

UNIVERSIDAD NACIONAL DE MOQUEGUA
INGENIERÍA DE SISTEMAS E INFORMATICA



TEMA:

INFORME DEL CARRITO BOMBER CON CONTROL PD Y PID

ESTUDIANTES:

HUAILLA RAMOS, GUSTAVO

CONTRERAS MELENDREZ, LEOPOLDO JAVIER

DOCENTE:

ROSAS CUEVAS, YESSICA

FECHA:

AGOSTO DE 2025

ILO - PERU

ÍNDICE

INTRODUCCIÓN	3
OBJETIVOS	3
DESCRIPCIÓN DEL SISTEMA	4
COMPARACIÓN CON CONTROL PID	5
MODO AUTOMÁTICO VS MANUAL	6
INTERFAZ WEB	6
FLUJO GENERAL DEL SISTEMA	7
SEGURIDAD DEL SISTEMA	7
CONCLUSIONES	8

INTRODUCCIÓN

El presente informe documenta el diseño, construcción y análisis técnico de un robot bombero móvil basado en Arduino, cuyo objetivo es detectar y extinguir focos de fuego. El sistema combina hardware de bajo costo, algoritmos de control y comunicación inalámbrica para ofrecer un prototipo funcional, robusto y controlable tanto de forma automática como remota. Este proyecto fue desarrollado en el contexto de la asignatura de Robótica II, integrando conocimientos de electrónica, programación, automatización y control.

OBJETIVOS

- Implementar un modelo de control PD que permita dirigir el servomotor hacia el foco de fuego de forma eficiente.
- Diseñar una interfaz web sencilla y efectiva para controlar y monitorear el carrito bombero.
- Visualizar en tiempo real los valores de los sensores de flama y de distancia.
- Garantizar un funcionamiento seguro limitando los ángulos del servomotor y protegiendo el hardware.

DESCRIPCIÓN DEL SISTEMA

El sistema se compone de:

- Arduino UNO: ejecuta la lógica de detección de fuego, movimiento del carrito, control del servo y la bomba.
- ESP8266 (Wemos D1 Mini): actúa como servidor web, recibe datos del Arduino y permite el control remoto.
- Sensores de flama (A0, A1, A2): detectan la dirección y presencia del fuego.
- Sensor ultrasónico (trigPin=5, echoPin=4): mide la distancia al objetivo.
- Servomotor (pin 2): dirige la manguera al foco de fuego.
- Relé (pin 12): activa la bomba de agua para extinguir el fuego.
- Driver L298N (pines 6-11): controla los motores para movimiento.

CONTROLADOR PD

El servomotor se encarga de orientar la manguera de agua hacia la dirección del fuego, basándose en los valores de los sensores de flama izquierdo y derecho. Para lograr un movimiento ágil pero estable, se implementó un controlador PD con la siguiente fórmula:

$$\text{error} = \text{val_izq} - \text{val_der}$$
$$\text{salida_pd} = (K_p * \text{error}) + (K_d * (\text{error} - \text{error_previo}))$$
$$\text{posicion_servo} = \text{salida_pd}$$

Donde el error se calcula como la diferencia entre las lecturas de los sensores izquierdo y derecho. Valores usados (ajustados empíricamente):

$K_p = 0.05$: determina la velocidad de respuesta al error presente.

$K_d = 0.04$: amortigua las oscilaciones del servo, reduciendo sobrepasos.

La posición del servo se actualiza constantemente y se limita entre 60° y 120° mediante la función `constrain()` para proteger el mecanismo físico y garantizar precisión.

COMPARACIÓN CON CONTROL PID

En versiones anteriores se implementó un controlador PID con K_i incluido:

$$\text{salida_pid} = (K_p * \text{error}) + (K_i * \text{error_acumulado}) + (K_d * (\text{error} - \text{error_previo}))$$

$K_p = 0.007$: sensibilidad al error.

$K_i = 0.001$: corrige errores acumulados a lo largo del tiempo.

$K_d = 0.04$: suaviza el movimiento.

Aunque el PID ofrecía teóricamente mayor precisión a largo plazo, en la práctica generaba oscilaciones debido al término integral. Por ello, se optó por el modelo PD, el cual resultó más eficiente y estable para la tarea puntual de apuntar al fuego de forma rápida.

MODO AUTOMÁTICO VS MANUAL

- Modo **automático**: el sistema detecta fuego y se aproxima, luego apunta el servo y activa la bomba automáticamente.
- Modo **manual**: el usuario controla el carrito desde la interfaz web mediante comandos: A (adelante), S (atrás), I (izquierda), D (derecha), T (detener), B (encender bomba), N (apagar bomba).

INTERFAZ WEB

- El ESP8266 funciona como un Access Point (CarritoBombero_WIFI) y hospeda una página web donde:
- Se muestran botones de dirección, bomba y parada total.
- Se actualizan en tiempo real los valores de los sensores, el estado del modo, del servo y de la bomba.
- Se envían comandos HTTP al ESP8266, que luego los reenvía al Arduino vía serial.
- Se usa JavaScript para actualizar el estado con peticiones cada segundo sin recargar la página.
- Esto permite una operación completamente inalámbrica, cómoda y visualmente comprensible para el usuario.

FLUJO GENERAL DEL SISTEMA

- El sistema se inicia en modo manual.
- Si los sensores detectan fuego (valor por debajo del umbral), cambia automáticamente al modo automático.
- El robot se aproxima al fuego mientras esté presente.
- Si la distancia es menor a 20 cm o el fuego es muy intenso, se detiene.
- Se ejecuta la rutina PD para apuntar el servo y se activa la bomba durante 5 segundos.
- Al finalizar, se apaga la bomba, se centra el servo, y si no hay más fuego, se regresa al modo manual.

SEGURIDAD DEL SISTEMA

- Protección del servomotor con límites angulares de 60° a 120°.
- Timeout de 30 ms en el sensor ultrasónico para evitar bloqueos.
- Diferenciación clara entre modos para evitar conflictos de lógica.
- Control suave del servo para evitar desgaste mecánico.

CONCLUSIONES

El uso del controlador PD permitió una solución más estable y eficiente que el PID en este caso.

La combinación Arduino–ESP8266 habilitó control autónomo y monitoreo remoto en tiempo real. El sistema integra correctamente hardware, software y teoría de control para resolver un problema real. Se logró un robot funcional, seguro, y capaz de responder de forma efectiva al fuego.