Spatial Irradiance Estimation in a Thermosolar Power Plant by a Mobile Robot Sensor Network

J. G. Martin[a,*], J.M. Maestre[a], E.F. Camacho[a]

[a] *Department of Systems Engineering and Automation, University of Seville, Seville, Spain*

Abstract

An algorithm for mapping Direct Normal Irradiance (DNI) in thermosolar power plants using a Mobile Robotic Sensor Network (RSN) is presented. The algorithm selects measurements spots and allocates the RSN accordingly to carry out the dynamic estimation of DNI. A generic thermosolar power plant with a fleet of vehicles is used as a simulated case study to assess the performance of the algorithm. The results show that the proposed method allows us to obtain a spatial estimation of the DNI that improves the flow control in the loops of the plant, outperforming estimations based on a single pyrheliometer.

Keywords: Multi-Robot System, Task planning, Sensor Networks, Direct Normal Irradiance, Distributed Estimation, Thermosolar plant.

1. Introduction

Thermosolar power plants are large-scale systems where solar collectors gather solar energy to generate electric power. In the case of Parabolic Trough Collector (PTC) solar plants, collectors are composed of parabolic mirrors and a tube located in the focal point of the parabola where a heat transfer fluid (HTF), usually thermic oil, is heated up to generate steam for a turbine (Camacho et al., 1997, Camacho and Berenguel, 2012). In these plants, there is typically a single pyrheliometer measuring Direct Normal Irradiance (DNI) to control the HTF flow, which is not efficient whenever the DNI received is not homogeneous across the solar field, e.g., due to clouds. In these circumstances, it is important to know the direct solar irradiance over the plant in order to increase or reduce the HTF flows accordingly to avoid avoid overheating in some parts of the

*Corresponding author
Email address: jgarmar@us.es (J. G. Martin)

Nomenclature

Acronyms

HTF Heat Transfer Fluid.

TES Thermic Energy Storage.

UAV Unmanned Aerial Vehicle.

UGV Unmanned Ground Vehicle.

DNI Direct Normal Irradiance.

MRS Multi-Robot System.

MRTA Multi-Robot Task Allocation.

WSN Wireless Sensor Network.

RSN Robotic Sensor Network.

GIS Geographic Information System.

PTC Parabolic Though Collector.

Sets

$\mathcal{T}(k)$ Set of tasks in instant k.

$\mathcal{V}$ Set of unmanned vehicles.

$\mathcal{G}$ Set of UGVs.

$\mathcal{A}$ Set of UAVs.

$\mathcal{C}$ Set of values that sensors can measure.

Algorithm variables

$\mathrm{CF}^{\mathrm{R}}_{ij}$ Real cloud factor in cell ij.

$\mathrm{CF}^{\mathrm{E}}_{ij}$ Expected cloud factor in cell ij.

$\mathrm{CF}^{\mathrm{est}}_{ij}$ Estimated cloud factor in cell ij.

H_{ij} Information entropy in cell ij.

CFE_{ij} Cloud factor effect in cell ij.

HE_{ij} Information entropy effect in cell ij.

WE_{ij} Wind effect in cell ij.

J_{ij} Cost function in cell ij.

Algorithm parameters

t^a Time in which algorithm erase unfinished tasks and generate new ones.

$\lambda_{1,2,3,4}$ Weights for CFE HE and WE respectively.

$a_{1,2}$ Parameters for $\mathrm{CF}^{\mathrm{est}}$.

$\mathrm{N}^{o}_{\mathrm{T}}$ Number of tasks generated each time the algorithm generates new tasks.

F Forgetting factor.

MRTA parameters

δ_j^{MRTA} Represents the urgency of the tasks, i.e., the penalization for the time taken in performing a certain task j.

γi^{MRTA} Represents the penalization for moving a certain robot i.

Other parameters

V_{mean} xy mean velocity of vehicles.

V_{Zmean} vertical mean velocity of vehicles.

σ^{o} Standard deviation of a measurement taken in the same cell.

σ^{max} Standard deviation of a measurement taken in the farthest cell within the range.

R_D Range of cells where a measurement affects other measurements.

plant and not hot enough HTF in other parts of the plant. HTF may reach temperatures over the maximum limits. and as a consequence, collectors may need to be defocused, with the undesirable waste of solar energy.

To deal with this issue, some works such as (Sánchez et al., 2018, Masero et al.) propose using local valves for controlling the flows in different collectors, requiring a spatial estimation of the irradiance throughout the plant. To this end, clouds movement can be forecasted using images (Radovan and Ban, 2014) and weather information (Zhang et al., 2018). Here, we consider DNI sensors mounted on unmanned vehicles as an integral part of the control system. Particularly this work presents a first assessment based on simulations for the use of mobile Robotic Sensor Network (RSN) to estimate solar irradiance on a thermosolar power plant. Even though unmanned vehicles are still expensive, their price is expected to continue decreasing at the same or even at a faster pace in the next years. Likewise, the performance increase of the power plant can easily outweigh the investment required.

Multi-Robot Systems (MRS) are formed by more than one robot and have the objective of performing a set of tasks in an efficient manner. These systems have been used in the last decades for purposes such as logistics (Farinelli et al., 2017), surveillance (Gohari et al., 2019), filming (Zema et al., 2017), agriculture (agr), inspection (Brusell et al., 2016), and mapping (Yang et al., 2017). Another relevant application of MRS is the generation of RSN, i.e., a Wireless Sensor Network (WSN) where sensors can move around the field (Akyildiz et al., 2002, Aydin et al., 2019). In particular, RSN have often been proposed for mapping and moniroting environmental variables and gathering information for Geographic Information System (GIS) (Bolstad, 2016). Some works using RSNs for spatial estimation are mentioned in (Roldán et al., 2016), where one Unmanned Ground Vehicles (UGV) and one Unmaneed Aerial Vehicles (UAV) are used to monitor temperature and humidity in a greenhouse, and in (Conesa-Muñoz et al., 2016), where aerial and ground vehicles are respectively used to gather environmental information and perform interventions. Also, in (Zhang and Leonard, 2010), a cooperative Kalman filter is used for both managing a RSN and estimating the state of a static planar scalar field.

Integration of the collected information is also a relevant topic in this context. For example, mapping environmental variables such as temperature, pressure and other geographic data has been largely approached using Kriging (Williams, 1998), which has became a *de facto* standard for many GIS, such as ArcGIS. Kriging is a technique that was first developed in (Matheron, 1963) and then generalized for all sort of spatial applications in (Cressie and Wikle, 2015). In particular, the spatio-temporal generalization is specially suitable for events with dynamic variables and shifting concentration over a certain area, e.g., fire, contaminating fluids in water, and DNI changes due to cloud coverage. Indeed, this technique has been proposed in the literature to estimate and forecast solar irradiance, e.g., see (Yang et al., 2013) and (Aryaputera et al., 2015). Also, spatio-temporal kriging has been proposed for simultaneous environmental mapping of dynamic variables and sensor placement in works such as (Roy et al., 2016), (Roy et al., 2018) and (Graham and Cortés, 2011).

In this work, we study how the RSN can be managed to collect information for the control system in the most efficient manner. In particular, we integrate several information layers comprising data measured by ground and air robots, and fixed sensors such as pyrheliometers. Then, we use a Bayesian approach to update the probability that a certain area of the solar field is covered. In this way, robots are moved to positions where the information gathered is maximized. This approach is aligned with other works in the literature that use Bayesian Inference (BI) and information theory to manage RSN such as (inf). Other works that follow this approach are (Julian et al., 2012), (Julian et al., 2013), and (Cui et al., 2015), where mutual information is used in lockstep with consensus based strategies for the control of a distributed fleet of vehicles acting as a RSN and then BI is applied to update the state of the environment.

