

Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia Mecânica

SEM 142 – Sensores e Sistemas de Medidas

Prof. Leopoldo P.R. de Oliveira

Prática #1 – Incerteza de Mediação

OBJETIVO

Familiarização com a plataforma Arduino e cálculo de propriedades estatísticas de dados

MATERIAIS E MÉTODOS

1. Modelo para simulação do circuito disponível no tópico do e-disciplinas
2. Software para cálculos e geração de gráficos a escolha do aluno (MatLab, SciLab, Octava, Python, Excel, Google Sheet, etc.)

Obs: Se tiver dificuldades em acessar o simulador (tinkercad) no e-disciplinas, abra uma área no site <https://www.tinkercad.com/> e monte um circuito similar. O código Arduíno pode ser copiado/colado do final deste documento.

INTRODUÇÃO:

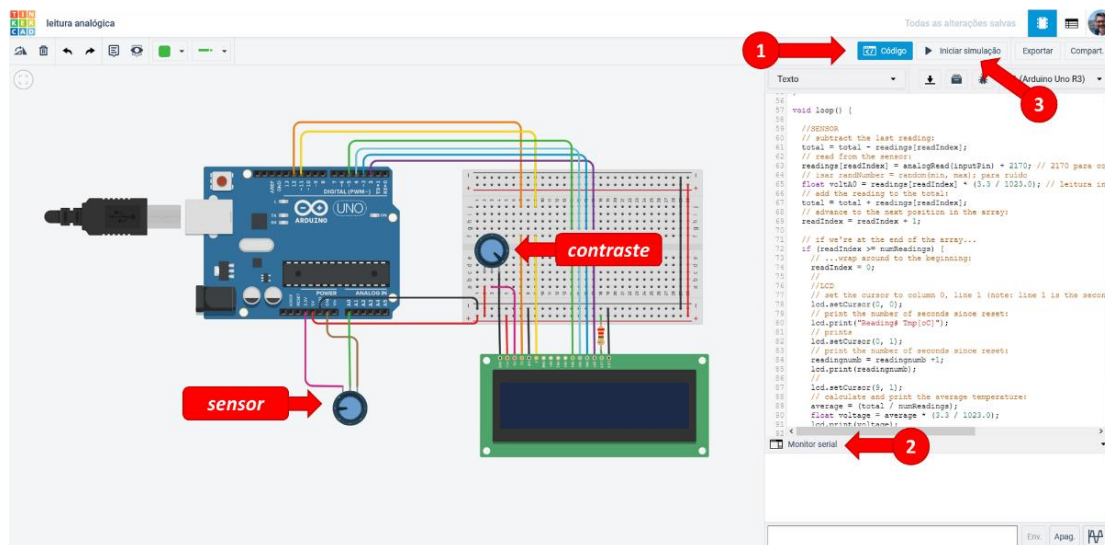
Esta prática (virtual) permite que vocês tenham contato com a plataforma Arduino, visualizem e manipulem algumas variáveis durante o experimento e utilizem esses dados para o cálculo de algumas propriedades estatísticas dos dados gerados.

Ela está relacionada com o tema de Incerteza de Medições e procura reproduzir o circuito de medição de temperatura, parte do sistema de controle de temperatura trazido como motivação na aula teórica. No circuito da Fig.1, o potenciômetro montado na protoboard tem a função de ajustar o brilho do display LCD. Já o segundo potenciômetro simula um sensor de temperatura analógico. A leitura deste valor é feita pela porta A0 e o valor de tensão é manipulado para resultar numa leitura de temperatura. Desta forma, ao variar o ângulo do pot. introduzimos variabilidade no valor da temperatura.

No tópico do e-disciplinas, acesse o conteúdo do Tinkercad, manipule o modelo para se familiarizar com os componentes, analise o código e habilite o monitor serial (vide Fig.1). Depois, inicie a simulação e observe as mensagens no display e monitor serial. Enquanto a simulação roda, você pode ajustar os potenciômetros e observar o efeito que eles provocam. Após esta etapa de familiarização, aponte o potenciômetro da “temperatura” aproximadamente para baixo, limpe o monitor serial (botão Apag.) e inicie novamente a simulação.

Assim como descrito na aula, este código toma 10 medidas de temperatura, uma por segundo e calcula o valor médio que é enviado ao display LCD. Ainda, cada medida individual que compõe esta média, é impressa no monitor serial, junto da sua etiqueta de tempo. Toda vez que o ângulo do Pot fica entre “2h e 6h” (numa analogia com o ponteiro de um relógio) a temperatura varia dentro da faixa aceitável. Entre “6h e 10h” ele registra um valor acima de 8°C. Se a média das 10 medidas supera o limite de 8°C, então fica registrado um alerta no contador.

Figura 1 – ambiente de simulação: (1) visualizar o código, (2) monitor serial e (3) iniciar a simulação.



