A Study of Advanced Reservation Scheduling Structure for Smartfarm Actuator Control

Anna Yang¹, Song-Gang Kim², Hee-Dong Park³

Abstract

With the growing aging and decline in population in rural areas, smart farm system-based on Information Technology (IT) and Internet of things (IoT) convergence are attracted as an alternative to improve agriculture productivity and self-sufficiency. The smart farm control system, an option to solve food problems in the future, manages various sensors and actuators in the greenhouse. We propose in this paper an advanced reservation scheduling structure for base-level smart farm automation having environmental condition dependency. Using this structure briefly, actuators are managed by a corresponding thread with a scheduling list that contains sensor-based constraint information. If there is a scheduling list for each actuator, the configuration items in the list are continuously checked including control mode and availability and operation time of the actuator. The operation condition should be examined to avoid the repetition of a command delivered. The pre-defined sensor list which is relevant to the actuator should be referred to for adapting quickly to sudden changes or constraints in the environment. With this system, the smart farm actuators can be controlled and managed and controlled efficiently with consideration of environmental sensor values.

Keywords : Actuator, Control, Dynamic, Configuration, Hybrid, Control, Reservation, Scheduling, Smartfarm

1. Introduction

The worldwide rapid progress of Information and Communication Technology (ICT) has dramatically revolutionized the quality of life for human needs. For instance, it is easily available to realize a service to support better connectivity and to guarantee convenience to the people such as smart cities, smart cars, IoT, and smart factories. Now, these can be the main solution for working out many economic, social, and cultural problems. The importance of food...
security is becoming critical and the development of a smart farm system is a possible alternative to prevent the decrease in agricultural productivity caused by the aging of the farming population and reliance on foreign products. There had been a traditional control method for a farm management system that has a reservation control using a hardware timer. Recently with the advancement in agricultural ICT technology, the adaptive farm automation function using environment sensor data inside a greenhouse can be possible. However, in Korea, because smart farm technologies have been developed by the leading small companies, there are many problems that occurred. For instance, it is frequently reported an actual damaged case by unpredictably caused unknown systematic errors of the smart farm system which was applied in the greenhouse. Also, because most existing commercial products for the smart farm have been developed without consideration of interoperability, most developers are not having any interest in the standardization for its scalability and compatibility. The research on the standardization of smart farm technology has been slow[1]. This brought out a disappointing result to the farmers who had been adopting the smart farm system on their greenhouses. Because the system was not compatible with interfaces of environmental sensors and actuators, it is not easy to apply additional sensors or actuators for their farm. In other words, interfaces applied in a specific smart farm system cannot be applied to other smart farm systems due to their low supportability by the manufacturer. This is even unavailable now due to the interruption or abandonment of the development of the smart farm system. Therefore, farmers cannot receive technical assistance anymore. Due to these problems, most farmers have un-positive stances to introduce a new system for their greenhouses. Nevertheless, much research and development are undergoing because the smart farm is the only option to cultivate crops efficiently as before.

For instance, a real-time monitoring system has been developed by transmitting data such as humidity and temperature from environmental sensors in the greenhouse via the internet[2] and a system for early warning has been developed as the result of data analysis of unusual circumstances change[3]. The use of an algorithm for extracting the environmental state for monitoring inside the greenhouse and algorithms to reactively control actuators depending on sensor values has been developed for the smart farm automated[4-7]. A study of the actuator control system in the greenhouse had been done for giving the way of supporting optimized growing conditions by an IoT-based sensor data collection and analysis using a scalable and configurable structure[8][9]. A general smart farm system for monitoring and controlling the farm has been developed and applied focusing on a dynamic control using environmental sensor data in the smart farm. However, after the growing condition data has been collected
large enough, static control can be efficiently and reliably used, except for unpredictable accidents such as natural disasters and crimes. Even so, a study of static control like a reservation scheduling control function has not been doing enough. Therefore, the study on reservation scheduling control should be conducted for establishing its data structure and function in terms of standardization. We previously proposed the reservation scheduling control method so that the minimum static control is available. However, it is not enough to utilize for adapting quickly to environmental conditions changes. A sensor data dependency should be considered to guarantee to properly respond to the sudden change in environments such as a shower, hail, strong wind, and a typhoon.

In this paper, an advanced reservation scheduling method for the smart farm system is introduced. With this, the actuator in the smart farm can be controlled by time intervals and specific time settings. In section 2, the structure and items for reservation scheduling are introduced and described which includes workflow and message processing diagrams. Concluding in section 3, more future work will be discussed.

