

การประยุกต์ใช้ตัวประมวลผลแบบคู่สำหรับระบบรวบรวมข้อมูล และระบบพลวัตแบบเวลาจริง

Application of Dual-Core Processors for Real-Time Data Acquisition and Dynamic System

ปรีชา สาคะรังค์^{1*}
Preecha Sakarung^{1*}

Received: 7 October 2024

Revised: 13 April 2025

Accepted: 21 January 2026

บทคัดย่อ

บทความนี้กล่าวถึงการสร้างอุปกรณ์ที่มีความสามารถในการรวบรวมข้อมูลและสร้างสัญญาณในระบบพลวัตแบบเวลาจริง ซึ่งอุปกรณ์ที่นำเสนอมีประโยชน์ต่อการเรียนการสอนในรายวิชาระบบควบคุมดังนี้ 1) สามารถสร้างสัญญาณแรงดันไฟฟ้าของระบบพลวัตหนึ่งอินพุตหนึ่งเอาต์พุตทั้งอันดับหนึ่งและอันดับสองได้ 2) สามารถใช้เป็นระบบรวบรวมข้อมูลหกลช่องสัญญาณอินพุตและสองช่องสัญญาณเอาต์พุต และ 3) สามารถปรับค่าตัวแปรต่างๆ ของระบบพลวัตได้ในรูปแบบบล็อกไดอะแกรมเมื่อทำงานร่วมกับโปรแกรม WinFACT/BORIS และในการประยุกต์ใช้ตัวประมวลผลแบบคู่ทำงานขนานกันในการรวบรวมข้อมูลและสร้างสัญญาณของระบบพลวัตใช้เวลาในการคำนวณต่อรอบ คือ 1) การรวบรวมข้อมูลระหว่างบอร์ดประมวลผล ESP32 และโปรแกรม WinFACT/BORIS ที่เวลา 0.05 วินาที และ 2) การสร้างสัญญาณของระบบพลวัตที่เวลา 0.01 วินาที จากการทดลองพบว่าอุปกรณ์นี้เหมาะสมที่จะนำไปใช้งานในห้องปฏิบัติการระบบควบคุม เนื่องจากสามารถรวบรวมข้อมูลและสร้างสัญญาณของระบบพลวัตแบบเวลาจริงได้อย่างมีประสิทธิภาพ โดยพิจารณาจากค่าความคลาดเคลื่อนเมื่อเทียบกับผลการทดลองด้วยโปรแกรม MATLAB ที่ไม่เกิน 4% และสามารถใช้งานได้สะดวกดังตัวอย่างการศึกษาเกี่ยวกับระบบพลวัตและการออกแบบตัวควบคุมด้วยการต่อบล็อกไดอะแกรมบนโปรแกรม WinFACT/BORIS

คำสำคัญ: การรวบรวมข้อมูล ระบบพลวัต เวลาจริง ระบบควบคุม

¹ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และสถาปัตยกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ สุพรรณบุรี ประเทศไทย 72130

¹Department of Electrical Engineering, Faculty of Engineering and Architecture, Rajamangala University of Technology Suvarnabhumi, Suphanburi, Thailand 72130

*ผู้รับผิดชอบประสานงาน (Corresponding author) e-mail: preecha.s@rmutsb.ac.

ABSTRACT

This paper introduces a device enhancing not only real-time data acquisition but also signals generation in real-time dynamic systems. The proposed device yields a great benefit for the instruction of control system due to the fact that it is able to 1) generate voltage signals of single-input single-output dynamic system of both first-order and second-order systems, 2) function as a data acquisition system of six input and two output channels, and 3) adjust the values of various variables of the dynamic system in the form of a block diagram while working with the WinFACT/BORIS. In having dual processors function parallelly in data acquisition and signal generation in the dynamic systems, the calculation time for each cycle were as follows 1) the data acquisition between the processing board ESP32 and the WinFACT/BORIS functions at 0.05 seconds, and 2) the signal generation in the dynamic system functions at 0.01 seconds. The experiments have shown that the proposed device can be implemented in a control laboratory as it is capable of functioning in both real-time data acquisition and signal generation in real-time dynamic systems, in which all of the gained results were in accordance with those gained from experiments with MATLAB with a maximum of 4% discrepancy. In addition, with an ease of using block diagrams and the WinFACT/BORIS, the proposed device is therefore appropriate in laboratory contexts as the examples of dynamic system study and controller design were also demonstrated.

Keywords: Data Acquisition; Dynamic Systems; Real-time; Control System

บทนำ

การรวบรวมข้อมูล (data acquisition) เป็นกระบวนการสำคัญในระบบควบคุมเพราะทำหน้าที่รับค่าที่ได้จากตัวตรวจรู้ ส่งค่าที่ได้เข้าคอมพิวเตอร์เพื่อประมวลผลตามกฎการควบคุมที่ต้องการและส่งสัญญาณควบคุมไปกระตุ้นกระบวนการในลำดับถัดไป ระบบรวบรวมข้อมูลขนาดเล็กที่ใช้สาย USB ต่อเข้ากับคอมพิวเตอร์ได้รับความนิยมเพราะมีความเหมาะสมในการใช้งานใน สามารถสรุปข้อดีและข้อเสียของระบบรวบรวมข้อมูลจากงานวิจัยที่นำมาเป็นตัวอย่างได้ดังนี้ การใช้ตัวประมวลผลเบอร์ PIC 18F4550 และโปรแกรม visual C++ สำหรับการรวบรวมข้อมูลมีข้อดีคือราคาไม่สูงแต่มีตัวอย่างโปรแกรมการรวบรวมข้อมูลจำกัด [1] การใช้ตัวประมวลผลแบบ FPGA และโปรแกรม MATLAB มีข้อดีคือทำงานได้เร็วและความละเอียดสูงแต่ราคาสูงและมีขั้นตอนในการสร้างซับซ้อน [2] การใช้ตัวประมวลผลแบบ AVR และโปรแกรม MATLAB หรือ LabView มีข้อดีคือราคาไม่สูงแต่ทำงานช้าและไม่สามารถสร้างสัญญาณเอาต์พุตได้ [3] การใช้ตัวประมวลผลเบอร์ AT90 USB1286/87 และโปรแกรม ATOOL-90 มีข้อดีคือราคาไม่สูงแต่สามารถใช้งานได้เฉพาะการรวบรวมข้อมูลเพียงอย่างเดียว [4] การใช้ตัวประมวลผล AVR ATMEGA644P และโปรแกรม LabView หรือ Python มีข้อดีคือมีความเร็วและความละเอียดสูงแต่ราคาสูงและมีขั้นตอนในการสร้างซับซ้อน [5] การใช้ตัวประมวลผลเบอร์ PIC 18F4550 และโปรแกรม LabView หรือ C# มีข้อดีคือราคาไม่สูงแต่ไม่มีช่องสัญญาณเอาต์พุตแอนะล็อก [6] การใช้ตัวประมวลผลเบอร์ MEGA 2560 และโปรแกรม MATLAB มีข้อดีคือราคาไม่สูงแต่โปรแกรมมี

