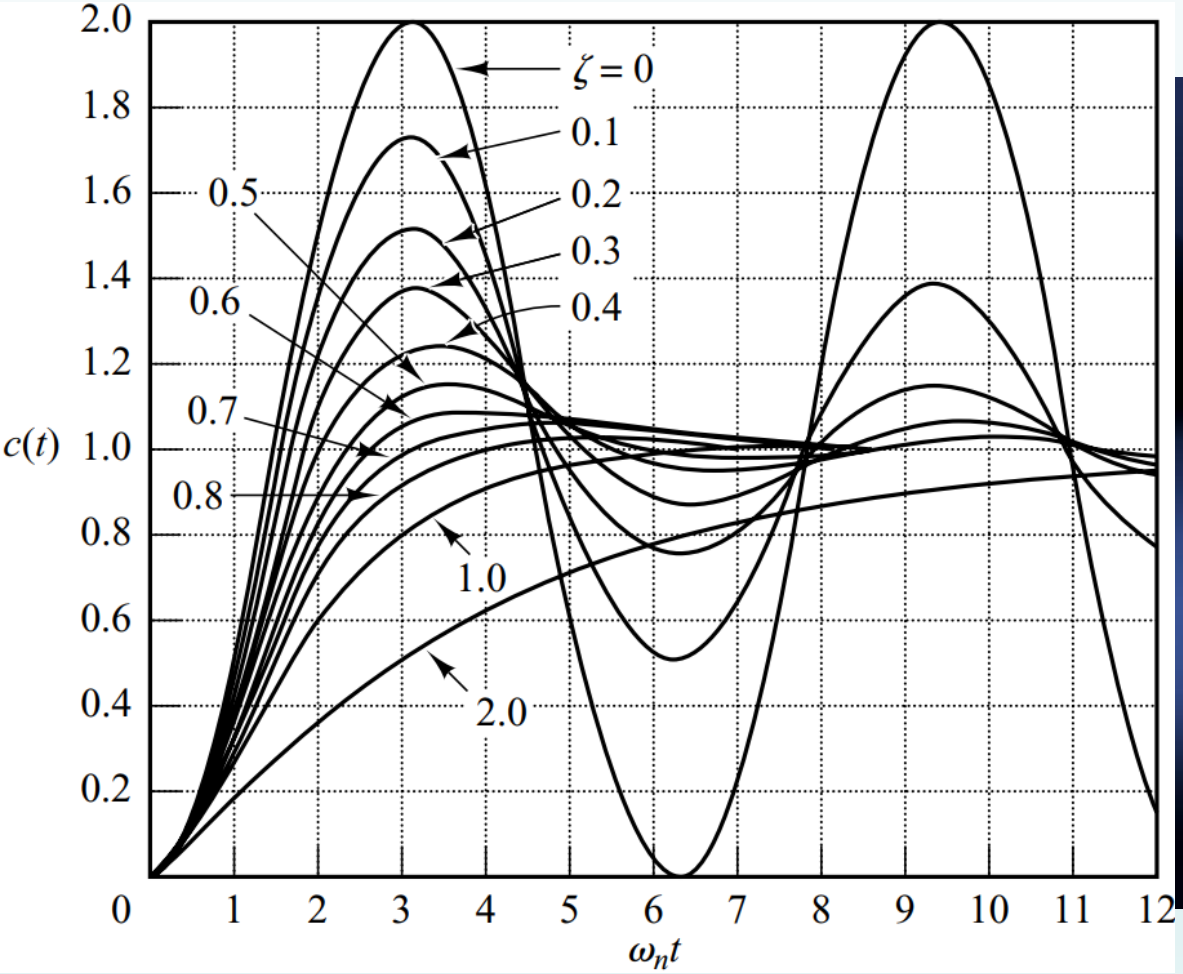


PMR 3100 – Introdução à Engenharia Mecatrônica

Módulo 04 – Meu Primeiro Robô

Aula 06 – Controle

Prof. Dr. Rafael Traldi Moura

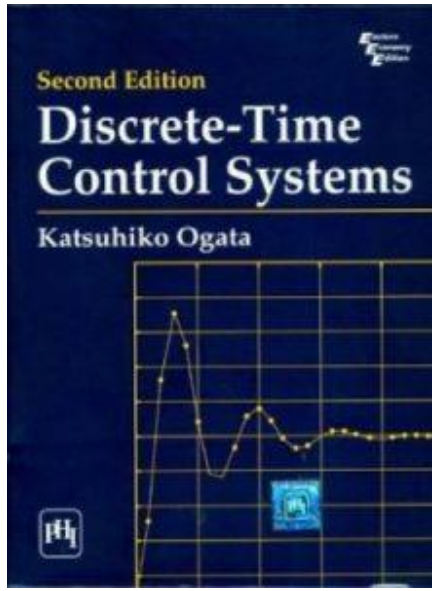
Controle

Vocês terão um caminho longo, estudando controle em algumas disciplinas:

- PMR3302 - Sistemas Dinâmicos I para Mecatrônica;
- PMR3306 - Sistemas Dinâmicos II para Mecatrônica;
- PMR3305 - Sistemas a Eventos Discretos;
- PMR3404 - Controle I;
- PMR3409 - Controle II;

Controle – bibliografia básica e **universal**

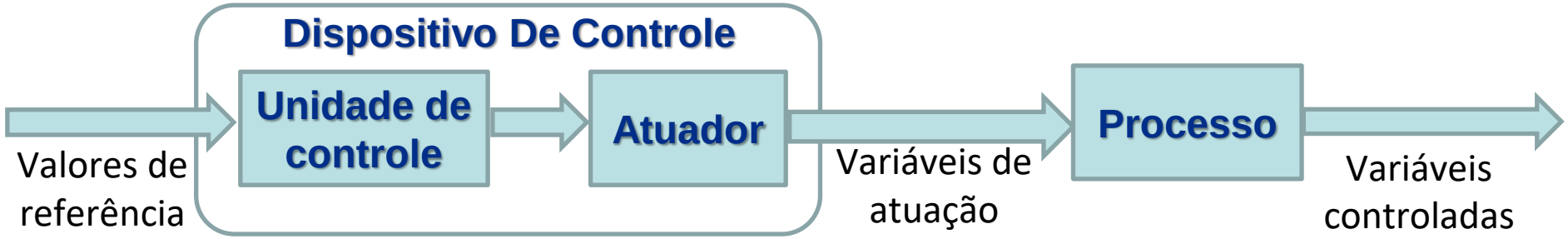
- Engenharia de Controle Moderno, Katsuhiko Ogata, Pearson Universidades; 5ª edição, 2010;



