

ENG1450 - Experiência 7 - Motor

Hugo Costa Machado - Marcos A van Boekel Marques

29/05/2017

1 Introdução

Esta experiência foi dividida em partes, com seu primeiro objetivo consistindo em simular no Proteus um projeto que fosse capaz de obter a leitura de uma onda quadrada e seu valor ser exibido no LCD. Após isto deveríamos montar um controlador de motor DC, que transformaria o sinal de PWM em tensão para o motor DC. Estas simulações foram efetuadas e estão nos arquivos anexos.

A Experiência 7 tinha como propósito principal a implementação da lei de controle PI, para comandar através do PIC a velocidade do motor DC. Este deveria se manter equilibrado em torno do SetPoint especificado.

2 Dificuldades encontradas

- A tensão mínima para o funcionamento correto do nosso motor DC foi de 10V. Tentamos alterar os resistores associados ao circuito porém com 5V o motor não conseguia alcançar rotações maiores que 800 RPM. Já com os 10V, conseguimos atingir as rotações solicitadas - 800, 1200 e 1400 RPM - indo além obtivemos rotações maiores até o limite de 1900 RPM.
- Ao entrarmos com valores para o SetPoint, através do teclado, em alguns casos o timer0 parava de responder, afetando a atualização da rotação do motor e da exibição do LCD que ficava congelada. A solução de contorno obtida foi pressionar uma tecla qualquer que não tenha efeito no programa, com isso o timer0 voltava a responder. Olhando o código fonte, não conseguimos identificar o conflito para a causa deste bug.

3 Funcionamento

A tensão do gerador determina o RPM máximo que o motor pode atingir. Qualquer SetPoint definido até esse valor máximo, funciona como esperado. Ao definirmos um SetPoint, o programa tenta ajustar a rotação do motor para este valor, eventualmente chegará a um estado que oscilará perto do SetPoint definido estabilizando-se. Em outras palavras o erro mantém-se estável. Para RPMs muito baixos, o comportamento é diferente do descrito acima. O motor gira até um RPM acima do SetPoint(wind-up) e depois para. Isto repete-se de forma que a rotação do motor não ficará próxima do SetPoint, pois o primeiro "tiro" sendo acima tenta corrigir para baixo, entretanto ao diminuir o valor este fica negativo, ocorrendo então a parada do motor pois não são permitidas rotações negativas. Neste caso o erro se mantém instável. Para um K_i muito pequeno, exemplo, 100x menor que K_p ($K_p=0.1$), o motor não consegue alcançar o SetPoint. Para um K_i 10x menor que K_p ($K_p=0.1$), o motor converge, porém lentamente. Então achamos melhor colocar $K_i = K_p$, onde $K_p = 0.01$. Dessa forma, a convergência ocorreu de forma mais acelerada do que nos experimentos anteriores. Testamos para os setpoints solicitados 800, 1200, 1400 RPM. O funcionamento foi conforme o esperado. Portanto, obtivemos o resultado esperado deste laboratório, em que consistia ter um controlador de motor DC que respeita-se a lei de controle PI.