

Mixed Open-loop / Closed-loop Control of a Slow Sampled Water Supply System

Petr Chalupa
Faculty of applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
chalupa@utb.cz, ORCID: 0000-0002-9229-7483

Jakub Novák
Faculty of applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic
jnovak@utb.cz

Filip Janota
Faculty of applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic

Martin Vláčil
Faculty of applied Informatics
Tomas Bata University in Zlin
Zlin, Czech Republic

Abstract—This paper deals with control of stand-alone water supply system for a small clearing-based settlement. The sensorics system works with very long sampling period and therefore the system is so-called slow sampled. The paper presents a design of control algorithm as well as hardware and software design of the controller itself.

Keywords—real-time control, slow sampled system, water supply system

I. INTRODUCTION

The paper builds on an article [1] where a liquid level monitoring system for a remote reservoir was presented. The monitoring system is aimed at accurately measuring liquid levels in remote reservoirs. The goal was to achieve high precision while maintaining low power consumption and cost-effectiveness in the final system design. The monitoring system is powered by batteries and therefore electricity consumption is crucial. To reduce power consumption a long sampling time is used and therefore classical closed-loop control cannot be applied.

From the control point of view, the system is so-called slow sampled. Slow sampled systems are characterized by their inability to capture fast dynamics due to insufficient sampling rates, which can lead to challenges in system identification and control design [2]. There are many various types of slow sampled systems and different approaches were proposed for their control.

Recent advancements in control methodologies have focused on multirate sampling strategies, which allow for different sampling rates for slow and fast dynamics. This approach can enhance the performance of closed-loop systems by ensuring that critical fast dynamics are adequately captured while reducing the computational burden associated with high-frequency sampling [3]. For instance, Munje and Patre propose a two-stage multirate state feedback control design that optimally balances the sampling rates for slow and fast eigenvalue modes, thereby improving system performance [3]. Similarly, hidden Markov model-based filtering techniques for slow-sampling singularly perturbed systems was explored, demonstrating the effectiveness of advanced filtering methods in managing slow dynamics [4].

Event-triggered control strategies have also gained traction in the context of slow sampled systems. These strategies aim to minimize communication and computational load by triggering control updates based on the state of the

system rather than at fixed intervals. Dynamic event-triggered output feedback control for networked nonlinear uncertain systems was proposed, emphasizing the practical implications of such approaches in real-world applications [5]. Additionally, investigation of dynamic event-triggered sliding mode control specifically for slow sampling singularly perturbed systems was performed highlighting the robustness of sliding mode control in the presence of slow dynamics [6].

Another significant aspect of controlling slow sampled systems involves the use of advanced monitoring techniques. For example, slow feature analysis (SFA) has been employed to extract slow-changing features from fast-varying data, which can be particularly useful in process monitoring applications [7]. This article demonstrates the application of SFA in fault detection for dynamic processes, showcasing its effectiveness in identifying slow features that may indicate system anomalies. Furthermore, article [8] propose a slow-varying batch process monitoring method based on canonical variate analysis, which enhances the detection of faults in complex systems.

In the context of practical applications, slow control systems are integral to various experimental setups, such as the PandaX-III experiment, where a centralized database collects data from multiple subsystems [9]. The development of slow control packages for experiments like Belle II further illustrates the importance of effective monitoring and control in high-energy physics applications [10]. These systems often rely on modular designs to facilitate data sharing and improve system reliability.

Overall, the control of slow sampled systems encompasses a diverse range of methodologies and applications, from multirate sampling and event-triggered control to advanced monitoring techniques. As the complexity of systems continues to grow, the development of robust control strategies that can effectively manage slow dynamics will remain a critical area of research.

The paper is organized into the following sections: Section II presents properties and configuration of the current monitoring system, Section III describes requirements for the control system, Section IV proposes control algorithm for the slow sampled water supply system, Section V describes hardware design of the controller as well as its software, Section VI contains results and Section VII concludes the paper.

II. INITIAL STATE OF THE WATER SUPPLY SYSTEM

The monitoring system proposed in [1] has been used for several years in water supply system of a small settlement consisting of approximately 20 consumers. Its simplified scheme is presented in Fig. 1.

