UNIVERSIDADE FEDERAL DE UBERLÂNDIA - UFU

FACULDADE DE ENGENHARIA ELÉTRICA – FEELT

SINAIS E SISTEMAS EM ENGENHARIA BIOMÉDICA

**Controlador Proporcional**

**Alunos:**

1. Ítalo Gustavo Sampaio Fernandes - 11511EBI004
2. Nathalia Rodrigues - 11411EBI018
3. Paulo Camargos Silva - 11611EBI023

**Prof.** Sérgio Ricardo de Jesus Oliveira

Uberlândia, **11** de **dezembro** de 2017

**1 – Introdução**

Em um controlador proporcional à saída do mesmo, chamada de ação de controle, é diretamente proporcional ao sinal do erro atuante. Para remover o sinal de proporcionalidade entre as duas variáveis, deve admitir a constante Kp de proporcionalidade, o ganho proporcional. Na qual a saída é igual a Kp multiplicada pelo erro atuante.

O controlador proporcional é basicamente um amplificador com ganho ajustável, onde o sinal de controle é o sinal de erro amplificado por Kp. Este tipo de controle é destinado quando queremos uma boa velocidade de resposta de atuação sem compromisso com o erro de regime igual a zero. O diagrama de blocos deste controlador pode ser observado na Figura 1 abaixo.

A medida que o valor de Kp aumenta o erro de regime e o tempo de acomodação diminuem. Porém se aumentar muito o valor de Kp provoca a saturação no sistema ou ainda produzir uma variável manipulada acima do limite tolerável para o processo. Assim, condicionamos o limite máximo de Kp às características do processo.

**2 – Materiais e Métodos**

**2.1 - Materiais**

O circuito montando neste experimento é apresentado abaixo na Figura 2 abaixo.

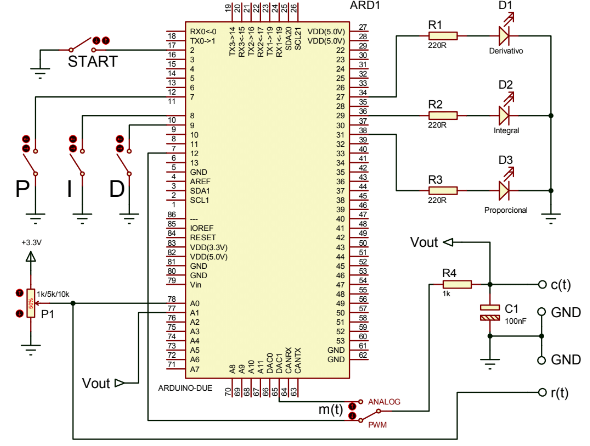


Figura 2: Esquema do circuito montado em laboratório

Os materiais utilizados neste experimento foram:

* Osciloscópio;
* Arduino;
* 3 LEDs;
* 3 Resistores de 220 Ω;
* 1 Resistor de 1 kΩ;
* 1 Capacitor de 100 nF;
* 1 Potenciômetro de 10 kΩ;
* Multímetro.

**2.2 - Obtenção da função de transferência do sistema.**

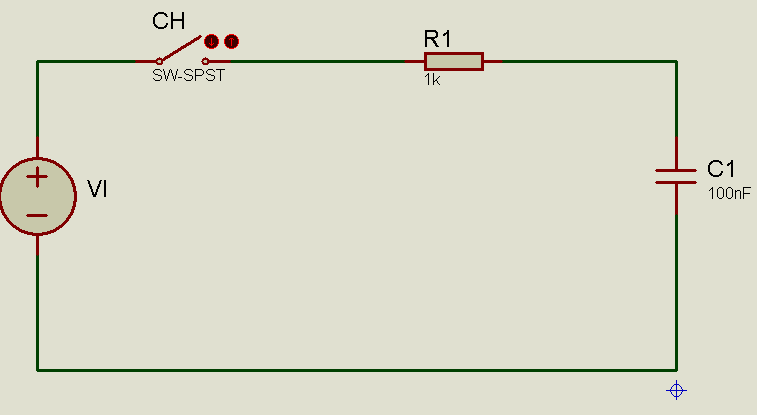
****

Fig. 3 - Representação do circuito

Para o sistema apresentando na Figura 3, temos a seguinte função de transferência:

Considerando a tensão no capacitor como um divisor de tensão, temos:

Fechando a malha temos a seguinte F(s):

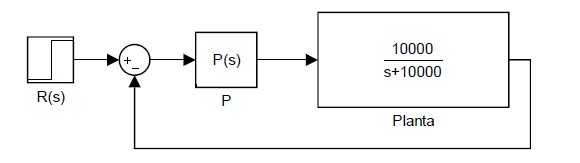


Figura 4 – Diagrama de blocos de malha fechada com Kp.

