A Low-Cost Distributed Network for Crop Growth Optimisation in Plant Factories

Antoine Bossard
Graduate School of Science, Kanagawa University, Hiratsuka, Kanagawa, Japan
Emial: abossard@kanagawa-u.ac.jp

***Abstract*—Crops, such as for rice, are first grown in plant factories before being planted and cultivated in a field. Efficient operation of such plant factories is thus key to a successful harvest, and several theoretical models and systems have been proposed in the literature to optimise the management and production capacities of these agricultural facilities. To this end, we describe in this paper the main idea of a low-cost monitoring solution for a plant factory, based on a network of sensing devices. Compared to previous works, our proposal keeps the system costs at a minimum and achieves a higher fault tolerance by relying on a distributed network. In other words, we present in this paper the main idea of a cost-minimal fault-tolerant monitoring solution.**

***Index Terms*—agritech, smart agriculture, plant factory, microcontroller, sensor, network, distributed, interconnection**

I. Introduction

Before being planted into the ground until the harvest, crops are usually grown inside plant factories, which are equivalent to plant nurseries but inside a building. The operation and management of such facilities are consequently critical as they condition a successful harvest.

Because of its economic importance, improvement and optimisation of plant factories, for instance for rice crops, is an actively researched topic. For example, Huang *et al.* proposed a mathematical model to refine the operation schedule of a plant factory considering the market value and various other properties, such as cultivation duration, of the selected crops [1]. Again focusing on the scheduling issue of plant factories, Yang *et al.* described a heuristic algorithm to maximise the plant factory revenue, taking into consideration additional environmental conditions such as cleaning, maintenance and the use of solar panels [2].

In our own previous research, we have considered the grid layout of a typical paddy field as commonly found in Asia, that is a two-dimensional surface into which young rice crops are evenly planted in a mesh fashion (see Fig. 1), to connect sensors according to the torus network topology [3]. Although this torus connection approach could be applicable to the crops of a plant factory as well, this would require first to carefully position each crop and second to place a sensing device next to each of all crops (or crop cells). Both requirements are labour-intensive, and the latter also induces a non-negligible impact on the plant factory building and operating costs, including maintenance. The approach proposed in this paper is more flexible and in comparison significantly reduces costs in that it does not have such requirements.

Figure 1. A photograph of a typical paddy field (i.e. to grow rice), showing the characteristic grid layout which is suitable for the interconnection of sensing devices according to the torus network topology (two-dimensional torus).

In this paper, aiming at optimising crop growth in a plant factory, we describe a low-cost agricultural system based on a wired network of microcontrollers that enables real-time monitoring of the cultivation environment of crops. While Huang *et al.* focus on a theoretical model to optimise plant factories [1] and Yang *et al.* on the description of a scheduling algorithm to this same end [2], hereinafter we instead consider concrete technical and engineering issues, and present the main idea of a low-cost practical solution for the realisation of such optimised plant factory operation and management.

Although there already exist monitoring systems for such plant factories (e.g. [4]-[6]), our proposal is based on a fully autonomous distributed network which enables smarter monitoring and diagnosis by collecting data from various locations, data which are then shared and analysed amongst sensing devices before, if needed, information is reported to the plant factory operator. Besides, the proposed system is, for instance compared to that of Chen *et al.* [6], minimally invasive (refer to Section III).

Manuscript received December 1, 2021; revised May 10, 2022.

doi: 10.12720/jait.13.4.320-325

The rest of this paper is organised as follows. The introduction of the core components and the logic of the proposal is continued in Section II. Then, in Section III we describe the proposal in detail. Next, the applicability of the proposal is shown in Section IV. Finally, this paper is concluded in Section V.

II. Preliminaries

Our proposal relies on microcontrollers as they are very cheap (e.g. about USD $2.50 for Microchip Technology's ATmega328P as of September 2021), widely available and have a very low power consumption, which makes them suitable as autonomous, self-powered sensing devices. It is common for microcontrollers to be used as sensing devices [7], [8]. So as to achieve such desirable properties, these small controllers rely on low-end processors, typically based on a Reduced Instruction Set Computer (RISC) architecture [9] such as AVR and ARM, and minimal interfacing (i.e. minimal input and output capabilities). They should not be confused with Systems on a Chip, which are more powerful devices [10]. The Arduino Uno and BBC Micro Bit microcontrollers are examples of such devices.