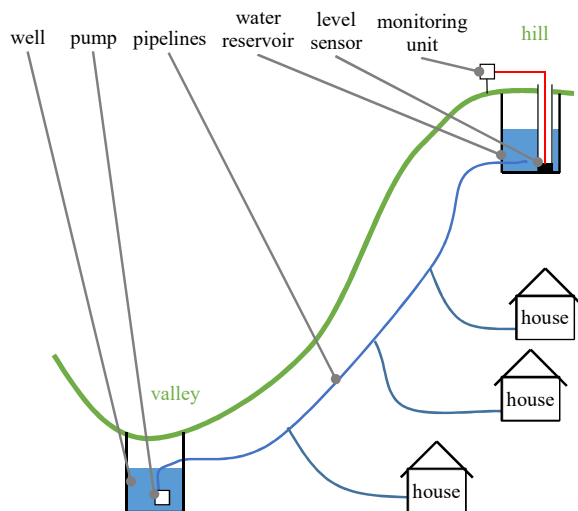


Fig. 1. Simplified scheme of water supply system [1]

The water is periodically pumped from the well in the valley to a water reservoir on the hill. Subsequently water flows through the same main pipeline to individual consumers (houses). The monitoring unit attached to the reservoir periodically measures water level and sends values to the cloud database using wireless communication. There is no wired electricity distribution available for monitoring unit. The monitoring unit is powered from batteries and therefore power consumption is crucial. To minimize power consumption the measurement unit is configured as follows:

- by default, all parts are in sleep mode (if possible)
- water level measurement part is activated every 10 minutes, measures the level and stores it locally
- communication part is activated every 60 minutes (once per 6 measurements) and sends the locally stored measurements to cloud database using mobile network

This configuration means that the measurement unit communicates with “external world” only once per hour. This configuration has been successfully used, particularly for fault detection. There are two basic types of fault that can be detected:

- water level below a defined limit
- burst detected, i.e. water level has decreased rapidly during last two measurement (20 minutes)

Both these faults can be detected locally by measurement unit and if the fault is detected, the measurement unit can send a message to system operator. An SMS message is used in this case.

Water pumping from the well is performed by electric pump without control of motor rotations – an on/off control is used. In the initial state, control of the pump is performed by

simple time switch with week scheduling period. This timer can switch the pump on and off up to 10 times each day. The main drawback of this solution is that it has to be configured locally. This time switch can be also used to manually start pumping.

The only feedback of the system is its human operator. The operator can change the period of pumping according to expected consumption. The pumping was configured as follows:

- pump 6 times per day: 0:00, 4:00, 8:00, 12:00, 16:00 and 20:00
- pump for specified number of minutes

The duration of each pumping has been adjusted several times per year to correspond to average consumption in the given season. Consumption during winter is significantly lower compared to summer. Adjusting the pumping time or duration requires presence of human operator, which is both time-consuming and inconvenient, making it impractical for daily adjustments. An example of course of water level during a week is presented in Fig 2. As the reservoir is of cylindrical shape, the water level is proportional to its volume. Red parts of the course correspond to extraordinary consumption – i.e. rapid decrease of the water level.

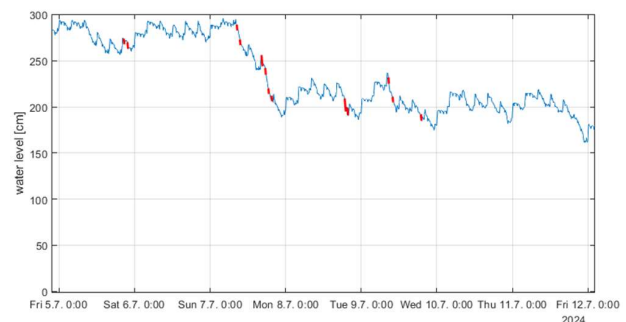


Fig. 2. Example of week course of water level

III. REQUIREMENTS FOR THE CONTROL SYSTEM

The main drawback of the initial state of the control of presented water supply system is the absence of the automatic feedback. The control system of the pump does not have information about current water level. This leads to water wastage because the pumping duration is configured to, first of all, ensure that there will be water in the reservoir for the consumers. The pumping duration is therefore longer than necessary at a particular time which, from time to time, causes the reservoir to overflow.

