

Experiência 1

Erro em Regime Permanente e Tipo do Sistema

1 – Introdução

Sistemas de controle em malha fechada podem ser classificados em função do **tipo do sistema**, considerando sinais típicos de referência e de perturbação. O tipo do sistema determina a capacidade de acompanhar sinais – degraus, rampas, parábolas etc., e o correspondente **erro em regime permanente**, e_{ss} . Sistemas do tipo 1, como o kit impressora (sistema carro sobre trilho) utilizado neste experimento, seguem referências em degrau com erro nulo, enquanto, para uma referência rampa, apresentam erro constante. A rejeição de perturbações depende de integradores no caminho direto, antes do ponto de incidência da perturbação.

2 - Objetivos

Calcular os erros em regime permanente, para referência degrau e rampa e perturbação degrau, em função dos parâmetros identificados do kit impressora, comparando-os de forma crítica com os valores medidos em laboratório.

2 – Kit impressora (Carro sobre trilho)

O kit impressora, mostrado na Figura 1, é composto por um motor CC acoplado a um carro (dos cartuchos de tinta) através de correia e polias, juntamente com um *encoder*, uma placa analógica, um arduino nano, uma ponte H e um protoboard. Visa a experimentação didática de baixo custo, boa robustez e alta repetibilidade de sistemas de controle analógico de posição linear.

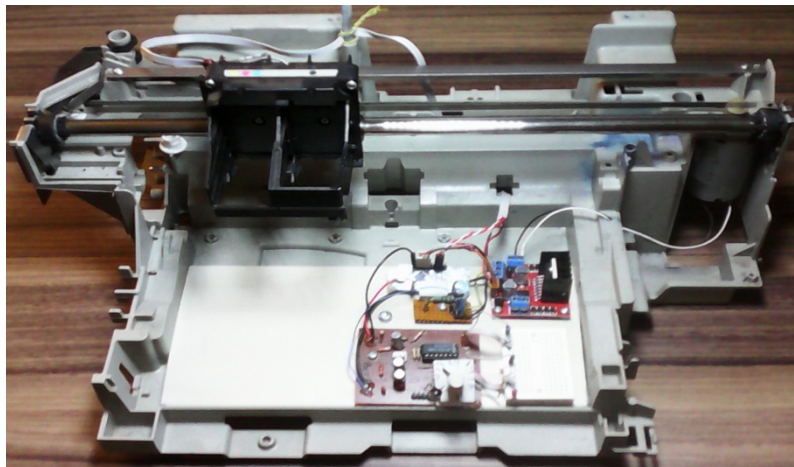


Figura 1 – Impressora adaptada para experimentos de controle, (Prof. Egito –ENE/UnB, 2017).

2.1 – Controle de posição linear em Malha Fechada

O diagrama de blocos da Figura 2 apresenta todos os subsistemas relevantes. O kit deve ser utilizado sempre em *malha fechada*, uma vez que em malha aberta o carro pode atingir os batentes danificando a correia. A medição da posição do carro é feita através de uma fita transparente com marcações e detector óptico (*encoder*). O detector óptico, preso ao carro, conta as marcações da fita gerando pulsos em 2 canais, o que permite detectar movimento do carro para a direita ou para a esquerda. Os pulsos são “lidos” pelo arduino em um contador que reflete a posição do carro. O conteúdo do contador de posição é então convertido em um sinal analógico via PWM₂ do arduino e um filtro passa-baixas. Note que a posição do carro, y , é dada em metros (ou cm) enquanto a referência (gerador de funções), r , e o erro são sinais de tensão em Volts.