Wired connections are utilized for their advantages over wireless ones: the induced power consumption is minimal since wave (radio) emission is not needed, unlike for wireless networks, and the proposed interconnection mechanism relies on a routing unit whose power consumption is negligible. The main idea of routing solution presented in this paper effectively relies solely on discrete transistor activation, activation whose frequency is as low as desired; it is set to one minute in the detailed descriptions below. The microcontroller unit is also significantly simplified since it includes no wireless component. As a result, the hardware architecture of the sensing device becomes simpler, which in turn reduces the risk of failure and malfunction.

The approach we follow here is that of a distributed network: there is no central machine to control the network. The advantages are a lower hardware cost due to the absence of a controlling node, and maybe more importantly, such a network structure provides a higher tolerance to faults: unlike with a distributed network, in a centralised network if the controlling node (i.e. central controller) fails, the entire network becomes out of order.

III. Methodology

Previous researches, such as that of Huang *et al.* [1], confirm that monitoring and adjusting temperature, humidity, lighting and so on are critical for a plant factory. Acknowledging these agricultural needs, we describe in this section a concrete real-time monitoring system for the crops of a plant factory, as previously presented and motivated.

Definition 1: A monitoring node consists of a microcontroller unit (MCU, typically mounted on a printed circuit board) and a routing unit (a.k.a. switch).

(As just defined, MCU hereinafter denotes the whole MCU–board set, not only the microcontroller chip).

A. The Microcontroller Unit and Its Environment

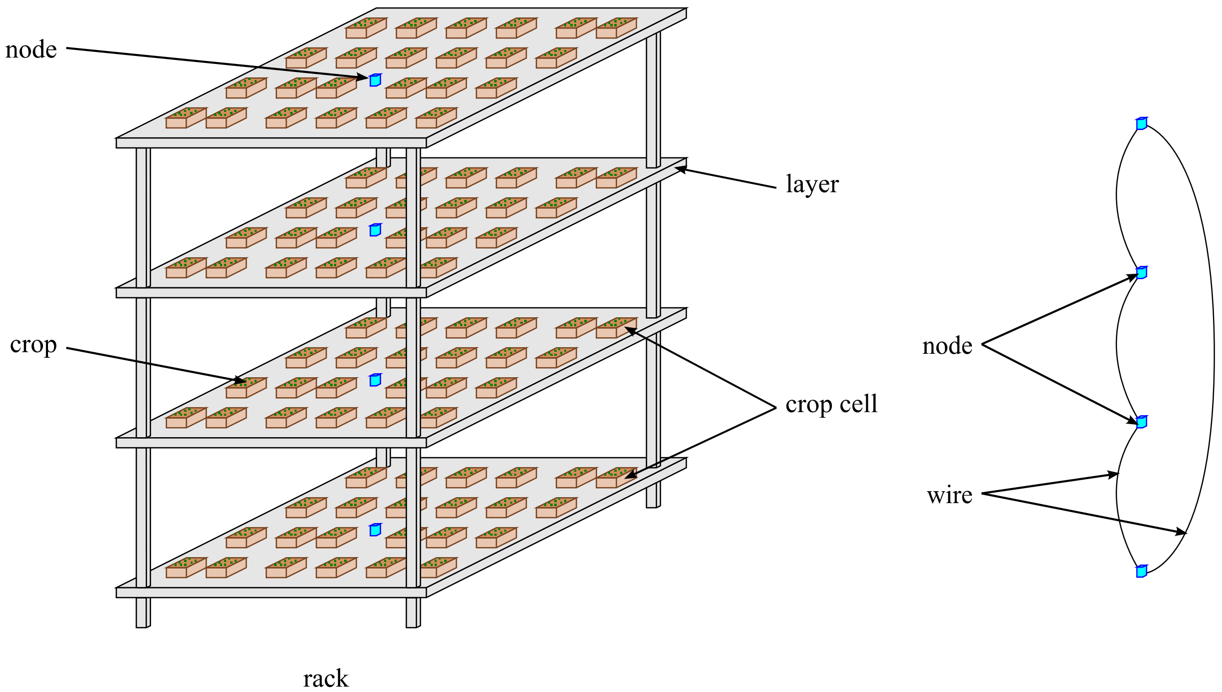
As recalled for instance by Yang *et al.* [2], the usage of racks in plant factories is common. Racks indeed provide a way to efficiently utilise the plant factory space by not only growing crops horizontally on plane surfaces, but also vertically by stacking several layers of crops, for instance crop trays or simply several layers of individual crop cells.