A Figura 5 abaixo exibe os gráficos de saída e erro do sistema.

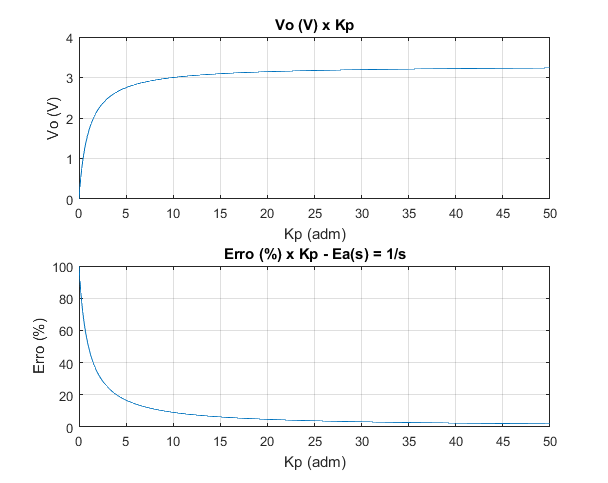


Figura 5 – Vo e erro em função de Kp.

**2.3 – Código**

Inicialmente, o código faz a inicialização dos pinos dos sensores, pinos de saída e da chave on/off do sistema. Em *void setup()* o regime de trabalho dos pinos são configurados e a leitura/escrita é configurada para 12 bits (0 - 4095). No interior de *void loop()* a rotina faz a leitura da tensão no capacitor, da tensão de referência e do valor de Kp. Posteriormente o erro é calculado. O sinal atuante então é calculado, multiplicando o erro pelo Kp. Após o cálculo, o sinal é enviado para as portas analógica (DAC1) e PWM.

**3 – Resultados e discussões**

Os gráficos obtidos no osciloscópio são exibidos abaixo.