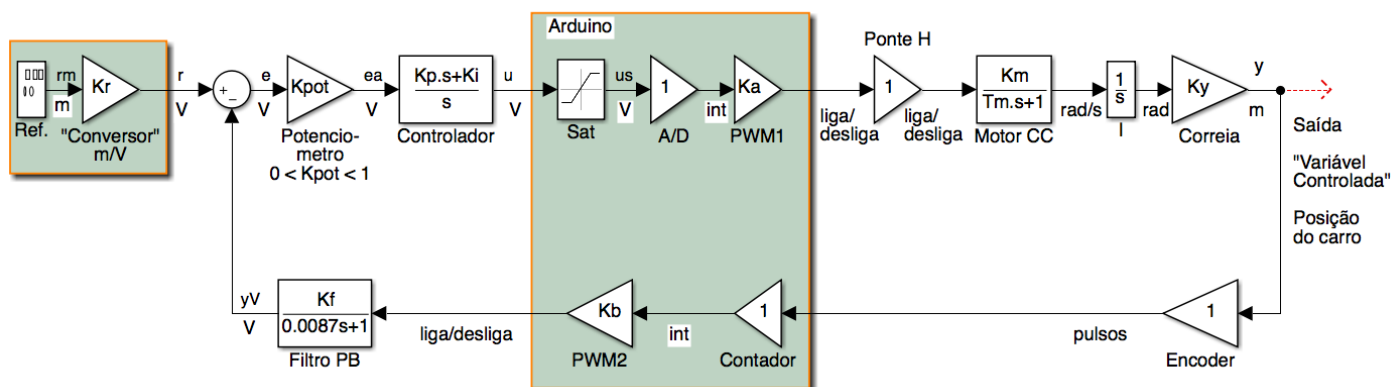


Figura 2 – Diagrama de blocos geral do kit impressora.

Blocos internos à linha pontilhada (arduino) são digitais e externos são basicamente analógicos (exceto *encoder* e ponte H). Nem todas as ligações elétricas estão representadas, apenas as que tem importância do ponto de vista da modelagem dinâmica. Observe que alguns blocos tem ganho “1”, o que corresponde a uma simplificação. Os blocos PWM, por exemplo, convertem uma informação digital em analógica e tem um ganho associado (e.g., $00000001_2 \rightarrow 5 \text{ V}/256$; conversão de 8 bits). O bloco ADC teria um ganho inverso, por este motivo foram incluídos os blocos K_a e K_b para corrigir possíveis distorções e acomodar os sinais na melhor faixa possível para a parte eletrônica.

2.2 – Saturações

A saturação compromete a modelagem linear utilizada neste experimento e deve ser evitada. A parte eletrônica do kit e software foram projetados para minimizar problemas com saturações, que podem ocorrer em:

- Entrada do conversor A/D: Procurar manter o sinal ADC da protoboard entre $-3 \text{ V} < \text{ADC} < +3 \text{ V}$.
- *Encoder* e PWM₂: Não deixar o carro se aproxime muito dos batentes.
- PWM₁, ponte H e fonte de alimentação do motor: Só ocorre durante breves momentos desde que o motor seja alimentado com o fio de 20V durante os experimentos (“turbo”).

2.3 – Simulação Eletrônica do kit impressora

A simulação do kit impressora, com destaque para as partes eletrônicas relevantes ao experimento, é mostrada na figura 3. O motor é simulado como função de transferência ideal acrescido da saturação. A zona morta do motor é compensada nominalmente pelo arduino e assim não aparece no diagrama. Observe que o potenciômetro que atenua o erro é de 10K e assim deve-se minimizar o carregamento pelo circuito subsequente.