ค่าลิขสิทธิ์สูง [7] การใช้ตัวประมวลผลเบอร์ MEGA 2560 และโปรแกรม Python มีข้อดีคือราคาไม่สูงแต่สามารถใช้รวบรวมข้อมูลได้เพียงอย่างเดียว [8] นอกจากนี้ถ้าแบ่งประเภทระบบรวบรวมข้อมูล ตามผู้ผลิตจะได้สองลักษณะ คือ ผลิตภัณฑ์ทางการค้า [9, 10] และระบบที่ประกอบขึ้นเอง [1-8] ซึ่งต่างมีข้อดีและข้อเสีย โดยผลิตภัณฑ์ทางการค้า มีประสิทธิภาพและราคาสูง ในขณะที่ระบบรวบรวมข้อมูลในระบบที่ประกอบขึ้นเอง มีต้นทุนสำหรับส่วนเครื่อง (hardware) ต่ำกว่าแต่ถ้าต้องการความสะดวกในการใช้งานราคาจะสูงขึ้นจากส่วนชุดคำสั่ง (software) ที่ใช้งานร่วม เช่น LabView หรือ MATLAB

สำหรับการทำงานแบบเวลาจริงนั้น ระบบรวบรวมข้อมูลที่ใช้ตัวประมวลผลเดี่ยวสามารถนำชุดโปรแกรมเสริม FreeRTOS มาใช้ร่วมได้ เพราะตัวประมวลผลเดี่ยวจะทำงานได้เพียงอย่างเดียวในช่วงขณะเวลาหนึ่งโดยพื้นฐาน ทั้งนี้โครงสร้างการสั่งงานของชุดโปรแกรมเสริม FreeRTOS จะสร้างตารางการทำงานให้กับงานที่มีความสำคัญแตกต่างกันและจะทำงานที่สำคัญที่สุดก่อน โดยใช้การขัดจังหวะเพื่อให้สามารถทำงานได้ทันตามเวลาที่กำหนด นอกจากนี้ชุดโปรแกรม FreeRTOS ยังรองรับการทำงานแบบหลายตัวประมวลผล ซึ่งส่งผลให้การทำงานโดยรวมมีประสิทธิภาพสูงขึ้น หลักการนี้เรียกว่าการประมวลผลแบบขนาน (Parallel Processing) ซึ่งมีข้อดีในด้านการประหยัดทรัพยากรหน่วยความจำ การบริหารจัดการพลังงาน และการควบคุมรีจิสเตอร์ด้วยบัสสื่อสารภายในที่ทำให้เสถียรภาพในการทำงานของตัวประมวลผลแบบคู่เพิ่มมากขึ้น อย่างไรก็ตามระบบชนิดนี้ต้องการชุดโปรแกรมการสั่งงานโดยเฉพาะเนื่องจากความซับซ้อนของโครงสร้างการทำงานและสภาพแวดล้อมที่จำกัด

สำหรับการประยุกต์ใช้ตัวประมวลผลแบบคู่ในระบบควบคุมนั้น Krause, et al. [11] ได้แสดงให้เห็นว่าเมื่อใช้ตัวประมวลผลตัวที่หนึ่งควบคุมกระบวนการและใช้ตัวประมวลผลตัวที่สองจำลองการทำงานของกระบวนการแบบที่มีส่วนเครื่องอยู่ในลูป (hardware in the loop, hil) ในการตรวจวัด แสดงผล และตรวจสอบจะพบว่าข้อมูลที่ได้จากทั้งสองระบบตรงกันที่ความถี่ต่ำหรือช่วงผลตอบสนองคงตัว แต่ข้อมูลที่ได้กลับมีความแตกต่างกันที่ความถี่ระดับกลางถึงระดับสูง

นอกจากนี้ในการศึกษาเปรียบเทียบการทำงานของโครงข่ายประสาทเทียม (neural network) บนตัวประมวลผลแบบหนึ่งตัวและตัวประมวลผลแบบคู่ในบอร์ด ESP32 Dokic, et al. [12] แสดงให้เห็นว่ามีความเร็วของการแพร่กระจายของโครงข่ายประสาทเทียมแบบป้อนไปด้านหน้า (feed forward neural network, FNN) แตกต่างกัน โดยตัวประมวลผลแบบคู่จะทำงานได้รวดเร็วกว่าการใช้ตัวประมวลผลแบบเดี่ยว ซึ่งตัวประมวลผลแบบคู่จะมีความเร็วเพิ่มมากขึ้นตามการเพิ่มขึ้นของจำนวนโครงข่ายประสาทเทียมที่อยู่ในชั้นซ่อน (hidden layer)

สำหรับข้อดีของโปรแกรม WinFACT/BORIS คือการทำงานแบบบล็อกไดอะแกรมที่มีข้อได้เปรียบด้านความสะดวกในการใช้งานเนื่องจากโปรแกรม WinFACT/BORIS นี้มีขนาดเล็กเพียง 62 kbyte จึงเหมาะสำหรับการติดตั้งในเครื่องคอมพิวเตอร์ที่มีทรัพยากรจำกัด และสามารถใช้งานร่วมกับอุปกรณ์เชื่อมต่อภายนอก เช่น บอร์ด ESP32 ได้อย่างสะดวกโดยการเรียกไฟล์ esp32.dll ซึ่งเป็นไฟล์ที่ใช้ติดต่อกับบอร์ดเท่านั้น จึงไม่จำเป็นต้องติดตั้งโปรแกรมอื่นเพิ่มเติม นอกจากนี้โปรแกรม WinFACT/BORIS บางรุ่นไม่มีค่าใช้จ่ายซึ่งถือเป็นข้อได้เปรียบด้านราคาที่สูงเสริมการนำมาใช้ในการเรียนการสอนทางด้านวิศวกรรมระบบควบคุมในระดับปริญญาตรี

ในวิชาระบบควบคุมระดับปริญญาตรี การสร้างสัญญาณของระบบพลวัตมีวัตถุประสงค์เพื่อจำลองกระบวนการโดยมีระบบอันดับหนึ่งและระบบอันดับสองเป็นกระบวนการที่สำคัญ Li [13] ได้แสดงการจำลองกระบวนการด้วยระบบแอนะล็อกซึ่งการเปลี่ยนค่าตัวแปรของกระบวนการทำได้ไม่เที่ยงตรงเพราะอุปกรณ์

ไฟฟ้าที่ใช้ เช่น ตัวต้านทานมีความผิดพลาด และถึงแม้การจำลองกระบวนการด้วยระบบดิจิทัลจะมีความเที่ยงตรงมากขึ้น แต่การเปลี่ยนค่าตัวแปรของกระบวนการทำได้ไม่สะดวกเพราะในการเปลี่ยนแต่ละครั้งต้องใช้เครื่องอัตโนมัติโปรแกรมลงบนบอร์ดไมโครคอนโทรลเลอร์ [14]

ดังนั้นบทความนี้จึงนำเสนออุปกรณ์ที่มีความสามารถในการรวบรวมข้อมูลและสร้างสัญญาณของระบบพลวัตแบบเวลาจริงโดยใช้ตัวประมวลผลแบบคู่ทำงานขนานกัน โดยแสดงตัวอย่างการต่อบล็อกไดอะแกรมบนโปรแกรม WinFACT/BORIS ในการศึกษาเกี่ยวกับระบบพลวัตและการออกแบบตัวควบคุมซึ่งสามารถทำได้อย่างสะดวกและใช้ต้นทุนต่ำ