To ensure better water management, feedback control is required for the presented water supply system. Following boundary conditions were required by system operator and consumers:

- The measurement system is well proofed, reliable, and will be not changed. Also, the sampling and the communication periods will be preserved.
- The control system should be of a low-cost.
- A Wi-Fi internet connection is available at the place of the well, i.e. at the place of the controller.

- The pump will be controlled by on / off signal, frequency changer will not be used.

IV. CONTROL ALGORITHM

The controller can obtain feedback from the cloud database or directly from the web server which retrieves data from the database – see Fig. 3.

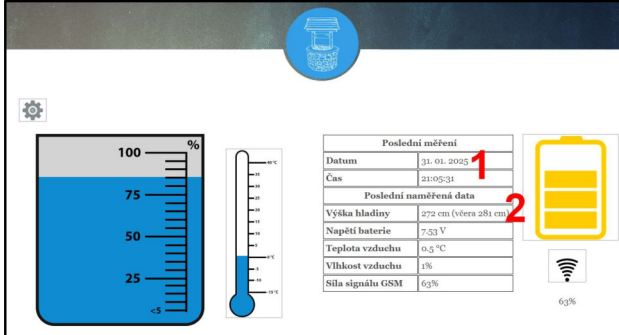


Fig. 3. Web interface of the measurement system

In addition to other information, this web page contains date and time of the last measurement (1) and measured water level (2). It was tested that when pumping is in progress the water level in the reservoir increases by approx. 1.5 cm / min assuming there is no consumption during pumping. This behaviour prevents usage of standard feedback control approach (e.g. PI controller or its discrete version) because water level will increase by up to 90 cm during sampling period. Note that the sampling period is equal to communication period described in Section II (60 minutes).

A. Basic control algorithm

A two-stage control approach was proposed combining closed loop and open loop control:

1. Wait for a new measured sample on the web (Fig. 3) – closed loop phase. The controller periodically accesses the web page to check whether the timestamp of the last measurement has changed. If a new sample is available, it is read, and the controller algorithm proceeds to stage 2.
2. Based on control error (i.e. difference between water level setpoint and its actual level) generate the appropriate control action. The control action will be applied during next sample time of measurement system (i.e. 60 minutes) – open loop phase.

Three approaches for generating the control action signal have been proposed. They are based on control error at each sample time (k), i.e. difference between desired water level height and $w(k)$ and current water level height $y(k)$: $e(k) = w(k) - y(k)$.

- I) Start pumping for a constant time if control error is positive:

$$t_{off}(k) = \begin{cases} 0, & \text{for } e(k) \leq 0, \\ const, & \text{for } e(k) > 0, \end{cases} \quad (1)$$

where $t_{off}(k)$ determines the time to switch the pump off, $const$ is user-defined constant (e.g. 5 minutes), and $e(k)$ is current control error. The pump is switched on at the beginning of stage 2 – at

time $t(k)$, and switched off at time $t(k) + t_{off}(k)$ and remains off till new sample of control variable is detected.

- II) Start pumping for a time period proportional to control error:

$$t_{off}(k) = \begin{cases} 0, & \text{for } e(k) \leq 0, \\ K \cdot e(k), & \text{for } e(k) > 0, \end{cases} \quad (2)$$

where K is used-defined constant corresponding to pumping power, e.g. $K = \frac{1}{1.5} \text{ min/cm}$.

- III) Generate PWM control of the pump. In this case PWM period is user defined (e.g. 10 minutes) and width of PWM pulses are proportional to control error:

$$width(k) = \begin{cases} 0, & \text{for } e(k) \leq 0, \\ K_{PWM} \cdot e(k), & \text{for } e(k) > 0, \end{cases} \quad (3)$$

where K_{PWM} is used-defined constant. It should be defined in that way that the sum of pumping time during the sampling period is the same as in case II).

B. Fault detection and handling

Previous sub-section described ideal case without faults. The most common fault is increase of the sampling period of measurement system. This corresponds to situation where the time of last measurement (see Fig. 3) does not change for a long time. It can be caused by problems with the wireless connection of the measurement system or by its battery discharge. This fault can be easily detected by the controller. Fault handling is performed by switching to open-loop control as described in Section II and periodically attempting to read the new value from the web page.

A similar fault and handling process occur if the web page is unavailable. This corresponds to problem with internet connection of the controller or cloud server error.