The rest of this work is organized as it follows: In Section 2, the problem statement and assumptions regarding the thermosolar plant, the vehicle fleet, and the clouds are presented; in Section 3, the proposed algorithm for the spatial estimation is detailed; in Section 4, the case study where the algorithm is tested is described; in Section 5, the results of the simulations are shown and discussed. Conclusions and future investigation lines are given in Section 6.

2. Problem Statement

PTC plants have an structured layout where the following elements can be found (see Fig. **??**):

- Collectors that gather solar energy and warm up the thermic oil.
- Buildings containing:
  - Steam generation plant.

    - Turbines.
    - Control rooms.
    - Workshops and storage buildings.
    - Offices.
- Cooling towers.
- HTF tanks.
- Thermic Energy Storage (TES) tanks (in some cases).
- Manifolds, which distribute (receive) the HTF to (from) the collectors.
- Parking lots.

In most cases, the buildings, the parking lots and the TES tanks are located in a central area for human operators. All the manifolds that distribute and collect oil come in and out of this area and the collectors are placed in rows to maximize the use of available area.

A grid of DNI measurement spots or *cells* located in the space between collectors is considered. We assume that information from previous measurements and wind vectors in cells is available (e.g., obtained from a numeric model like *Harmonie Arome* in Spain (Kalnay, 2003, AEMET)). Moreover, it could be considered that there are other sources of wind information such as anemometers in the plant and sensors in the UAVs.

Clouds are assumed to be composed of a cluster of ellipsoids contained in a larger ellipsoid with random dimensions following a Gaussian distribution according to (Kulemin, 2003), with velocity interpolated from the wind field in each time iteration. An example of the clouds considered can be seen in Fig. 1. Note that Spencer equations (Spencer, 1971) can be used to obtain solar rays at each measurement spot and calculate their interference with clouds.

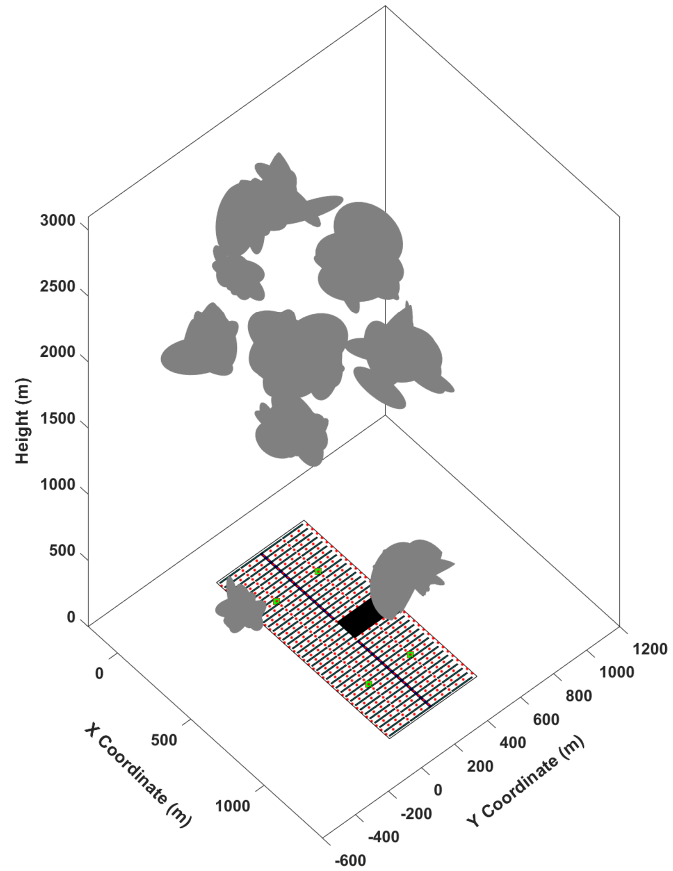
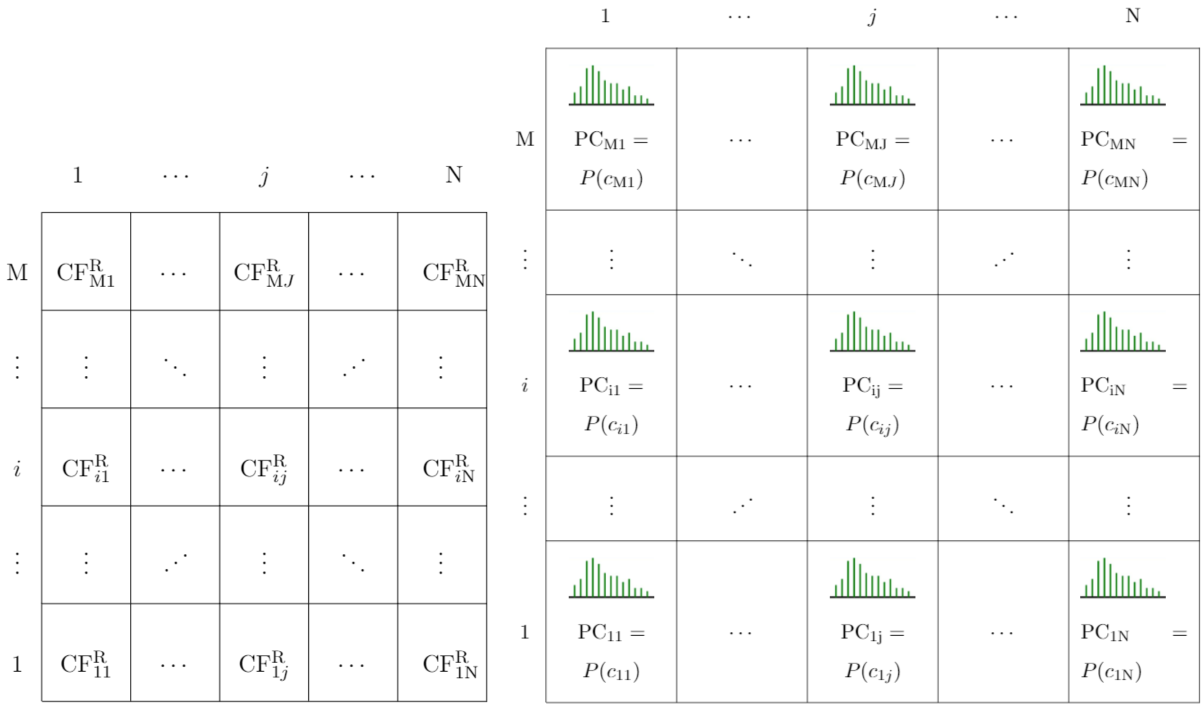


Figure 1: Example of clouds above the power plant.

Since nominal clear-sky irradiance can be computed with clear sky models, irradiance losses can be modelled by a *cloud factor* (CF) in the range $0 - 1$ in each measurement spot, creating the CF grid of Fig. 2a. If $\mathrm{CF}_{ij} = 0$, spot ij receives nominal DNI, whereas $\mathrm{CF}_{ij} = 1$ implies that spot ij receives 0 DNI. Our aim is to estimate the CF grid as precisely as possible. Discretized real values are denoted as CF_{ij}^{R} and are approached via a *discrete probability distribution* for each cell, forming the so-called PC field, which is depicted in Fig. 2b. When there is no information regarding the field, a uniform distribution for each cell is assumed.



(a) CF^{R} grid.

(b) PC grid.