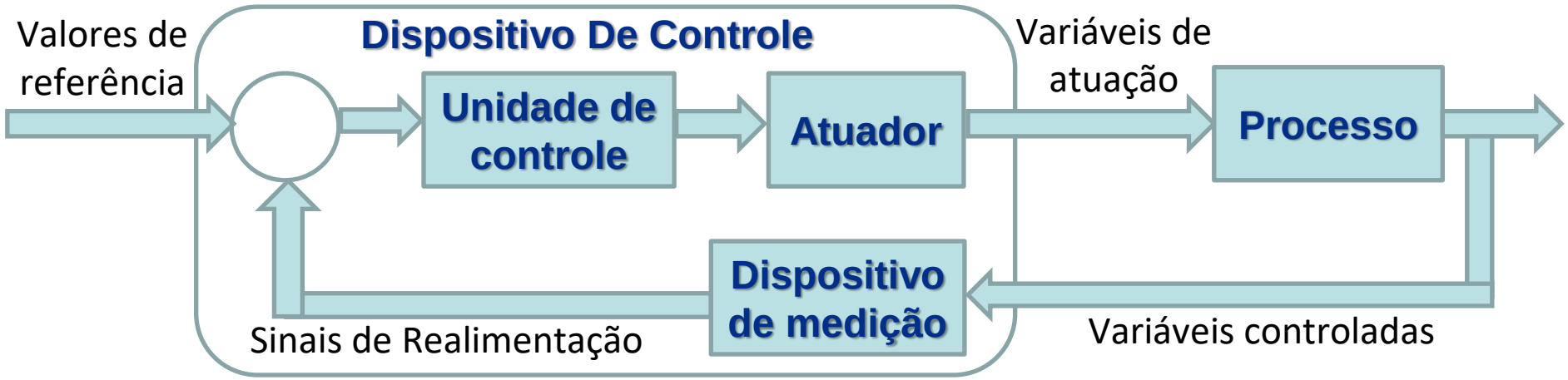
- Discrete-Time Control Systems, Katsuhiko Ogata, Pearson Universidades; 2ª edição, 1994;

Controle – malha aberta e fechada

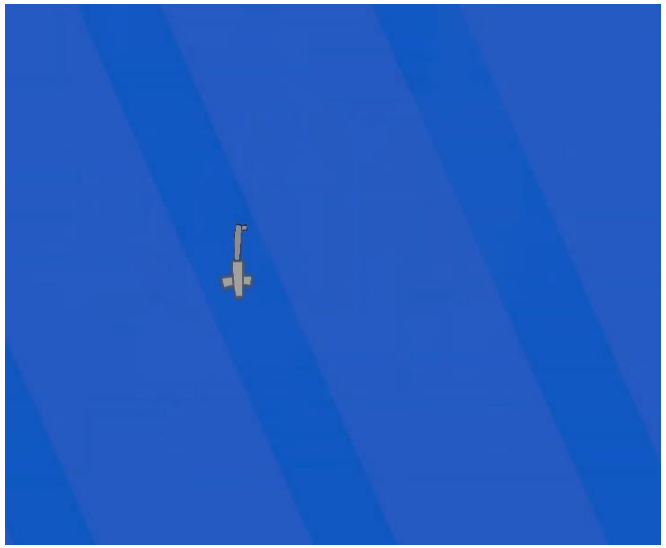
- Sistema mecatrônico com controle de malha aberta:

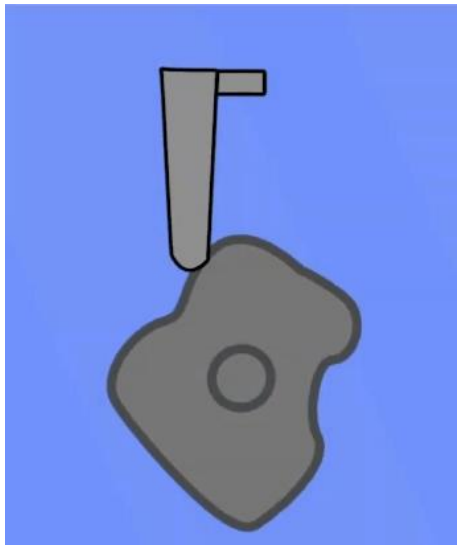


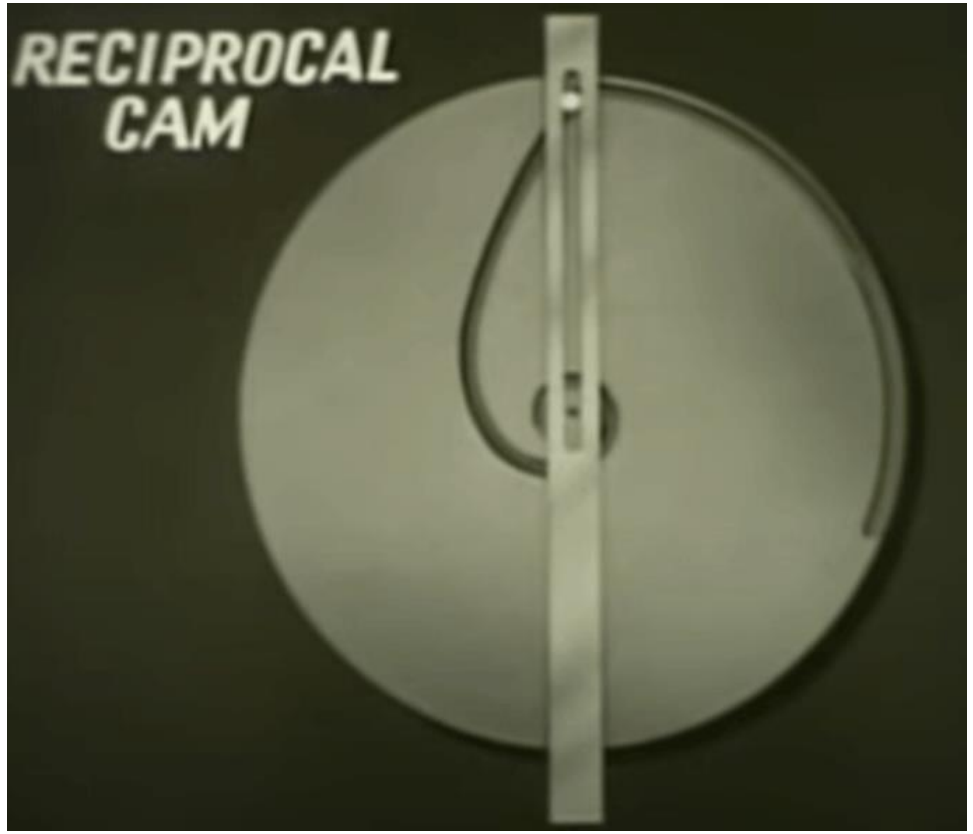
- Sistema mecatrônico com controle de malha fechada:

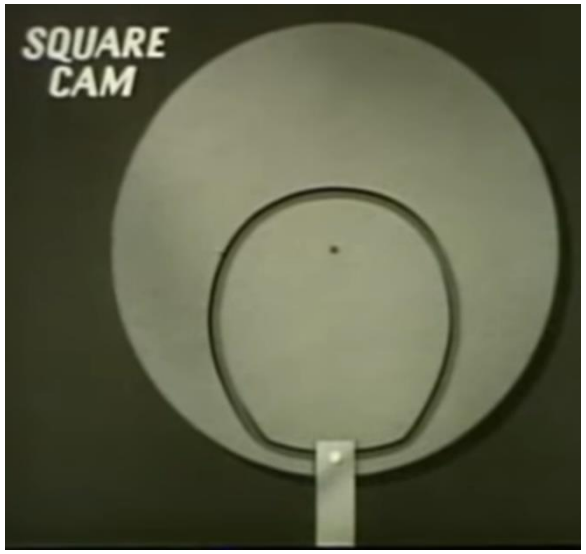


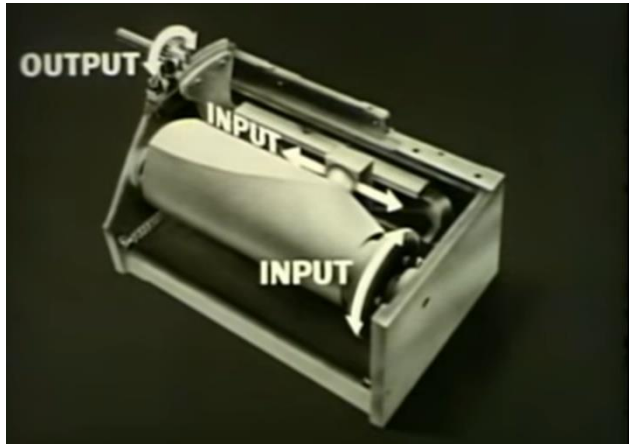
Malha aberta – Came Seguidor











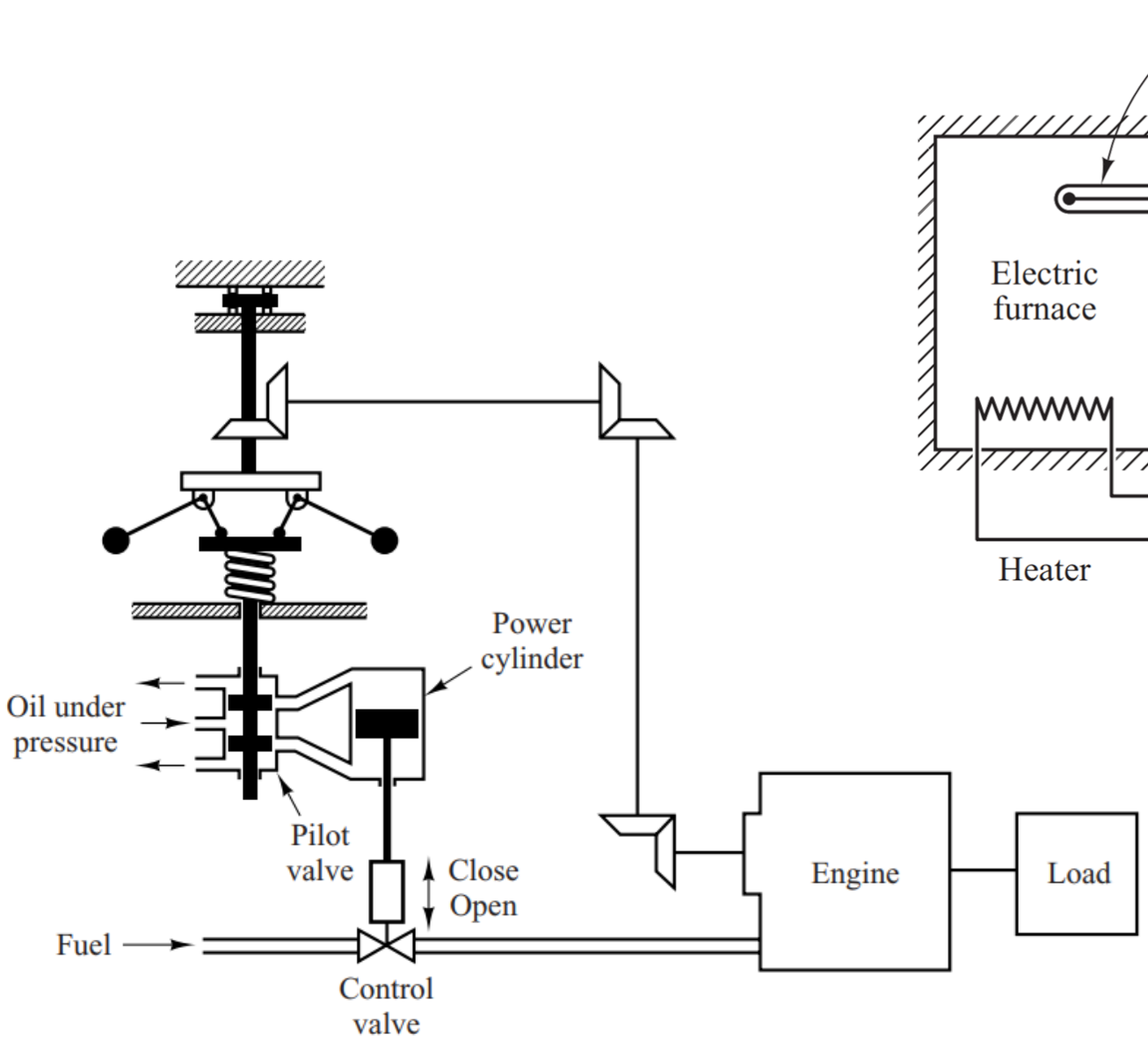
- VIDEO: COMPUTADOR MECÂNICO

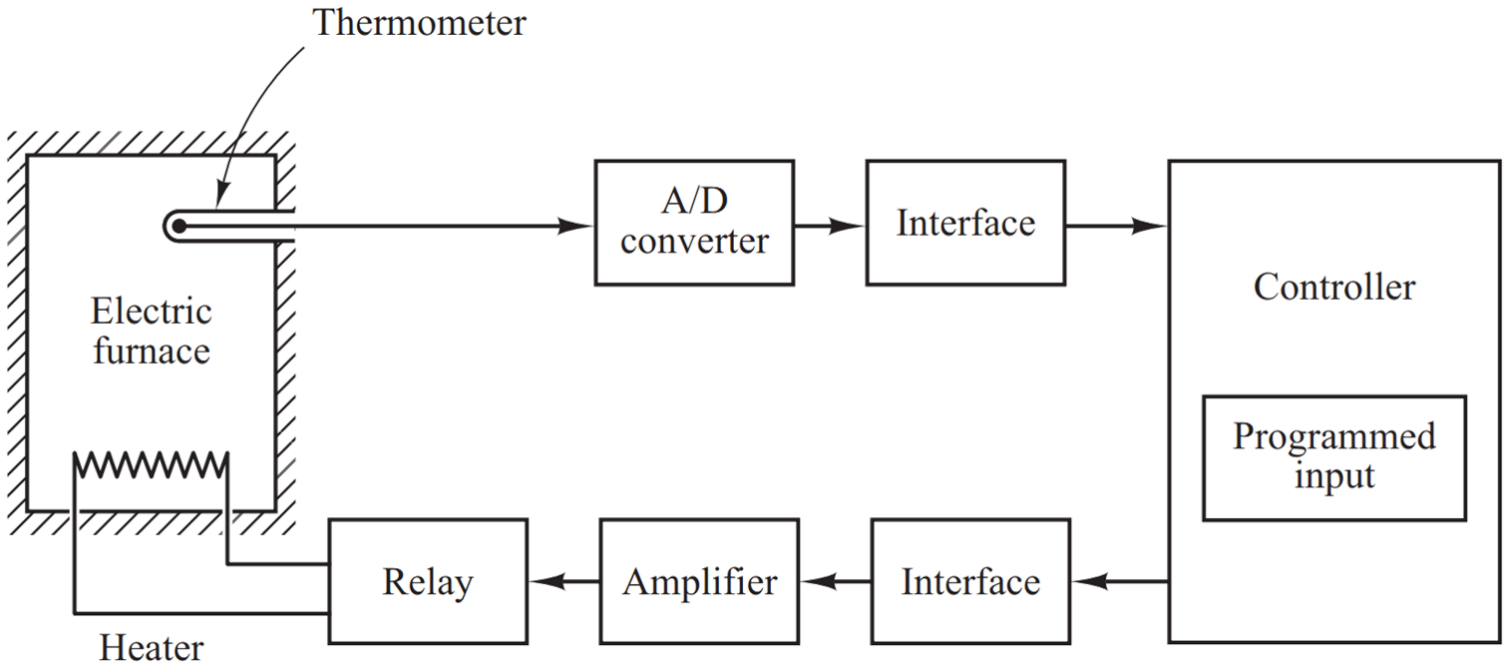