The main loop of the control algorithm can be written in the following pseudo-code:

```
repeat_forever
1. level = check_new_sample_from_web()
2. if is_valid(level)
3.   control_error = setpoint - level
4.   control_action = generate_action(control_error)
5.   reset_time()
6.   reset_fault()
7. apply_action(control_action)
8. if time > time_threshold and not fault
9.   control_action = generate_safe_action()
10. set_fault()
11. if user_input()
12. break
```

The function `check_new_sample_from_web()` accesses the web page (see Fig 3) and returns valid level only if the web can be accessed and a new value of water level is available. If time from the last time reset exceeds a threshold (e.g. 1.5 x sampling period of measurement system), controller switches to safe mode where action signal is independent from water level.

V. CONTROLLER HARDWARE AND SOFTWARE

A. Hardware

The purpose of the device is to control the level of the remote reservoir. The data required for the control is obtained by using values from a web page. The device is required to have control and display elements, and an output signal to control the pump. The power supply circuit for the whole system is also an essential part. The microcomputer system will therefore consist, apart from the microcomputer itself, also of a display device in the form of an LCD display, control buttons, a Wi-Fi module for communication with the website or for reading data, a power supply circuit for all necessary components and a galvanically isolated power output for pump control (relay). Possibly also a backup battery in case of power failure.

An important component for our application is the ability to connect to the Internet, so a microcomputer with Wi-Fi module has to be chosen. A suitable candidate is, for example, the ESP 8266 NODE MCU. The device has 10 input/output pins, which is sufficient for our case and is fully compatible with the Arduino IDE development environment [11].

For power supply, a coaxial connector was used, to which 6-20 VDC can be applied. It is used to power all components used in the system. It is equipped with LED indication and a reversible short-circuit fuse. The bridge consists of an LM7805 linear regulator which reduces the output voltage to 5 VDC. A combination of faster ceramic and larger electrolytic filter capacitors are used to smooth out the output voltage ripple that occurs in the voltage regulator. An LED with a biasing resistor to limit the current through the diode is used to indicate the on/off state of the device and the possible discharge of the capacitors after shutdown. The circuit also includes a switch to turn the circuit on and off.

The input circuit for the control buttons consists of disconnectable pins, an LED with a biasing resistor to indicate pressing, and a pull-down resistor connected to GND to limit the current flowing to the microcomputer. The pushbuttons are then connected to digital pins 1, 2, and 3 which take the values log 1 and log 0. The buttons are used to control the program.

The output circuit consists of a terminal block to which the pump will be connected, which will be used to control the level using the output signal. The relay is switched by the digital output of the microcomputer. Since the output current at the digital output is not sufficient, an NPN transistor is used to trigger the relay, which amplifies the signal and also allows a higher voltage to be used for triggering than 5 VDC. The rectifier diode is used to discharge the energy that has accumulated on the relay coil after the relay has been turned off, and to avoid large fluctuations in current that could damage the transistor. The main reason for using a relay is to galvanically isolate the low-current device from the power part, which is also expected to be supplied with alternating current.

A 20x4 LCD display was used to display the status and control the device. The display is connected via I2C bus to save the number of digital pins required for connection (from 10 to 4 including power supply). The I2C bus reduces the number of wires required, for non-high-speed devices, and is based on bidirectional full-duplex data transfer, where one

signal wire is called SDA (serial data) and the other SCL (serial clock).

The overall scheme of the board connection is presented in Fig. 4.

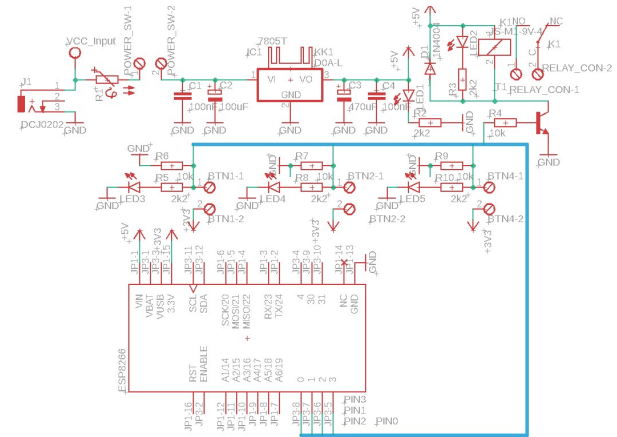


Fig. 4. Scheme of ESP8266 connection

All the parts from the scheme in Fig. 4 were assembled to the main board as depicted in Fig 5.