2. Structure for Reservation Scheduling

The reservation scheduling processing is a basic control function of the smart farm system. Using this process, actuators in the greenhouse can be easily controlled by the setting of time to operate. Two modes are supported, one is the configuration of a specific time with duration to be active and the other is the time of interval with duration. The advanced function with a reservation scheduling dependency can be supported by pre-configuration of constraints referencing environmental condition data in the sensor list. This is the method to more reliably adjust the environment of the greenhouse by the setting constraints according to specific sensor values which have a range. For example, if we assume that a farmer sets the time to be driving a water pump per 5 hours on that day, thundershowers may hit, unfortunately. In this case, the humidity would be increasing. In this case, it is better not to activate the water pump. To support this, the structure and workflow for the reservation scheduling process should be defined. In this section, we will introduce and describe items for the advanced reservation scheduling control and its workflow.

2.1 Items for Reservation Scheduling

There are some considerations for the reservation scheduling structure. The first will be the time scheduling mode. To reserve the time to be active, we support some modes such as
A Study of Advanced Reservation Scheduling Structure for Smartfarm Actuator Control

TIME_ON_DEMAND, EVERY_SECOND, EVERY_MINUTE, PERIODIC_HOUR, and PERIODIC_DAY. Two different modes are supported, the one is the on-demand mode, and the other is the periodic control mode (every second, every minute, and every hour). Secondly, the dependency or priority is assigned by considering the association of the scheduling function for the status and interface of a smart farm system. This indicates that the actuator management can be supported by the setting of a specific time zone and the status of sensors. It is needed for preventing a system collision as well as ambiguity with the automatic control system. Thirdly, objects to be scheduled should be defined. The objects can be one or more actuators and, in some cases, sensors in a smart farm system. All of the objects should be devices that can be pre-configured and operated. Lastly, the sensor list corresponding to each specific actuator as well as its constraint condition should be properly packed. To support the advanced reservation scheduling, as shown in Table 1, there are some configurable items such as interval, start time, speed, direction, and duration. There are also some more configurable items to adapt more easily against the unpredictable external environment conditions. Each of the actuators can be enabled or disabled, and the constraints for the actuator are derived by referring to the max/min value of sensors and deviation value from the pre-configured sensor value. These configurable items are related to actuator data.

Table 1: Items for Reservation Scheduling Function

<table>
<thead>
<tr>
<th>Items</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>Identification(32-bit)</td>
<td>Actuator identifier</td>
</tr>
<tr>
<td>mode</td>
<td>TIME_ON_DEMAND</td>
<td>actuator activated on specific time (reservation mode)</td>
</tr>
<tr>
<td></td>
<td>EVERY_SECOND</td>
<td>actuator activated in every second</td>
</tr>
<tr>
<td></td>
<td>EVERY_MINUTE</td>
<td>actuator activated in every minute</td>
</tr>
<tr>
<td></td>
<td>PERIODIC_HOUR</td>
<td>actuator activated in every hour</td>
</tr>
<tr>
<td></td>
<td>PERIODIC_DAY</td>
<td>actuator activated in every day</td>
</tr>
<tr>
<td>enable</td>
<td>1/0</td>
<td>enable/disable</td>
</tr>
<tr>
<td>depend</td>
<td>0 - n</td>
<td>association(dependancy)</td>
</tr>
<tr>
<td>op data</td>
<td>interval</td>
<td>sec., min., hour., hh:mm</td>
</tr>
<tr>
<td></td>
<td>start_time: 2018-03-02 00:00:00</td>
<td>start time</td>
</tr>
<tr>
<td></td>
<td>speed: 0 - 255</td>
<td>speed</td>
</tr>
<tr>
<td></td>
<td>direction: 0/1</td>
<td>direction</td>
</tr>
<tr>
<td></td>
<td>duration: ms</td>
<td>duration</td>
</tr>
<tr>
<td>scheduled data</td>
<td>sid</td>
<td>sensor identifier</td>
</tr>
<tr>
<td></td>
<td>enable</td>
<td>sensor availability</td>
</tr>
<tr>
<td></td>
<td>mode</td>
<td>mode for deriving the range of actuator driving depending on a sensor value</td>
</tr>
<tr>
<td></td>
<td>mvalue</td>
<td>median value of sensor</td>
</tr>
<tr>
<td></td>
<td>hysteresis</td>
<td>variance depending on the median value</td>
</tr>
</tbody>
</table>
2.2 Data structure

To schedule what actuator should be working, some information should be configured by the user or system manager. The configuration for the advanced reservation scheduling is constructed by values defined in Table 1. The schedule list(schd_list) used for actuator constraints can be more than one. The configuration can be done by not only using the configuration file but also setting on-the-fly. In Table 2, JavaScript Object Notation (JSON) objects to be configured for scheduling are presented. JSON is a lightweight marked-up language and is quite quickly parsed, and easily dealt with.

