All Sciences Proceedings
http://as-proceeding.com/

3rd International Conference on Innovative Academic Studies

September 26-28, 2023 : Konya, Turkey

ICIAS
https://www.icias.net/

Active Disturbance Rejection Control of a Coupled Tank System with Raspberry Pi Implementation

Erdem Ilten

Electrical-Electronics Engineering, Balikesir University, Turkey

erdemilten@balikesir.edu.tr

Abstract – A coupled tank system is designed for liquid level control application with Raspberry Pi 4B board. To control of the liquid level, Active Disturbance Rejection Control (ADRC) method is used. Control algorithm is prepared on Simulink environment by using *Simulink Support Package for Raspberry Pi hardware* tool of Matlab. Then, the algorithm is loaded to Raspberry Pi board to run as microcontroller. The level control system is tested in real-time and results are obtained. It is seen that the ADRC method works stably in the system.

Keywords – Liquid Level Control, Raspberry Pi, Simulink, External Mode, ADRC

I. Introduction

The coupled tank system, often referred to as a two-tank system or dual-tank system, is a fundamental example in the field of control engineering and process control. It serves as a simple yet illustrative model for studying and understanding the principles of dynamic systems, feedback control, and fluid dynamics [1]. At its core, the coupled tank system consists of two interconnected tanks, typically cylindrical in shape, filled with liquid. These tanks are connected in such a way that the fluid level in one tank affects the fluid level in the other. By manipulating valves, pumps, or other control mechanisms, the flow of liquid between the tanks can be regulated, offering a practical and dynamic system for control engineering experiments and demonstrations [2]. The coupled tank system finds applications in various industrial processes, such as chemical engineering, water treatment, and manufacturing, where precise control of fluid levels, flow rates, and interactions between interconnected tanks is critical. Furthermore, it serves as a valuable teaching tool in academic settings, helping students grasp the concepts of feedback control, system dynamics, and the practical challenges involved in real-world control systems.

In the realm of control engineering, where precision and stability are paramount, the pursuit of innovative control strategies has led to the development of various advanced techniques. Among these, Active Disturbance Rejection Control (ADRC) stands out as a promising and dynamic approach that has gained considerable attention and recognition in recent years [3]. ADRC represents a departure from traditional control methods by addressing disturbances in real-time with an adaptive and robust framework. Unlike classical control techniques, which often rely on precise mathematical models of the system, ADRC takes a fundamentally different approach by treating disturbances as the central focus of control, regardless of the complexity or variability of the underlying system [4]. At its core, ADRC operates on the principle that any dynamic system is subject to external disturbances, noise, and uncertainties. These disturbances can often lead to deviations from desired setpoints and degrade system performance. ADRC seeks not only to mitigate the effects of disturbances but to actively estimate, predict, and counteract them in real-time, thereby enhancing system stability and performance [5].

The Raspberry Pi, originally designed as an affordable and compact single-board computer for educational purposes, has evolved into a versatile platform that extends far beyond its initial scope. One of its remarkable capabilities is serving as a highly capable microcontroller for a wide range of electronics and IoT (Internet of Things) projects. This transformation has empowered hobbyists, engineers, and innovators to harness the computational power and connectivity of the Raspberry Pi in a microcontroller form factor. Traditionally, microcontrollers have been the workhorses of embedded systems, handling tasks that require real-time control, sensing, and interfacing with the physical world [6]. However, the Raspberry Pi, with its quad-core processor, ample memory, and comprehensive I/O options, presents an enticing alternative to traditional microcontrollers. While it may not match the power efficiency of dedicated microcontrollers, it excels in flexibility, computational capability, and the ability to run full-fledged operating systems like Linux. Using a Raspberry Pi as a microcontroller opens up a realm of possibilities. It allows you to leverage a familiar computing environment for your embedded projects, complete with programming languages like Python, C, and C++ for application development [7]. Additionally, its HDMI output, USB interfaces, and networking capabilities enable easy debugging, remote access, and integration with other devices and services.

