

UNIVERSIDADE FEDERAL DE SANTA CATARINA CENTRO TECNOLÓGICO  
ENGENHARIA DE CONTROLE E AUTOMAÇÃO  
DAS 5332 - Arquitetura e Programação de Sistemas Microcontrolados  
Prof. Marcelo Menezes Morato

ISAQUE LUIS GALLINA VERONA, JOÃO VITOR ORCHULHAK, ARTHUR SWENSON  
COUTINHO, CEZAR AUGUSTO DUDA

**REGULAÇÃO DE NÍVEL EM TANQUE PULMÃO  
MODELAGEM E CONTROLE**

**Florianópolis  
2024**

## 1. Introdução

Este relatório detalha o experimento do primeiro trabalho experimental da disciplina de Arquitetura e Programação de Sistemas Microcontrolados: implementação de um sistema de controle embarcado para a regulação de nível no tanque tipo pulmão. O trabalho foi feito como uma simulação no software MATLAB SimuLink, com o uso do kit de microcontrolador para o controle do processo. Os objetivos do experimento incluem a observação global das estratégias de controle embarcado e a análise do comportamento do processo.

O módulo de desenvolvimento utilizado foi o ESP32-WROOM-32 com chip de porta serial CH9102X. O ESP32 tem uma arquitetura de 32 bits baseada em um modelo Harvard modificado, o que significa que possui memórias de dados e instruções separadas. Ele inclui dois núcleos de processamento Xtensa LX6 (dual-core) com capacidades de 80MHz até 240 MHz. Quanto à memória, ele possui até 520 KB de SRAM e uma vasta gama de periféricos ADC, DAC, interfaces UART, I2C, SPI, PWM e sensor capacitivo. Alguns pinos (GPIO 6 a 11) estão reservados para o uso com a flash SPI integrada e não devem ser usados para outras funções(ESP 32 Wroom).

## 2. Desenvolvimento

Para programação do microcontrolador foi utilizado o driver da empresa Nanjing Qinheng Microelectronics Co., Ltd. para o respectivo chip de comunicação serial. A IDE utilizada foi o V.S Code utilizando a extensão Platform I/O que fica responsável pela programação do microcontrolador. Ademais, as bibliotecas para manuseio da comunicação via WiFi foram utilizadas a biblioteca própria WiFi.h do arduino que permite receber e enviar pacotes, bem como WiFiServer.h que cria um servidor para escutar uma porta.

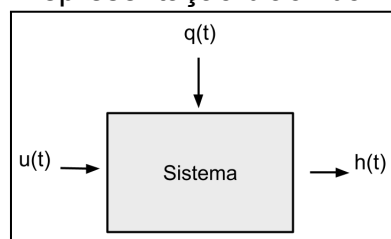
Já para a parte de modelagem, foi utilizado o Simulink, um software de desenvolvimento de simulações utilizando blocos e funções do matlab. Foi utilizada uma biblioteca para manuseio da comunicação WiFi da parte da simulação chamada Instrument Control Toolbox.

### 1. Modelagem

Para desenvolver o modelo do sistema, foi necessário uma equação que representasse o modelo desejado, para isso utilizamos as construções fenomenológicas do processo que foram disponibilizadas, relacionando as características físicas do processo em forma de equações, seguindo o modelo básico de Volume sendo igual a vazão de entrada menos a vazão de saída.

Sobre as relações entre as variáveis do sistema, teríamos duas que são relacionadas com a variável controlada, que seria o  $r(h(t))$  (raio do tanque) que varia com a altura  $h$ , a vazão de saída da qual sua fórmula é representada pela equação de bernoulli, a vazão de entrada  $u(t)$  e também  $q(t)$  a vazão de entrada não controlada.

**Figura 1** - Representação black box do sistema



Fonte: Autor (2024)

A partir da abstração, conseguimos tomar os primeiros passos para a modelagem, relacionando o  $h(t)$  com as variáveis de entrada e saída, retornando. Transformando a equação, chegamos em um modelo de equação contínua não linear equação 1. E também em um modelo de equação linearizada, equação 2.