Malha aberta – Exemplos

Leonardo da Vinci

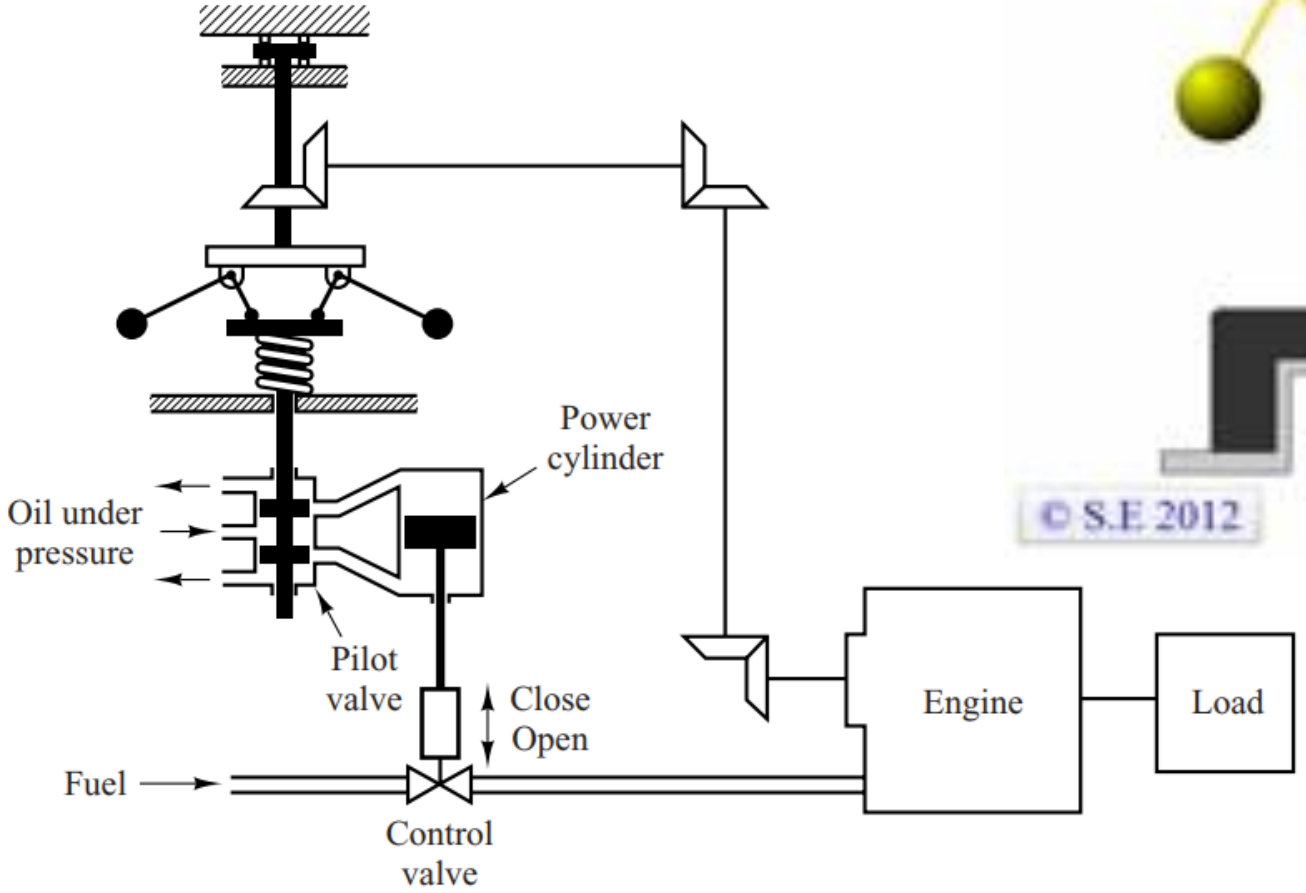
Automato little writer (Jaquet Droz, Sec. 18)

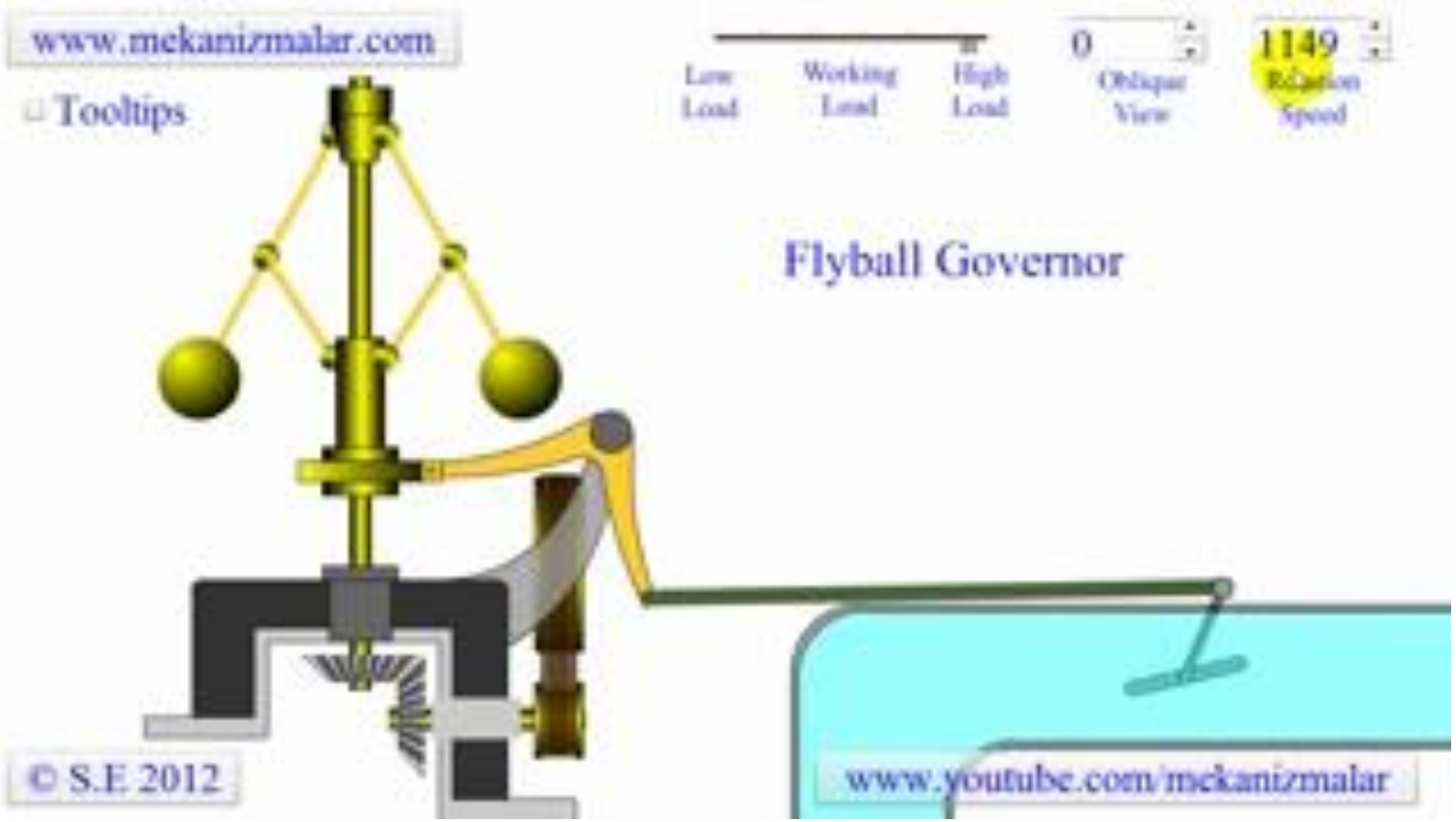
Malha fechada – Regulador Watt de velocidade