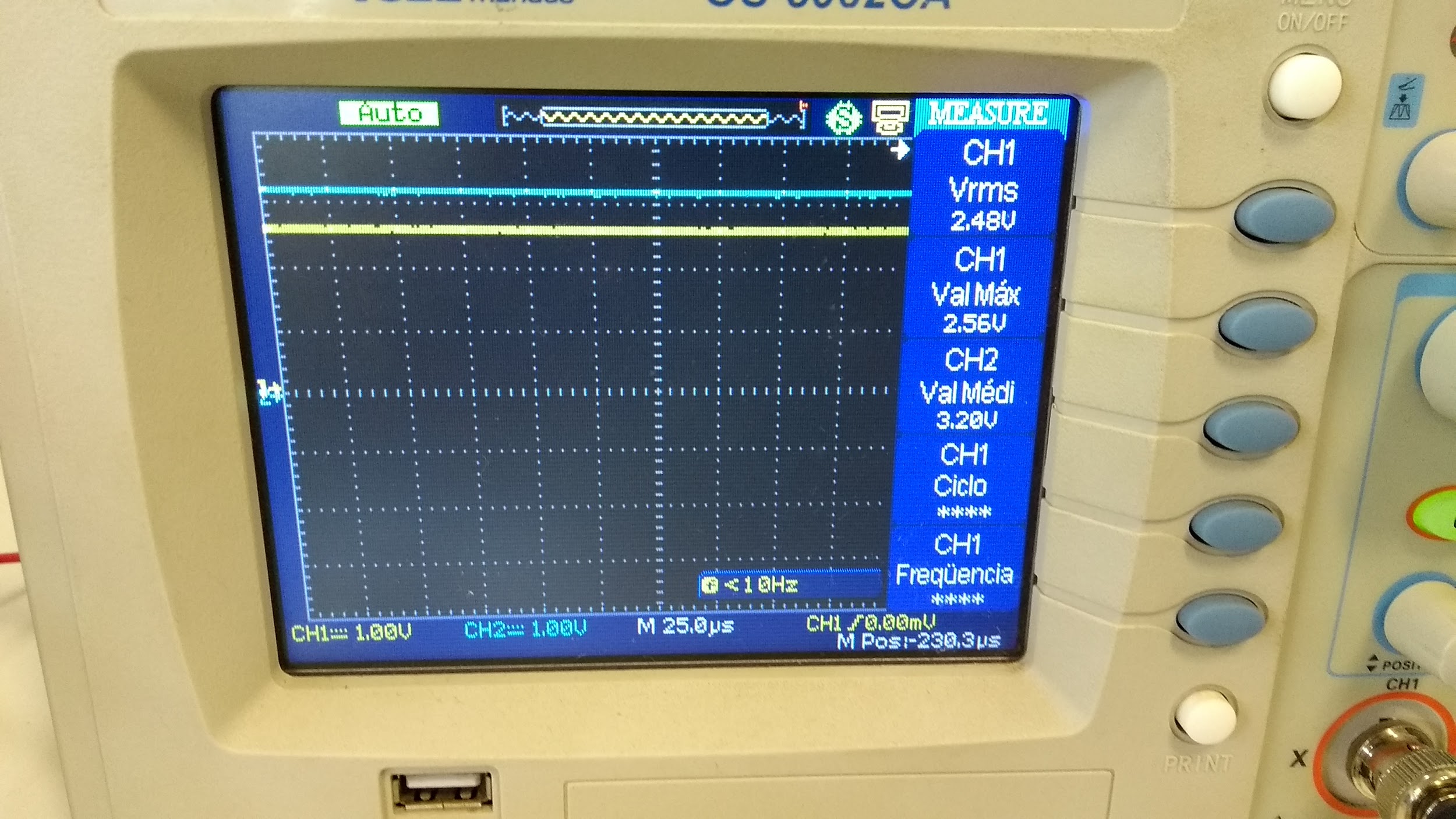


Figura 6 - Sinal referência (azul) e Vout (amarelo). Kp = 19 com sinal aplicado por DAC1.