วิธีดำเนินการวิจัย

การประยุกต์ใช้บอร์ดประมวลผล ESP32 ที่มีหน่วยประมวลผลแบบคู่และหลักการเขียนโปรแกรมแบบเวลาจริงโดยใช้ชุดโปรแกรมเสริม FreeRTOS ซึ่งรอบการคำนวณของแต่ละงานใช้กำหนดเวลาแตกต่างกันดังนี้ 1) การรวบรวมข้อมูลระหว่างบอร์ดประมวลผล ESP32 และโปรแกรม WinFACT/BORIS กำหนดเวลาทำงานที่ 0.05 วินาที และ 2) การสร้างสัญญาณของระบบพลวัตกำหนดเวลาทำงานที่ 0.01 วินาที ซึ่งค่าเวลาที่ใช้ในการคำนวณนี้ได้จากการทดลอง โดยมีวัตถุประสงค์เพื่อหาค่าเวลาที่ใช้ในการคำนวณที่ต่ำที่สุดที่ยังคงทำให้ระบบโดยรวมสามารถทำงานแบบเวลาจริงและทำงานได้ต่อเนื่องจนจบการทำงาน

1. บอร์ดประมวลผล ESP32 และการใช้โมดูล MCP4725

บอร์ดประมวลผล ESP32 รุ่น Xtensa ขนาด 32 บิต มีสถาปัตยกรรมแบบ LX6 มีหน่วยประมวลผลแบบคู่ คือ protocol core (core 0) และ application core (core 1) โดยหน่วยประมวลผล core 0 ใช้สำหรับการทำงานสื่อสารแบบไร้สาย และหน่วยประมวลผล core 1 ใช้สำหรับการทำงานกับโปรแกรมประยุกต์แบบต่างๆ จุดเด่นของหน่วยประมวลผลแบบคู่คือสามารถทำงานทั้งสองอย่างขนานกันได้ จากการทดลองเปรียบเทียบการให้งานทั้งสองอย่างกับหน่วยประมวลผลแบบหนึ่งตัวและหน่วยประมวลผลแบบคู่ พบว่าหน่วยประมวลผลแบบคู่ทำงานได้รวดเร็วและมีเสถียรภาพมากกว่า ดังนั้นบทความนี้ได้กำหนดให้งานของ core 0 คือการสร้างสัญญาณของระบบพลวัต และงานของ core 1 คือการรวบรวมข้อมูลระหว่างบอร์ดประมวลผล ESP32 และโปรแกรม WinFACT/BORIS ดังรูปที่ 1a

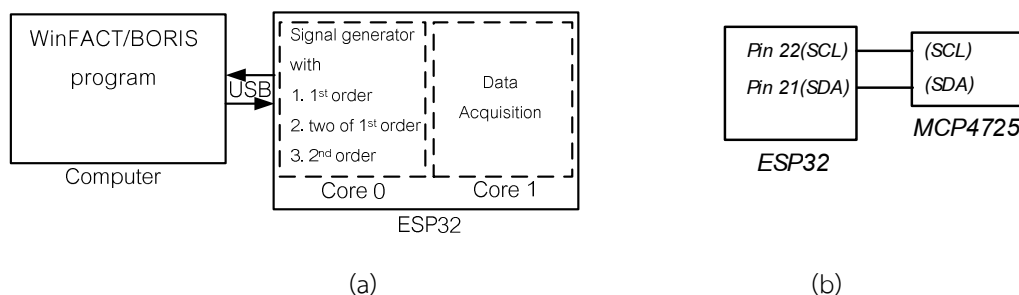


Figure 1 a) Two tasks for ESP32 dual core b) Electrical wiring between ESP32 and MCP4725

ระบบที่นำเสนอประกอบด้วยบอร์ดประมวลผล ESP32 และโมดูล MCP4725 โดยมีการต่อสายเข้าด้วยกันดังรูปที่ 1b บอร์ดประมวลผล ESP32 มีระบบการแปลงข้อมูลดิจิทัลไปเป็นข้อมูลแอนะล็อก (Digital

to Analog Conversion, DAC) สองช่องสัญญาณที่มีความละเอียด 10 บิต และมีระบบการแปลงข้อมูลแอนะล็อกเป็นข้อมูลดิจิทัล (Analog to Digital Conversion, ADC) สิบห้าช่องสัญญาณที่มีความละเอียด 12 บิต เพื่อให้ระบบรวบรวมข้อมูลของอุปกรณ์ที่นำเสนอมีความละเอียด 12 บิต ทั้ง ADC และ DAC งานวิจัยนี้จึงใช้โมดูล MCP4725 ซึ่งทำหน้าที่ DAC ที่มีความละเอียด 12 บิตต่อกับบอร์ดประมวลผล ESP32 ผ่านช่องสัญญาณ I2C

2.การสร้างสัญญาณโดยใช้เทคนิคการแปลง Z และสมการผลต่าง [15]

การสร้างระบบเวลาวิฤต คือการแปลงจากระบบเวลาต่อเนื่องโดยใช้เทคนิคการสุ่มข้อมูลด้วยการคงค่าอันดับศูนย์ (sampler and zero-order hold) ที่แสดงดังรูปที่ 2a เมื่อ $e(t)$ คือสัญญาณอินพุตในโดเมนเวลา, n คือเลขจำนวนเต็ม, $\bar{e}(t)$ คือสัญญาณเอาต์พุตในโดเมนเวลา, $E(s)$ คือสัญญาณอินพุตในโดเมนลาปลาซ, T คือคาบเวลาสุ่มข้อมูลและ $\bar{E}(s)$ คือสัญญาณเอาต์พุตในโดเมนลาปลาซที่ผ่านการสุ่มข้อมูลแล้วหรือเขียนได้ด้วยสมการที่ (1)

$$\bar{E}(s) = \frac{1 - e^{-Ts}}{s} \left[\sum_{n=0}^{\infty} e(nT) e^{-nTs} \right] \quad (1)$$

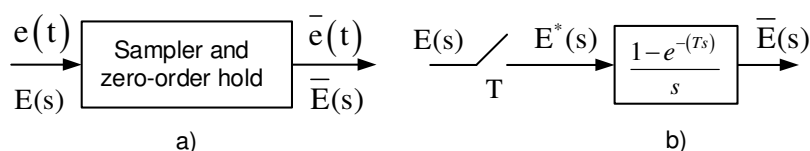


Figure 2 a) Sampler and zero-order hold b) Mathematical model of sampler and zero-order hold