Fig. 5. Assembled prototype board

The assembled prototype board, pushbuttons and the display were placed to 3D-printed case to form the final prototype. The photo of the final prototype is presented in Fig. 6.



Fig. 6. Final prototype of the controller

B. Controller software

The ESP8266 controller can be programmed in standard Arduino Language using the Arduino programming environment.

The program contains functionality of the controller as described in Section IV. Moreover, setup of the program using pushbuttons and the display was implemented. The parameters that can be changed by setup can be divided into 2 groups:

- system parameters
- controller parameters

The system parameters affect behaviour of the whole program, not the control algorithms itself. For example, SSID and password of the Wi-Fi connection belongs to this group. The controller parameters are directly connected to control algorithm (e.g. setpoint).

The structure of the menu of the controller program is overviewed in Fig. 7.

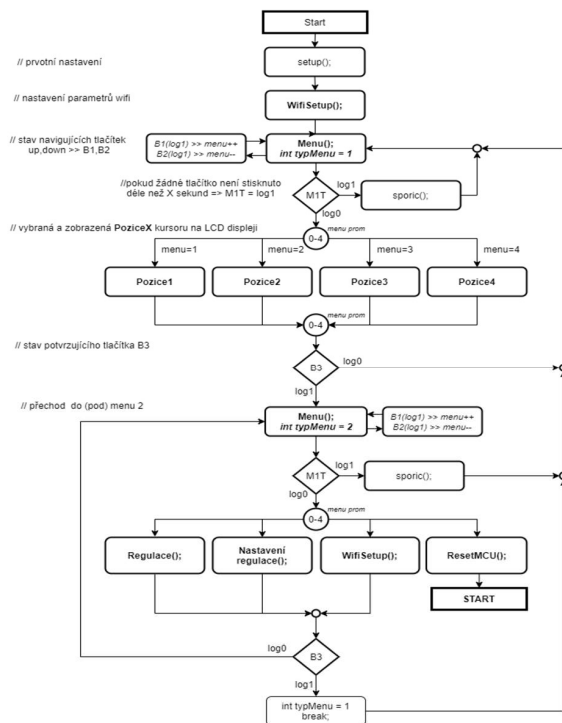


Fig. 7. Structure of the program

It can be seen that the program starts by connecting to the Wi-Fi and then waits for input via pushbuttons. The user (operator) can choose to setup the controller, to setup the Wi-Fi, to reset the MCU or to start the control process. Each of these parts contain specific subroutines where setups are based on user interactions, with pushbuttons for operator input and the display providing feedback to the operator.

VI. CONTROL RESULTS

Several courses are presented in this section to demonstrate the behaviour of the controller in both simulation and real conditions. The system was modelled as pure integrator with gain 1.5 cm/min, input is a binary signal (0 / 1) stating whether pumping is activated and the consumption is a disturbance input to the system. The course of consumption is the same for all simulations and represents two consecutive

days: a week day and a weekend day (Friday and Saturday). The second day contains excessive consumption which might represent watering a grass by some consumer.

A. Open loop control

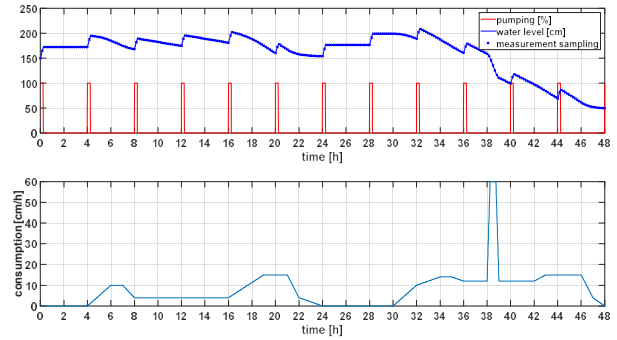


Fig. 8. Simulation of open-loop control

Fig. 8 presents simulation of the open-loop. The pump is activated for 15 minutes every 4 hours. It can be seen that this control scheme is suitable for the first day but it is insufficient for the second day.