Then, we assign to each layer of the rack one monitoring node equipped with sensing capabilities such as a temperature sensor, a light sensor, a pressure sensor and a humidity sensor, amongst other possible features. The node is physically positioned near the centre of the layer.

The nodes are connected according to the ring network topology: each node has exactly two neighbours, and there exist exactly two (internally) node-disjoint paths between any two nodes of the network [11]. In other words, the monitoring nodes are connected according to a one-dimensional torus network [3], [11]. Because the number of such crop layers is physically limited to a few, the ring network topology poses no performance problem: the network diameter stays small. Besides, data communication between nodes is simplified with such a simple interconnection scheme: there is no need for complex routing algorithms. It can be noted that even in the case of a large number of nodes, there would be no problem with respect to the diameter of the network considering the communication method employed as detailed in Section III-D.

An overview of the proposed monitoring solution is given in Fig. 2a. This figure does not show the connecting wires for the sake of clarity. Node interconnection is instead illustrated in Fig. 2b. A rack of four layers is illustrated, but this number is of course arbitrary and can be freely adjusted upon needs.

Next, we detail the microcontroller unit part of each node; the routing unit is detailed in the next section. Since our proposal relies on standard technologies, the MCU can be realised with various microcontroller chips and various printed circuit boards (PCB). We have selected the ATmega328P chip to make a prototype since it is embedded on an Arduino Uno board, which is open-source and thus comparatively easy to operate; this board is commercially available. The specification sheet provided by the maker of the board gives 68.6 mm × 53.4 mm for the dimensions and a weight of 25 g. The board operating voltage is 5 V (an input voltage of 7–12 V is recommended). Again, this is a prototype board: a smaller one, such as the Arduino Micro board (48 mm × 18 mm, only 13 g) and its ATmega32U4 chip or even smaller the Arduino Nano board (45 mm × 18 mm, only 7 g) which features the ATmega328P chip just as our prototype, could be used for on-site deployment. The ATmega328P chip implements the AVR instruction set architecture (RISC) and contains an 8-bit processor.



(a) (b)

Figure 2. (a) An overview of the proposed monitoring solution: a rack consists of several layers, each including several crops (or crop cells) and one node. (b) Nodes are interconnected according to the ring network topology, which is isomorphic to a one-dimensional torus network.

A photograph of the prototype board is given in Fig. 3. Two sample sensing devices are included in this prototype: a temperature sensor, which is embedded inside the microcontroller, and a photodetector which we have connected to the MCU board. The liquid-crystal display panel has been set up for debugging purposes, and while it can be used for on-site deployment as well as an output interface for the proposed monitoring solution, it is not required.

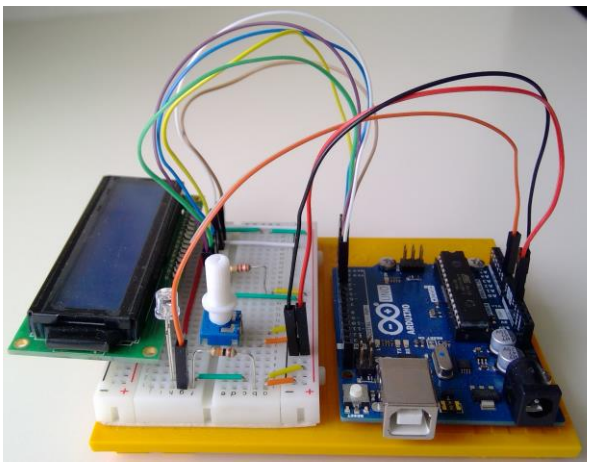


Figure 3. A photograph of the prototype board with two sample sensing devices (a temperature sensor and a photodetector). Initially for debugging purposes, the display panel could also be used as an output interface; it is entirely optional.

B. The Main Idea of the Routing Unit

Data communication between any two adjacent nodes is done through the Serial Peripheral Interface (SPI) of each microcontroller unit. Hence, each wire shown in Fig. 2b consists in practice of four lines: in short, the input line, the output line, the clock line and one control line.