แบบจำลองทางคณิตศาสตร์ของการสุ่มข้อมูลและการคงค่าอันดับศูนย์แสดงได้ดังรูปที่ 2b เมื่อ $E^*(s)$ คือสัญญาณอินพุตที่ผ่านการสุ่มข้อมูลซึ่งแทนได้ด้วยสมการที่ (2) และการสุ่มข้อมูลแทนด้วยสวิตช์ T ร่วมกับฟังก์ชันถ่ายโอนที่เรียกว่าการคงค่าข้อมูล (data-hold)

$$E^*(s) = \sum_{n=0}^{\infty} e(nT) e^{-nTs} \quad (2)$$

พิจารณาตัวอย่างการสร้างระบบเวลาวิฤตของระบบอันดับหนึ่งดังรูปที่ 3

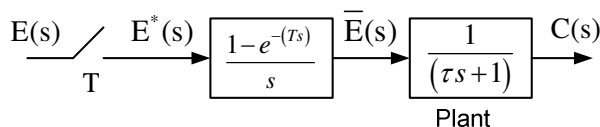


Figure 3 Discrete time of first-order system

เมื่อค่าคงตัวเวลา (τ) มีค่าเท่ากับหนึ่งจะสามารถเขียนได้ดังสมการที่ (3) และเมื่อใช้การแปลง Z กับสมการที่ (3) จะได้ผลลัพธ์ดังสมการที่ (4)

$$C(s) = G(s)E^*(s) = \frac{1-e^{-Ts}}{s(s+1)}E^*(s) \quad (3)$$

$$C(z) = G(z)E(z) = \frac{(1-e^{-T})}{(z-e^{-T})}E(z) \quad (4)$$

ดังนั้นถ้าพิจารณาเฉพาะฟังก์ชันถ่ายโอนในโดเมน Z ($G(z)$, pulse transfer function) ของระบบอันดับหนึ่ง จะได้ดังสมการที่ (5)

$$G(z) = \frac{(1-e^{-T})}{(z-e^{-T})} \quad (5)$$

เมื่อกำหนดให้คาบเวลาสุ่มข้อมูล (T) เท่ากับ 0.01 วินาที และเขียนสมการที่ (5) ใหม่ในรูปแบบสมการผลต่าง จะได้ดังสมการที่ (6) เมื่อ k คือจำนวนเต็มที่แสดงลำดับข้อมูล, $C(k)$ คือสัญญาณเอาต์พุตลำดับที่ k, $C(k-1)$ คือสัญญาณเอาต์พุตลำดับที่ (k-1), $E(k-1)$ คือสัญญาณอินพุตลำดับที่ (k-1) และจากสมการที่ (6) สามารถแสดงให้อยู่ในรูปแบบทั่วไปของไดอะแกรมการจำลองสมการผลต่างซึ่งแสดงได้ดังรูปที่ 4 โดยให้บล็อกไดอะแกรม T คือเวลาหนึ่ง

$$C(k) = 0.00995 * E(k-1) + 0.99 * C(k-1) \quad (6)$$

3. โปรแกรม WinFACT/BORIS และการติดตั้งบอร์ดประมวลผล ESP-32

โปรแกรม WinFACT/BORIS รุ่น 2016 เป็นโปรแกรมพื้นฐานที่ใช้ในการเรียนการสอนรายวิชาการควบคุมที่มีประสิทธิภาพดี ใช้งานง่ายและสามารถเลือกใช้รุ่นที่ไม่เสียค่าลิขสิทธิ์ได้ [16] การใช้งานโปรแกรมทำได้ด้วยการต่อสายสัญญาณเชื่อมระหว่างช่องสัญญาณเอาต์พุตของบล็อกไดอะแกรมตัวหนึ่งเข้ากับช่องสัญญาณอินพุตของบล็อกไดอะแกรมอีกตัวหนึ่ง โดยการใส่ค่าตัวแปรสามารถทำบนบล็อกไดอะแกรมได้โดยตรง

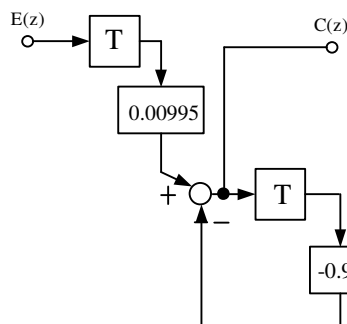


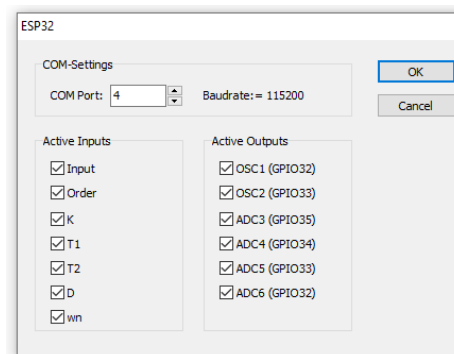
Figure 4 Difference equation of first-order system in block diagram

การใช้งานบอร์ดประมวลผล ESP32 และโปรแกรม WinFACT/BORIS [16] ได้พัฒนาเพิ่มเติมโดยมีขั้นตอนการสร้างดังนี้ 1) เขียนโปรแกรมฝังไว้ในบอร์ดประมวลผล ESP32 เพื่อรับส่งข้อมูลระหว่างอุปกรณ์สองตัวโดยใช้ภาษา C 2) เขียนโปรแกรมเพื่อสร้างบล็อกไดอะแกรม ESP32 บนโปรแกรม WinFACT/BORIS โดยใช้ภาษา Delphi ซึ่งบล็อกไดอะแกรม ESP32 ที่มีช่องสัญญาณอินพุตและเอาต์พุตที่สร้างบนโปรแกรม WinFACT/BORIS สามารถแสดงได้ดังรูปที่ 5a โดยมีหน้าต่างการกำหนดค่าตัวแปรต่างๆ รวมทั้งหมายเลขของช่อง USB ที่เชื่อมต่อกับบอร์ด ESP32 ดังที่แสดงในรูปที่ 5b

นอกจากนี้ความเร็วที่ใช้ในการรับส่งข้อมูลระหว่างบอร์ดประมวลผล ESP32 และโปรแกรม WinFACT/BORIS เท่ากับ 115200 (baudrate) เมื่อช่องต่อทางด้านซ้ายของบล็อกไดอะแกรม ESP32 คือช่องสัญญาณสำหรับกำหนดค่าตัวแปรต่างๆ ของระบบพลวัตซึ่งรายละเอียดของตัวแปรทุกตัวได้แสดงดังตารางที่ 1 และสมการ (7-9) และช่องต่อทางด้านขวาของบล็อกไดอะแกรม ESP32 คือช่องสัญญาณที่ใช้วัดค่าแรงดันเปรียบเทียบกับเวลาซึ่งสามารถวัดพร้อมกันได้หกช่องสัญญาณ แต่เมื่อใช้งานพร้อมกันจะทำให้ความเร็วในการทำงานโดยรวมลดลง



a)



b)

Figure 5 a) Block diagram of ESP-32 on WinFACT/BORIS b) Configuration

Table 1 Parameters in ESP32 block diagram

Parameters	Description
Input	Reference signal
Order	Select 0: first order, 1: cascade of two first-order, 2: second order
K	Plant gain
T1	Time constant No.1
T2	Time constant No.2
D	Damping ratio
wn	Natural frequency

ในการกำหนดระบบพลวัตจะใช้ตัวแปรจากตารางที่ 1 โดยสามารถแบ่งได้ดังนี้ 1) ระบบอันดับหนึ่ง $G_1(s)$, 2) ระบบอันดับหนึ่งสองตัวต่ออนุกรม $G_2(s)$, และ 3) ระบบอันดับสอง $G_3(s)$ ซึ่งอธิบายได้ดังสมการ (7) - (9) ตามลำดับ

$$G_1(s) = \frac{K}{T_1s + 1} \quad (7)$$

$$G_2(s) = \frac{K}{(T_1s + 1)(T_2s + 1)} \quad (8)$$

$$G_3(s) = \frac{K(w_n)^2}{s^2 + 2Dw_ns + (w_n)^2} \quad (9)$$

เมื่อ K คืออัตราขยายของกระบวนการ, T_1 คือค่าคงตัวเวลาของระบบอันดับหนึ่งที่หนึ่ง, T_2 คือค่าคงตัวเวลาของระบบอันดับหนึ่งที่สอง, D คือค่าการหน่วง และ w_n คือความถี่ธรรมชาติ