Figure 2: Grids

To perform the estimation, an heterogeneous set of unmanned vehicles, $\mathcal{V}$, is considered, which is composed of a set $\mathcal{G}$ of UGVs and a set $\mathcal{A}$ of UAVs, i.e., $\mathcal{V} = \mathcal{G} \cup \mathcal{A}$. Each vehicle has a DNI sensor integrated and can perform measurements at the cell where it is located by using sensors such as that in Fig. 3. Task consist of moving a sensing vehicle to a different cell and our aim is to generate at each time instant a set of tasks $\mathcal{T}$ to improve DNI estimations, i.e., $\mathcal{T} = \mathcal{T}(k)$.

Figure 3: ISS Solar-MEMS sun sensor (Solar MEMS Webpage).

Vehicles are assumed to have low level controllers and move always at cruise velocity with some constraints:

- Unmanned vehicles must not enter the buildings area. Consequently, there are no measurement spots inside this area.
- Manifolds can only be crossed by UGVs using the bridges designed for this purpose.
- Flying over collectors is not allowed for security.

Furthermore, each vehicle is assumed to consume and charge its battery in a linear way, with rates depending on its features. Likewise, there are different charging stations available for UGVs and UAVs.

Also, some considerations have been assumed regarding the sensors:

- Sensors have a predefined resolution and range, i.e., they can only take values from an *alphabet* $\mathcal{C}$, and not intermediate values.
- The probability function of the sensor $\mathrm{p}(z|c)$ is known and modelled using truncated normal distributions, where z is the value of the measurement and c denotes the discretized value of the field. A subscript ij can be used to denote the value at a given cell (see Table 1). Hence, given the prior probability contained in PC field and the conditional sensor probability contained in PZ, the posterior probability (probability after a measure is taken) can be easily calculated using BI as

$$P(c_{ij}^r|z_{ij}) = \frac{P(c_{ij}^r)\cdot P(z_{ij}|c_{ij}^r)}{\sum\limits_{c_{ij}^r\in\mathcal{C}} P(c_{ij}^k)\cdot P(z_{ij}|c_{ij}^k)} \quad c_{ij}^r \in \mathcal{C}. \tag{1}$$

- Measurements taken while vehicles are moving are less reliable than when they are still for a certain time, t^{cm}. Also, measurements taken by UAVs are less reliable than those taken by UGVs.
- Measurements taken by fixed pyrheliometers are the most reliable ones.
- Measurements in a certain location will affect the beliefs of nearby locations and during a time window, i.e., it is assumed that there is a spatio-temporal correlation in the DNI. Therefore,

to take nearby cells into account, BI becomes

$$P(c_{ij}^r|z_{mn}, z_{op}) \quad = \quad \frac{P(c_{ij}^r)\cdot P(z_{mn}|c_{ij}^r)\cdot P(z_{op}|c_{ij}^r)}{P(z_{mn})\cdot P(z_{op})} \quad c_{ij}^r \in \mathcal{C}, \tag{2}$$

with z_{op} and z_{mn} being measurements taken in cells inside the influence area of cell ij.

Table 1: PZ.

PZ	$z_{ij} = c^1$	$\cdots$	$z_{ij} = c^r$	$\cdots$	$z_{ij} = c^{\vert\mathcal{C}\vert}$
$c_{ij} = c^1$	$P(z_{ij} = c^1 \vert c_{ij} = c^1)$	$\cdots$	$P(z_{ij} = c^r \vert c_{ij} = c^1)$	$\cdots$	$P(z_{ij} = c^{\vert\mathcal{C}\vert} \vert c_{ij} = c^1)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$⋰$	$\vdots$
$c_{ij} = c^r$	$P(z_{ij} = c^1 \vert c_{ij} = c^r)$	$\cdots$	$P(z_{ij} = c^r \vert c_{ij} = c^r)$	$\cdots$	$P(z_{ij} = c^{\vert\mathcal{C}\vert} \vert c_{ij} = c^r)$
$\vdots$	$\vdots$	$⋰$	$\vdots$	$\ddots$	$\vdots$
$c_{ij} = c^{\vert\mathcal{C}\vert}$	$P(z_{ij} = c^1 \vert c_{ij} = c^{\vert\mathcal{C}\vert})$	$\cdots$	$P(z_{ij} = c^r \vert c_{ij} = c^{\vert\mathcal{C}\vert})$	$\cdots$	$P(z_{ij} = c^{\vert\mathcal{C}\vert} \vert c_{ij} = c^{\vert\mathcal{C}\vert})$

Finally, a regulation factor F is considered to make the probability return to maximum uncertainty if no new information is gathered during several time steps by using the update filter

$$P(c_{ij}^r(k+1)|c_{ij}^r(k)) \quad = \quad P(c_{ij}^r(k)) + (\tfrac{1}{|\mathcal{C}|} - P(c_{ij}^r(k)))\cdot F, \quad c_{ij}^r \in \mathcal{C}. \tag{3}$$

2.1. Information Sources and Processing

In this subsection, we review the considered information sources and the corresponding processing to obtain the PC field.

2.1.1. Wind Effect

The aim of this layer is to take into account the appearance of *new* clouds above the plant. Averaging the measurements of anemometers, a homogeneous wind in the plant is considered to find out where a cloud is more likely to enter the field. These cells will be assigned a value of $WE_{ij} = 1$. The value of the rest of the cells will be determined so that the previously mentioned value is $WE_{ij} = 0$ and that the value of WE_{ij} decays as a function of distance to the area where clouds get to the plant and the intensity of the wind.

2.1.2. CF Effect

The expected value of the cell probability distributions, i.e., $CF^{\mathrm{E}}_{ij} = \mathbb{E}(\mathrm{PC}_{ij})$, is updated applying a mask or *kernel* h^{CF}, generated using the average velocity of the clouds, taking into account cells from where a cloud might come from according to wind direction and intensity.

This kernel is generated taking into account the already calculated layer WE. The size of the kernel depends on the intensity of the wind, since the key idea here is to modify CF according to the movements of clouds, so that new tasks can anticipate this movement.

Once this kernel has been applied to CF, a new layer CFE is obtained. Notice that in layer CFE, the value of each cell ij depends not only on the value of CF^{E}_{ij}, but also on the values of upwind cells.

2.1.3. H Effect

This layer identifies points where a measure can provide more information to the system. In information theory, Shannon Entropy (inf), H, measures the uncertainty of an information source, being 0 if the result is certain and 1 if there is total uncertainty on the result. From this viewpoint, it is clear that taking a measurement in a cell of the grid will reduce the information entropy on that cell, H_{ij}, and also the cells nearby. The kernel h^H can be computed by adding up all the entropy values in a circle around the measurement spot and normalizing to keep the value between 0 and 1. Once this kernel has been applied to H, a new layer called HE is obtained.

2.1.4. Shadow Detection

Cloud shadows can alternatively be found using cameras on top of towers as in Kuhn et al. (2017), UAVs and even a hot-air balloon. Either way, we consider a Shadow Effect (SE) layer, with $SE_{ij} = 1$ if there is shadow in the cell and $SE_{ij} = 0$ otherwise. This camera provides us with very relevant information, discriminating in advance those points where there may be a fall of DNI due to a cloud passing by.

2.2. Outcomes

The outcomes obtained after processing the information gathered by the information sources are detailed next.

2.2.1. *Computation of J*

Layer J allows us to generate new tasks and is calculated as the convex sum of the four layers:

$$\begin{aligned} J_{ij} = \quad & \lambda_1 \cdot CFE_{ij} + \lambda_2 \cdot HE_{ij} + \lambda_3 \cdot WE_{ij} + \lambda_4 \cdot SE_{ij}, \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{4} \lambda_i = 1. \end{aligned} \tag{4}$$

Since the four layers are normalized, it holds that $J_{ij} \in [0, 1]$.