Under the SPI communication model, two types of nodes are distinguished: master devices and slave devices. A master device can exchange data with one slave device. The master device initiates the data transfer. Hence, two versions of the microcontroller code are needed, one for the master nodes and one for the slave nodes. The master–slave role assignment is detailed in Section III-D.

Data is exchanged between the two devices (i.e. the master data is sent to the slave device and simultaneously the slave data is sent to the master device) as soon as the transfer is initiated by the master device. Besides, each node set as master device uses a control line – the slave select (SS) line – to determine with which of its two neighbour nodes it is about to exchange data (it is recalled that each node has two neighbours; refer to Section III-A).

The main idea of the electronic circuit for the routing unit of master nodes is given in Fig. 4 and that of slave nodes in Fig. 5. The two circuits simply rely on Field-Effect Transistors (MOSFET), both P-channel and N-channel. When the control line is driven high (resp. low), communication with the second (resp. first) neighbour is activated. (It is recalled that when the slave select pin is driven high at the slave device, the SPI component of the slave node is passive, and when driven low, the SPI component becomes active [12].) We however emphasise that this is an early design which describes the circuit logic and that further electronic adjustments remain required: we have only partially successfully implemented this circuit idea (simplex communication from the master device towards two slave devices, with one control line and one signal (data) line, without SPI), the activation time of MOSFETs likely requiring additional analysis and adjustments to support the transmission frequency of the SPI component, amongst other possible electronical issues.

Finally, it is recalled that this main idea of routing unit does not require a separate power supply: activation of the field-effect transistors by the microcontroller would suffice.

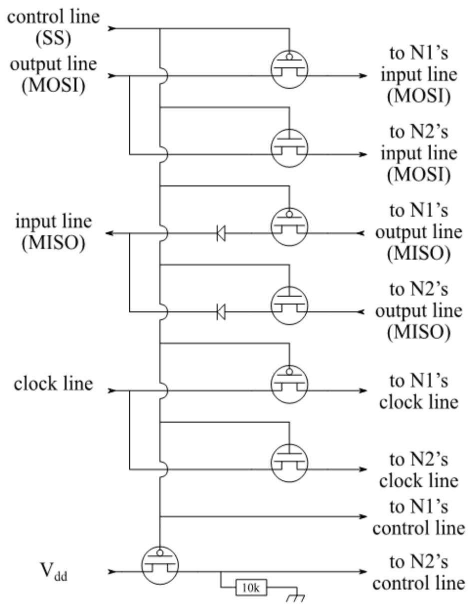


Figure 4. The main idea of the routing unit of master nodes, based on field-effect transistors, with thus no separate power supply needed.

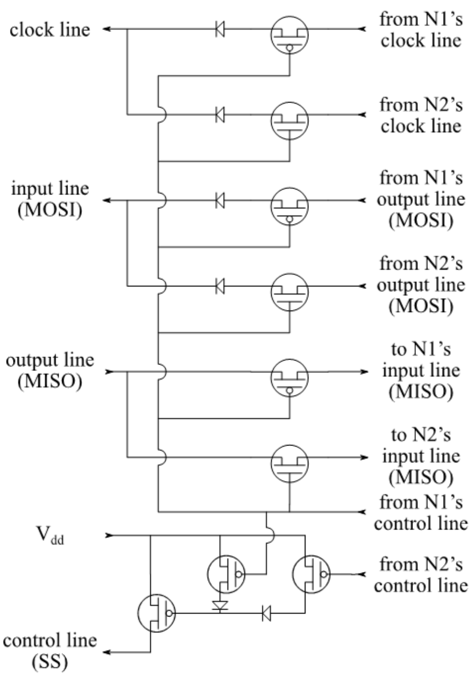


Figure 5. The main idea of the routing unit of slave nodes, based on field-effect transistors, with thus no separate power supply needed.

C. Data Analysis

The main idea of the data analysis process is as follows. Each node gathers data (e.g. temperature, light, humidity, pressure) which correspond to its own layer, and receives those of its two neighbour nodes, that is, data which correspond to the two neighbour layers in the rack.

The data collected are then analysed by each node, and if predefined conditions are met, the node signals itself to the plant factory operator. Node signalling can be, for example, realised with simple LEDs or larger lights, or even with a display panel if detailed information on the current state of the layer is needed.