4.การทำงานแบบเวลาจริงโดยใช้โปรแกรมเสริม FreeRTOS

เนื่องจากหน่วยความจำและทรัพยากรต่างๆ บนบอร์ดไมโครคอนโทรลเลอร์มีจำกัด ดังนั้นการเขียนคำสั่งเพื่อสร้างระบบการทำงานแบบเวลาจริงจึงไม่ใช่เรื่องง่าย การใช้โปรแกรมเสริม เช่น FreeRTOS จึงเป็นแนวทางที่ผู้พัฒนาระบบการทำงานแบบเวลาจริงนิยมใช้งาน เนื่องจากเป็นชุดโปรแกรมที่ช่วยให้บอร์ดไมโครคอนโทรลเลอร์สามารถทำงานแบบเวลาจริงได้อย่างเต็มประสิทธิภาพ

ในบทความนี้ได้แสดงการแบ่งงานที่แตกต่างกันให้หน่วยประมวลผล core 0 และ core 1 โดยใช้คำสั่ง `xTaskCreatePinnedToCore()` เมื่อกำหนดให้การรวบรวมข้อมูลระหว่างบอร์ดประมวลผลกับโปรแกรม WinFACT/BORIS มีความสำคัญสูงสุด และการสร้างสัญญาณของระบบพลวัตทำได้โดยใช้การขัดจังหวะ ซึ่งจากการทดลองพบว่าคาบของการขัดจังหวะในระบบอันดับหนึ่งและระบบอันดับหนึ่งสองตัวต่ออนุกรมกันอยู่ที่ 0.001 วินาทีแต่คาบของการขัดจังหวะในระบบอันดับสองอยู่ที่ 0.01 วินาทีเนื่องจากมีการคำนวณที่ซับซ้อนจึงต้องใช้เวลามากกว่า ดังนั้นในการวิจัยนี้จึงเลือกคาบเวลาในการขัดจังหวะอยู่ที่ 0.01 วินาทีเพื่อให้ระบบโดยรวมสามารถทำงานได้

ผลการวิจัย

1.การศึกษาระบบพลวัตแบบเวลาจริง

งานวิจัยนี้ได้ศึกษาระบบพลวัตแบบเวลาจริงผ่านการทดลองสามหัวข้อดังนี้ 1) การสร้างสัญญาณของระบบอันดับหนึ่ง 2) การสร้างสัญญาณของระบบอันดับหนึ่งต่ออนุกรมกันสองตัว และ 3) การสร้างสัญญาณของระบบอันดับสอง โดยแสดงภาพอุปกรณ์ที่นำเสนอขณะพร้อมใช้งานในรูปแบบที่ 6a และการใช้งานอุปกรณ์ดังกล่าวด้วยโปรแกรม WinFACT/BORIS ในรูปแบบที่ 6b

1.1 การสร้างสัญญาณของระบบอันดับหนึ่ง

จากรูปที่ 6b เมื่อกำหนดให้อุปกรณ์ที่นำเสนอสร้างสัญญาณของระบบอันดับหนึ่งที่มีค่าตัวแปร ดังนี้ $Order = 0$, $K = 1$ และ $T_1 = 1$ จะได้ผลการทดลองที่สามารถนำไปใช้งานได้สองรูปแบบคือ 1) การ

แสดงผลของกราฟแรงดันต่อเวลาบนคอมพิวเตอร์และ 2) การวัดแรงดันที่ชั่วต่อสัญญาณโดยใช้ ออสซิลโลสโคป ซึ่งแสดงได้ดังรูปที่ 7a และรูปที่ 7b ตามลำดับ

รายละเอียดของกราฟในรูปที่ 7a ที่แสดงโดยเส้นกราฟประจักษ์ (reference) คือสัญญาณอินพุตแบบ ขนาด 1 โวลต์ เส้นกราฟทึบ (MATLAB) คือสัญญาณที่สร้างขึ้นจากโปรแกรม MATLAB เพื่อจำลองการทำงาน (simulation) และเส้นกราฟประจักษ์ (device) คือสัญญาณแรงดันไฟฟ้าที่สร้างจากอุปกรณ์ที่ นำเสนอ และในรูปที่ 7b ที่แสดงโดยเส้นสัญญาณ ได้แก่ สัญญาณสี่เหลี่ยมคือสัญญาณอินพุตแบบขนาด 1 โวลต์ และสัญญาณสีฟ้า (กราฟเส้นโค้ง) คือสัญญาณที่สร้างจากอุปกรณ์ที่นำเสนอ

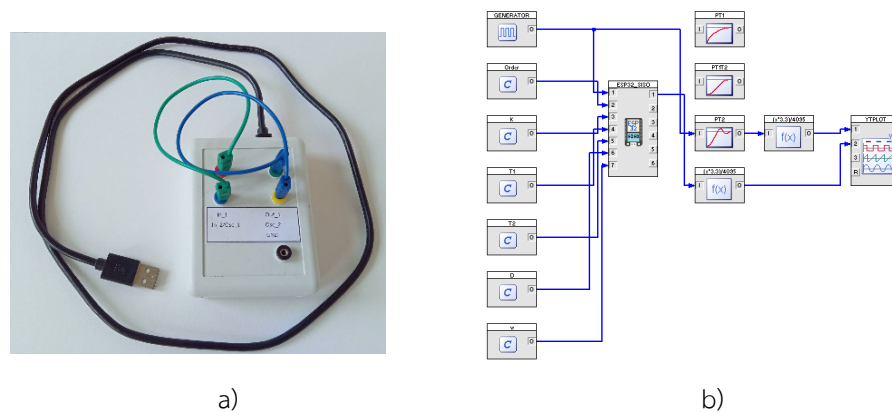


Figure 6 a) Hardware b) Software configuration of real-time data acquisition and real-time dynamic system

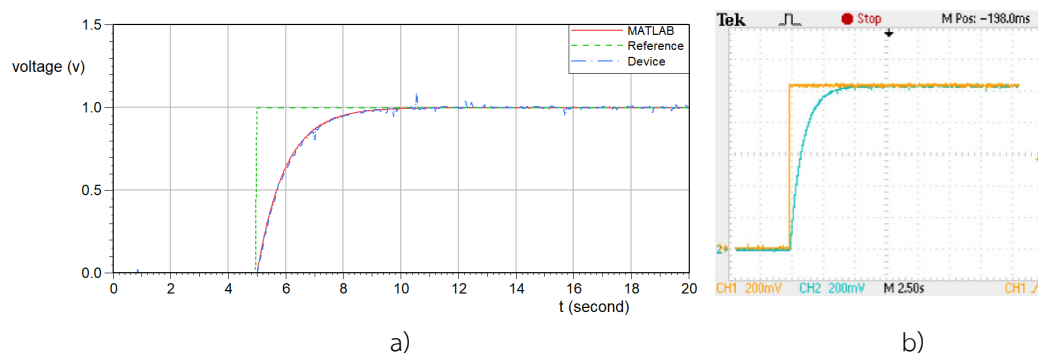


Figure 7 Step responses of first-order system derived from a) WinFACT/BORIS plot and b) Oscilloscope

