



Pontifícia Universidade Católica do Rio de Janeiro

ENG1450 - Microcontroladores e Sistemas Embarcados – 2017.1

Controle PID de motor DC com PIC18F452

Matheus Caldas – 1312760

Ian Albuquerque - 1310451

Rio de Janeiro, 29 de maio de 2017

Índice

1.0 – Objetivos	3
2.0 – Montagem	4
2.1 - Teclado	4
2.2 – Motor	4
2.3 – Feedback	5
2.4 – Controle PID	5
3.0 – Testes do Valor das Constantes	6
3.1 – Teste RPM x Duty	6
3.2 – Teste RPM x Tensão	8
3.3 – Valores das Constantes	10
4.0 – Resultados	11

1.0 – Objetivos

O objetivo do experimento é controlar um motor DC por meio do controle PID da razão dos ciclos (*duty*) de ondas PWM. Através de um teclado, deve ser escolhido o valor das rotações por minuto (RPM) e, através do uso de um sensor que mede a velocidade de rotação do motor, o PIC deve ajustar a velocidade de saída com o ajuste do PWM. No display LCD serão exibidas as RPM atuais e o valor desejado, valor este encontrado pelo feedback do motor dado por um sensor ótico. Dessa forma é possível acompanhar o controle sendo feito. Devem ser realizados testes para 800, 1200 e 1400 RPM.

2.0 – Montagem

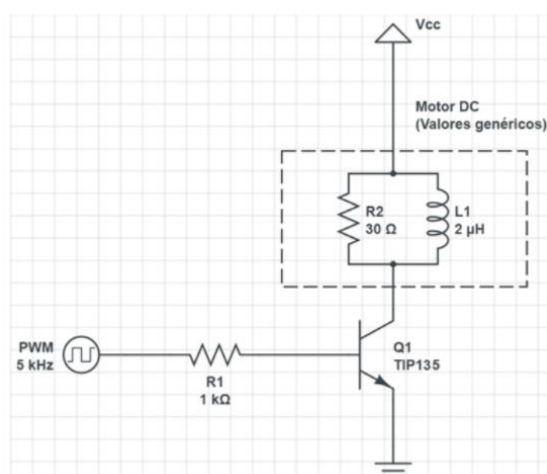
2.1 - Teclado

O teclado é basicamente o mesmo utilizado nos experimentos anteriores. Ele é feito por interrupção. Quando uma tecla é pressionada, uma interrupção de nível é acionada e é possível detectar qual a linha que foi pressionada através da alteração do nível lógico das colunas, uma por uma e a verificação do nível lógico nas linhas. No instante que for detectado que uma determinada linha mudou de estado lógico, sabemos exatamente qual foi a linha e qual a coluna corresponde ao botão pressionado.

Como simplificação deste projeto, mapeamos três teclas distintas do teclado para três valores distintos de RPM desejados. Mapeamos as teclas com valor numérico 7, 8 e 9 para os valores de RPM de 800, 1200 e 1400 respectivamente, uma vez que esses foram os três valores sugeridos no enunciado. Também foi implementado um anti bouncing para a leitura do teclado para evitarmos que oscilações mecânicas interferissem na nossa leitura do teclado.

2.2 – Motor

A implementação do motor DC consiste em ligar um dos dois terminais numa fonte externa que o outro terminal é ligado na saída de um transistor que permite a passagem de corrente caso a PWM que vem do PIC tenha um valor médio mínimo. Isso é feito controlando o *duty* da onda que pode variar de 0 (0%) a 255(100%). Quanto maior o duty estabelecido mais rápido o motor irá girar. O esquema básico está abaixo.



2.3 – Feedback

O feedback do motor, que informa o PIC qual é a velocidade corrente de rotação, é feito por um sensor óptico. Ele nada mais é do que um sensor de luz que a cada volta do 64 pulsos, cada uma delas correspondendo a um buraco em uma roda de plástico fixa ao motor que permite a passagem de luz. Para o tratamento do sinal desse sensor, utilizamos uma entrada por interrupção do PIC. A cada novo pulso uma nova interrupção é contada. Ao final de um tempo estipulado (200ms) é feita a conta de quantos pulsos foram recebidos no intervalo de tempo, chegando portanto na frequência e consequentemente no número de rotações por minuto.

2.4 – Controle PID

O controle foi formulado utilizando o conceito de um controle PI, onde temos uma variação proporcional e outra integrativa (somadora). A parte derivativa (que faria que o controle fosse um PID e não somente um PI) que mede a variação instantânea não foi usada conforme solicitado no enunciado. Sua expressão analítica conforme dado é:

$$u[k] = u_0 + K_p * e[k] + K_i \sum_{k=1}^n e[k]$$

A “parte proporcional” atua de modo a ser proporcional com a dimensão do erro (dado pela diferença entre o valor buscado e o valor atual). K_p é a constante de proporcionalidade que deve ser definida empiricamente. Caso o K_p seja muito alto, o motor apresentará um comportamento muito oscilatório e caso seja muito baixo pode demorar muito a se estabilizar.