Concretely, the sensor data of a node *b* are compared with those of its two neighbour nodes, say *a* and *c*. This way, we can detect possible anomalies in the environmental conditions of the rack: if the sensor data (e.g. temperature, humidity) of one layer are significantly different from those of the neighbouring layers, the node signals itself as explained, enabling an appropriate response by the operator. To this end, two threshold values need to be defined at the node, one corresponding to each neighbour node. This threshold value definition depends on the plant factory architecture (e.g. whether racks and layers share the same, global environmental conditions) and cultivated crops themselves.

For example, the signalling conditions at node *b* can be defined as satisfied when

$$(2v(a) + v(c)) / 3 \leq v(b) \leq (v(a) + 2v(c)) / 3$$

is not satisfied, where $v(u)$ denotes the sensing data value of the node *u*. Here, we have applied linear interpolation between $v(a)$ and $v(c)$, with the two threshold values, corresponding to the nodes *a* ($t = 0$) and *c* ($t = 1$), being set at $t = 1/3$ and $t = 2/3$, respectively.

This is illustrated in Fig. 6. The two threshold values are predefined at parameters $t = 1/3$ and $t = 2/3$ of the linear interpolation between the values $v(a)$ and $v(c)$ of the nodes *a* and *c*, respectively (this is freely adjustable). If $v(b)$ the value of the node *b* falls within the two bounds induced by these two interpolation parameters, then the environmental conditions are assumed satisfactory, and otherwise, that is if $v(b)$ falls outside of the two bounds, the node *b* signals itself to the plant factory operator for checking and, if needed, for refining the environmental conditions (e.g. adjusting the surrounding temperature and humidity).

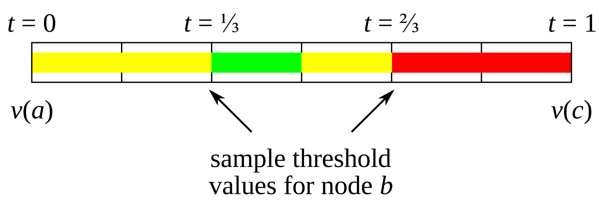


Figure 6. Node signaling conditions: here, the bounds are set at parameters $t = 1/3$ and $t = 2/3$ of the linear interpolation between the values $v(a)$ and $v(c)$ of the nodes *a* and *c*, respectively. Outside of this range, the node *b* signals itself to the plant factory operator.

The predefined threshold values can be separately set on each node. But for facilitated deployment, the microcontroller code can be the same for each of all nodes. In this case, if each layer has its own lighting, etc. equipment, then there is nothing to do. Otherwise, that is all the rack layers share the same, global environmental conditions, it should be noted that because nodes are connected according to the ring topology as explained, two of them, precisely the nodes of the topmost and bottommost layers are at a greater physical distance. So, for each rack, these two nodes are more likely to signal themselves than others. This is not problematic since an expected behaviour: this can be considered as increased sensibility of the two extremal nodes of a rack.

D. Data Exchange Algorithm

Thanks to the ring network topology, the proposed system can rely on a minimal data transfer protocol, thus easily implemented in the microcontroller software code and less prone to errors.

It is assumed that communication channels are duplex (i.e. two-way). This is the case with the SPI communication model provided by the microcontroller and which we have selected to build the node routing unit (see Section III-B). Simple communication is achieved as follows: each node is assigned an ID in the form of a natural number, such that we can count from node 0 and traverse completely the ring of nodes by each time adding 1 to the node ID. Hence, we have even and odd node IDs; we simply say even and odd nodes. For the sake of simplicity, we assume that the number of nodes is even. All the even nodes are started simultaneously first ($t = 0$) and one minute later, all the odd nodes are started simultaneously ($t = 60$), with t in seconds. This is easily feasible since nodes are started by simply powering them on. So, two power strips of multiple sockets, both having a switch, would suffice.

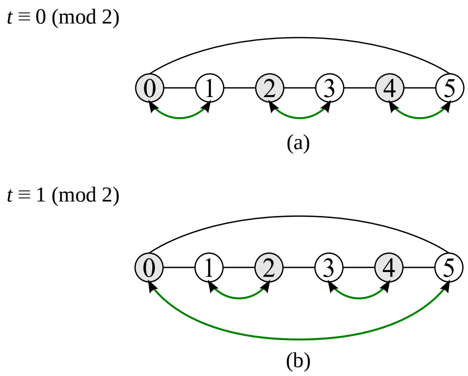


Figure 7. Data exchange mechanism when the number of nodes is even. (a) $t \equiv 0 \pmod 2$; (b) $t \equiv 1 \pmod 2$. The node ID is shown inside each device (master nodes are greyed).

