, promoting interactions and synergies due to the similarities, but at the same time taking into account their differences. In this paper a description of the work done so far in the farming domain is presented, exploring the use of mature technologies combined with more sophisticated ones in order to give solutions to different problems encountered. It is in the aim of the authors that by the end of the project, many of the ideas explained in this work will be completed and totally operative in order to be applied to final commercial products. The use of drones, hyperspectral and radar sensors, autonomous vehicles, advanced simulation environments with different weather models and simulated sensor stimuli and robust communications will open new avenues in a domain with a significant number of social, economic and environmental impact.

Acknowledgment

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under Grant Agreement No 692455. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

References

[1] ENABLE-S3: European Initiative to Enable Validation for Highly Automated Safe and Secure Systems, [Online], https://www.enable-s3.eu/, last accessed: January 2018

[2] Precision agriculture and the future of farming. Annex 1: Technical Horizon Scan, European Parliamentary Research Service (EPRS), August 2016.

[3] CarMaker: Virtual testing of automobiles and light-duty vehicles, [Online], https://ipg-automotive.com/products-services/simulation-software/carmaker/, (last accessed: February 2018)

[4] VTD – Vires Virtual Test Drive, [Online], https://vires.com/vtd-vires-virtual-test-drive/, (last accessed: February 2018)

[5] Dynacar, [Online], http://www.dynacar.es/en/home.php, (last accessed: February 2018)

[6] xMOD – Multi-Modal Integration & Virtual experimentation platform, [Online], http://www.fev.com/fileadmin/user_upload/FEV_xMOD_2017.pdf, (last accessed February 2018)

[7] ICOS – Independent Co-Simulation Framework, [Online], http://www.v2c2.at/en/products/icos/, (last accessed: February 2018)

[8] M. Althoff, O. Stursberg, M. Buss: "Model-Based Probabilistic Collision Detection in Autonomous Driving", IEEE Transactions on Intelligent Transport Systems, 10(2): 299-310 (2009)

[9] Matrice 600, [Online], https://www.dji.com/matrice600, (last accessed: February 2018)

[10] Specim FX10, [Online], http://www.specim.fi/products/specim-fx10/, (last accessed: February 2018)

[11] Chein-I Chang, "Hyperspectral Imaging", Springer, 2003

[12] Sebastian Lopez, Tanya Vladimirova, Carlos Gonzalez, Javier Resano, Daniel Mozos, Antonio Plaza, "The promise of reconfigurable computing for hyperspectral imaging onboard systems: A review and trends", Proceedings of the IEEE, Vol. 101, Issue 3, pp. 698-728, 2013.

[13] Prasad S. Thenkabail and John G. Lyon, "Hyperspectral Remote Sensing on Vegetation", CRC Press, 2011

[14] Lattarulo, R., Pérez, J., & Dendaluce, M. (2017), "A complete framework for developing and testing automated driving controllers.", IFAC-PapersOnline, 50(1), 258-263

[15] Pedro, A., Pinto, J.S., Pereira, D., and Pinho, L.M., "Runtime Verification of autopilot systems using a fragment of MTL-∫", pp. 1-17, Springer Berlin Heidelberg, August 2017

[16] Pedro, A., Pinto, J.S., Pereira, D., and Pinho, L.M., "RMTLD3Synth: Runtime Verification toolchain for generation of monitors based on the restricted Metric Temporal Logic with Durations", https://github.com/cistergit/rmtld3synth (last accessed: February 2018)

[17] Pedro, A., Pinto, J.S., Pereira, D., and Pinho, L.M., "RMTLD3Synth Web Demonstrator", [Online], https://anmaped.github.io/rmtld3synth/ (last accessed: February 2018)

LoRa Alliance – Wide Area Networks for IoT, [Online], http://www.lora-alliance.org/, (last accessed: February 2018)