B. Mixed Open-loop / Closed-loop Control

The setpoint was held constant at 200 cm for the following experiments. Fig. 9 presents a more realistic course of consumption with proposed controller of type I (i.e. constant pumping duration – see Section IV). If the control error was positive, 10 min pumping was applied. The performance is significantly better compared to open loop control, but the controller still has problems with excessive consumption.

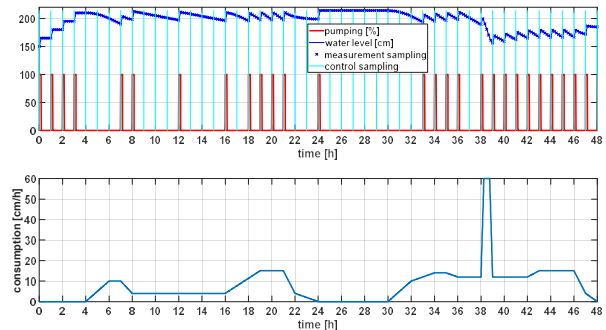


Fig. 9. Simulation of the proposed control, type I

Fig. 10 presents usage of proportional pumping duration – type II (see Section IV). In this case, the controller can cope even with excessive consumption.

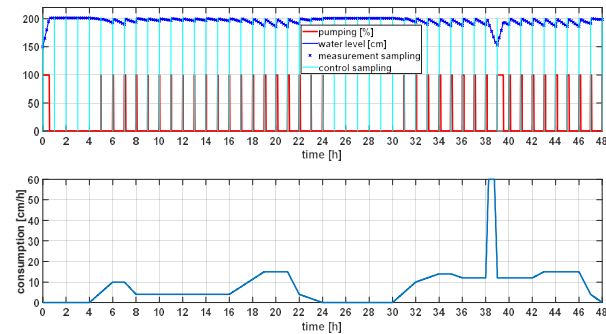


Fig. 10. Simulation of the proposed control, type II

Fig. 11 presents results obtained by type III controller. The PWM period was set to 10 minutes. Description of the controller can be found in Section IV. The course of water level is smoother compared to type II control.

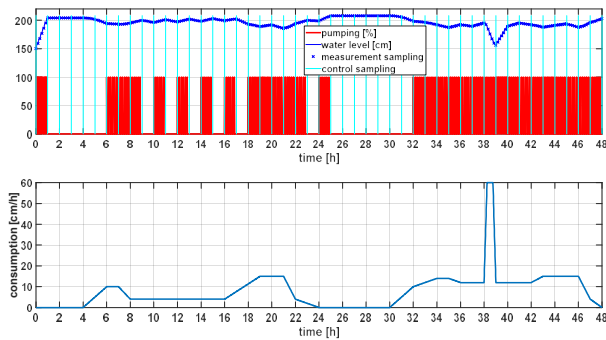


Fig. 11. Simulation of the proposed control, type III

The difference between type II control and type III control can be observed from Fig. 12. A narrower time range of 38 – 41 hours is presented. It can be seen that whole pumping is performed at the beginning of sampling period when using type II controller while type III controller spreads the control action across the whole sampling period.

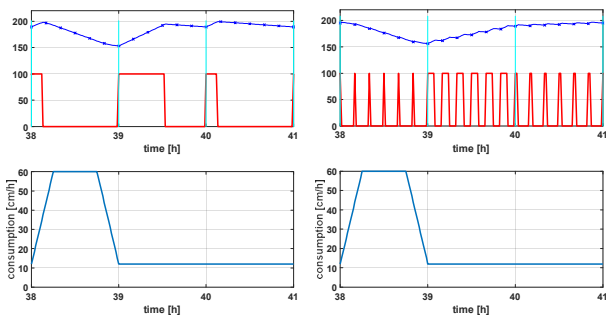


Fig. 12. Comparison of type II (left) and type III (right) control

C. Real-time control

Fig. 13 presents the course of water level of the real system during 2 days where controller of type II was applied and the setpoint was 300 cm.

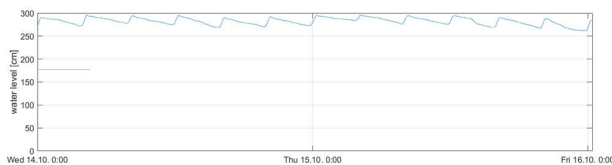


Fig. 13. Real course of water level

VII. CONCLUSION

A control algorithm for the slow-sampling water supply system was proposed. A hardware platform for the controller was selected and the controller was assembled and programmed. Subsequently it was tested in real conditions.