**Equação 1** - Equação continua manipulada isolando  $\frac{dh(t)}{dt}$

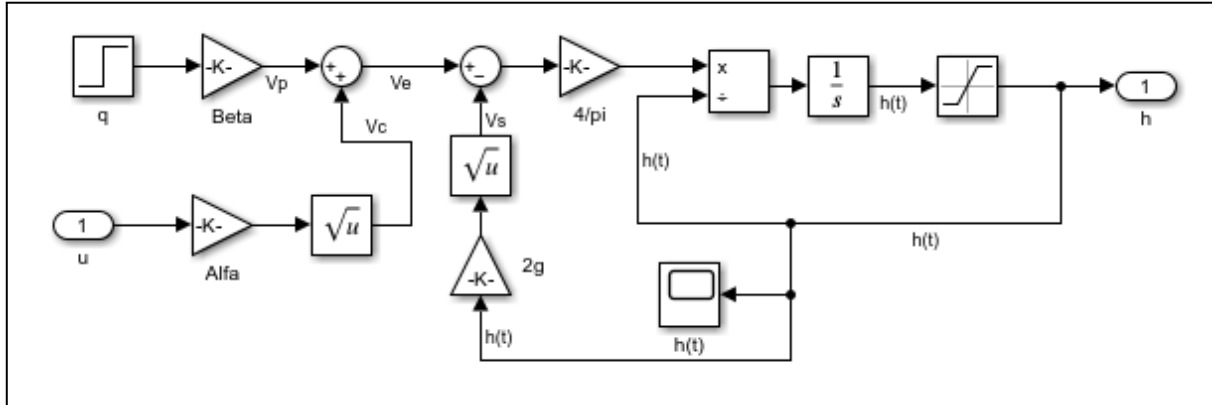
$$\frac{dh(t)}{dt} = \frac{4(\sqrt{2g} u(t) + \sqrt{0.1g} q(t) - \sqrt{2g} h(t))}{\pi h^2(t)}$$

**Equação 2** - Equação linearizada

$$\frac{d\Delta h}{dt} = (\sqrt{2g} u_o + \frac{\sqrt{2g}}{2\sqrt{u_o}} \Delta u + \sqrt{0.1g} q_o + \sqrt{0.1g} q_o \Delta q - \sqrt{2g} h_o - \frac{\sqrt{2g}}{2\sqrt{h_o}} \Delta h) (\frac{4}{\pi h_o^2} - \frac{8}{\pi h_o^3} \Delta h)$$

A partir da equação 1 montou-se um diagrama de blocos no simulink para a simulação do processo conforme a variação das entradas do sistema.

**Figura 2** - Diagrama de blocos no Simulink da equação da Figura 1



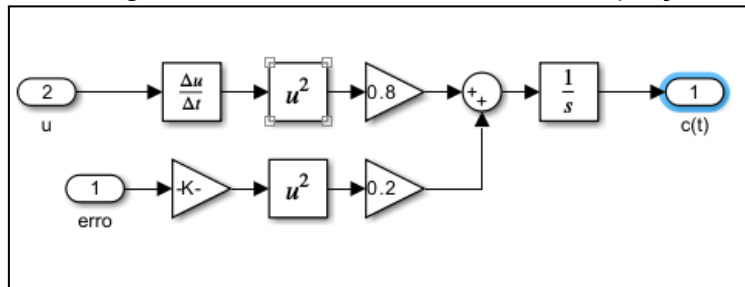
Fonte: Autor (2024)

O diagrama de blocos da função custo foi feito de modo semelhante, utilizando a equação 3.

**Equação 3** - Custo de operação

$$c(t) = \int_0^t 0.8 \left( \frac{du(\mu)}{d\mu} \right)^2 + 0.2 (e(\mu))^2 d\mu.$$

**Figura 3** - Diagrama de blocos no Simulink da equação de custo



Fonte: Autor (2024)

## 2. Controle

O sistema de controle do processo foi desenvolvido seguindo a lei de controle dada pela equação 4.