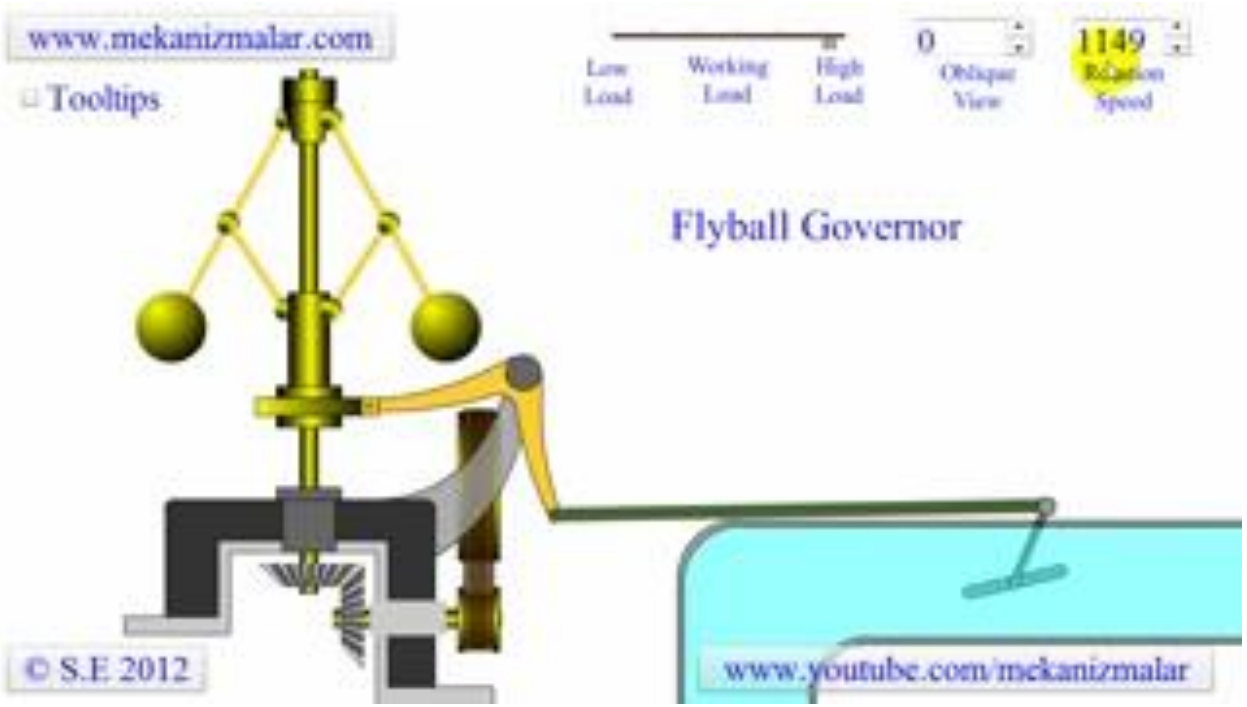


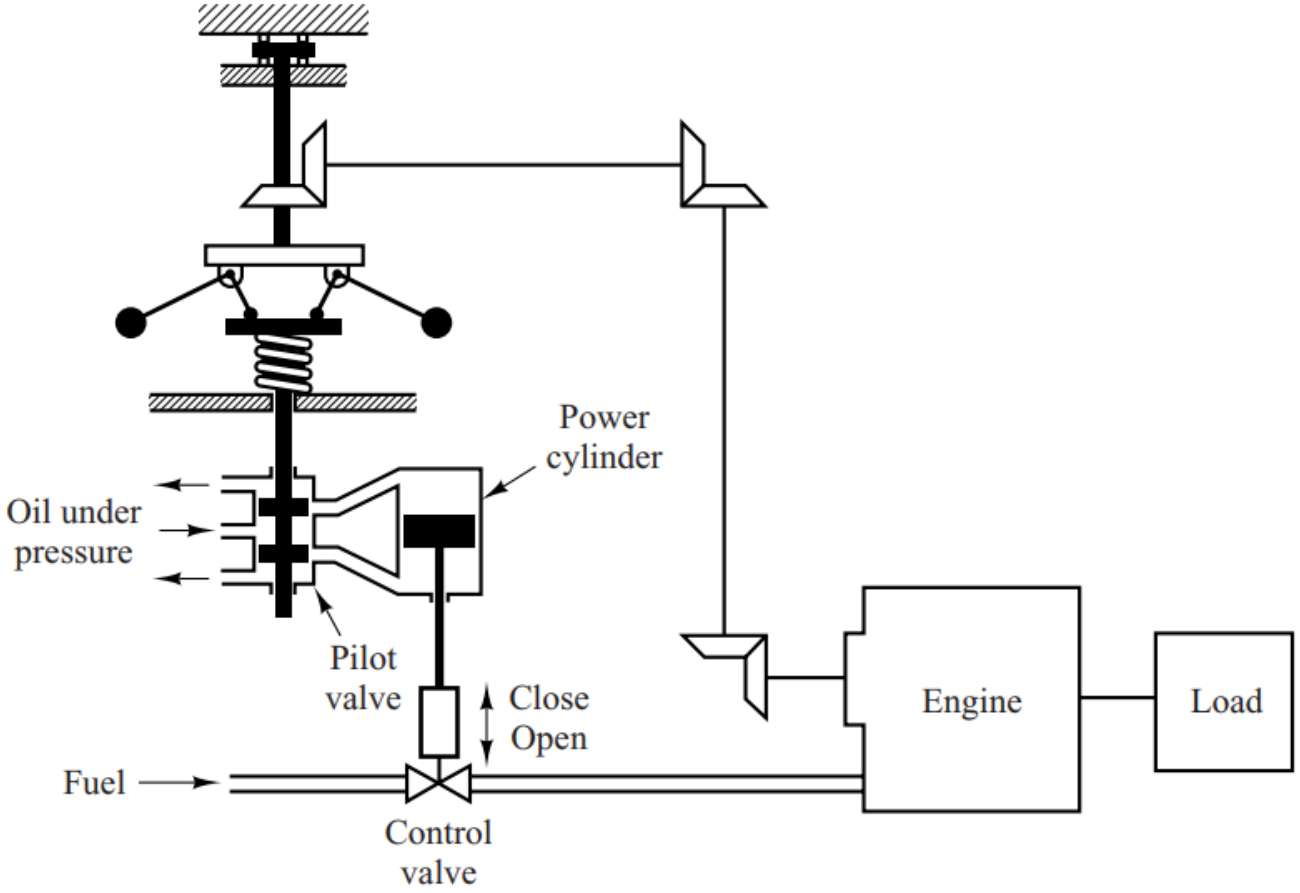
Malha fechada – Regulador Watt de velocidade



