



Projeto USV

Relatório USV - Controlo Autónomo Arduino



Prof. Pedro Teodoro

Trabalho realizado por:

- Christian Rodrigues 15202
- Rodrigo Salgueiro 15213
- Diogo Paixão 15208
- Vasco Ferreira 15200

ÍNDICE

Introdução	2
Desenvolvimento	2
Explicação do código.....	2
Conclusão	6

Introdução

O presente trabalho teve como principal objetivo a implementação de um sistema de controlo de um motor, recorrendo à integração de dois dispositivos: o Arduino e o Raspberry Pi.

Para alcançar esse objetivo, foi necessário estabelecer uma comunicação eficaz entre os dois dispositivos, assegurando a troca de dados e comandos de forma estável e fiável.

Para além disso, foi também estabelecida a comunicação com uma câmara, de forma a permitir a aquisição de imagem como complemento ao controlo do sistema.

Desenvolvimento

O sistema desenvolvido consiste num protótipo de controlo de motores com capacidade de operação em dois modos distintos — manual e automático — integrando três subsistemas principais: comunicação via rádio com um módulo **nRF24L01**, processamento de imagem através de uma **câmara ligada à Raspberry Pi**, e atuação de **três servomotores** controlados por um **Arduino Mega**. A comunicação entre o Arduino e a Raspberry Pi é feita por **interface serial (UART)**, permitindo o envio contínuo de coordenadas provenientes da análise de imagem.

Explicação do código

Durante o desenvolvimento do sistema de controlo para o modo autónomo do catamarã, foi necessário repensar por completo a lógica de atuação dos motores principais. A versão inicial baseava-se em comandos manuais simples e numa estrutura de código pouco modular. Com o progresso do projeto, essa abordagem revelou-se insuficiente para responder aos desafios reais de navegação em ambientes com perturbações (como correntes e variações na deteção da boia).

Foi então desenvolvida de raiz a função `calcular_controlo_automatico()`, responsável por gerar os sinais PWM a aplicar aos motores A e B de forma inteligente, tendo como base a posição da boia detetada pela Raspberry Pi. Esta nova abordagem introduziu um controlador proporcional (P) como forma de alinhar e aproximar o veículo da boia. O controlo atua sobre dois eixos:

- Erro horizontal (X): utilizado para corrigir a direção (viragem) do barco.
- Erro vertical (Y): utilizado para ajustar a aproximação ou manter a posição.

Consoante a proximidade da boia e o alinhamento detetado, a função adapta o comportamento dos motores entre duas fases distintas:

Fase de aproximação, com valores de ganho que permitem correções mais agressivas para reduzir a distância.

Fase de posicionamento, com correções mais suaves para garantir estabilidade junto da boia.

Além da lógica de controlo, a função integra mecanismos de tolerância e filtragem de erro para evitar oscilações desnecessárias e responder apenas a desvios relevantes. Os valores PWM calculados são ainda limitados a intervalos seguros, prevenindo danos nos motores.

Este desenvolvimento está intimamente ligado a outras melhorias implementadas ao longo do projeto:

A integração de um terceiro motor (motor C), usado apenas no modo manual ou para tarefas auxiliares;

A criação de uma estrutura modular, com funções específicas para leitura de dados, controlo de LED, comunicação e controlo de motores;

A implementação de comunicação serial entre o Arduino e o Raspberry Pi, permitindo ao sistema reagir em tempo real às deteções visuais da boia.

Estes avanços resultaram numa navegação mais precisa, responsiva e robusta, mesmo em condições adversas. A validação do sistema foi realizada com base em testes práticos e ajustamentos iterativos, com contributos valiosos do Professor Carlos Gonçalves e de toda a equipa de desenvolvimento.

Abaixo apresentam-se os blocos mais relevantes da função `calcular_controlo_automatico`, que permite ao veículo ajustar a sua trajetória com base na posição da boia detetada pela Raspberry Pi. Esta lógica representa um controlo proporcional (P) com adaptação de ganhos consoante a fase de navegação.