Simulink, developed by MathWorks, is a powerful and widely-used graphical modeling and simulation environment for engineers, scientists, and researchers [8]. It provides an intuitive and visual platform for designing, simulating, and analyzing complex dynamic systems, making it an invaluable tool in various fields, including control systems, signal processing, communications, and more [9]. With its extensive library of pre-built blocks and a user-friendly interface, Simulink allows users to create models that represent real-world systems, enabling them to gain insights, perform simulations, and optimize designs before physical implementation. Whether you're working on academic research, industrial projects, or product development, Simulink's versatility and robust features make it an essential tool for tackling complex engineering challenges.

In the ever-evolving landscape of embedded systems and software development, Simulink Embedded Coder stands as a powerful and indispensable tool [10]. Developed by MathWorks, the same company behind MATLAB and Simulink, Simulink Embedded Coder is a specialized software solution designed to bridge the gap between model-based design and the efficient, real-world implementation of embedded systems [11]. Embedded systems, which are the backbone of countless modern devices and technologies, require precise, reliable, and often resource-efficient software. Simulink, a graphical modeling environment within MATLAB, provides a visual means for engineers and developers to design and simulate complex systems [12]. Simulink Embedded Coder takes the models created in Simulink and translates them into highly optimized, production-quality C and C++ code, ready for deployment on embedded microcontrollers, processors, and FPGAs (Field-Programmable Gate Arrays). The power of Simulink Embedded Coder lies in its ability to automate and streamline the code generation process, significantly reducing the time and effort required to transition from simulation to real-world deployment [13]. It generates code that is not only highly efficient but also adheres to industry standards and safety-critical guidelines, making it suitable for a wide range of applications, from automotive control systems to aerospace avionics and beyond [14].

The aim of this study is to control the liquid level of the second tank of the coupled tank system at desired level with ADRC by using Raspberry Pi.

II. Materials and Method

Coupled tank system is prapared to control the liquid level of second tank at desired value by using ADRC method. Block diagram of the system is demonstrated in Fig. 1. The experimental setup is illustrated in Fig. 2. The ADRC system can be defined as below [15].

$$\begin{aligned} \dot{e}(t) &= A_n e(t) + b_n\left[\hat{b}(\hat{e},t)u(t) + d(e,\hat{e},u,t)\right], \\ y(t) &= c_n^T e(t) - \omega(t) \end{aligned} \tag{1}$$

$$\begin{aligned} \dot{z}(t) &= A_{n+1} z(t) + b_{n+1}\dot{d}(z,\hat{z},u,t) + d_{n+1}\hat{b}(\hat{e},t)u(t), \\ y(t) &= c_{n+1}^T z(t) - \omega(t) \end{aligned} \tag{2}$$

$$\dot{\tilde{z}}(t)(A_{n+1}-l_{n+1}c_{n+1})\tilde{z}(t)+b_{n+1}\dot{d}(t)-l_{n+1}w(t),$$
$$\dot{e}(t)=(A_n-b_nk_n)e(t)+[k_n \quad 1]\tilde{z}(t). \qquad (3)$$

Where $e(t)$ is the error function, $y(t)$ is the output, $z(t)$ is the extended state. $\tilde{z}(t)$ is the observation error $\tilde{z}(t)=z(t)-\hat{z}(t)$. In Eq. (3), l_{n+1} and k_n can be defined with ω_o and ω_c as in Eq. (4).

$$l_i=\frac{(n+1)!}{i!(n+1-i)!}\omega_o^2 \text{ for } i\{1,\ldots,n+1\}$$
$$k_i=\frac{n!}{i!(n-i)!}\omega_c^2 \text{ for } i\in\{1,\ldots,n\} \qquad (4)$$

Here $\hat{b}$ is the input gain, ω_o is the observer bandwidth and ω_c is the controller bandwidth.

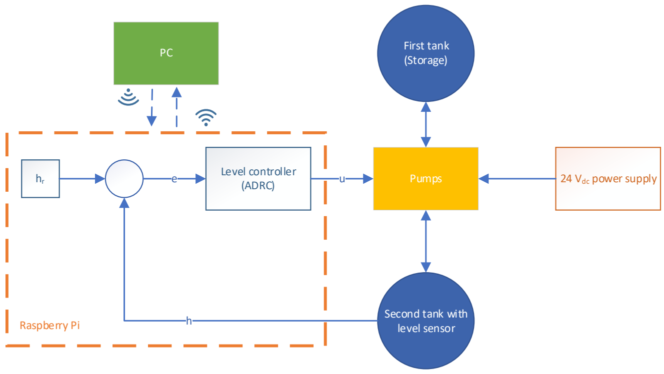

Fig. 1 Block diagram of coupled tank system.

