

# Taller de Programación

## Sesión II

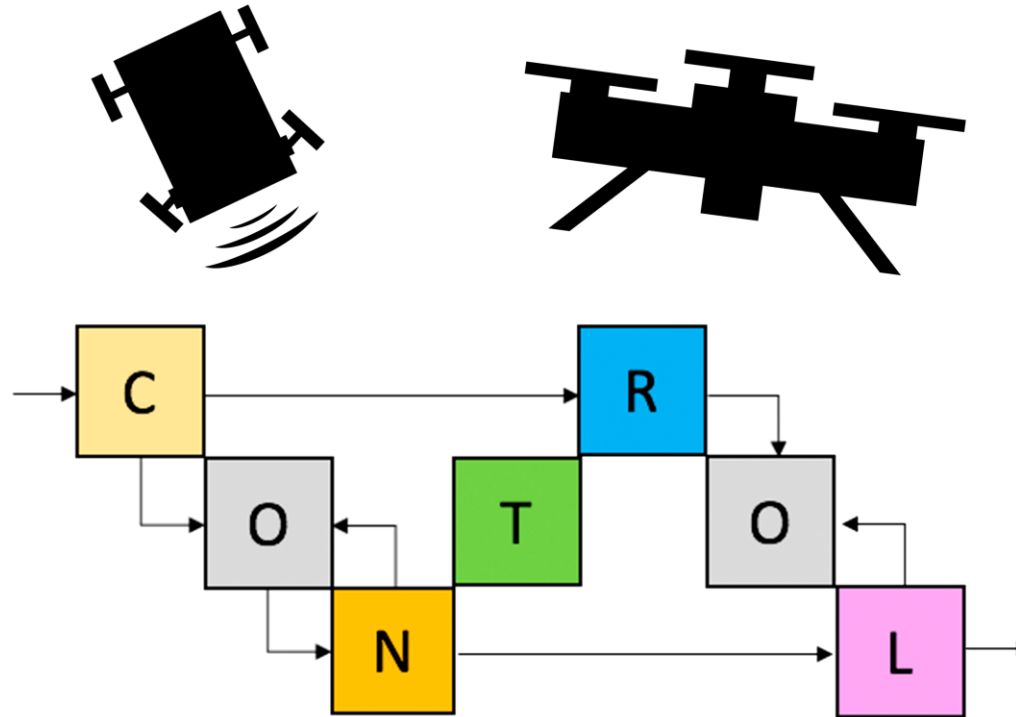
SASIM

Mes de la robótica

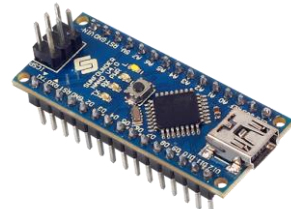
# ¿Cómo logramos que se siga la línea en tiempo real?

- Distintos tipos de controladores...

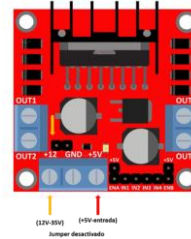
- PID.
- On-Off.
- Condicionales.
- Bang Bang.