PROCEDIMENTO:

Após se familiarizar com a prática, limpe o monitor serial e inicie a simulação novamente. Deixe que rode por, pelo menos, 100 iterações (“100s na contagem do monitor serial”). Durante o processo, varie a posição do potenciômetro para introduzir variabilidade na medida. Como cada aluno deve fazer isso de forma arbitrária, espero resultados diferentes de todos vocês. Fica a seu critério provocar ou não “Alarmes”.

Copie os dados do monitor serial que serão manipulados utilizando algum programa de cálculo a sua escolha (matlab, octave, python, excel, etc.).

1) calcule a média ' μ ' e o desvio padrão amostral ' σ ' para toda a população e guarde o valor para futuras comparações.

2) Histogramas (cf Figs. 2 e 3)

(a) Construa um histograma com estes valores. Teste alguns valores diferentes para o intervalo (ou número de bins) até encontrar uma distribuição com aspecto que lhe agrade. Note que, neste estágio, valores do eixo Y deste gráfico devem mostrar a incidência de ocorrências de indivíduos da população para aquele intervalo (bin), e a soma de todos os valores deve ser 100, ou seja, o total de amostras na sua população.

(b) Faça outro gráfico, com o valor da amplitude normalizado e compare com o gráfico da distribuição normal, usando os valores médio e do desvio padrão encontrado.

Figura 2 – Exemplos de histogramas de uma população de 1000 indivíduos com $\mu = 120$ e $\sigma = 1.47$ e (a) 20 bins e (b) 50 bins

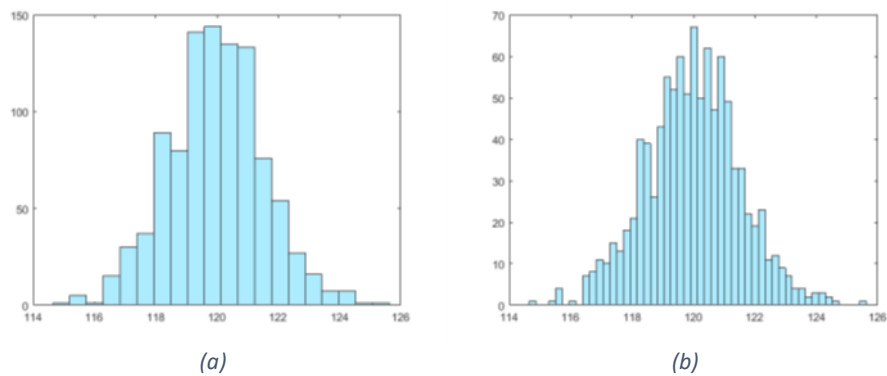
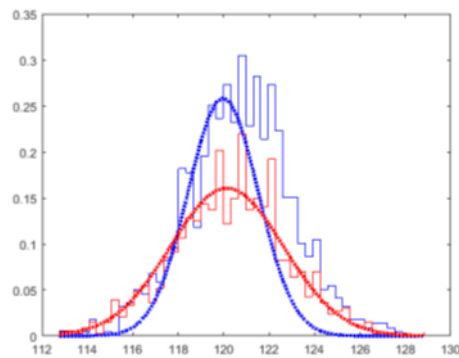
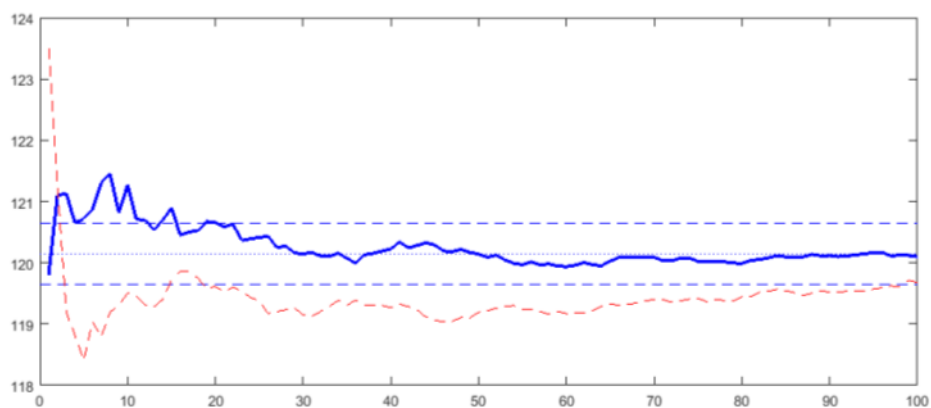


Figura 3 – Comparação entre histogramas normalizados e curva de distribuição normal para valores diferentes de desvios padrão



3) Agora calcule o valor médio (\bar{x}) para amostras com número crescente de medidas, ou seja, seguindo sua sequência de medidas, inicie com 1 elemento (cuja média será o próprio valor e, de forma incremental, inclua novas medidas até atingir a média das 100 medidas). Faça um gráfico com estes valores e note a partir de quantos indivíduos na amostra sua média converge. Para convergência observe se o valor médio fica limitado em $\pm 5\%$ da amplitude máxima dos dados (Fig. 4).