[Table 2] JSON Objects for Reservation Scheduling

```json
{
  "target": Actuator/Sensor ID
  "depend": dependency
  "enable": 0/1
  "mode": reservation mode
  "interval": period of event (in one-minute intervals)
  "stime": starting time of operation of the actuator (in this case, interval should be "0")
  "speed": drive speed pr on/off(255/0)
  "direction": direction
  "duration": The amount of elapsed time between start and end (ms)
  "sen dependancy": number of sensor
  "schd_list":
  [  
    {
      "sid": Sensor ID
      "enable": logic operation
      "mode": Sensor condition mode
      "mvalue": Median value
      "hyst": Hysteresis value
    },
    ......
  ]
}
```

The list for the reservation scheduling is processed by the type of linked list. Each scheduling list of each actuator is executed by cycling through the checking repeatedly. In Fig. 1, the data structure for the advanced reservation scheduling is presented. Above all, if there is a configuration the parsing is processed. With the result, each parsed data is stored in each data structure depending on the actuator identifier. Through this process, the data structure is organized and for accessing the resource, the actuator can be activated by the value of The list
A Study of Advanced Reservation Scheduling Structure for Smartfarm Actuator Control

for the reservation scheduling is processed by the type of linked list. Each scheduling list of each actuator is executed by cycling through the checking repeatedly. In Fig. 1, the data structure for the advanced reservation scheduling is presented. Above all, if there is a configuration the parsing is processed. With the result, each parsed data is stored in each data structure depending on the actuator identifier. Through this process, the data structure is organized and for access to the resource, the actuator can be activated by the value of configurations items. To permit data independence when transmitting from driver to actuator, the control is divided as separate space on the sidelines of data relating to the setting for each scheduling list. This is efficiently utilized for transmission via MODBUS, Inter-Integrated Circuit (I2C), and RF (Radio Frequency) communication channels.

![Data Structure of Reservation Scheduling List](image)

The message processing flow for reservation scheduling is shown in Fig. 2. When there exists a request for reservation scheduling from a network or internal operation, the message is parsed and analyzed to process the request.

The types of messages include addition, deletion, and modification of scheduling information. The payload according to the topic is analyzed and processed by JSON name/value items when using Message Queuing Telemetry Transport (MQTT) protocol. Depending on the parsed JSON objects and values, the reservation scheduling for auto-control can be processed to generate and reply to messages, and modifying, adding, and deleting functions are executed according to the command.
2.3 Work Flow of Reservation Scheduling Control

As mentioned above, to support data independence, the structure for control data should be designed separately and dealt with independently from setting data by scheduling list. The configurations for each actuator can be done for the scheduling. The smart farm system requires accuracy rather than real-time or high performance. The control command could be effective when the status information and the control command are not the same as the previous ones to eliminate the redundant process. This feature can provide more advantages for the system performance and efficient use of system resources.

Fig. 3 illustrates the workflow of the reservation scheduling operation. In the case of on-demand mode, the interval item is ignored while the start_time is referred to. When using periodic mode which supports sub-modes such as EVERY_SECOND, EVERY_MINUTE, PERIODIC_HOUR, and PERIODIC_DAY, item start_time is ignored, and the interval is referred to. There exists a related sensor list for each reservation scheduling list. Even though the actuator is scheduled to be driven at a specific time, it may not be actuated depending on the corresponding sensor values. In other words, the actuator control command will be generated only when all the prerequisite conditions have been met. Due to the possibility of the ability to set more than one sensor, several sensors can be referred by their priority. In Fig. 3, because there are actuators, configuration settings should exist. So, times of calculation would be done. There also could be a k setting to restrict driving-specific actuators.

The configuration of static and dynamic control can be easily deleted, modified, and added.
A Study of Advanced Reservation Scheduling Structure for Smartfarm Actuator Control

by the user. This indicated that the modification of scheduling lists can be done during system operation so that this information can be applied immediately to the system configuration. To manage and configure a remote client or server, it needs network connections and will be executed by several threads such as network thread as well as configuration threads. To prevent the race condition of shared resources, the critical section, mutual exclusion (MUTEX), signal/wait, and semaphore could be considered.

![Diagram of Reservation Scheduling Operation]

[Fig. 3] Process of Reservation Scheduling Operation

3. Conclusion

In this paper, the advanced reservation scheduling structure for the smart farm system is introduced. Using the reservation scheduling control, base-level automation is available to control actuators in the smart farm. We also provide designing the structure to control actuators in the greenhouse with constraints depending on relevant sensor values. The message processing for advanced reservation scheduling has been described. After parsing JSON formatted configuration messages through the network interface, the reservation schedules can be processed by the actuator thread. It can be used for adaptively controlling the system referring to environmental data. Concluding, the structure can support scalability as well as compatibility. In the future, it is necessary to develop algorithms considering the performance based on this structure to develop standardization.
References