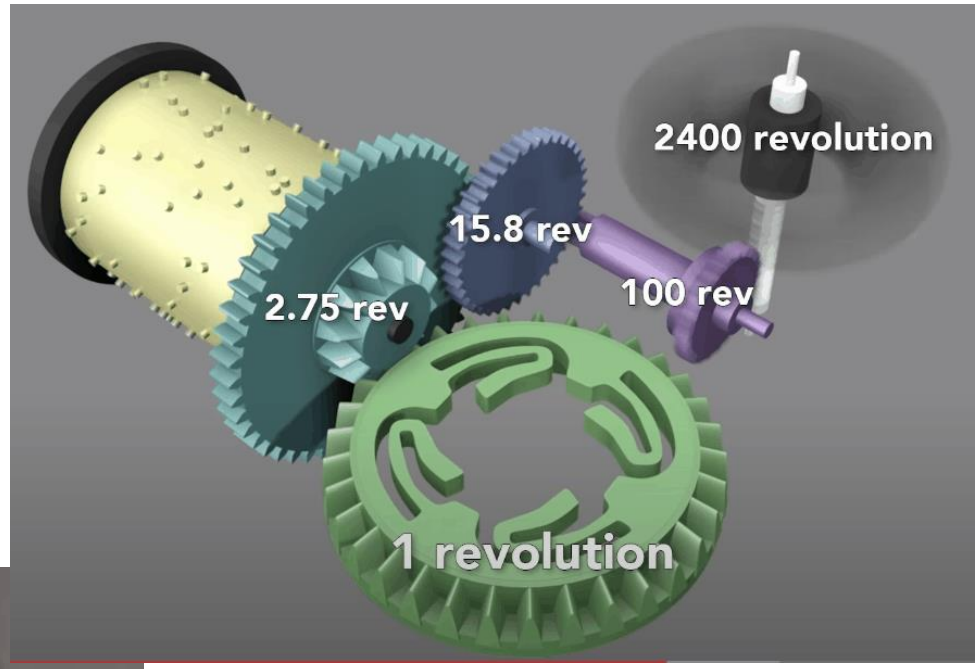


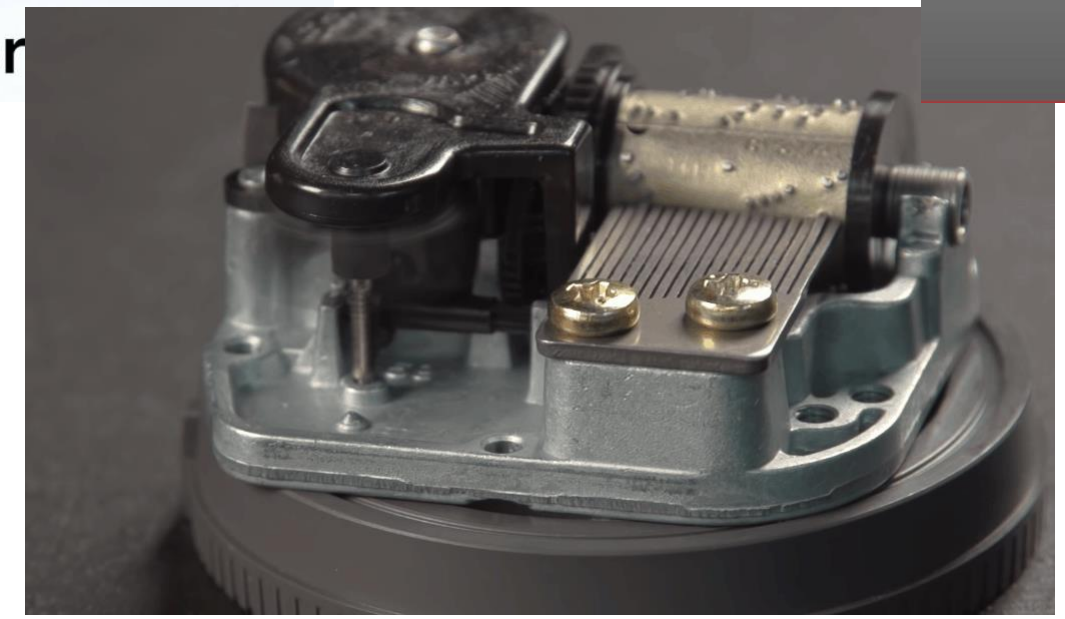
Malha fechada – Regulador Watt de velocidade