The values of these weights change the behavior of the algorithm. The higher λ_1 is, the more important the already known shadows become; exploring unknown areas is adjusted via λ_2; λ_3 deals with surveilling the borders of the plant, where new shadows can appear; finally, λ_4 ensures that non-shadowed cells have low value of J and it can be set to 0 if no shadows are detected.

2.2.2. *Cloud Factor Estimation*

To generate a spatial estimation of DNI at any time instant, it is necessary to consider the information available from the measurements taken previously, i.e., the information contained in PC. As stated previously, both the expected value CF_{ij}^{E} and the entropy value H_{ij} can be obtained from $\mathrm{PC}_{ij}, \ \forall i, j$. Since the most likely value of a cell is zero when no other information is available, the following filter function is used:

$$CF_{ij}^{\mathrm{est}} \quad = \quad \frac{CF_{ij}^{\mathrm{E}}}{1+e^{a_1 \cdot (H_{ij} - a_2)}}, \tag{5}$$

where a_1 and a_2 are tuning parameters. In this way, if the entropy in a cell is high, the estimation will be nearer to 0 and if the entropy is low, the estimation will be corrected towards the value CF_{ij}^{E}. Notice that equation (5) is an inverted logistic function that assign CF_{ij}^{est} the value of CF_{ij}^{E} if there is little uncertainty, and value 0 otherwise.

Remark 1. *Note that equation (5) is based on the activation function of a neuron. Implementing a complete neural network to taking into account not only the probability distribution in each cell but also that of the nearby cells its a matter of current research.*

Furthermore, when cameras are available we can assume that in all the cells where the cameras are not detecting any shadows $CF_{ij}^{\mathrm{est}} = 0$ disregarding the value of (5).

Finally, in Fig. 4, a summary of the different information sources considered and the processing flow is shown.

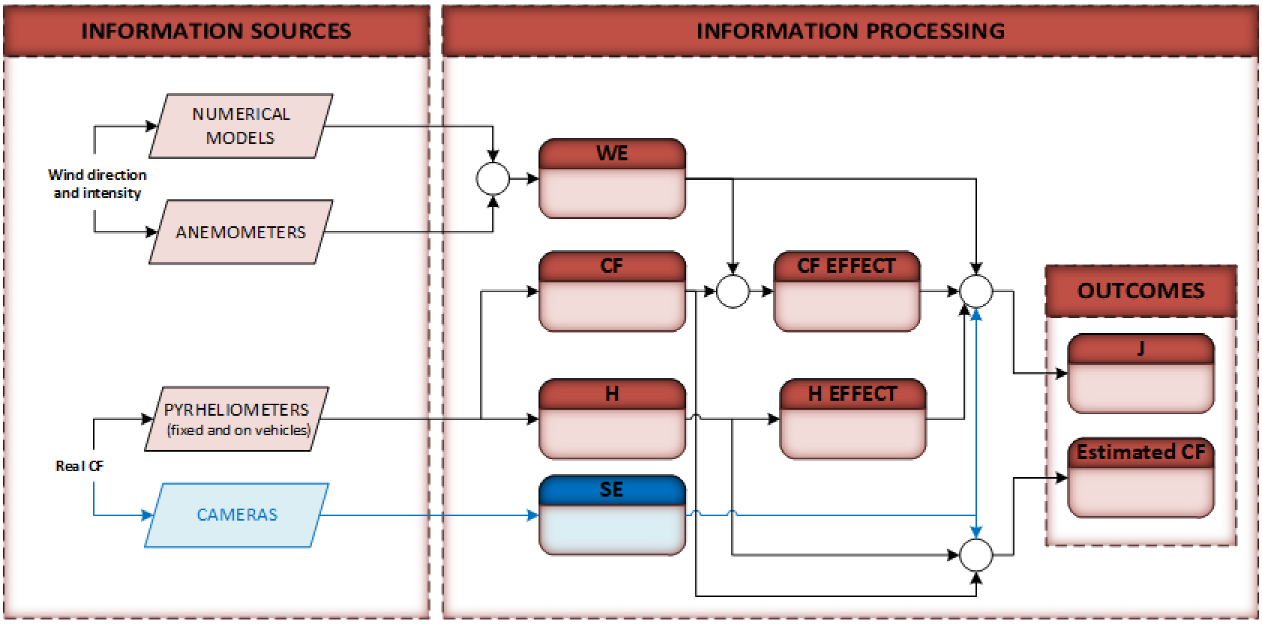


Figure 4: Data gathered from information sources and processing flow. The WE block is explained in Subsubsection 2.1.1. Both CF and CFE blocks are detailed in Subsubsection 2.1.2. H and HE are both described in Subsubsection 2.1.3. Information gathered by cameras and the corresponding processing can be found in Subsubsection 2.1.4. Finally, the computation of layer J and of the estimated CF appear in Subsubsections 2.2.1 and 2.2.2, respectively.

3. Tasks Generation

In this section, an algorithm is proposed to generate new tasks for the vehicle fleet, so that a spatial estimation of the DNI in the field can be obtained. The proposed algorithm follows the block diagram of Fig. 5, where the first stage is to update the previously introduced layers.

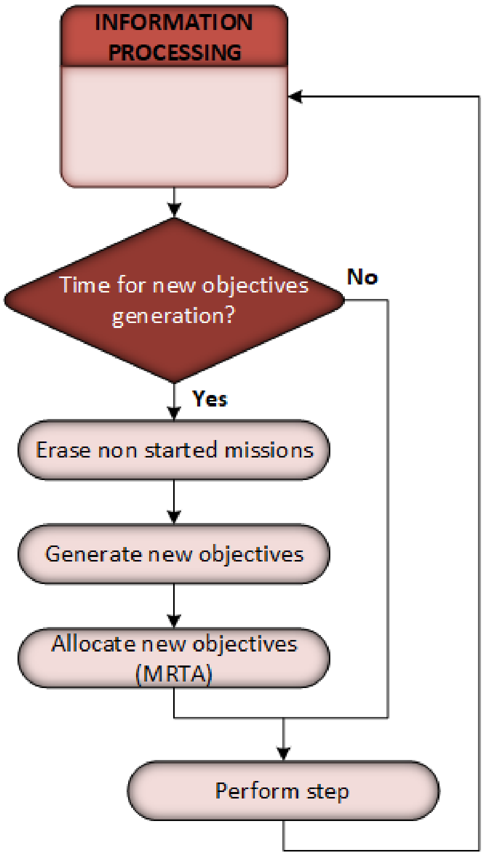


Figure 5: Proposed Algorithm. Note that performing a step include all type of actions regarding the robots, the sensors and the environment (moving to allocated tasks, taking new measures, updating the clouds and the sun position, etc.)