1. Cálculo dos Erros e Verificação de Centragem

```
int erroX = CENTRO_IMAGEM_X - cx;  
int erroY = Y_MINIMO_PROXIMO - cy;  
bool centradoX = abs(erroX) < TOLERANCIA_CENTRADO_X;  
bool centradoY = abs(erroY) < TOLERANCIA_CENTRADO_Y;
```

- Calcula a diferença entre a posição atual do objeto detetado (coordenadas `cx`, `cy`) e os valores de referência definidos no centro da imagem.
- Verifica se o veículo está suficientemente alinhado e próximo para transitar para o modo de posicionamento.

2. Determinação do Modo de Atuação

```

if (cy >= Y_MINIMO_PROXIMO && centradoX && centradoY) {
    modo_posicionamento = true;
} else {
    modo_posicionamento = false;
}

```

- Define se o veículo deve manter a posição (modo de posicionamento) ou se ainda está em fase de aproximação à boia.

3. Ajuste dos Erros e Definição dos Ganhos

```

if (!centradoX) { /* mantém erroX */ } else { erroX = 0; }
if (!modo_posicionamento && centradoY) { erroY = 0; }

```

```

float KpX = modo_posicionamento ? 0.5 : 0.5;

```

```

float KpY = modo_posicionamento ? 0.5 : 0.7;

```

- Elimina pequenos erros que já não requerem correção, evitando oscilações.
- Define ganhos diferentes para os eixos X e Y, ajustados conforme o modo de atuação.

4. Cálculo e Aplicação dos Sinais aos Motores

```

int sinalVirar = erroX * KpX;

```

```

int sinalAvancar = erroY * KpY;

```

```

int motorA_output = PWM_CENTRO + sinalAvancar + sinalVirar;

```

```

int motorB_output = PWM_CENTRO + sinalAvancar - sinalVirar;

```

```

motorA_output = constrain(motorA_output, 1000, 2000);

```

```

motorB_output = constrain(motorB_output, 1000, 2000);

```

```

motorA.writeMicroseconds(motorA_output);

```

```

motorB.writeMicroseconds(motorB_output);

```

- Aplica os sinais de controlo calculados aos motores A e B.
- O valor sinalVirar ajusta a direção (rotação), e sinalAvancar regula o avanço.
- Os valores PWM são limitados aos intervalos seguros entre 1000 e 2000 microsegundos.

5. Saída para Depuração

```
Serial.print("MOD0: ");  
Serial.print(mod0_posicionamento ? "POSICIONAMENTO" :  
"APROXIMACAO");  
Serial.print(" | erroX: ");  
Serial.print(erroX);  
Serial.print(" | erroY: ");  
Serial.print(erroY);  
Serial.print(" | PWM A: ");  
Serial.print(motorA_output);  
Serial.print(" | PWM B: ");  
Serial.println(motorB_output);
```

- Esta secção envia informações de diagnóstico pela porta série para facilitar testes e ajustes do sistema durante a fase de desenvolvimento.

Este controlo proporcional poderá futuramente ser expandido para um sistema PID completo (Proporcional-Integral-Derivativo), permitindo maior robustez e suavidade na resposta, especialmente em condições com perturbações ambientais mais acentuadas.

Conclusão

O desenvolvimento deste sistema de controlo para um veículo de superfície não tripulado (USV) permitiu explorar e aplicar conceitos fundamentais de eletrónica, programação embarcada e robótica aplicada ao meio marítimo. Através da integração entre o Arduino Mega, o módulo de comunicação nRF24L01 e a Raspberry Pi, foi possível conceber um protótipo funcional capaz de operar tanto em modo manual como automático.

Durante o projeto, destacaram-se várias evoluções importantes: a modularização do código, a introdução de um terceiro motor para apoio em manobras específicas e, sobretudo, a criação de uma função de controlo automático baseada em lógica proporcional (P), com ganhos adaptativos consoante o estado da aproximação à boia. Esta função demonstrou ser eficaz ao garantir estabilidade e resposta fluida mesmo em condições de perturbação, como ondulação ou variações na deteção visual.

O trabalho colaborativo da equipa e o apoio contínuo do professor orientador foram essenciais para a superação dos desafios encontrados ao longo do desenvolvimento. Ficou ainda identificada a possibilidade de evolução futura do sistema para um controlo PID completo, aumentando a precisão e robustez da navegação autónoma.

No geral, o projeto permitiu consolidar conhecimentos técnicos e demonstrar a viabilidade de soluções autónomas para navegação em ambientes reais, constituindo uma base sólida para trabalhos futuros no domínio da robótica marítima.