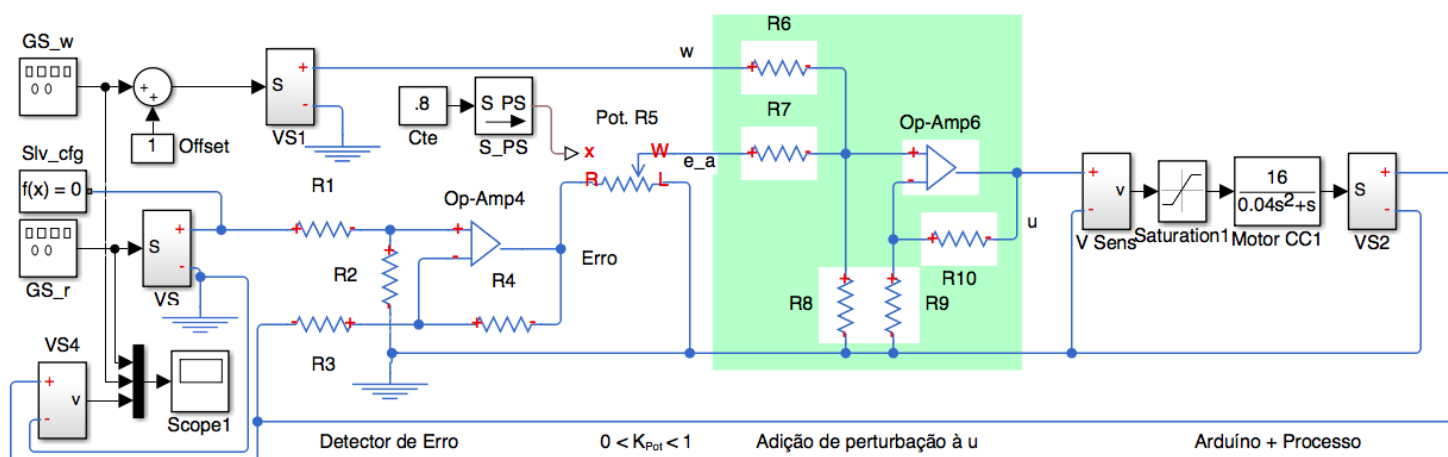


Figura 3 – Esquemático eletrônico-motor kit impressora (adaptado da documentação Prof. Egito –ENE/UnB).

2.4 – Simulação do Sistema - Diagrama de Blocos Simplificado

A introdução do Arduino na malha permite não apenas a leitura do *encoder* digital, mas também a correção da zona morta do motor, pois uma tensão abaixo de certo limiar não movimenta o carro. No diagrama da Fig. 2 a correção é feita pelo bloco de K_a . Desta maneira o software do Arduino sempre soma o valor correspondente à zona morta à voltagem que deve ser aplicada ao motor (positiva para um lado negativa para o outro). Considerando que a taxa de atualização dos blocos do Arduino é suficientemente alta, o kit se aproxima de um sistema de controle de posição linear com o diagrama de blocos da Fig. 3. Onde os demais ganhos de malha fixos no kit foram incorporadas às constantes K_m e K_v . Observe que nos dois diagramas anteriores consideramos que o Arduino e o filtro passa baixas na realimentação são muito rápidos que o processo podendo ser modelados sem dinâmica. Isto será melhor avaliado mais adiante.

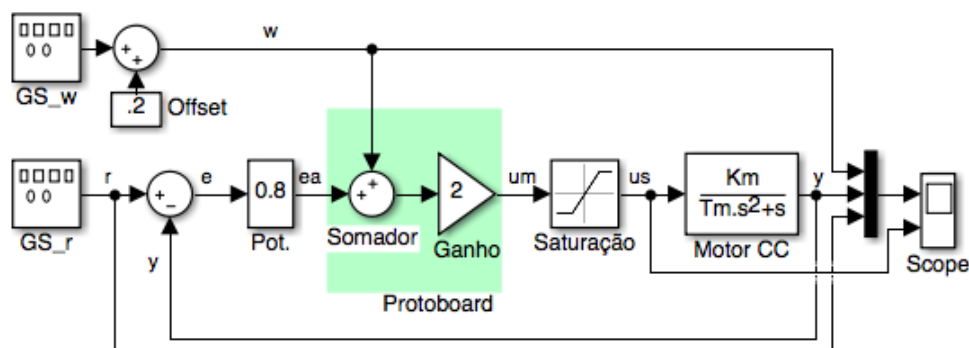


Figura 4 – Diagrama de blocos simplificado do kit impressora – Experimento 1.

3 – Identificação dos Parâmetros do Processo

A simulação do processo, Fig. 4, permite verificar a resposta esperada em laboratório, Fig. 5. A Fig. 6 ilustra respostas típicas obtidas com o kit.