Then, every time that timer t^a ends, incomplete tasks are erased and new tasks are generated using layer J and Algorithm 1:

Algorithm 1: New Tasks Generation

Let N_T^o be the number of tasks to be generated;

$\mathrm{N}_\mathrm{T} = 1$;

while $\mathrm{N}_\mathrm{T} \leq \mathrm{N}_\mathrm{T}^o$ **do**

  Find the cell with the maximum value of J;

  Update field H considering that a measurement has been taken in that cell by subtracting to field H^a a conic surface centered in the chosen cell with a base radius of R_D;

  Compute HE^a from H^a;

  Compute J^a by means of (4) replacing HE for HE^a;

  $\mathrm{N}_\mathrm{T} = \mathrm{N}_\mathrm{T} + 1$;

end

Once there is a new set of tasks, new assignments for the vehicles can be obtained using a MRTA algorithm based on (MRT), which solves

$$\begin{array}{ll} \min\limits_{U} & J^{\mathrm{MRTA}}(U) = \sum\limits_{j=1}^{|\mathcal{T}|} \delta_j^{\mathrm{MRTA}} \cdot t_j(U) + \sum\limits_{i=1}^{|\mathcal{V}|} \gamma_i^{\mathrm{MRTA}} \cdot d_i(U) + \Omega(U) \\ \text{s.t.} & u_i(n) \in \mathcal{T} \cup \{0\} \ \forall\ i, n \end{array} \tag{6}$$

using a genetic algorithm (i.e., the allocations obtained may be suboptimal but feasible), where $U = \begin{bmatrix} u_1 & \cdots & u_V \end{bmatrix}$ aggregate vectors $u_i \in \mathbb{R}^{1\times M}$ representing the allocation of robot i, i.e., each element in u_i represents a task and in a given allocation robot i will perform the tasks in u_i sequentially (not considering zeros); δ_j^{MRTA} are weights corresponding to the priority given to task j; $t_j(U)$ is the time that it takes to complete the task j in a given allocation; γ_i^{MRTA} corresponds to the penalty of using robot i; and $d_i(U)$ corresponds to the distance traveled by robot i, and function $\Omega(U)$ implements soft restrictions related to power feasibility of the allocation and no repetition of tasks. In the approach presented in this work, we will consider layer J to set δ_j^{MRTA} values for the tasks.

Finally, this MRTA algorithm does not ensure fulfillment of all tasks before the next allocation, t^a, but note that tasks may lose relevance as new information comes in. Hence, only truly relevant tasks will be generated again by the algorithm.

4. Case-Study

In this section, we first present the simulated thermosolar plant layout and vehicles considered. Then, the proposed algorithm is tested.

4.1. Thermosolar Plant Layout

The thermosolar power plant is based on a section of the plant *Solacor I* in *El Carpio, Spain* (Abengoa) and can be seen in Fig. 6.

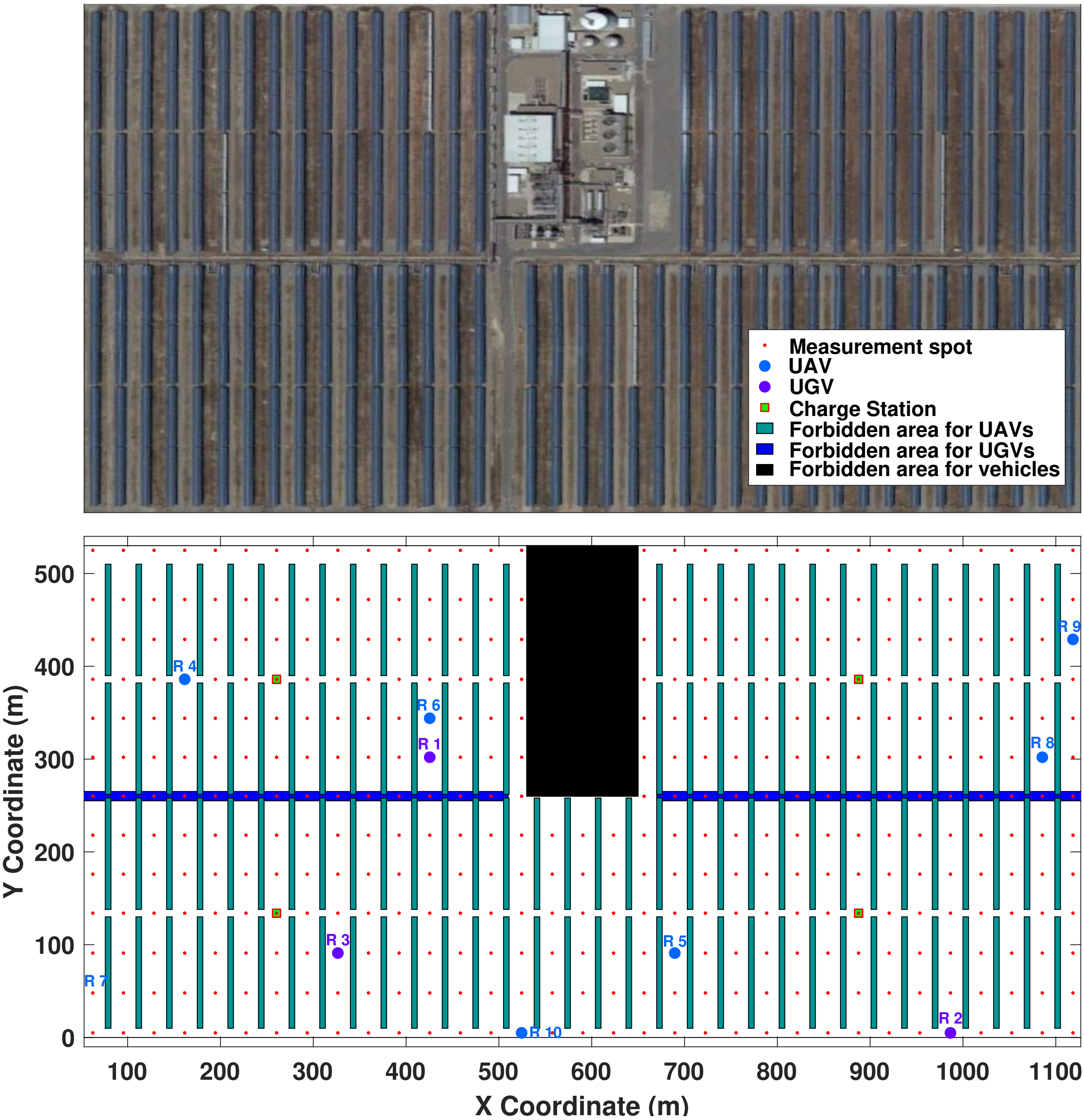


Figure 6: Section of the real Thermosolar power plant of *El Carpio* (Spain) and layout considered in this Case-Study

This 30 MW thermosolar plant covers 63 ha (1180 × 540 m), containing its operation zone, where the buildings, parking lots, TES and electric station are placed, in the upper centre of the plant. Likewise, cold and warm HTF manifolds run paralell from the southern part of the operation zone to the extremes of the plant, dividing the plant into three sectors. There are 18 loops in the south sector of the plant and 8 loops in each one of the north sectors and a bridge for the ground vehicles to go from the south to the north at each side of the operation zone. Each loop has 4 collectors in "U" shape, because loops start and end at the manifolds.

4.2. Unmanned vehicles parameters

The parameters of ground and aerial vehicles are given in Table 2 and are based on commercial models [1], with V_{mean} being the velocity in the horizontal plane, V_{Zmean} the vertical velocity, and λ_i the weight that each agent has in the MRTA step.

Table 2: Value of the vehicle parameters based on commercial models.

Parameter	**UGV**	**UAV**
V_{mean}	1.5 m/s	10 m/s
V_{Zmean}	0 m/s	3 m/s
Discharge rate	0.005	0.1
Charge rate	0.0025	0.03
γ_i^{MRTA}	1.5	1

The initial positions of the vehicles can be seen in Fig. 6 and have been randomly generated. As for the pyrheliometer, it is fixed and located next to the plant.

4.3. Bayesian estimation parameters

It has been considered that the precision of the DNI sensors equipped in the vehicles is 10% so that c^r can take 11 values from 0 to 1. The measurements follow normal distributions depending on the equipment and on how they are takenwith relevant features are presented in Table 3. The standard deviation σ is assumed to vary for adjacent cells as

$$\sigma \quad = \quad \frac{\sigma^{\max} - \sigma^{\text{o}}}{R_D^3} \cdot D^3 + \sigma^{\text{o}}, \tag{7}$$

[1]UAVs are based on DJI Matrice 200 and UGVs on Summit-XL.

where σ^{o} is the standard deviation in the cell where the measurement has been taken, R_D is the maximum distance from which a cell is affected by a measurement in another cell, and $\sigma^{\max}$ is the value of σ at the farthest affected cell.