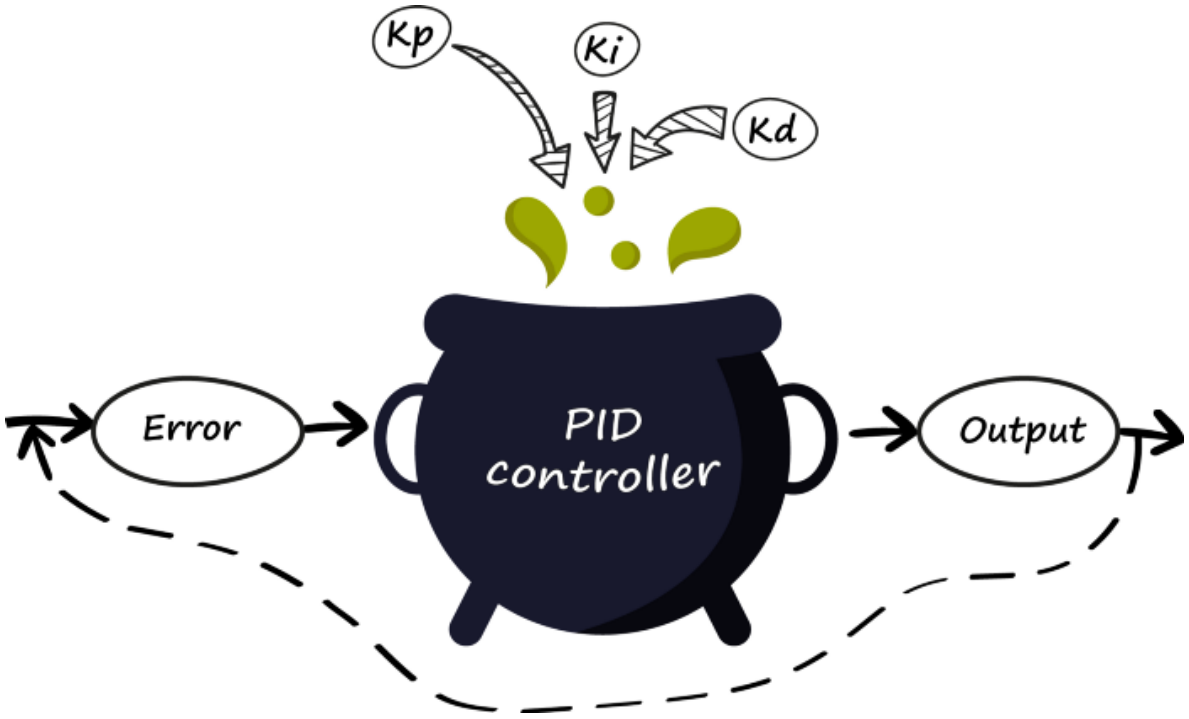
Malha fechada – Regulador de velocidade

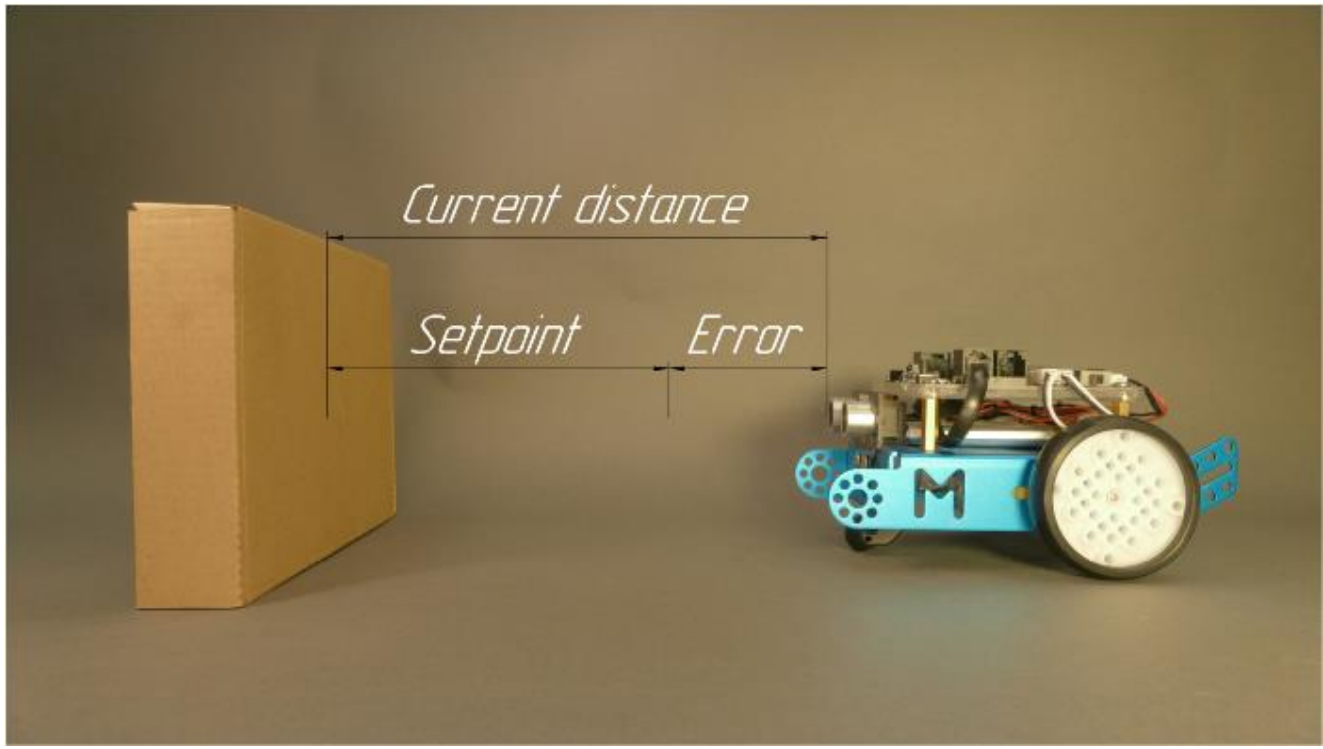




VIDEO:
Regulador de Velocidade

Controle Proporcional Integrativo Derivativo





$$erro = x - setpoint$$

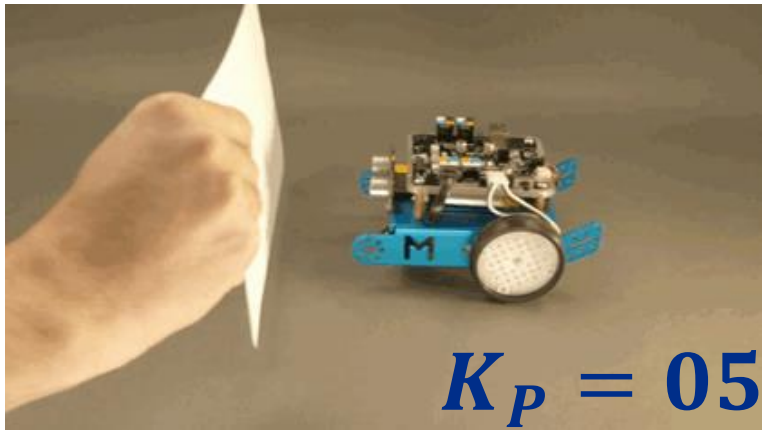
$$erro_i = \sum (x - setpoint)$$

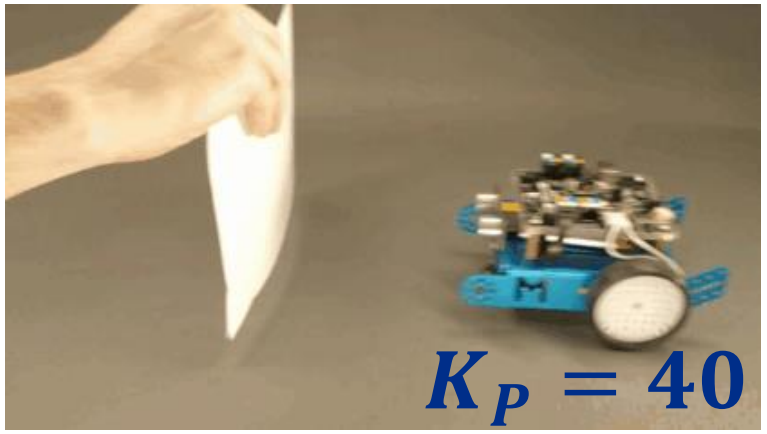
$$erro_d = \frac{\Delta erro}{\Delta t}$$