As funções de transferência $Y(s)/R(s)$ e $Y(s)/W(s)$ permitem calcular e_{ss} .

$$\frac{Y(s)}{R(s)} = \frac{\frac{K_p K K_m}{s(sT_m + 1)}}{1 + \frac{K_p K K_m}{s(sT_m + 1)}} = \frac{K_p K K_m}{K_p K K_m + s(sT_m + 1)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

$$\frac{Y(s)}{W(s)} = \frac{\frac{K K_m}{s(sT_m + 1)}}{1 + \frac{K_p K K_m}{s(sT_m + 1)}} = \frac{K K_m}{K_p K K_m + s(sT_m + 1)} = \frac{K_w \omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

Onde K_p – ganho do potenciômetro; K – ganho do circuito com AmpOP; K_m – ganho do motor (inclui, de forma pragmática, “*todos os ganhos*” do kit que não sejam K_p ou K).

A identificação dos parâmetros do processo, K_p , K , K_m e T_m pode ser feita de diversas formas. Neste experimento será utilizada uma identificação determinística baseada no tempo de pico t_p , o sobrepasso percentual M_p , e os valores em regime permanente de r_{ss} e y_{ss} . Com estes valores é possível calcular ζ e ω_n . Sendo K_p o ganho estabelecido pelo potenciômetro do Kit, e K o ganho do circuito com AmpOp, obtêm-se facilmente os parâmetros do motor, K_m e T_m .

Consideração Prática:

A equação acima $Y(s)/R(s)$ mostra que, para degraus de referência, em regime permanente, a saída segue a referência sem erro. No mundo real pode haver uma pequena diferença entre os sinais medidos, devido a alguma(s) parte(s) do sistema não bem calibradas. Neste caso é melhor utilizar os termos do denominador da seguinte função de transferência nos cálculos, uma vez que há um ganho K' distorcendo o numerador.

$$\frac{Yr(s)}{Rr(s)} = \frac{K' \omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}$$

Formulário:

Observe que para um sistema de 2ª ordem sem zeros vale:

$$\text{Sobrepasso Percentual: } M_p = 100 \times e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$\text{Fator de Amortecimento: } \zeta = \frac{-\ln(M_p/100)}{\sqrt{\pi^2 + \ln^2(M_p/100)}}$$

$$\text{Tempo de pico: } t_p = \pi/\omega_d$$

$$\text{Frequência natural: } \omega_n = \omega_d / \sqrt{1 - \zeta^2}$$

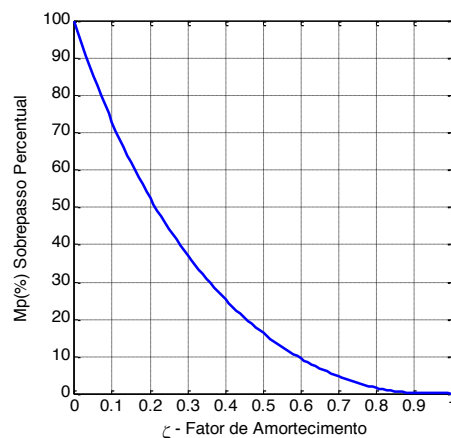


Figura 4 – Relação $\zeta \times M_p$ sistema 2ª ordem sem zeros.