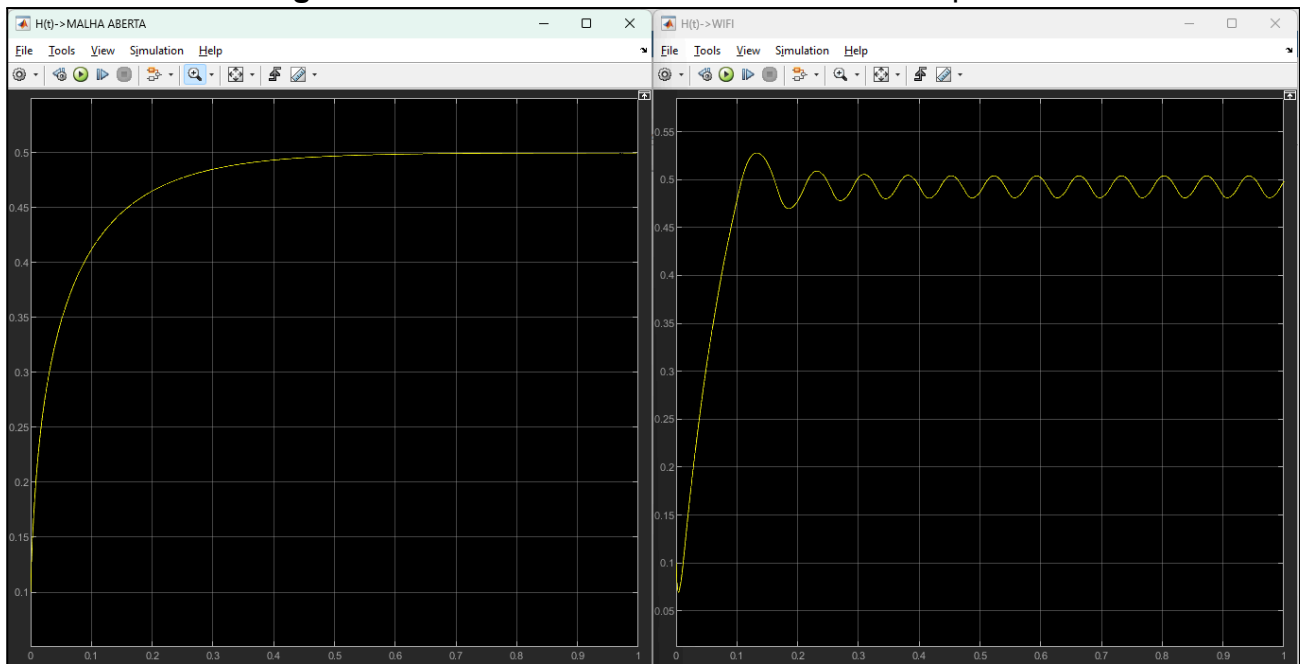
**Equacao 4** - Lei de controle

$$u(t) = (Kp e(t))^2$$

Onde  $Kp$  é uma constante arbitrária,  $e(t) = (ref(t) - h(t))$ , referência e o nível que queremos que o tanque atinja,  $h(t)$  e o nível real do tanque no tempo  $t$ .

Utilizando o resultado de testes empíricos, o  $Kp$  foi definido como 5 para valores de erro variando entre  $[0, 100]$ , assim o sistema em malha fechada (com o controle atuando) respondia 2x mais rápido que em malha aberta, como pode ser observado na figura 4 abaixo.

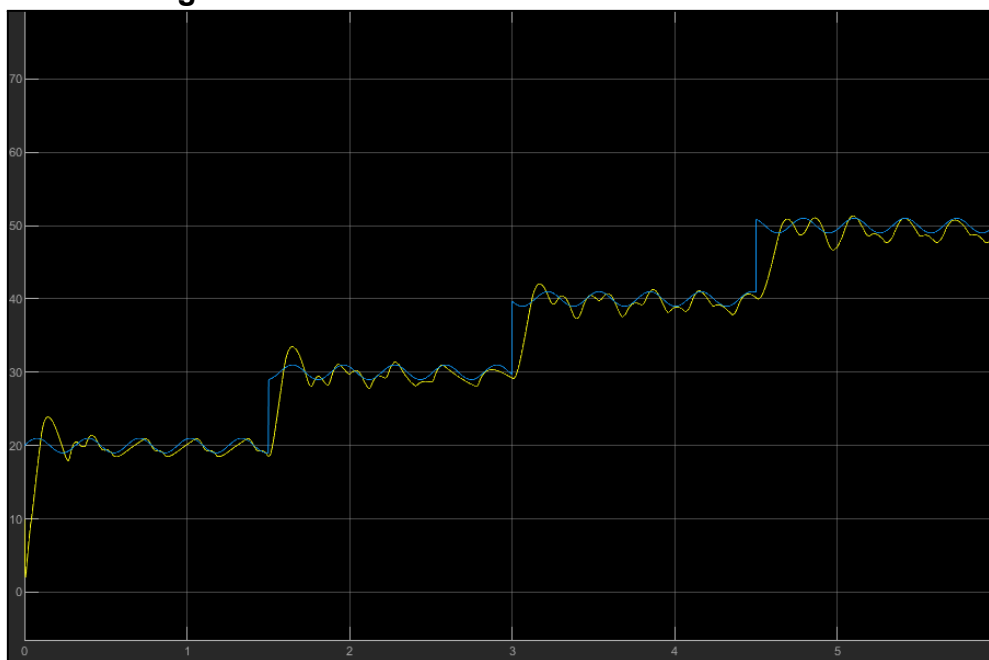
**Figura 4 - malha aberta X malha fechada com  $K_p = 5$**



Fonte: Autor (2024)

Já para a validação da lei de controle, em uma simulação de 6 segundos, foi aplicado uma referência em forma de degrau somado a uma fonte de sinal senoidal, a dinâmica entre a referência e o nível pode ser observada na figura 5, a referência em linha azul e o nível do tanque em linha amarela.