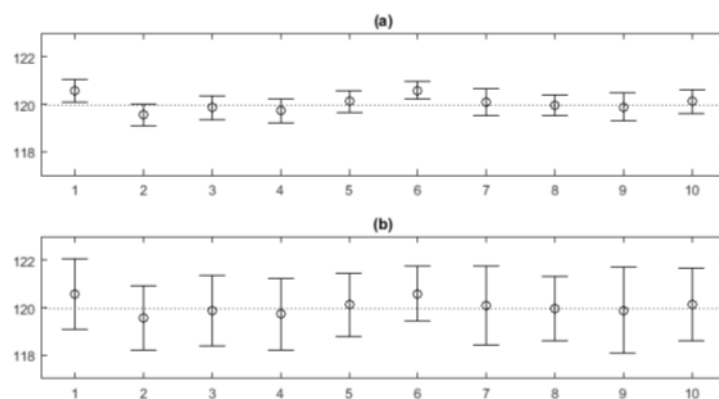
Figura 4 – Evolução da média de dois processos, a medida em que se aumenta o número de indivíduos na amostra.



4) Dentre os 100 indivíduos da população, selecione arbitrariamente 10 amostras de 5 indivíduos cada:

- (a) Calcule as 10 médias relativas a estas amostras, os 10 desvios padrão amostrais e os 10 desvios padrão das médias.
- (b) Compare o valor dos desvios padrão amostrais com o desvio padrão da população.
- (c) Compare cada valor médio e seu erro (desvio padrão da média) com o “média real” μ .
- d) repita o exercício anterior (a, b e c), agora com 10 amostras de 10 indivíduos cada. Compare os valores das médias e dos desvios padrão das médias com os obtidos para amostras de 5 indivíduos.

Figura 5 – comparação entre médias de diferentes conjuntos de indivíduos:
(a) com 25 e (b) com 10 indivíduos cada



ENTREGA:

Redija um relatório sucinto desta prática, incluindo:

- Gráficos gerados com os seus dados, similares aos das Figs. 2, 3 e 4;
- Uma tabela sintetizando os valores calculados nos itens 1 e 4 (itens a, b, c e d) - o gráfico da Fig.5 é opcional

Dicas

Matlab

funções úteis no matlab (na linha de comando, digite '`>> help nome da função`')

`hist` (calcula e plota um histograma)

veja também: `bar`, `stairs` e `plot` para outras opções de gráficos

use '`[h,x] = hist(...)`' para obter os valores do histograma na variável 'h' e a correspondente abscissa em 'x'.

`normpdf` (calcula os valores de uma PDF para distribuição normal, dados μ e σ da população e os valores abscissa 'X').

Arduino

Utilize o Tinkercad e monte seu próprio circuito (ou utilize um hardware). O código utilizado pra esta aula está copiado abaixo. Utilize como base para modificar e fazer ajustes a seu critério.

```
// DISPLAY DE CRISTAL LIQUIDO [LCD] /////
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// VARIABELS /////
const int numReadings = 10;      // number of readings in an average
int readings[numReadings];       // the readings from the analog input
int readIndex = 0;               // the index of the current reading
int total = 0;                   // the running total
int average = 0;                 // the average
int alarms = 0;                  // number of alarms
int randNumber;                  // random number (noise)
int inputPin = A0;

void setup() {
  // LCD /////
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.setCursor(0, 0);
  lcd.print("keep cool! ;-)");
  delay(1000);
  lcd.setCursor(0, 0);
  lcd.print("Alarm# Temp[oC]");
  lcd.setCursor(0, 1);
  lcd.print("0      calc.");

  // SENSOR /////
  // initialize serial communication with computer:
  Serial.begin(9600);
  // initialize all the readings to 0:
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
  // randomSeed() will then shuffle the random function.
  randomSeed(analogRead(A0));
  delay(1000);      // delay in between reads for stability
}

void loop() {
  // SENSOR /////
  // subtract the last reading:
  total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = analogRead(inputPin) + 2170; // 2170 para correção da temperatura
  // isar randNumber = random(min, max); para ruído
  float voltA0 = readings[readIndex] * (3.3 / 1023.0); // leitura instantanea no A0
  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;
  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
    // calculate and print the average temperature:
    average = (total / numReadings);
    float voltage = average * (3.3 / 1023.0);
    // LCD /////
    // set the cursor to column 0, line 1 (note: line 1 is the second row, since counting
    begins with 0):
    lcd.setCursor(0, 1);
    // print headings:
    lcd.print("Alarm# Temp[oC]");
    // checks for overheating:
```

```

        if (voltage >= 8){
            alarms = alarms + 1;
        }
        // prints number of alarms
        lcd.setCursor(0, 1);
        lcd.print(alarms);
        // prints temperature value
        lcd.setCursor(9, 1);
        lcd.print(voltage);
    }
    //
    //SERIAL /////
    // send it to the computer as ASCII digits
    Serial.print("time [s] = \t");
    Serial.println(millis() / 1000);
    Serial.print("Temp [oC] = \t");
    Serial.println(voltA0);

    delay(1000);          // delay in between reads for stability
}

```