1.2 การสร้างสัญญาณของระบบอันดับหนึ่งสองตัวต่อแบบอนุกรม

จากรูปที่ 6b เมื่อกำหนดให้อุปกรณ์ที่นำเสนอสร้างสัญญาณของระบบอันดับหนึ่งสองตัวต่ออนุกรมกันโดยเลือกค่าตัวแปร $\text{Order} = 1$ ซึ่งระบบอันดับหนึ่งทั้งสองตัวจะมีค่าตัวแปรของอัตราขยาย (K) และค่าคงตัวเวลาเหมือนกันคือ $K = 1$, $T_1 = 1$ และ $T_2 = 1$ ซึ่งทำให้ได้ผลการทดลองดังรูปที่ 8

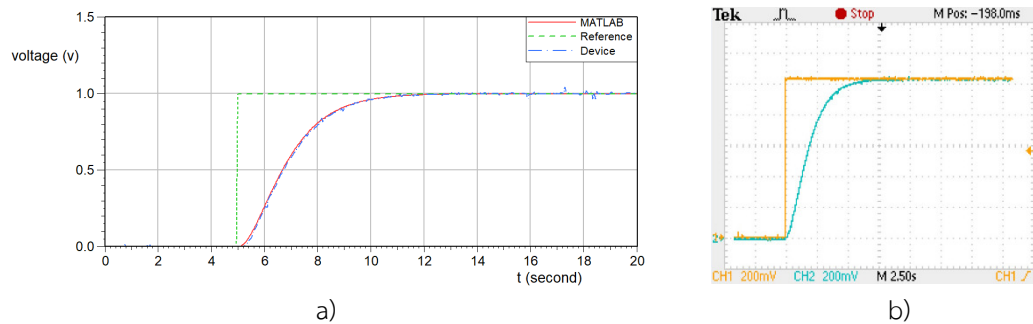


Figure 8 Step responses of cascade system with two first-orders derived from a) WinFACT /BORISplot and b) Oscilloscope

1.3 การสร้างสัญญาณของระบบอันดับสอง

จากรูปที่ 6b เมื่อกำหนดให้อุปกรณ์ที่นำเสนอสร้างสัญญาณของระบบอันดับสอง โดยเลือกค่าตัวแปร $\text{Order} = 2$ ซึ่งระบบอันดับสองจะมีค่าตัวแปร คือ $K = 1$, $D = 0.5$ และ $\omega_n = 1$ ซึ่งทำให้ได้ผลการทดลองดังรูปที่ 9

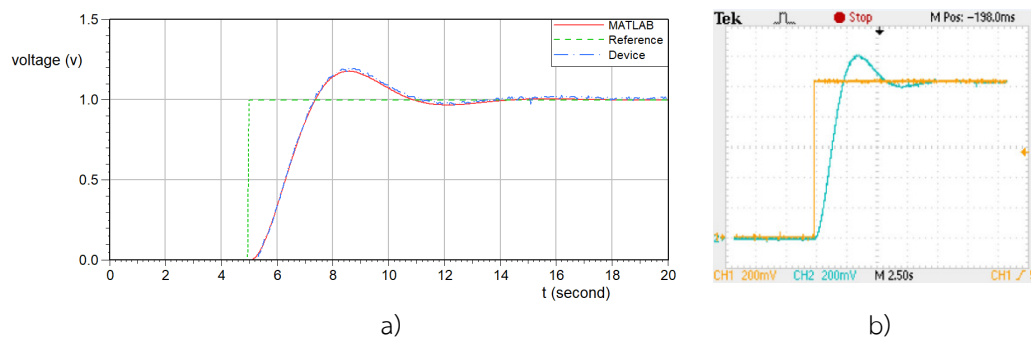


Figure 9 Step response of second-order system derived from a) WinFACT/BORISplot and b) Oscilloscope

Table 2 Time response specifications for figures 7 - 9

Specifications	Figure 7		Figure 8		Figure 9	
	Device	Oscilloscope	Device	Oscilloscope	Device	Oscilloscope
Rise Time	2.24	2.25	3.29	3.31	1.56	1.57
Peak Time	-	-	-	-	3.43	3.47
Maximum Overshoot (%)	-	-	-	-	18.9	18.2
Settling Time	3.83	3.86	5.53	5.55	7.62	7.81

จากการเปรียบเทียบการตอบสนองทางเวลาของระบบทั้งสามแบบตามรูปที่ 7 - 9 ในตารางที่ 2 แสดงให้เห็นว่าระบบพลวัตแบบเวลาจริงที่น่าเสนอนี้มีผลตอบสนองทางเวลาใกล้เคียงกับผลที่แสดงโดยกราฟที่ได้จากออสซิลโลสโคป ซึ่งเป็นหลักฐานที่ยืนยันได้ว่าเป็นการทำงานแบบเวลาจริงที่มีค่าความคลาดเคลื่อนสูงสุดไม่เกิน 4% โดยพิจารณาจากสมการ (7 - 9) ว่าสามารถคำนวณสัญญาณเอาต์พุตของระบบอันดับหนึ่ง ระบบอันดับหนึ่งสองตัวต่อแบบอนุกรมกัน และระบบอันดับสอง เมื่อสัญญาณอินพุตเป็นฟังก์ชันแบบขั้นหนึ่ง หน่วยได้ตั้งสมการ (10 - 13) [15] ตามลำดับ ซึ่งผลจากการจำลองบนโปรแกรม MATLAB ตามที่แสดงในรูปที่ 7 - 9 ให้ผลที่มีความสอดคล้องกันกับสมการ (10 - 13)

$$c_1(t) = k - ke^{-\frac{t}{T_1}} \quad (10)$$

$$c_2(t) = k_1 + k_2 e^{-\frac{t}{T_1}} + k_3 e^{-\frac{t}{T_2}} ; T_1 \neq T_2 \quad (11)$$

$$c_2(t) = k_1 + k_2 t e^{-\frac{t}{T_1}} + k_3 e^{-\frac{t}{T_2}} ; T_1 = T_2 \quad (12)$$

$$c_3(t) = k - \frac{k}{\sqrt{1-D^2}} e^{-D\omega_n t} \cos\left(\omega_n t \sqrt{1-D^2} - \tan^{-1}\left(\frac{D}{\sqrt{1-D^2}}\right)\right) \quad (13)$$

เมื่อ k_1, k_2, k_3 ได้จากการแยกเศษส่วนย่อย (partial fraction expansion) ก่อนทำลาปลาซผกผัน จากสมการ (12) การคำนวณหาค่า $c_3(t)$ ที่เวลา 3.43 วินาทีได้ค่าแรงดัน 1.159 V และค่าความคลาดเคลื่อนเท่ากับ $((1.159 - 1.189)/1.159) * 100 = 2.58\%$

2.การออกแบบตัวควบคุม

การควบคุมวงปิดด้วยตัวควบคุมพีไอดีเมื่อสัญญาณอ้างอิง (reference) คือเป้าหมายของการควบคุม ตัวควบคุมพีไอดี (PID controller) จะสร้างสัญญาณควบคุมเพื่อส่งไปกระตุ้นกระบวนการ (plant) ตามการเปลี่ยนแปลงของสัญญาณความผิดพลาดระหว่างสัญญาณอ้างอิงและสัญญาณเอาต์พุต ดังแสดงในรูปที่ 10