A “parte integral” atua de modo a ser proporcional com a dimensão e também a duração do erro. Dessa forma, o erro pode acumular, mesmo que lentamente, que a parte integral (somadora) se responsabilizará por consertar isso. K_i é a constante integrativa. Vale notar que um K_i muito alto pode tornar o sistema muito instável, enquanto que um valor muito baixo, pode resultar em um sistema que demora demais a se estabilizar. O valor ideal foi também encontrado empiricamente.

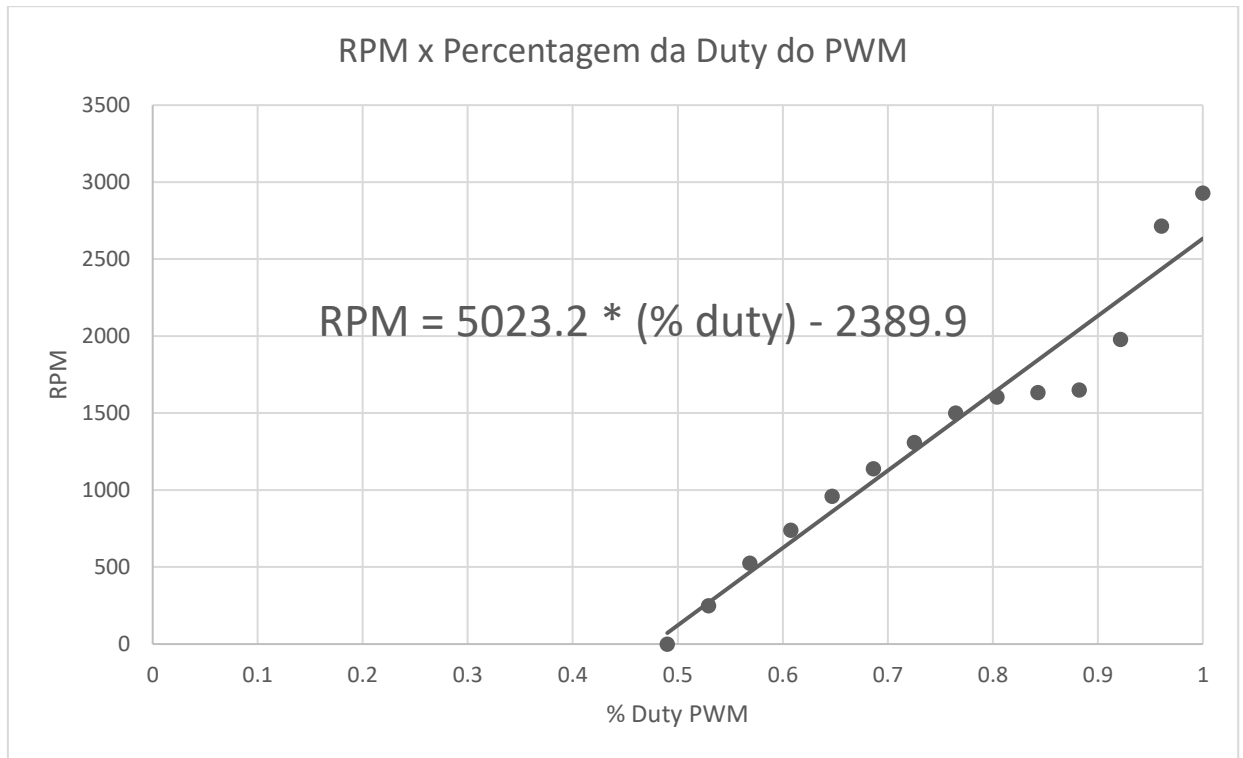
3.0 – Testes do Valor das Constantes

3.1 – Teste RPM x Duty

Com o objetivo de basear o valor das constantes em algo, foram tiradas algumas medidas do sistema. Segue a seguir, as relações de tensão externa aplicada ao motor e sua rotação assim como da porcentagem da *duty* aplicada, junto com a rotação do motor para uma tensão de alimentação do motor fixa. O valor da tensão fixa foi de 10V, enquanto que foi variada a porcentagem da *duty*, 255 sendo 100% e 0 sendo 0%;

Valor de Saída da PWM	% Duty	RPM
255	1	2930
245	0,960784	2715
235	0,921569	1980
225	0,882353	1650
215	0,843137	1635
205	0,803922	1605
195	0,764706	1500
185	0,72549	1310
175	0,686275	1140
165	0,647059	960
155	0,607843	740
145	0,568627	525
135	0,529412	250
125	0,490196	0

Modelamos os dados como lineares e obtemos a seguinte equação de reta, utilizando o método dos mínimos quadrados.