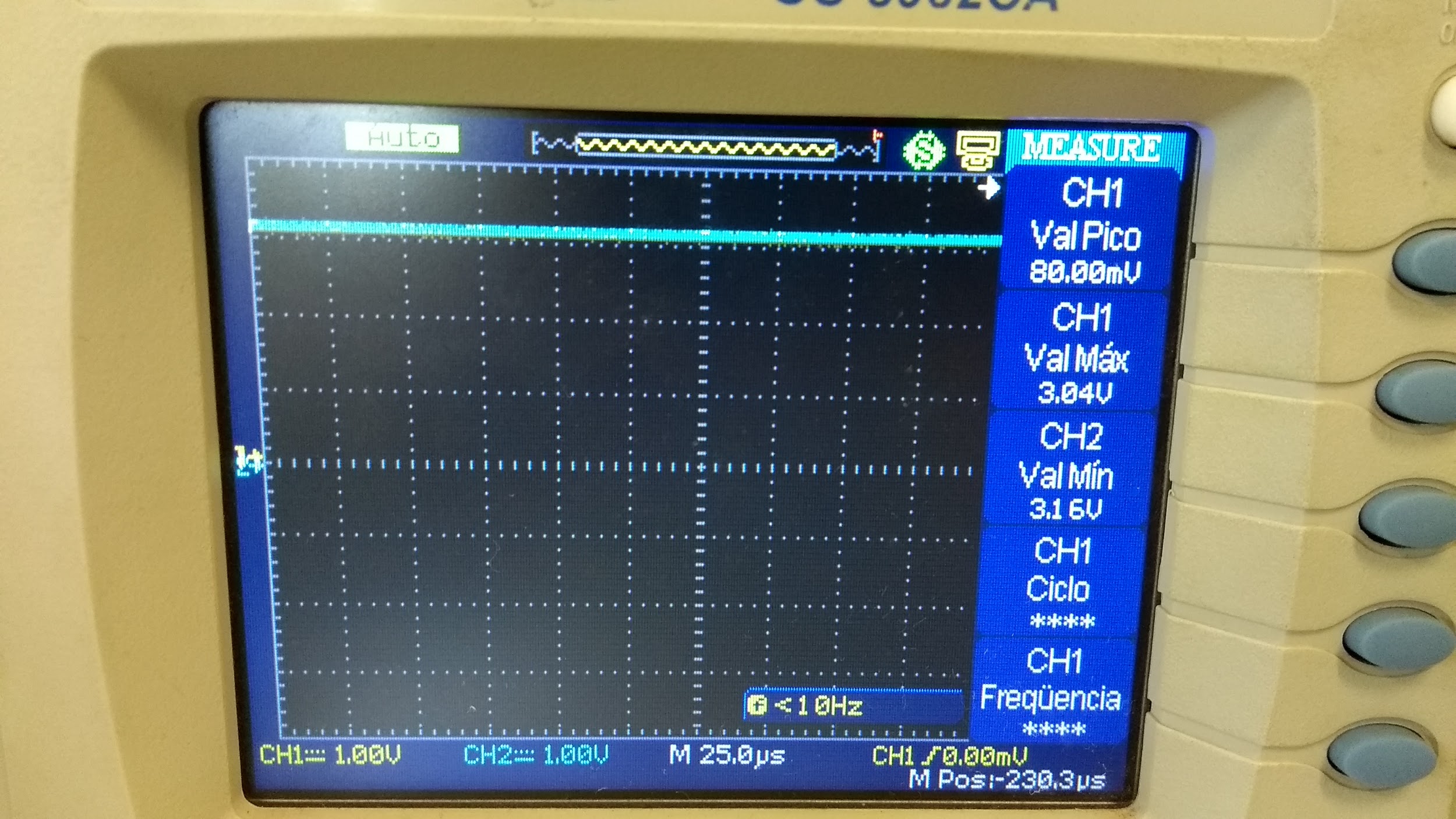


Figura 7 - Sinal de referência e Vout juntos (mesma amplitude) com Kp = 19, porém sinal aplicado através da porta PWM.

Podemos observar que o erro se torna menor à medida que aumentamos o valor de Kp. Isto respeita a função de transferência calculada e o gráfico desenhado em MatLab. Podemos observar também que existe um erro residual nas portas analógicas. Este erro se deve a própria construção do Arduino Due, onde o intervalo de tensão de saída das portas DAC’s está limitado entre 0.55 - 2.75 V. Desta forma, quando o sinal atuante for 0, tem-se uma tensão residual de 0.55 V aplicada ao sistema (mesmo raciocínio para o limite superior). Este fato evidencia a melhor precisão das portas PWM neste quesito, onde os limites de 0 - 3.3 V são respeitados. Nas Figuras 6 e 7 podemos observar que o erro na porta PWM é menor.

**4 – Conclusão**

A partir do experimento montado e testado em laboratório foi possível visualizar e entender o princípio de funcionamento de um controlador proporcional. Foi possível também entender e obter a função de transferência do sistema. A partir da programação e do funcionamento do controlador tivemos capacidade de obter e observar a forma de onda capturada pelo osciloscópio do controlador proporcional e assim justificar as mesmas. Assim como também foi possível entender os resultados de e(t) obtidos pelo sistema.

**5 – Referências**

Links acessados em 11/12/2017:

https://www.embarcados.com.br/controlador-proporcional-eletronico/

https://www.arduino.cc/reference/en/language/functions/math/constrain/

**Anexo I – CÓDIGO**

1. #define pinSensor         A9 // Vout pin
2. #define pinReferencia     A8
3. #define pinStart          8
4. #define pinKp             A2
5. #define pinPWM            11
7. void setup() {
8. pinMode(pinStart, INPUT\_PULLUP);
9. pinMode(pinPWM, OUTPUT);
10. Serial.begin(115200);
11. analogReadResolution(12);
12. analogWriteResolution(12);
13. }
15. void loop() {
16. if (!digitalRead(pinStart)) {
17. int Vout = analogRead(pinSensor);   // Reading capacitor voltage
18. int ref = analogRead(pinReferencia);  // Reading reference voltage
19. int erro = ref - Vout;
20. int Kp = analogRead(pinKp) \* 20 / 4095.0; // 0<Kp<20. Ajustable as potenciometer
21. int P = erro \* Kp;
23. // Returns
24. // # P, if 0 < P < 4095;
25. // # 0, if P<0;
26. // # 4095, if P>4095.
27. P = constrain(P, 0, 4095);
28. analogWrite(DAC1, P);
29. Serial.println("Kp: " + (String) Kp +
30. "| P: " + (String) P + "| Ref: " + (String) ref + "| Vout: " + (String) Vout);
31. analogWrite(pinPWM, P);
32. }
33. }