Figure 10 Closed loop control system with PID controller

ตัวควบคุมพีไอดีมีสมการของการสร้างสัญญาณเอาต์พุตดังสมการ(14) เมื่อ $e(t)$ คือสัญญาณอินพุต $c(t)$ คือสัญญาณเอาต์พุต K_P คือค่าอัตราขยายของตัวควบคุมพี T_N คือค่าคงที่เวลาของตัวควบคุมไอและ T_V คือค่าคงที่เวลาของตัวควบคุมดี สำหรับการเลือกใช้ตัวควบคุมพีไอดีในการควบคุมกระบวนการที่มีเสถียรภาพสามารถทำตามตัวอย่างดังนี้ เมื่อต้องการให้ผลตอบสนองแบบคงตัวปรับปรุงดีขึ้นสามารถใช้ตัวควบคุมพีไอทำการควบคุม และเมื่อต้องการให้ผลตอบสนองแบบชั่วครู่ปรับปรุงดีขึ้นสามารถใช้ตัวควบคุมพีดีทำการควบคุมแทน

$$c(t) = K_P \left(e(t) + \frac{1}{T_N} \int_0^t e(t) dt + T_V \frac{de(t)}{dt} \right) \quad (14)$$

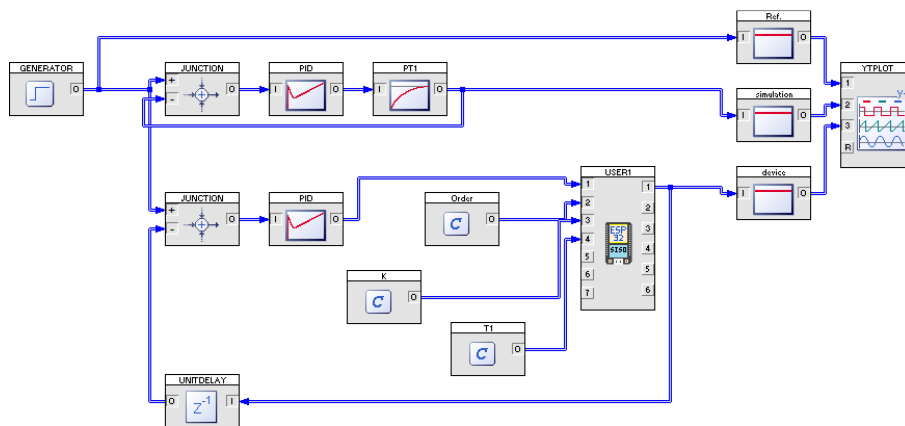


Figure 11 Closed loop control system with PID controller

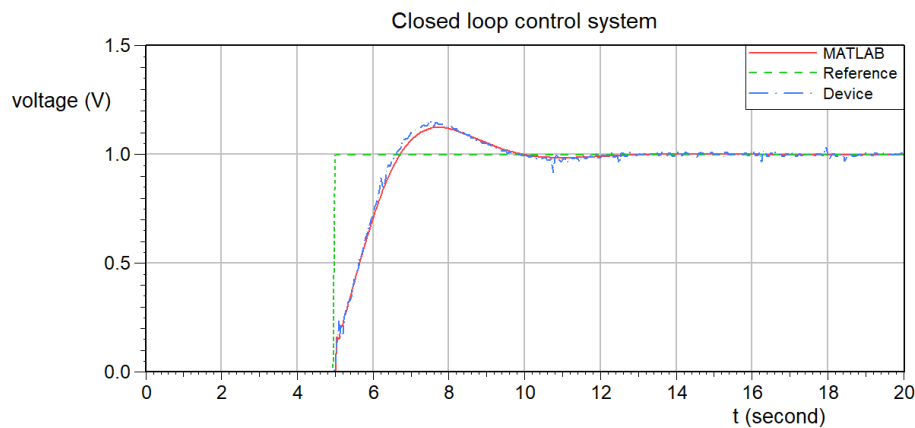


Figure 12 Step response of closed loop control system with PID controller

จากรูปที่ 10 เมื่อต้องการควบคุมกระบวนการระบบอันดับหนึ่ง (PT1) ด้วยตัวควบคุมพีไอดี (PID) สามารถต่อเป็นบล็อกไดอะแกรมได้ดังรูปที่ 11 โดยลูปควบคุมด้านล่างคือการควบคุมที่มีส่วนเครื่องอยู่ในลูป (hardware in the loop , hil) โดยกระบวนการระบบอันดับหนึ่ง (PT1) จะถูกสร้างขึ้นเหมือนรูปที่ 7 ที่มี

ค่าตัวแปรดังนี้ Order = 0, K = 1 และ T1 = 1 สำหรับตัวควบคุมพีไอดีถูกออกแบบโดยใช้วิธีลองผิดลองถูก (trial and error) ซึ่งมีค่าตัวแปรดังนี้ KP = 1, TN = 0.5 และ TV = 0.5 ส่วนลูปควบคุมด้านบนเป็นการจำลองด้วยคอมพิวเตอร์โดยมีวัตถุประสงค์เพื่อทดสอบความถูกต้องในการทำงาน ดังนั้นตัวแปรของตัวควบคุมพีไอดีและกระบวนการระบบอันดับหนึ่งจึงใช้ค่าเดียวกันกับลูปควบคุมด้านล่าง ซึ่งให้ผลการทดลองดังรูปที่ 12 โดยใช้เส้นกราฟแสดงค่าสัญญาณต่าง ๆ ดังนี้ เส้นกราฟประจักษ์ (reference) คือสัญญาณอินพุตแบบขั้นขนาด 1 โวลต์ เส้นกราฟฟีดแบ็ก (MATLAB) คือสัญญาณเอาต์พุตของการควบคุมวงปิดที่ได้จากโปรแกรม MATLAB และเส้นกราฟประจักษ์ (device) คือสัญญาณเอาต์พุตของการควบคุมวงปิดที่ได้จากโปรแกรม WinFACT/BORIS ร่วมกับอุปกรณ์ที่น่าเสนอ

อภิปรายและสรุปผลการวิจัย

จากการศึกษาระบบพลวัต ผลที่ปรากฏในรูปที่ 7 - 8 แสดงให้เห็นว่ารูปร่างของสัญญาณที่สร้างจากอุปกรณ์ที่น่าเสนอมีลักษณะเหมือนกันกับสัญญาณที่สร้างขึ้นจากโปรแกรม MATLAB โดยมีความผิดพลาดสูงสุดไม่เกิน 1% และผลที่ปรากฏในรูปที่ 9 ยังแสดงให้เห็นว่าสัญญาณของระบบอันดับสองที่สร้างขึ้นนั้นอยู่ในเกณฑ์ที่ดี ถึงแม้จะมีค่าความผิดพลาดสูงกว่าแต่ค่าสูงสุดก็ไม่เกิน 4% ซึ่งค่าความผิดพลาดที่สังเกตได้นั้นเกิดจากความซับซ้อนของระบบอันดับสองที่มีมากกว่า และจากค่าความผิดพลาดสะสมที่เกิดจากการแปลงตัวเลขทศนิยมเป็นตัวเลขจำนวนจริงของตัวประมวลผลที่มี 12 บิตที่ใช้ในแรงดันการสร้างสัญญาณ