Even nodes are set as master nodes, and odd ones as slave nodes. At $t \equiv 0 \pmod 2$, even nodes use the control line of their routing unit to select the lines with their +1 neighbour (thus an odd node) and exchange data according to the SPI. At $t \equiv 1 \pmod 2$, even nodes use the control line of their routing unit to select the lines with their −1 neighbour (thus an odd node) and exchange data according to the SPI. This is illustrated in Fig. 7.

As a result, all data transfers are done in a synchronous data exchange mechanism which is minimal in terms of complexity and thus cost. It should be noted that data exchanged in this way are limited to one or a few bytes. This is enough to transfer at once, for example, both light and temperature information: after calibration, only offset values (i.e. difference from the initial value) are sent, thus requiring only a few bits. Now, it is important to note that because no broadcasting operation is needed under this operational scheme, the number of nodes involved can be as large as desired: it will not impact the performance of the system.

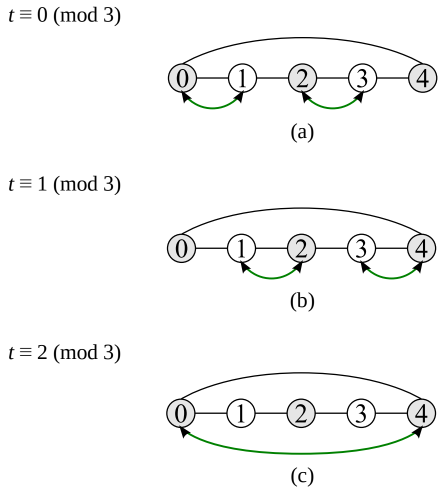


Figure 8. Data exchange mechanism when the number of nodes is odd. (a) $t \equiv 0 \pmod 3$; (b) $t \equiv 1 \pmod 3$; (c) $t \equiv 2 \pmod 3$. The node ID is shown inside each device (master nodes are greyed).

In the case the number of nodes is odd, the above communication scheme can be adjusted as follows. At $t \equiv 0 \pmod 3$, even nodes except node $n - 1$ (e.g. node 4 in our figures) use the control line of their routing unit to select the lines with their +1 neighbour (thus an odd node) and exchange data according to the SPI. At $t \equiv 1 \pmod 3$, even nodes except node 0 use the control line of their routing unit to select the lines with their −1 neighbour (thus an odd node) and exchange data according to the SPI. At $t \equiv 2 \pmod 3$, node 0 uses the control line of its routing unit to select the lines with its −1 neighbour (thus an even node as well) and exchanges data according to the SPI. This is illustrated in Fig. 8.

So, for greatest simplicity an even number of nodes is recommended, but the proposed system would correctly operate as well with an odd number of nodes.

IV. Evaluation

The proposed monitoring solution has been designed to minimise both its production cost and its power consumption. In this section, we evaluate the proposed system with respect to these two criteria.

A. Power Consumption

As per the Arduino Uno board specification sheet [13], there is a maximum current of 20 mA per I/O pin, so, assuming that, say, five I/O pins are used (four for the routing unit, plus another one for, say, signalling purposes) we can assume that one node draws a current of at most 100 mA. The microcontroller unit operating at 5 V, this thus gives a maximal power consumption of 5 × 0.1 = 0.5 W, multiplied by the number of nodes, say five in a rack, we obtain only 2.5 W in total and in the worst case for the five nodes. This can be compared to the power consumption of a computer CPU: for example, the power consumption of a single Intel Core i7 processor is in the range from 35 W to 165 W [14].

B. Production Cost

The cost is also minimal: for one node, the microcontroller unit is about $23 and enough field-effect transistors about $10 (as of September 2021, all prices in USD). The cost of additional components such as diodes is negligible. And this can be further reduced by transitioning from a node prototype to the final node design. This is significantly lower than, for instance, the cost required by the system proposed in [5] (the features provided by the respective systems of course differ) as at least one dedicated monitoring computer is required, in addition to the ZigBee nodes. We have not considered the sensor costs since negligible.

Finally, it is recalled that since our approach relies on a distributed network, it features a higher fault tolerance than a centralised network such as that of [5].