Table 3: Measurement parameters

	σ^{o}	$\sigma^{\max}$	$\text{t}^{\text{m}}(s)$	$\text{R}_D(cells)$
Ground Vehicle	0.05	3	1	4
Moving Ground Vehicle in march	1	5	0.5	4
Aerial Vehicle	0.05	3	1	4
Moving Aerial Vehicle	1.5	6.5	0.5	4
Pyrheliometer	0	3	0.3	2

The algorithm is run every 60 s (t^a) and the weights used are $\lambda_1 = 0.2$, $\lambda_2 = 0.7$ and $\lambda_3 = 0.1$.

4.4. Wind and clouds

For the case study, wind velocity has been introduced as a vectorial field extracted from a numerical model, as stated in Section 2. In particular, the field is defined for a square area encompassing the plant and its surrounding area with different values at different heights. The mean value of this field is given by the vector $\begin{bmatrix} 0.554 & 0.318 & 0 \end{bmatrix}^{\text{T}}$ m/s.

In order to simulate dynamically a shadow field in the power plant, cumulus clouds have been randomly simulated above the thermosolar plant field between 500 m and 2000 m, which is the altitude where this type of clouds usually appear.

5. Results

In this section, the results of applying the algorithm proposed for task generation and spatial estimation to the case study are presented.

A set of 30 minute simulations have been run with 3 different CF inputs chosen to test the algorithm (multiple cloud shadows with different intensities). Since the proposed algorithm solves the MRTA problem heuristically, simulations has been run multiple times. Also, results have been compared with the same estimation method but with random generation of the tasks.

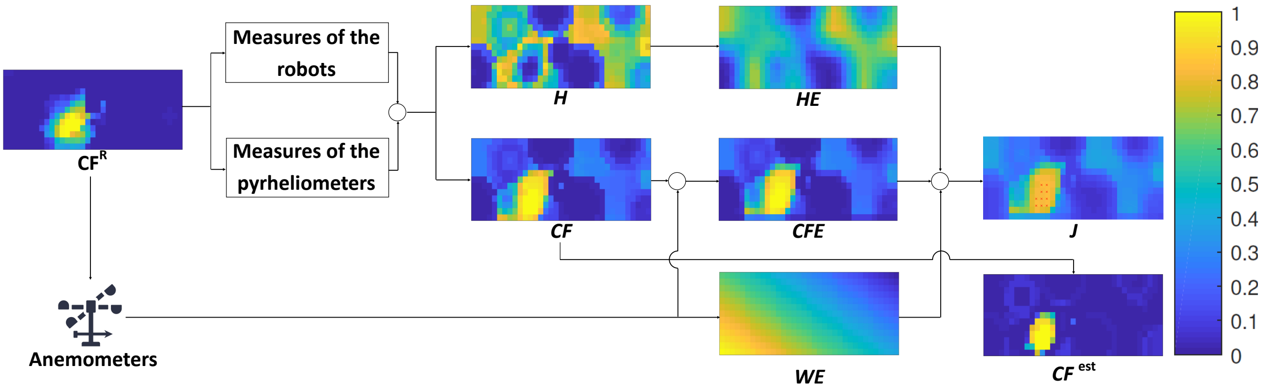


Figure 7: The *real* cloud factor value, CF^{R}, appears on the upper left corner. We can see that a cloud is covering a big area of the field and the expected CF is extracted from the real map through the sensors (here we can see that there is a strong belief of a big cloud shadowing the southwest of the plant). The H field shows the areas of the field with greater uncertainty in yellow. Even though there is a strong certainty of the presence of a cloud, there are other places where more clouds could be discovered. In addition, anemometers measure wind direction and intensity, which will be used to process the expected CF field (using the kernel h^{CF}) and to update field WE. In this case, since wind is coming from the lower left corner, this area has higher priority, which decreases linearly with distance. The HE layer is obtained from correcting H using kernel h^{H}. By adding the weighed CFE, HFE and WE layers we can obtain J (cells with red crosses represent active tasks). Finally, the current estimation can be seen in the lower right corner.

An example of the simulations, extracted from the video that can be checked here, can be seen in the following figure: In Fig. 9, the real value of the cloud factor in each cell is represented. The information gathered until that moment in the PC grid can be seen seen in the two fields obtained directly from it, namely, the expected CF field and the H field. From the information contained in these fields, the CFE and HE layers are obtained. Likewise, the WE layer is obtained from the wind direction and intensity. Then, using the previous layers, J is obtained. Besides, the estimation for the time instant is extracted from CF by means of (5). These active tasks and the locations of the vehicles in the plant can be seen in Fig. 8.

Considering the heuristic nature of the algorithm proposed for solving the MRTA step, the same simulations have been run again for the same and different CF inputs. The complete set of simulations can be consulted here.

In case a camera is available as an information source to detect cloud shadows, the algorithm can improve its performance. This information provides an extra layer that assigns 1 to shadowed cells and 0 to cloudless cells. This way, cloudy areas receive higher priority. The results can be

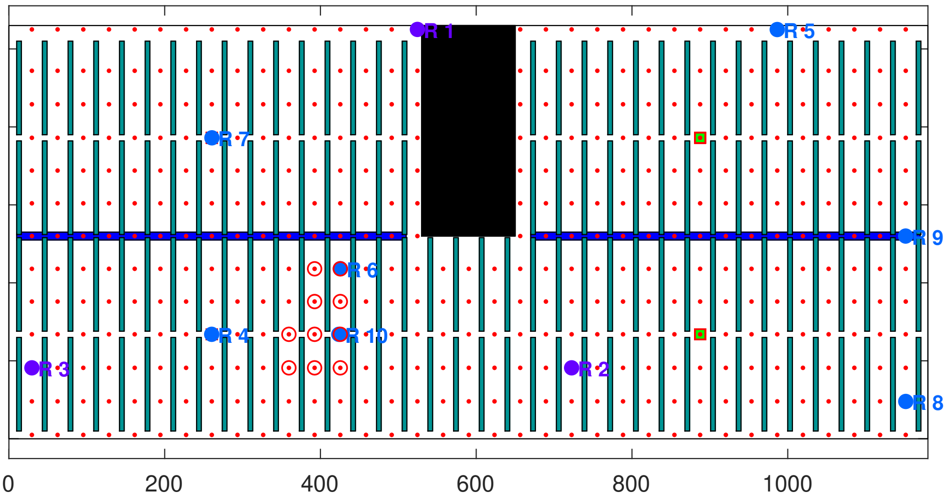


Figure 8: Location of the vehicles and the active tasks (second 1620)

verified here.

For the sake of assessment, the error has been computed as the sum of the absolute difference between the estimated and the real cloud factor in every cell during the simulation. Fig. 10 compares the proposed method with and without cameras to the random tasks generation approach and to the case where it is assumed that there are no clouds. Although the error made with the proposed approach without cameras can be greater at some points, when there is a cloud it improves its performance. Also, the proposed method with cameras outperforms all other methods during all the simulation length. Likewise, the random tasks approach can be worse than simply considering zeros in all cells.

On the other hand, in Fig. 11, the proposed method (without cameras) is compared to using a single pyrheliometer to estimate the irradiance in the whole plant. Notice that whenever the pyrheliometer is shadowed, its error explodes. Likewise, the error using only the pyrheliometer and the error considering the complete grid zero are the same except for the cases previously mentioned.