Further work will involve verifying the controller from both hardware and software perspectives.

ACKNOWLEDGMENT

The work was supported by grant from the Technology Agency of the Czech Republic, identification code TK04020222 “Decentralisation of energy sources in an existing district heating system”. This support is greatly appreciated.

REFERENCES

- [1] P. Chalupa, M. Vláčil, and J. Novák, "A Liquid Level Monitoring System for a Remote Reservoir," *2019 22nd International Conference on Process Control (PCI9)*, Strbske Pleso, Slovakia, 2019, pp. 80-85, doi: 10.1109/PC.2019.8815027.
- [2] M. Haren, L. Mirkin, L. Blanken, and T. Oomen, "Beyond Nyquist in Frequency Response Function Identification: Applied to Slow-Sampled Systems", *IEEE Control Systems Letters*, vol. 7, p. 2131-2136, 2023. <https://doi.org/10.1109/lcsys.2023.3284344>
- [3] R. Munje and B. Patre, "Two-stage multirate state feedback control designs for systems with slow and fast eigenvalue modes", *Transactions of the Institute of Measurement and Control*, vol. 46, no. 6, p. 1027-1037, 2023. <https://doi.org/10.1177/01423312231187445>
- [4] C. Yang, J. Xia, X. Huang, X. Song, H. Shen, and J. Wang, "Hm-Based Filtering for Slow-Sampling Singularly Perturbed Jumping Systems", *IET Control Theory and Applications*, vol. 14, no. 13, p. 1797-1805, 2020. <https://doi.org/10.1049/iet-cta.2019.0661>
- [5] J. Sun, J. Yang, S. Li, and W. Zheng, "Estimate-Based Dynamic Event-Triggered Output Feedback Control of Networked Nonlinear Uncertain Systems", *IEEE Transactions on Systems Man and Cybernetics Systems*, vol. 51, no. 5, p. 3148-3158, 2021. <https://doi.org/10.1109/tsmc.2019.2918547>
- [6] J. Song and Y. Niu, "Dynamic Event-Triggered Sliding Mode Control: Dealing With Slow Sampling Singularly Perturbed Systems", *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 67, no. 6, p. 1079-1083, 2020. <https://doi.org/10.1109/tcsii.2019.2926879>
- [7] C. Zhai, X. Sheng, and W. Xiong, "Multi-Block Fault Detection for Plant-Wide Dynamic Processes Based on Fault Sensitive Slow Features and Support Vector Data Description", *IEEE Access*, vol. 8, p. 120737-120745, 2020. <https://doi.org/10.1109/access.2020.3006282>
- [8] S. Zhang and X. Bao, "Slow-Varying Batch Process Monitoring Based on Canonical Variate Analysis", *The Canadian Journal of Chemical Engineering*, vol. 101, no. 1, p. 400-419, 2022. <https://doi.org/10.1002/cjce.24401>
- [9] X. Yan, X. Chen, Y. Chen, B. Dai, H. Lin, T. Li, K. Han, K. Ni, F. Wang, S. Wang, Q. Zheng, and X. Zeng, "Slow Control System For Pandax-III Experiment", 2020. <https://doi.org/10.48550/arxiv.2012.13275>
- [10] C. Kim, S. Kim, I. Lee, H. Cho, Y. Kim, J. Ahn, E. Jang, S. Choi, Y. Unno, and B. Cheon, "Development of Slow Control Package for the Electromagnetic Calorimeter Trigger System at the Belle II Experiment", 2019. <https://doi.org/10.22323/1.340.0760>
- [11] NodeMcu V3 ESP8266 Lua WiFi CH340G - HWKITCHEN. Váš parták ve světě tvoření (in Czech) | HWKitchen.cz [online]. Available from: <https://www.hwkitchen.cz/nodemcu-v3-esp8266-lua-wifi-ch340g>.
- [12] ESP8266 Node MCU board specifications and pins descriptions. Frequently asked questions related to WI-FI board esp8266 node MCU. Electronics Curiosities [online]. Available from: <https://www.electronicscuriosities.com/2021/08/esp8266-node-mcu-board-specifications.htm>