V. Conclusion

The optimisation of the operation of factory plants is an important, and thus well researched topic. Several theoretical models and monitoring systems have been described to this end in the literature. In this paper, we have described and evaluated a low-cost monitoring solution for a plant factory. Compared to conventional approaches, one key aspect of our proposal is the distributed nature of the network of sensing nodes. By relying on the ring network topology, we were able to achieve a high system fault tolerance and a simple data exchange mechanism.

Regarding future works, after completing the analysis and implementation of the described main idea of routing unit, we plan to transition from a prototype board to a node design that is suitable for on-site deployment. Besides, we are going to investigate if even cheaper microcontroller boards could be employed.

Conflict of Interest

The author declares no conflict of interest.

Acknowledgment

This research was partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under grant no. 19K11887.

References

[1] K. L. Huang, C. L. Yang, and C. M. Kuo, "Plant factory crop scheduling considering volume, yield changes and multi-period harvests using Lagrangian relaxation," *Biosystems Engineering*, vol. 200, pp. 328-337, Dec. 2020.

[2] C. L. Yang, S. J. Huang, and C. H. Ang, "Recursive heuristic scheduling method for multi-crop plant factory with solar panel roof," *Computers and Electronics in Agriculture*, vol. 165, p. 104941, Oct. 2019.

[3] A. Bossard and K. Kaneko, "Cluster-fault tolerant routing in a torus," *Sensors*, vol. 20, no. 11, p. 3286, Jun. 2020.

[4] A. P. Montoya, F. A. Obando, J. A. Osorio, J. G. Morales, and M. Kacira, "Design and implementation of a low-cost sensor network to monitor environmental and agronomic variables in a plant factory," *Computers and Electronics in Agriculture*, vol. 178, p. 105758, Sep. 2020.

[5] F. Ijaz, A. A. Siddiqui, B. K. Im, and C. Lee, "Remote management and control system for LED based plant factory using ZigBee and internet," in *Proc. 14th International Conference on Advanced Communication Technology*, PyeongChang, 2012, pp. 942-946.

[6] W. T. Chen, Y. H. F. Yeh, T. Y. Liu, and T. T. Lin, "An automatic plant growth measurement system for plant factory," *IFAC Proceedings Volumes (5th IFAC Conference on Bio-Robotics; 27–29 March, Sakai, Japan)*, vol. 46, no. 4, pp. 323-327, 2013.

[7] P. R. Nagarajan, B. George, and V. J. Kumar, "Improved single-element resistive sensor-to-microcontroller interface," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 10, pp. 2736-2744, Jun. 2017.

[8] A. Bossard, "Autonomous on-chip debugging for sensors based on AVR microcontrollers," *Journal of Sensor Technology*, vol. 11, no. 2, pp. 19-38, Jun. 2021.

[9] D. A. Patterson and D. R. Ditzel, "The case for the reduced instruction set computer," *ACM SIGARCH Computer Architecture News*, vol. 8, no. 6, pp. 25-33, Oct. 1980.

[10] Y. K. Chen and S. Y. Kung, "Trend and challenge on system-on-a-chip designs," *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, vol. 53, no. 1-2, pp. 217-229, Nov. 2008.

[11] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks*, San Francisco, CA, USA: Morgan Kaufmann Publishers, 2002.

[12] *ATmega48A/PA/88A/PA/168A/PA/328/P megaAVR Data Sheet*, DS40002061A, Microchip Technology, Chandler, AZ, USA, 2018.

[13] UNO R3, Arduino, Monza, Italy. [Online] Available: https://docs.arduino.cc/hardware/uno-rev3

[14] *Intel Core i7 Desktop Processors Comparison Chart*, Intel, Santa Clara, CA, USA, 2012.

Antoine Bossard is an Associate Professor of the Graduate School of Science, Kanagawa University (Japan) and a part-time lecturer at Tokyo University of Agriculture and Technology (Japan). He received his B.E. and M.E. degrees from Université de Caen Basse-Normandie (France) and his Ph.D. degree from Tokyo University of Agriculture and Technology. His research is focused on graph theory, parallel and distributed computing and interconnection networks, addressing issues such as routing and fault tolerance. Recently, he has started considering applying his research on interconnection networks to smart agriculture. For several years, he has also been conducting research regarding Chinese characters, their representation and processing on computer systems. He is a member of ACM, ISCA and TUG.