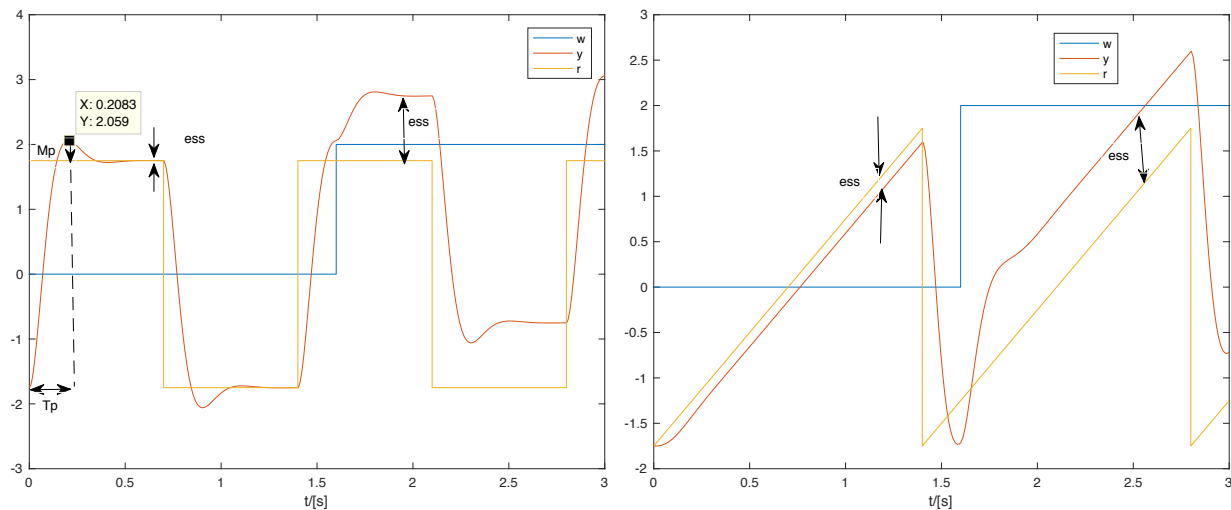


Figura 5 – Típica resposta à referência onda quadrada e dente de serra. Degrau de perturbação em 1,6 s. O pico da resposta subamortecida ao degrau de referência permite calcular os parâmetros do processo, K_m e T_m .

Para efeito de comparação apresentam-se a seguir respostas obtidas com o kit impressora.

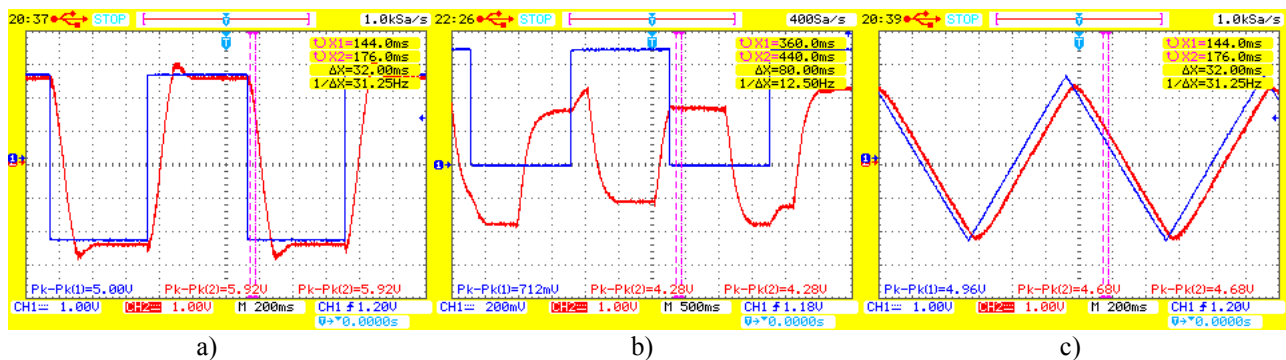


Figura 6 – Respostas típicas obtidas com o kit impressora.

a) y, onda quadrada de referência; b) y, onda quadrada de perturbação e c) y, onda triangular de referência.

4 – Pré-relatório

Para este experimento não será necessário **um pré-relatório**.

5 - Procedimento Experimental

3.1 – Verificação inicial: Conecte o sinal de erro atenuado ao pino do divisor de tensão que adequa o sinal à entrada do ADC do Arduino (PROTB_2 pino 6 ao PROTB_1 pino 1). Posicione o carro bem à direita para zerar o *encoder*. Energize o kit, inicialmente com o $\pm 12 V$, *GND* e depois no modo turbo 18 V. O carro de se posicionar no centro do trilho.

3.2 – Monte no protoboard utilizando PROTB_1 ou PROTB_2 os resistores que implementam o somador de perturbações (ganho não inversor = 2, potenciômetro em 0,5). Verifique que, ao ligar (sem gerador de sinais), que o carro se posiciona no centro do trilho.

3.3 – Resposta ao degrau. São necessários aqui, dois geradores de funções. Obtenha uma resposta sub-amortecida ($M_p \sim 30\%$) a degraus de referência – onda quadrada ($\sim 2V_{pp}$, 1 Hz) à referência e onda quadrada ($\sim 1V_{pp}$; 0,25 Hz) como perturbação. Note que **não deve haver saturação significativa** do sinal u_s (ver Fig.4) para manter a operação linear do sistema.