**Figura 5 - Controle atuando variando a referência**



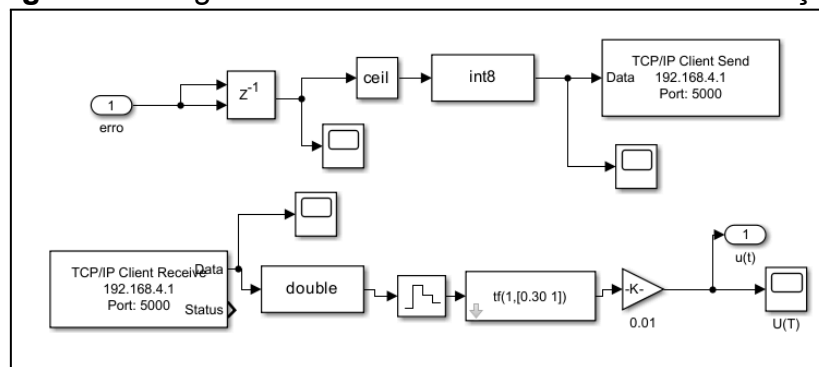
Fonte: Autor (2024)

### 3. Comunicação

O protocolo de comunicação utilizado foi o TCP/IP onde o microcontrolador cria uma rede de acesso de nome NodeMCU, e cria um servidor que fica escutando na porta 5000. O IP padrão do ESP32 é 192.168.4.1. Já do lado da simulação, existem dois blocos, O "Send" que cria um cliente no endereço 192.168.4.1:5000, e envia um byte que representa o valor do erro  $e(t)$  calculado, e o "Receive" fica escutando a resposta do servidor (microcontrolador), que chega 1 byte também, já saturado, variando entre  $[0,100]$  que é o sinal de controle  $u(t)$ . Esse sinal discreto é então transformado em contínuo com

uma função do tipo ZOH (zero order hold), filtrado com um filtro passa baixa e por fim realimentado no sistema.

**Figura 6** - Diagrama de blocos no Simulink da comunicação



Fonte: Autor (2024)

### 3. Considerações finais

Vale destacar que dependendo do ganho  $K_p$  da lei de controle, o sistema respondia de forma mais eficiente ou menos dependendo do erro, logo podemos supor que uma relação de proporcionalidade entre o  $K_p$  e o erro seria uma relação boa para a lei de controle, então uma nova proposta para a lei de controle a ser explorada seria:

$$u(t) = (K_p(t) e(t))^2, K_p(t) = \varphi e(t) \Rightarrow u(t) = \varphi^2 (e(t))^4. \text{ Sendo } \varphi \text{ uma constante.}$$

Outra consideração importante foi que quanto mais tempo de simulação passava, mais instável o sinal de controle ficava, uma hipótese levantada sobre foi: realimentação de erro do sinal  $u(t)$  no sistema, a implementação de um filtro ajudou mas não fez o problema acabar, esse erro parece estar associado com a velocidade de comunicação WiFi e transferência de dados.

Em suma, o grupo não apresentou uma dificuldade tão grande ao longo da primeira parte do trabalho que consistia em modelar o sistema e implementá-la no SimuLink do MATLAB, o maior problema enfrentado foi para realizar a comunicação entre o microcontrolador ESP32 e o Simulink. Entretanto, após algum tempo de trabalho, diferentes formatos de dados enviados, muitos testes e debugs, foi possível realizar uma comunicação estável, porém com algumas perdas de dados e alguns outliers, entre os dois e em seguida feitos experimentos para comprovar a eficácia do modelo e da lei de controle. Muitos erros cometidos ao longo do trabalho tem uma forte relação com a falta de conhecimentos sobre as ferramentas do Matlab, bem como a lógica da programação em C, principalmente sobre a declaração de variáveis e os diferentes tipos. Ao fazer o debug dos códigos, era necessário rever dois códigos e isso foi a tarefa mais árdua, muitas horas de testes e compilação. Desse modo, pode-se afirmar que o grupo obteve novos conhecimentos acerca de como se dá a comunicação com um microcontrolador e adquiriu mais noções acerca da ferramenta SimuLink do MATLAB.

### 4. Referências

- ESP32-WROOM-32 Datasheet, disponível em: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_data\\_sheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_data_sheet_en.pdf)
- Comunicação Wireless TCP com ESP, disponível em: <https://portal.vidadesilicio.com.br/comunicacao-wireless-esp-protocolo-tcp/>
- Send and Receive Data over TCP/IP Network disponível, em: <https://www.mathworks.com/help/instrument/send-and-receive-data-over-a-tcpip-network.html>
- Communication WiFi, disponível em: <https://docs.arduino.cc/libraries/wifi/#Client%20class>