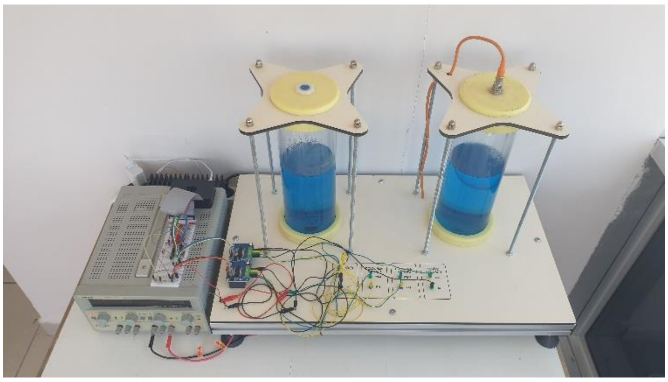

Fig. 2 The experimental setup.

Control system is designed on Matlab/Simulink program for Raspberry Pi board. The designed control panel and block diagram are presented in Fig. 3 and Fig. 4, respectively. *Simulink Support Package for Raspberry Pi hardware* tool of Matlab [16] is used for design of these control panel and block diagram on Simulink environment.

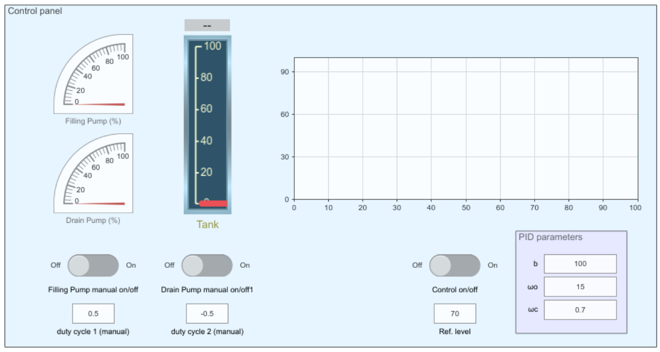

Fig. 3 Control panel.

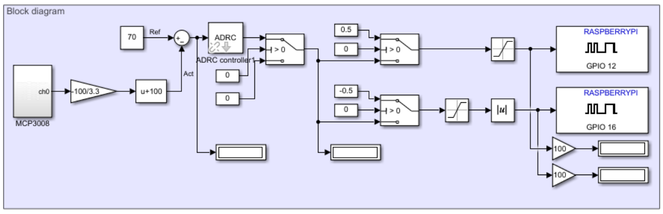

Fig. 2 Block diagram.

III. RESULTS

The designed coupled tank system is tested with ADRC for liquid level control. Test result is demonstrated in Fig. 5.

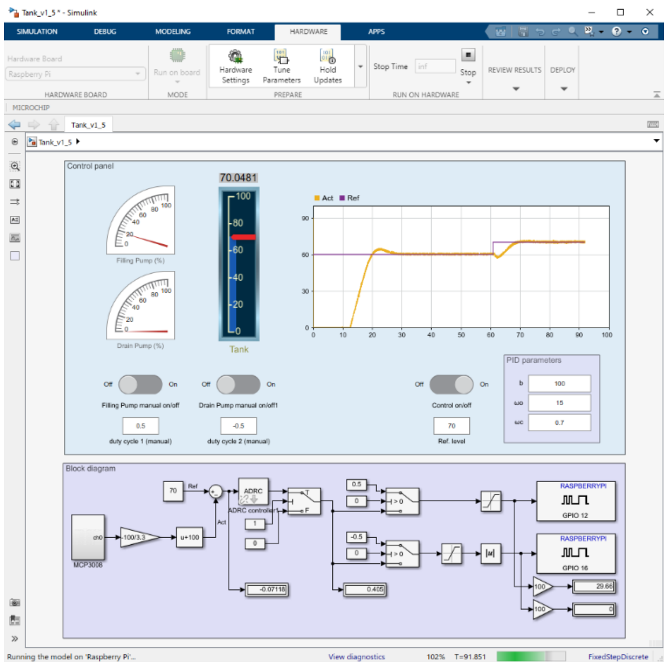

Fig. 5 Test result.