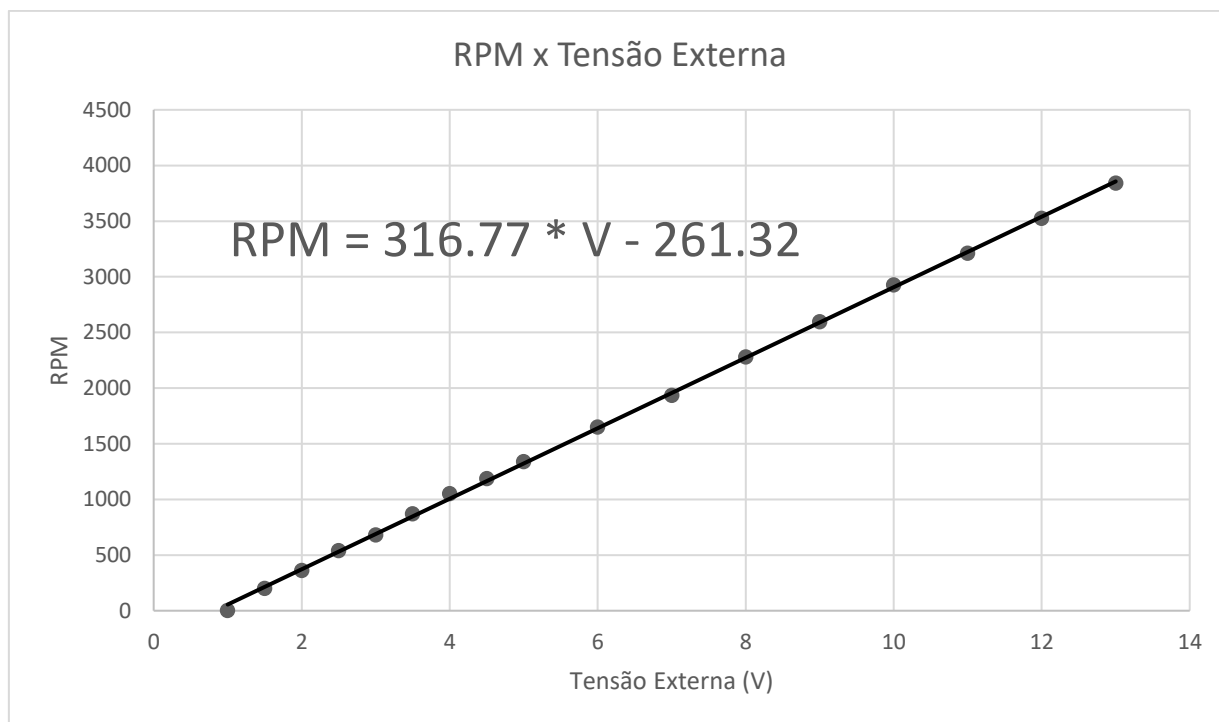


3.2 – Teste RPM x Tensão

Outro experimento realizado foi feito mantendo o duty fixo em 100% (255) e variando a tensão da fonte externa. Os resultados foram os seguintes:

Tensão (V)	RPM
1	0
1,5	200
2	360
2,5	540
3	680
3,5	870
4	1050
4,5	1185
5	1340
6	1650
7	1935
8	2280
9	2595
10	2925
11	3210
12	3525
13	3840

Também modelamos o resultado obtido com um modelo linear.



3.3 – Valores das Constantes

Uma vez com alguns valores empíricos, podemos estimar uma tentativa inicial de constante para K_p .

$$\text{Para } K_i = 0$$

$$u[k] = u_0 + K_p * e[k]$$

$$\text{Estando } u_0 \text{ em "unidades de PWM" (0 a 255)}$$

$$\text{Estando } e[k] \text{ em RPM}$$

Utilizando a equação encontrada pelo gráfico para converter PWM em RPM

$$RPM = 5023,3 (\% \text{ duty}) - 2389$$

$$\text{Assim,} \quad \left(5023,3 \left(\frac{u[k]}{255} \right) - 2389 \right) = \left(5023,3 \left(\frac{u_0}{255} \right) - 2389 \right) + \widetilde{K}_p * e[k]$$

$$\widetilde{K}_p \text{ sendo } K_p \text{ agora nas novas unidades}$$

$$\text{Para } \widetilde{K}_p = 1$$

$$\left(5023,3 \left(\frac{u[k]}{255} \right) \right) = \left(5023,3 \left(\frac{u_0}{255} \right) \right) + e[k]$$

$$u[k] = u_0 + \left(\frac{255}{5023,3} \right) e[k]$$

$$K_p = \left(\frac{255}{5023,3} \right) = 0,05076$$

Assim, experimentamos com diversos valores, começando com $K_p = 0,05076$ e $K_i = 0$. Os valores iniciais já demonstraram um comportamento razoável. Após diversas iterações, chegamos nas seguintes constantes empiricamente:

K_p	0.015
K_i	0.0005

4.0 – Resultados

Para testar o pleno funcionamento do sistema completo, variamos o valor da RPM desejada definindo-a com os valores de 800, 1200 e 1400 RPM. O sistema demorou em média cerca de 5 segundos para estabilizar. Importante notar que a precisão ficou muito melhor para 1200 e 1400 RPM. Isso fica claro ao analisar que quanto mais rotações, menos suscetível à erros está o sensor óptico acoplado ao motor. Outro teste aplicado foi alterar a alimentação externa do motor. Ao testar para variações da alimentação do motor entre 4V e 12V, o PIC conseguiu variar o duty da PWM de modo que a velocidade do motor voltasse para a RPM desejada. Consideramos o resultado um sucesso.