As can be seen, in most cases, once a cloud is detected, vehicles make a good estimation and follow it correctly. However, in some cases, particularly when a shadow is too light or the size of the cloud is not big enough, they may lose its track. Also, in other cases, clouds are only detected when they are already over the plant and not as they start covering it. Likewise, it may occur that the allocation finally performed by the MRTA algorithm was unable to fulfill nor regenerate some tasks due to the forgetting factor.

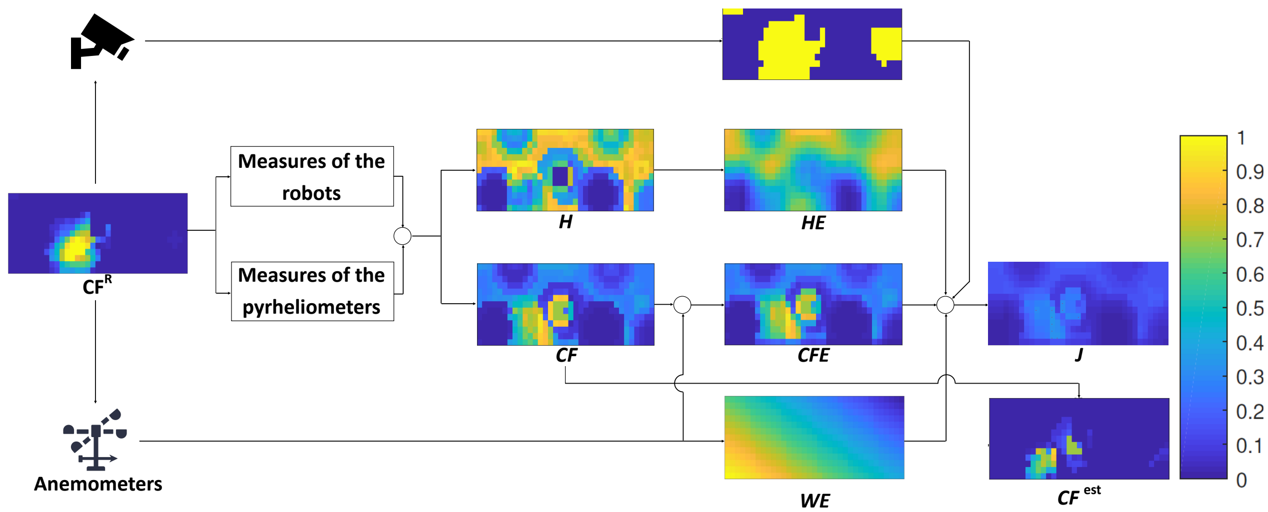


Figure 9: Information processing at second 1620 (CF Input No 1) using cameras. Weighing parameters are $\lambda_1 = 0.09$, $\lambda_1 = 0.6$, $\lambda_1 = 0.01$ and $\lambda_4 = 0.6$. The main difference with the case with no cameras is that, in this case, there is an extra layer that give us information about the places where there can be a shadow.

Finally, a relatively good estimation can be performed without cameras. Even when some noise is introduced, it can be eliminated using a filter. Also, a heuristic considering zero all measurements below a certain threshold has been included. The results of this approach can be verified here.

6. Conclusions

In this work, an algorithm that solves both the generation of new tasks and the spatial estimation of the DNI has been developed. This algorithm deals with areas where there is little information and with those where there is bigger probability of finding shadows according to the accumulated knowledge and considering the effect of the wind. The proposed framework is modular an can also include additional information sources as cameras capable of detecting shadows due to clouds. Likewise, note that this MSN is not designed to be operating at all times, but only during cloudy periods. It is in these moments when the control system can make the most from the information provided by the proposed system.

As future development, parametric continuous probability distributions will be considered for sensors and for the field probability. Likewise, taking into account another measurements sources as the temperature sensors located in the collectors, will be considered. Another future line is performing nowcasting using the information gathered and processed by our algorithm. This

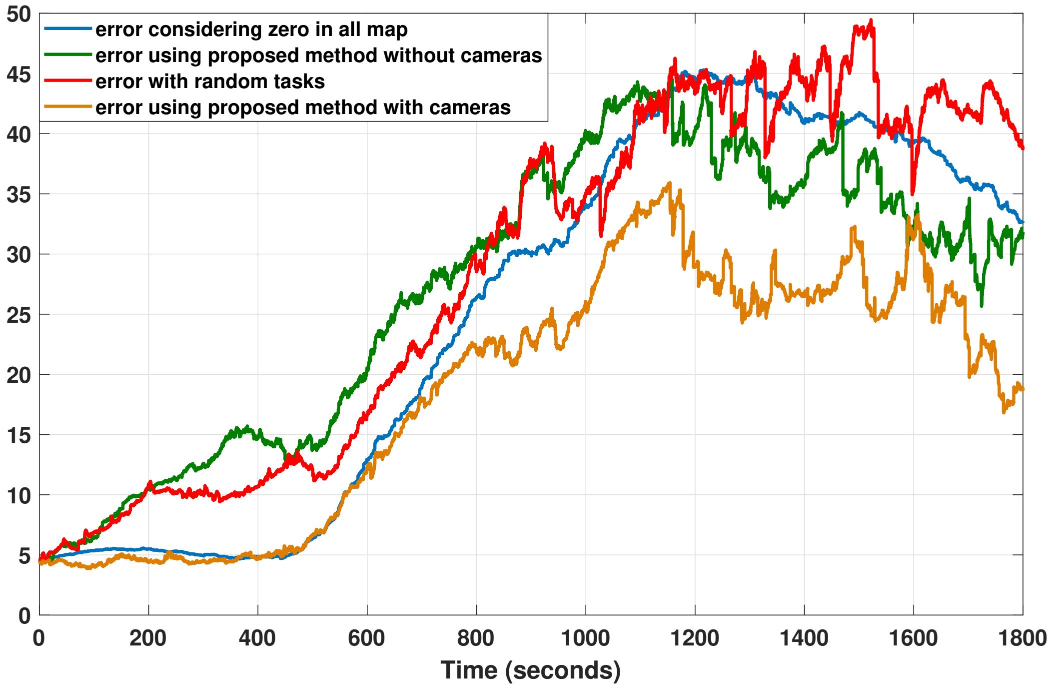


Figure 10: Comparison of errors.

nowcast will not only be useful for controlling the thermosolar power plant but will also be able to refine the algorithm proposed in this work, particularly the task generation.

Acknowledgment

This project has received funding from the European Research Council (ERC) project OCONTSOLAR (grant agreement No 789051) under the European Union Horizon 2020 Research and Innovation Programme.

References

M.

Robots in Agriculture: State of Art and Practical Experiences, author=Roldán, Juan Jesús and del Cerro, Jaime and Garzón-Ramos, David and Garcia-Aunon, Pablo and Garzón, Mario and de León, Jorge and Barrientos, Antonio, journal=Service Robots, year=2018, publisher=InTech.

E.

Abengoa. A.

AEMET. Armonie Harome. URL http://www.aemet.es/es/eltiempo/prediccion/modelosnumericos/harmonie_arome#.