ADRC coefficients are selected as 100, 15 and 0.7 for $\hat{b}$, ω_o and ω_c, respectively. The system is started with reference level for 50. Then switched to 60. ADRC controls the liquid level as seen in Fig. 5. Unfortunately, the performance is not very successful because the ADRC coefficients are not optimized.

IV. Conclusion

Liquid level control experimental setup for a coupled tank system is designed in this study. Raspberry Pi 4B board is used to control the system. Liquid level of the second tank is controlled with using ADRC method. In the first experiments, it is seen that the system can be operated stably with ADRC. In order to achieve successful control of the system, it is planned to optimize the ADRC coefficients in future studies.

References

[1] Ilten E, Demirtas M. Liquid Level Control Interface Design on Simulink External Mode with Raspberry Pi. Int. Conf. Mod. Adv. Res., 2023, p. 85–8.

[2] E. GK, J. A. Control of nonlinear two-tank hybrid system using sliding mode controller with fractional-order PI-D sliding surface. Comput Electr Eng 2018;71:953–65. https://doi.org/https://doi.org/10.1016/j.compeleceng.2017.10.005.

[3] Feng H, Guo B-Z. Active disturbance rejection control: Old and new results. Annu Rev Control 2017;44:238–48.

[4] Huang Y, Xue W. Active disturbance rejection control: Methodology and theoretical analysis. ISA Trans 2014;53:963–76.

[5] Fareh R, Khadraoui S, Abdallah MY, Baziyad M, Bettayeb M. Active disturbance rejection control for robotic systems: A review. Mechatronics 2021;80:102671.

[6] Ilten E. DC motor position control with Raspberry Pi on Simulink wireless external mode. Int. Conf. Pioneer Innov. Stud., vol. 1, 2023, p. 228–31.

[7] Ilten E, Demirtas M. Fuzzy Logic Position Control of DC Motor with Raspberry Pi and Real-Time Monitoring on Simulink External Mode. 1st Bilsel Int. World Sci. Res. Congr., 2023, p. 189–95.

[8] Ilten E, Demirtas M. Fractional order super-twisting sliding mode observer for sensorless control of induction motor. COMPEL-The Int J Comput Math Electr Electron Eng 2019;38:878–92.

[9] İlten E. Asenkron motor için adaptif kesirli kayan kipli gözlemci tasarımı 2019.

[10] Ilten E. Permanent Magnet Synchronous Motor Control Interface Design and Implementation with TI F28335 DSP on Simulink External Mode. Int. Conf. Appl. Eng. Nat. Sci., vol. 1, 2023, p. 67–70.

[11] Ilten E, Demirtas M. Permanent Magnet Synchronous Motor Flux Observer Implementation with TI F28335 DSP for Electric Vehicle System. Int. Conf. Mod. Adv. Res., 2023, p. 89–92.

[12] Ilten E. Conformable fractional order controller design and optimization for sensorless control of induction motor. COMPEL-The Int J Comput Math Electr Electron Eng 2022;41:1528–41.

[13] Ilten E. Conformable Fractional Order Controller Design and Implementation for Per-Phase Voltage Regulation of Three-Phase SEIG Under Unbalanced Load. Electr Power Components Syst 2022;50:636–48.

[14] Ilten E. Induction Motor Speed Controller Interface Design Via Simulink External Mode. II. Int. Jt. Conf. Eng. Sci. Artif. Intell., 2022, p. 59–63.

[15] Lakomy K, Giernacki W, Michalski J, Madonski R. Active disturbance rejection control (adrc) toolbox for matlab/simulink. ArXiv Prepr ArXiv211201614 2021.

[16] MathWorks. MATLAB documentation 2023.