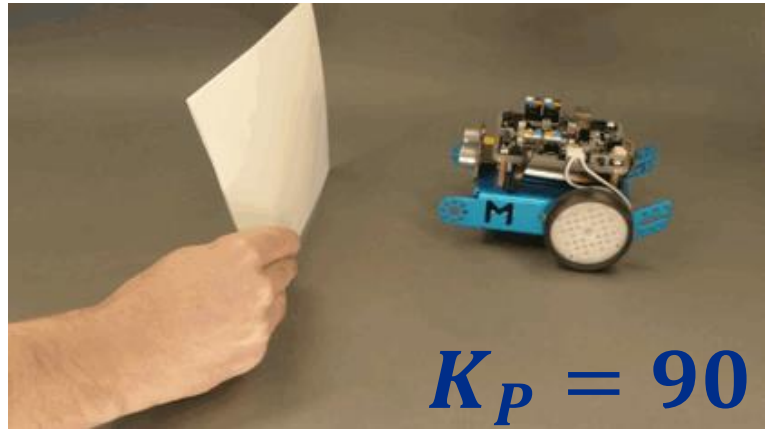
Fonte: XODlang

Controle Proporcional

$$ação = K_P \cdot erro$$

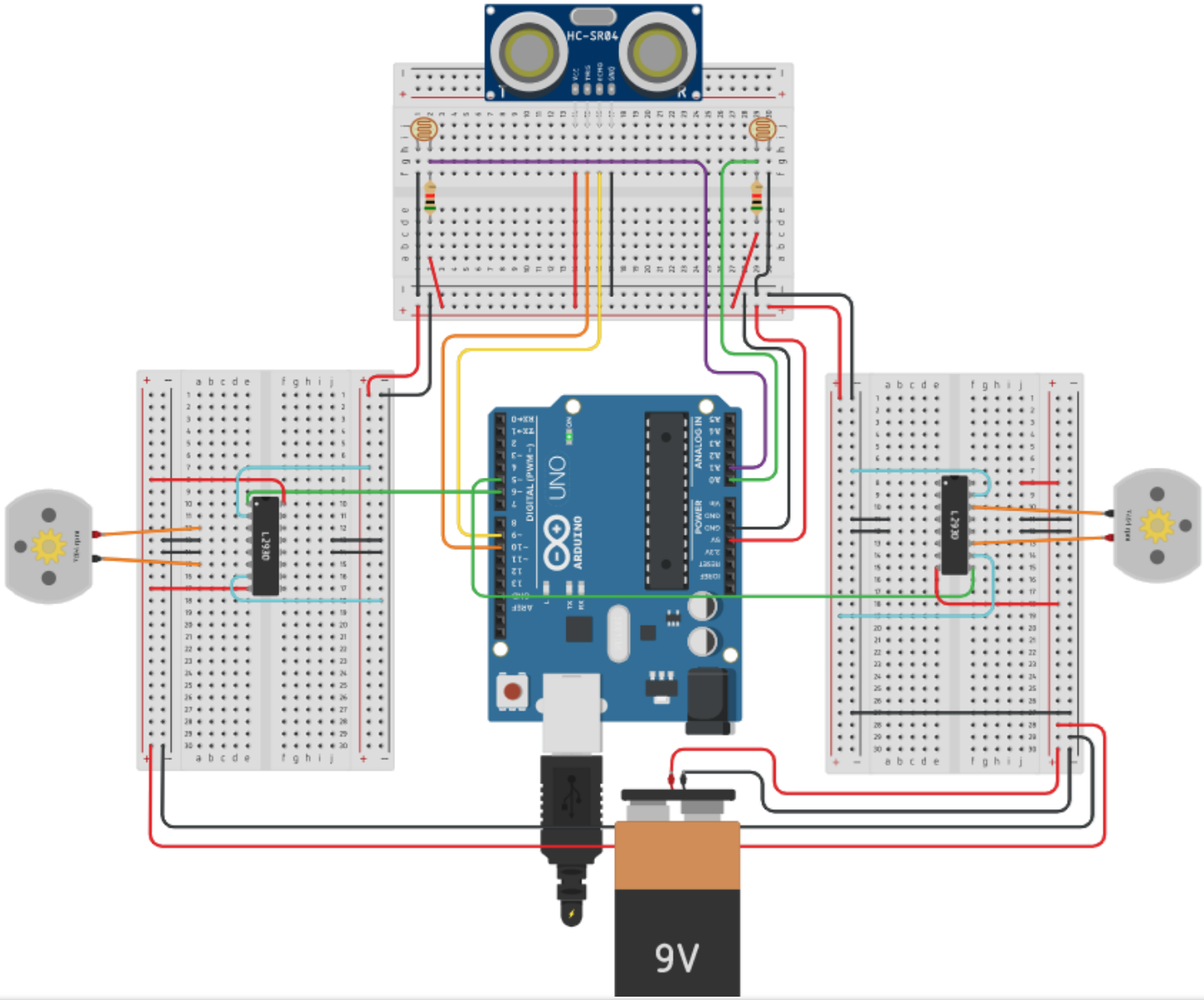






Fonte: XODlang

Carrinho 1 – eletrônica



Carrinho 1 – código parte 1

- A primeira parte do código contém:
  - define
  - declarações de variáveis globais

```
#define EN_dir 5
#define EN_esq 6
#define trigPin 10
#define echoPin 9
#define sensorDireita A0
#define sensorEsquerda A1
#define V0 120
#define V_DEVAGAR 50
#define offset 94
#define V_max 240
#define V_min 0

float velE,velD;
int leituraEsquerda, leituraDireita;
float distancia;
float erro,integral,der,erro_anterior,PID;
float Kp,Ki,Kd;
```

Carrinho 1 – código parte 2

- A segunda parte do código contém a função `le_sensor_ultrassonico` que implementa a sequência necessária para emitir a onda, que bate no objeto e volta, medir o tempo necessário para a ida e volta da onda, além de transformar em distância a partir da velocidade do som no ar.

```
float le_sensor_ultrassonico() {
  long duracao;
//Limpa o pino de Trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
//Coloca o pino de Trigger em HIGH por
//10 microseg
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
//Le o pino de echo, retornando o tempo
//em microseg da onda sonora ir e vir
  duracao = pulseIn(echoPin, HIGH);
//Calcula a distancia
  return ((float)duracao)*0.0174;
}
```

Carrinho 1 – código parte 3

- A terceira parte do código contém: a função `setup()` com:
  1. inicialização da comunicação serial em 9600kbps;
  2. definição dos pinos usados como entrada ou saída;
  3. Atribuição de valor inicial para cada variável.

```
void setup() {
  Serial.begin(9600);
  Serial.println("Esperando 5 segundos");
  pinMode (sensorEsquerda, INPUT);
  pinMode (sensorDireita, INPUT);
  pinMode (EN_dir, OUTPUT);
  pinMode (EN_esq, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  velE = V0;
  velD = V0;
  erro = 0;
  integral = 0;
  der = 0;
  erro_anterior = 0;
  Kp = 0.1;
  Kd = 0.0;
  Ki = 0.0;
  Serial.println("Esperando 1 segundos");
  delay(1000);
  Serial.println("Setup completo");
}
```

Carrinho 1 – código parte 4

- Na quarta parte do código começa a função loop.
- Primeiro é implementada a leitura dos sensores de linha, com atenção para o offset devido ao valor lido quando o mesmo está totalmente no branco (devido ao divisor resistivo);
- Depois é implementada a leitura dos sensores ultrassônico de distância;

```
void loop() {
// LÊ OS SENSORES
  //Lê o senor de distancia ultrassonico
  leituraDireita =
analogRead(sensorDireita)-offset;
  leituraEsquerda =
analogRead(sensorEsquerda)-offset;
  Serial.print("S_Esq= ");
  Serial.print(leituraEsquerda);
  Serial.print(" S_Dir= ");
  Serial.print(leituraDireita);
  //Lê o senor de distancia ultrassonico
  distancia = le_sensor_ultrassonico();
  Serial.print(" Dist=");
  Serial.print(distancia);
  Serial.print("cm ");
// fim de LÊ OS SENSORES -------
```

Carrinho 1 – código parte 5

- Na quinta parte do código continua a função loop.
- Nesta parte é implementado o controle;

```
// CALCULA O CONTROLE  ------------------------
// NESTE controle, vamos assumir
// que virar para a esquerda é positivo
//calcula os termos do PID
  erro = leituraDireita - leituraEsquerda;
  integral = integral + erro;
  der = erro - erro_anterior;
  erro_anterior = erro;
//calcula o valor do PID e a velocidade dos motores
  PID = Kp*erro + Ki*integral + Kd*der;
  velE =(int)(V0 + PID);
  velD =(int)(V0 - PID);
// se estiver muito perto da parede, desacelera ate a distancia limite
  if ((distancia > 10) && (distancia < 20)) {
    velE = V_DEVAGAR;
    velD = V_DEVAGAR;
  }if (distancia < 10) {
    velE = 0;
    velD = 0;
  }
// fim de CALCULA O CONTROLE  ------------------------
```

Carrinho 1 – código parte 6

- Na sexta parte do código termina a função loop.
- Nesta parte temos a verificação se os atuadores estão saturados;
- E atualizamos os valores dos atuadores.

```
// ATUALIZA VALOR DOS ATUADORES  ------------
  //verifica saturacao dos motores
  if (velE>V_max){
    velE = V_max;
  }if (velE<V_min){
    velE = V_min;
  }if (velD>V_max){
    velD = V_max;
  }if (velD<V_min){
    velD = V_min;
  }
  //atualiza a velocidade dos motores
  analogWrite(EN_esq,int(velE)); //MOTOR E PWM
  analogWrite(EN_dir,int(velD)); //MOTOR D PWM
  Serial.print("  M_Esq= ");
  Serial.print(velE);
  Serial.print(" M_Dir= ");
  Serial.println(velD);
// fim de ATUALIZA VALOR DOS ATUADORES  ------
  delay(50);
} //fim loop()
```

Exercício para entrega

Implemente o controle no seu carrinho 1 e:

1. Explique a função dos defines `V0`, `V_DEVAGAR`, `offset`, `V_max` e `V_min`.
2. Explique a função da variável `Kp`. Como ela influência o sistema? Coloque valores diferentes na mesma e as mudanças de `M_Esq` e `M_Dir` em diferentes posições dos sensores de linha para justificar sua resposta.
3. O que acontece se forem colocados valores `0.001`, `0.01` e `0.1` em `Kd`?
4. O que acontece se forem colocados valores `0.001`, `0.01` e `0.1` em `Ki`?