3.4 – Resposta à rampa – Substitua a referência quadrada por uma onda triangular. Mantenha a perturbação como no item anterior.

3.5 – Com $K_{pot}=1$ e K projetado, meça o **ganho de malha**, K_m , diretamente no Kit ou pela razão V_{out}/V_{in} no osciloscópio. (Abra a malha antes do ADC).

3.6 – Meça a relação da posição do carro, y em metros com a tensão y_v em Volts.

Obs: *Salve todos os sinais do osciloscópio no formato .csv (comma separated value) que pode ser lido, para posterior avaliação no matlab, com o comando csvread().*

6 - Relatório

O relatório deve apresentar de forma clara e direta um relato objetivo do que foi realizado no laboratório, fazendo uso sempre que possível de rigor teórico para explicar os procedimentos seguidos. São componentes importantes do relatório uma análise crítica dos resultados e das condições de experimentação.

Deve ser entregue um relatório por grupo incluindo:

- 1) Número do kit utilizado (Este mesmo kit deverá ser utilizado nos experimentos 2 e 3. *A utilização de outro kit demandará nova identificação dos parâmetros!!*)
- 2) Gráficos dos sinais adquiridos pelo osciloscópio, plotados a partir do formato .csv.
- 3) Valores e métodos utilizados para os cálculos dos parâmetros:
Considerando K_{pot} adotado e a resposta ao degrau subamortecida, calcule K_m e T_m .
- 4) Simule o processo (Figura 3) com os parâmetros identificados.
- 5) Preencha a tabela a seguir:

Obs: *Assume-se que, com exceção de K_{pot} , K_m engloba todos os demais ganhos de malha e será adotado assim como simplificação em todos os projetos subsequentes.*

Tabela 1: Erros em regime permanente

e_{ss}	à referência (experimental)	à perturbação (experimental)	à referência (simulação)	à perturbação (simulação)
Degrau				
Rampa				

Questões (a serem respondidas como parte da discussão no relatório do experimento)

- a) De que maneira o “turbo” afeta o experimento? e_{ss} ? Valores de K_m e T_m ?
- b) Que alteração haveria nos erros calculados (Tabela. 1) se fosse utilizado um controlador com canal integral? Qual a alteração na tipologia (para R e para W)?
- c) Considerando o controlador P utilizado neste experimento e um controlador PI, que diferença haveria na dinâmica (i.e., no tempo de pico e no tempo de subida) da resposta à perturbação?