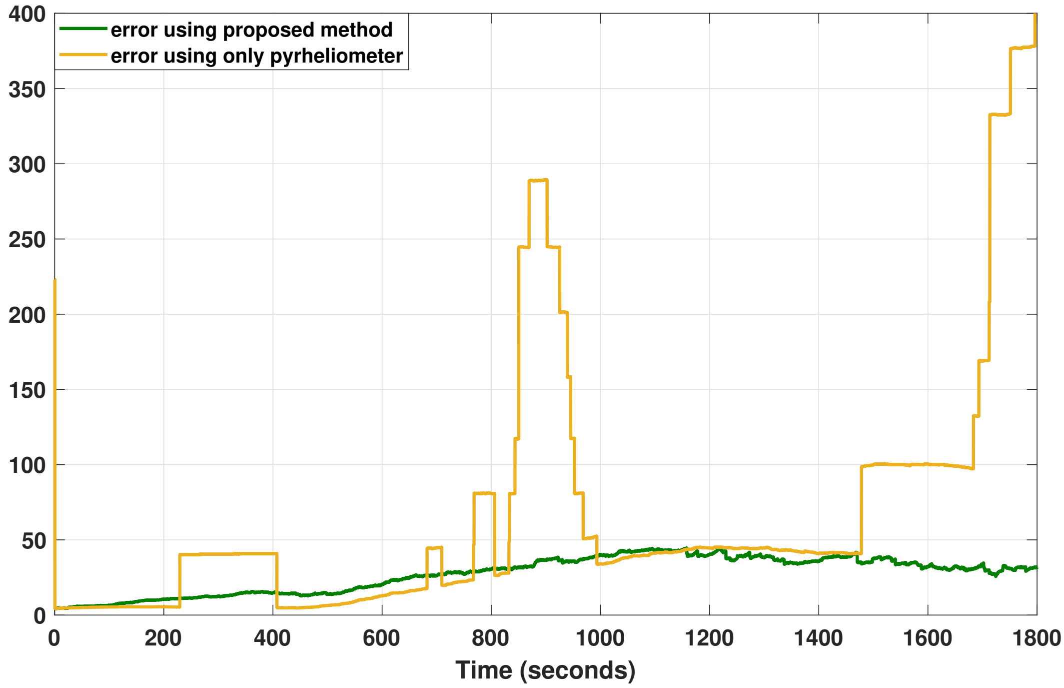


Figure 11: Comparison of errors.

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, 2002.

A. W. Aryaputera, D. Yang, L. Zhao, and W. M. Walsh. Very Short-Term Irradiance Forecasting at Unobserved Locations Using Spatio-Temporal Kriging. *Solar Energy*, 122:1266–1278, 2015.

B. E. Aydin, H. Hagedooren, M. M. Rutten, J. Delsman, G. H. Oude Essink, N. van de Giesen, and E. Abraham. A Greedy Algorithm for Optimal Sensor Placement to Estimate Salinity in Polder Networks. *Water*, 11(5):1101, 2019.

P. Bolstad. *GIS Fundamentals: A First Text on Geographic Information Systems*. Eider (PressMinnesota), 2016.

A. Brusell, G. Andrikopoulos, and G. Nikolakopoulos. A Survey on Pneumatic Wall-Climbing Robots for Inspection. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 220–225, 2016.

E. F. Camacho and M. Berenguel. Control of Solar Energy Systems. *IFAC Proceedings Volumes*, 45(15):848–855, 2012.

E. F. Camacho, M. Berenguel, and F. R. Rubio. *Advanced Control of Solar Plants*. Springer Berlin, 1997.

J. Conesa-Muñoz, J. Valente, J. del Cerro, A. Barrientos, and Á. Ribeiro. Integrating Autonomous Aerial Scouting with Autonomous Ground Actuation to Reduce Chemical Pollution on Crop Soil. In *Robot 2015: Second Iberian Robotics Conference*, pages 41–53. Springer, 2016.

N. Cressie and C. K. Wikle. *Statistics for Spatio-Temporal Data*. John Wiley & Sons, 2015.

R. Cui, Y. Li, and W. Yan. Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):993–1004, 2015.

A. Farinelli, E. Zanotto, E. Pagello, et al. Advanced Approaches for Multi-Robot Coordination in Logistic Scenarios. *Robotics and Autonomous Systems*, 90:34–44, 2017.

P. S. Gohari, H. Mohammadi, and S. Taghvaei. Using Chaotic Maps for 3D Boundary Surveillance by Quadrotor Robot. *Applied Soft Computing*, 76:68 – 77, 2019. ISSN 1568-4946.

R. Graham and J. Cortés. Adaptive Information Collection by Robotic Sensor Networks for Spatial Estimation. *IEEE Transactions on Automatic Control*, 57(6):1404–1419, 2011.

B. J. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed Robotic Sensor Networks: An Information-Theoretic Approach. *The International Journal of Robotics Research*, 31(10):1134–1154, 2012.

B. J. Julian, S. L. Smith, and D. Rus. Distributed Approximation of Joint Measurement Distributions Using Mixtures of Gaussians. *Robotics: Science and Systems VIII*, page 169, 2013.

E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge university press, 2003.

P. Kuhn, S. Wilbert, C. Prahl, D. Schüler, T. Haase, T. Hirsch, M. Wittmann, L. Ramirez, L. Zarzalejo, A. Meyer, et al. Shadow Camera System for the Generation of Solar Irradiance Maps. *Solar Energy*, 157:157–170, 2017.

G. P. Kulemin. *Millimeter-Wave Radar Targets and Clutter*. Artech House, 2003.

E. Masero, J. R. D. Frejo, J. M. Maestre, and E. F. Camacho. A Light Clustering Model Predictive Control Approach to Maximize Thermal Power in Solar Parabolic-Trough Plants. *Solar Energy*, 214:531–541.

G. Matheron. Principles of Geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

A. Radovan and Ž. Ban. Predictions of Cloud Movements and the Sun Cover Duration. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1210–1215. IEEE, 2014.

J. J. Roldán, P. Garcia-Aunon, M. Garzón, J. De León, J. Del Cerro, and A. Barrientos. Heterogeneous Multi-Robot System for Mapping Environmental Variables of Greenhouses. *Sensors*, 16(7):1018, 2016.

V. Roy, A. Simonetto, and G. Leus. Spatio-Temporal Sensor Management for Environmental Field Estimation. *Signal Processing*, 128:369–381, 2016.

V. Roy, A. Simonetto, and G. Leus. Spatio-Temporal Field Estimation Using Kriged Kalman Filter (KKF) with Sparsity-Enforcing Sensor Placement. *Sensors*, 18(6):1778, 2018.

A. Sánchez, A. Gallego, J. Escaño, and E. Camacho. Temperature Homogenization of a Solar Trough Field for Performance Improvement. *Solar Energy*, 165:1–9, 2018.

Solar MEMS Webpage. Solar-MEMS. URL https://www.solar-mems.com/solar-tracking/.

J. Spencer. Fourier Series Reprensentation of the Position of the Sun. *Search*, 2(5):172, 1971.

C. K. Williams. Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In *Learning in graphical models*, pages 599–621. Springer, 1998.

D. Yang, C. Gu, Z. Dong, P. Jirutitijaroen, N. Chen, and W. M. Walsh. Solar Irradiance Forecasting Using Spatial-Temporal Covariance Structures and Time-Forward Kriging. *Renewable Energy*, 60:235–245, 2013.

J. Yang, A. Dani, S.-J. Chung, and S. Hutchinson. Vision-Based Localization and Robot-Centric Mapping in Riverine Environments. *Journal of Field Robotics*, 34(3):429–450, 2017.

N. R. Zema, E. Natalizio, and E. Yanmaz. An Unmanned Aerial Vehicle Network for Sport Event Filming with Communication Constraints. 2017.

F. Zhang and N. E. Leonard. Cooperative Filters and Control for Cooperative Exploration. *IEEE Transactions on Automatic Control*, 55(3):650–663, 2010.

R. Zhang, M. Feng, W. Zhang, S. Lu, and F. Wang. Forecast of Solar Energy Production-A Deep Learning Approach. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 73–82. IEEE, 2018.