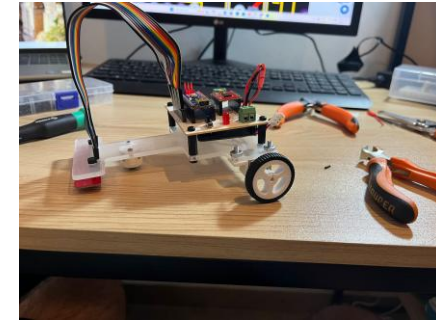
# Entendiendo el sistema



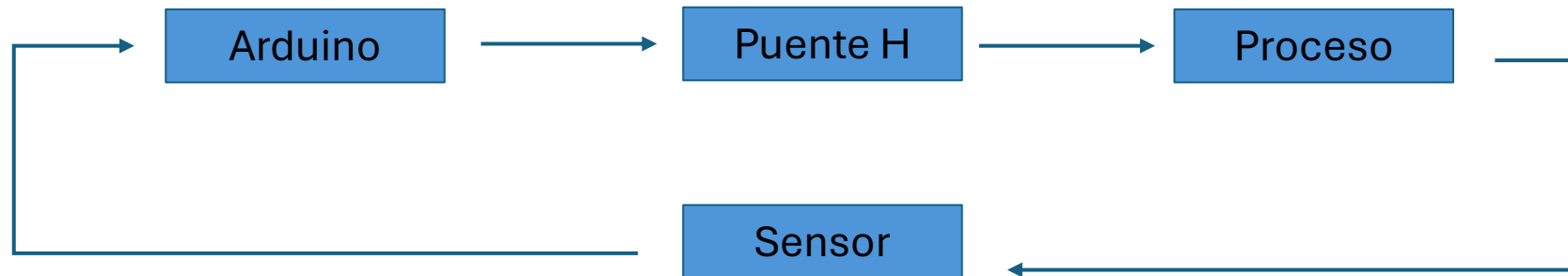
Controlador



Actuador

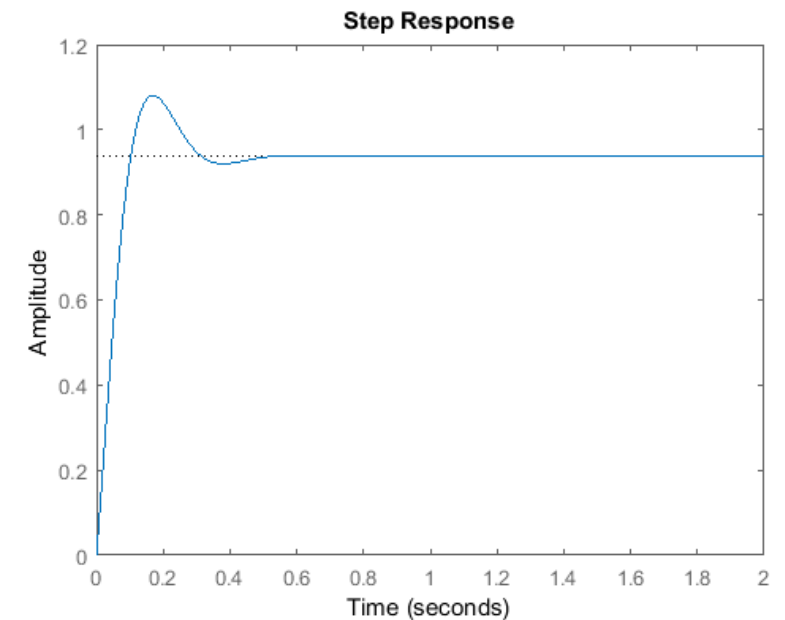


Proceso,  
Variable a controlar



# ¿Qué es un controlador PID y por qué lo utilizamos?

- El control PID consiste en operar un error con tres operaciones matemáticas y sumar sus efectos.
  - Proporcional: multiplicar el error por una constante  $K_p$ , nos aumenta la velocidad del sistema.
  - Integral: toma errores pasados y genera una señal de control,  $K_i$ .
  - Derivada: amortigua la señal y los impulsos, es decir reduce las oscilaciones.



# Aplicando la programación

- Antes del programa, necesitamos añadir variables que nos permitirán cambiar el sistema.

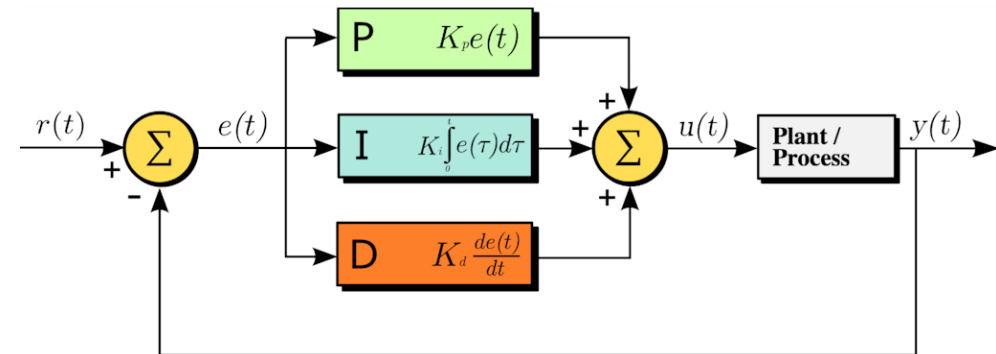
```
float Kp = 0.05;  
float Ki = 0.0;  
float Kd = 0.1;  
int setPoint = 3500;  
int posicion = 0;  
int error = 0;  
int errorAnterior = 0;  
float integral = 0;  
float derivada = 0;  
float salidaPID = 0;
```

**Nota importante:** uno tiene que modificar el setpoint dependiendo de la calibración, y modificar las constantes a prueba y error.

# Modificando el programa

- Iremos calculando y modificando la velocidad de nuestro seguidor en función de lo que de mejores resultados.
- Cada 10 ms se hará un nuevo registro.

```
void loop() {  
  // 1. Calcular posición y error  
  posicion = calcularPosicionLinea();  
  error = setPoint - posicion;  
  
  // 2. Cálculos PID  
  integral += error;  
  derivada = error - errorAnterior;  
  errorAnterior = error;  
  
  salidaPID = Kp * error + Ki * integral + Kd * derivada;  
  
  // 3. Calcular velocidades de los motores  
  int velocidadBase = 150;  
  int velocidadIzq = velocidadBase + salidaPID;  
  int velocidadDer = velocidadBase - salidaPID;  
}
```



# Modificando el programa

- Tras haber calculado las velocidades tenemos que cambiar la velocidad del sistema.

// 5. Aplicar velocidades a los motores

```
analogWrite(ENlzq, velocidadlzq);  
analogWrite(ENDER, velocidadDer);
```

La mejor manera de generar el mejor resultado es probando.

// 6. Dirección hacia adelante

```
digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);  
digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
```

// 7. Imrpimir valores

```
Serial.print("Posición: "); Serial.print(posicion);  
Serial.print(" | Error: "); Serial.print(error);  
Serial.print(" | PID: "); Serial.print(salidaPID);  
Serial.print(" | Vell: "); Serial.print(velocidadlzq);  
Serial.print(" | VelD: "); Serial.println(velocidadDer);
```

```
delay(10);} 
```

Tiempo en el que se repetirá

Gracias por su atención