Apêndice A - Código Arduino

```
// Encoder Library http://www.pjrc.com/teensy/td_libs_Encoder.html

#include <Encoder.h>

// Valimentacao ponte H -> 18V
// Arduino NANO
// 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output

#define Ss1 3 // definicao pinos arduino (Interrupcao)
#define Ss2 4 // definicao pinos arduino (NInterrupcao)

#define HORARIO true
#define ANTIHORARIO false
#define ENA 6 // definicao pinos arduino L298 (PWM)
#define IN1 7 // definicao pinos arduino L298
#define IN2 8 // definicao pinos arduino L298
#define Posicao 9 // definicao pinos arduino (PWM)
#define Tpro 13 // informacao tempos processamento
#define Entrada A0 // definicao pinos arduino (A/D)
#define Zmorta 142 // definicao zona morta 20V alim
// ajuste melhor possivel com compensador prop c/ Kp=.1

float Apos=255./7000.,Bpos=0.5, //Constantes encoder
      Aent=(255-Zmorta)/512.,Bent=Zmorta-254.5; //Constantes entrada
//Calculo para maximizar saturacoes [SatKitImpress2.mws]
int Vel,Pos;

// Inicializacao Encoder
Encoder Sens(Ss1, Ss2); //1 interupcoes

void setup() {
  // Serial.begin(9600);
  // setPwmFrequency(Posicao, 1); // ~31kHz
  TCCR1B = TCCR1B & 0b11111000 | 0x01;
  // setPwmFrequency(ENA, 8); // ~8kHz
  TCCR0B = TCCR0B & 0b11111000 | 0x02;
  for (int i = 6; i < 13; i++)
    pinMode(i,OUTPUT);
}

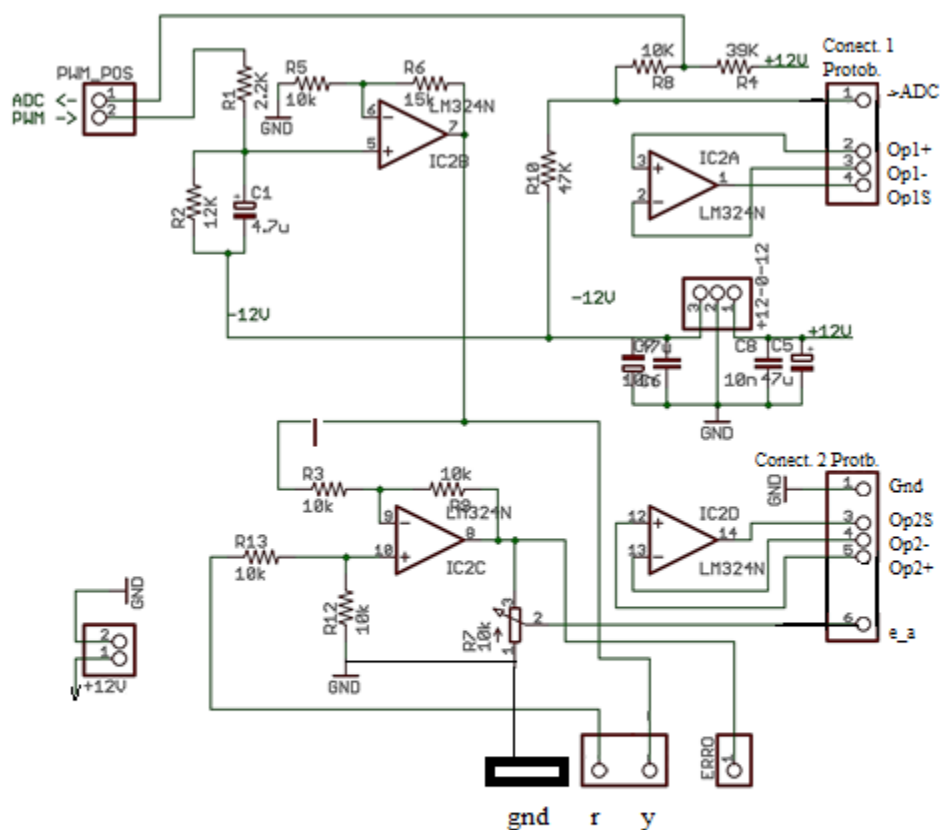
void loop() {
  // atualizacao saida posicao
  Pos=Sens.read()*Apos+Bpos;
  analogWrite(Posicao,Pos); //analogWrite de 0 a 255

  digitalWrite(Tpro,true); // marcador tempo 1

  // atualizacao saida motor
  Vel = (float)analogRead(Entrada)*Aent+Bent; // analogRead de 0 a 1023
  acionaMotor(Vel); //Ganho em Vel diminui a saturacao na entrada
  //que já é baixa <=> evitar [SatKitImpress.mws]
  digitalWrite(Tpro,false); // marcador tempo 2
}

//funcoes para acionamento individual de cada motor
void acionaMotor(int velocidade) {
  if(velocidade >= 0) {
    velocidade += Zmorta;
    if(velocidade > 255) velocidade = 255; // saturacao
    analogWrite(ENA, velocidade);
    digitalWrite(IN1,true); digitalWrite(IN2,false); // sentido horario
  }
  else {
    velocidade -= Zmorta;
    if(velocidade < -255) velocidade = -255; // saturacao
    analogWrite(ENA, -velocidade);
    digitalWrite(IN1,false); digitalWrite(IN2,true); // sentido anti-horario
  }
}
```

Apêndice B - Diagrama elétrico da placa analógica



Apêndice C - Diagrama elétrico geral