นอกจากนี้การควบคุมวงปิดแบบที่มีส่วนเครื่องอยู่ในลูปสามารถทำได้อย่างสะดวกดังรูปที่ 11 ที่แสดงถึงความยืดหยุ่นในการปรับเปลี่ยนชนิดของตัวควบคุมได้ในหลายลักษณะ เช่น ตัวควบคุมแบบปิดเปิด ตัวควบคุมแบบพีซี ซึ่งผลการทดลองที่แสดงในรูปที่ 12 ปรากฏความผิดพลาดสูงสุดที่ไม่เกิน 4% ที่จุดสูงสุดของรูปคลื่น โดยจุดเด่นของอุปกรณ์ที่น่าเสนอคือ 1) ความสามารถในการสร้างสัญญาณของระบบพลวัตแบบเวลาจริงและ 2) ความสามารถในการควบคุมวงปิดแบบที่มีส่วนเครื่องอยู่ในลูปได้ โดยสามารถปรับเปลี่ยนทั้งชนิดและตัวแปรของระบบพลวัตและตัวควบคุมได้ง่ายด้วยโปรแกรม WinFACT/BORIS

บทความนี้นำเสนอการประยุกต์ใช้ตัวประมวลผลแบบคู่ทำงานขนานกันในการรวบรวมข้อมูลและสร้างสัญญาณของระบบพลวัตซึ่งกำหนดใช้เวลาในการคำนวณแต่ละรอบดังนี้ 1) การรวบรวมข้อมูลระหว่างบอร์ดประมวลผล ESP32 และโปรแกรม WinFACT/BORIS ที่เวลา 0.05 วินาทีและ 2) การสร้างสัญญาณของระบบพลวัตที่เวลา 0.01 วินาที โดยอุปกรณ์ที่น่าเสนอนี้มีประโยชน์ในการเรียนการสอนในรายวิชาระบบควบคุมเนื่องจากสัญญาณแรงดันไฟฟ้าที่เกิดจากอุปกรณ์ดังกล่าวสามารถนำไปจำลองระบบที่แตกต่างกันได้สามชนิด ได้แก่ 1) ระบบอันดับหนึ่ง 2) ระบบอันดับหนึ่งต่อแบบอนุกรมสองตัว และ 3) ระบบอันดับสองและยังสามารถนำอุปกรณ์นี้ไปใช้ในการควบคุมวงปิดแบบที่มีส่วนเครื่องอยู่ในลูปได้ซึ่งครอบคลุมเนื้อหาที่เป็นสาระสำคัญในรายวิชาระบบควบคุมในระดับปริญญาตรี นอกจากนี้อุปกรณ์ที่น่าเสนอยังสามารถนำไปใช้ทดแทนออสซิลโลสโคปเพื่อใช้วัดสัญญาณแรงดันไฟฟ้าเปรียบเทียบกับเวลาได้ด้วยความถี่ไม่เกิน 20 Hz และด้วยความสะดวกในการใช้งาน ประกอบกับราคาที่ถูกลงเมื่อเทียบกับเครื่องมือที่ขายเชิงพาณิชย์ อุปกรณ์นี้จึงมีความเหมาะสมในการนำไปใช้เรียนรู้ทั้งในชั้นเรียนและเรียนรู้ด้วยตนเองที่บ้าน

การพัฒนาต่อยอดจากงานวิจัยนี้มีความเป็นไปได้สามแนวทาง ได้แก่ 1) การเพิ่มวงจรกรองความถี่สูงเพื่อกรองสัญญาณรบกวนที่มีความถี่สูงออกจากสัญญาณอินพุตซึ่งมีแนวโน้มที่จะทำให้รูปคลื่นสัญญาณที่วัดได้มีความเรียบมากขึ้น 2) การเพิ่มระดับแรงดันให้สูงขึ้น ซึ่งอาจจะทำให้การประยุกต์ใช้งานกับระบบที่มีระดับแรงดันแตกต่างกันสะดวกมากขึ้นและ 3) การลดเวลาการทำงานต่อรอบให้ต่ำลง ซึ่งเป็นไปได้ที่จะทำให้การคำนวณแสดงค่าความละเอียดมากขึ้น

References

- [1] Ruprez Fombellida, J. L. (2007). USB Data Acquisition Card. *Elektor Electronics Worldwide*, 33(371), 16-20.
- [2] Abdallah, M., et al. (2011). A Low-Cost Stand-Alone Multichannel Data Acquisition, Monitoring, and Archival System With On-Chip Signal Preprocessing. *IEEE Transactions on Instrumentation and Measurement*, 60(8), 2813–2827.
- [3] Sharma, S., & Kumar, G. (2011). Low Cost Data Acquisition System for Students. *International Journal of Electrical and Computer Engineering*, 2(2), 41–44.
- [4] Beltrán, A. E. (2013). Low-Cost Acquisition and Development Board: An Open Source Hardware Proposal. In *Symposium of Signals, Images and Artificial Vision* (p. 1–4). 11–13 September, 2013, Bogota, Colombia.
- [5] Blanco, J. R., et al. (2014). A Low-Cost Open-Source Data Acquisition System. In *IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)* (p. 1–6). 11–14 February, 2014, Barcelona, Spain.
- [6] Tehami, S., et al. (2015). Hardware and Software Designing of USB Based Plug n Play Data Acquisition Device With C# and LabVIEW Compatibility. In *IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)* (p. 143–146). 22–25 October, 2015, Brasov, Romania.
- [7] Haizad, M., et al. (2016). Development of Low-Cost Real-Time Data Acquisition System for Process Automation and Control. In *2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)* (p. 1–5). 25–27 September, 2016, Ipoh, Malaysia.
- [8] Wali, S., & Areeb, M. (2018). Development of Low-Cost DAQ for Power System Signals Using Arduino. In *IEEE 21st International Multi-Topic Conference (INMIC)* (p. 1–5). 1–2 November, 2018, Karachi, Pakistan.
- [9] National Instruments. (2023). User Guide NI USB-6001/6002/6003 Low-Cost DAQ USB Device. National Instruments.
- [10] Measurement Computing Corporation. (2000). *CD-7021/CB7021P & CB-7022, CB-7024 User's Manual*. Measurement Computing Corporation.
- [11] Krause, N., et al. (2021). Multi-Core Microcontroller Hardware In the Loop System for

- Electric Machine Control. In *IEEE Energy Conversion Congress and Exposition (ECCE)* (p. 4711–4718). 10–14 October, 2021, Vancouver, BC, Canada.
- [12] Dokic, K., et al. (2020). Neural Networks With ESP32 - Are Two Heads Faster Than One?. In *6th Conference on Data Science and Machine Learning Applications (CDMA)* (p. 141–145). 4–5 March, 2020, Riyadh, Saudi Arabia.
- [13] Li, J.-H. (2018). Control System Laboratory With Arduino. In *International Symposium on Computer, Consumer and Control (IS3C)* (p. 181–184). 6–8 December, 2018, Taichung, Taiwan.
- [14] Dalapati, S. (2020). A Control Systems Laboratory Experiment on Transfer Function Emulation. In *IEEE Applied Signal Processing Conference (ASPCON)* (p. 51–55). 7–9 October, 2020, Kolkata, India.
- [15] Phillips, C. L., & Harbor, R. D. (1999). *Feedback Control Systems* (2nd ed.). Prentice-Hall.
- [16] Kahlert, J. (2024). *WinFACT Documentation*. Available from <http://www.kahlert.com>. Accessed date: 10 January 2024.