

# Descrição temporal de forças de colisão: um modelo didático para laboratório de física assistido por sistema embarcado

Temporal description of colision forces: a didactic model for physics laboratory aided by embedded system

João F. Nascimento Júnior\*<sup>1</sup>, Vanessa E. S. Borges<sup>1</sup>, Raphael M. M. F. Nascimento<sup>2</sup>

<sup>1</sup>IPOG - Instituto de Pós-Graduação e Graduação, Departamento de Engenharia Civil, Goiânia, GO, Brasil

<sup>2</sup>Universidade Federal de Goiás, IPTSP, Goiânia, GO, Brasil

Recebido em 30 de Julho, 2018. Revisado em 24 de Setembro, 2018. Aceito em 12 de Novembro, 2018.

Nesse trabalho é modelado um experimento para laboratório de física para descrever temporalmente a força de colisão em uma bola. O modelo emprega um sensor piezoelétrico, um singelo circuito programável (*Arduino*) ligado a um computador e um suporte para a liberação da bola em queda livre. Um programa em linguagem *C* foi confeccionado para ler o sinal do sensor e calcular a intensidade da força, bem como para calibrar o sistema. O algoritmo do processamento, empregando as leis da física, é apresentado, assim como o código-fonte. Após a colisão o sistema fornece uma tabela *força x tempo* com resolução de 120  $\mu$ s, permitindo o aluno elaborar o gráfico da força ou fazer análise numérica, comprovando as leis da conservação do momento e da energia. Testes de desempenho são discutidos, mostrando que o modelo fornece uma exatidão muito boa, validando assim o método de calibragem. De baixo custo e montagem simples esse sistema embarcado, ou seja, controlado por programa, permite uma exploração minuciosa da força durante a colisão. Pode também ser estendido a outros fenômenos da física, bastando trocar o sensor e ajustar o programa.

**Palavras-chave:** força, colisão, sistema embarcado, *Arduino*, laboratório de física.

In this study is modelled an experiment for physics laboratory to describe the temporal force in a ball collision. The model employs a piezoelectric sensor, a modest programable circuit (*Arduino*) connected to a computer and a vertical support to release the ball in free fall. A C-language program was created to read the sensor signal and calculate the force intensity as well as to calibrate the system. The processing algorithm employing the physical laws is presented as the source-code. After the collision a 120  $\mu$ s resolution *force x time* table is provided, allowing the student to plot the graphic of force or to do numeric analysis, verifying the momentum and energy conservation laws. Performance tests are discussed showing that the model offers very good accuracy, thus validating the calibration method. At low cost and simple assembly, this embedded system, that is, program-controlled, allows a detailed exploration of force during the colision. As well it can be extended to another physics effects, merely changing the sensor and adjusting the program.

**Keywords:** force, colision, embedded system, Arduino, physics laboratory.

## 1. Introdução

O estudo de colisões é um tema clássico em experimentos de física. Nesse fenômeno a força de interação começa do zero, atinge um valor máximo e cai a zero novamente. Estudos podem ser concebidos mediante teoria de oscilações e conservação do momento e energia a fim de identificar a forma temporal da força [1, 2],  $F(t)$ . Essa curva pode ser mapeada em laboratório mediante um sensor piezoelétrico, que gera um sinal elétrico quando pressionado, e uma placa *Arduino*<sup>[1]</sup> [3], um modesto módulo eletrônico de fácil programação. Ambos baratos e facilmente encontrados no mercado.

Ligado a um computador via cabo *USB* [4] o *Arduino*, programado em linguagem Wiring (semelhante à linguagem *C++*), permite interação com o operador do experimento mediante uma janela de console. Ali são solicitados dados como massa da bola, altura da queda, entre outros, bem como mostra resultados. Uma rotina de calibragem calcula a constante de proporcionalidade entre o sinal elétrico e a força aplicada, gravando o parâmetro na memória *EEPROM*<sup>[2]</sup> do módulo e ali permanecendo mesmo com o sistema desligado. A listagem gerada permite o aluno elaborar o gráfico da força assim como comprovar as leis da mecânica clássica. A seguir serão abordados os requisitos para a compreensão desse trabalho

\*Endereço de correspondência: [joao.felipe@ipog.edu.br](mailto:joao.felipe@ipog.edu.br)

<sup>1</sup> Nome dado ao dispositivo

<sup>2</sup> Electrically-Erasable Programmable Read-Only Memory

## 2. Teoria da Conservação da Energia e Momento Linear

A energia mecânica  $E_m$  em um sistema compreende a energia cinética  $K$  e a energia potencial  $U$ , ou seja,

$$E_m = K + U \quad (1)$$

onde

$$K = \frac{mv^2}{2} \quad e \quad U = mgh \quad (2)$$

Em queda livre [5] sua energia potencial gravitacional se converte em cinética. Assim, igualando as Eq. (2) e desprezando as forças dissipativas no ar tem-se

$$v = \sqrt{2gh} \quad (3)$$

e o momento linear  $p = mv$  passa a ser

$$p = m\sqrt{2gh} \quad (4)$$

Durante a colisão a força de interação  $F(t)$  assume uma forma semelhante à exibida na Figura 1. A bola desce de uma altura  $h_1$  e sobe até uma altura menor  $h_2$ , devido a dissipação de energia no choque.

Nessa condição a bola recebe um impulso total  $I$  [6] do solo, que equivale à variação da quantidade de movimento  $m\Delta v$  [7], ou seja,

$$I = m(v_2 - v_1)$$

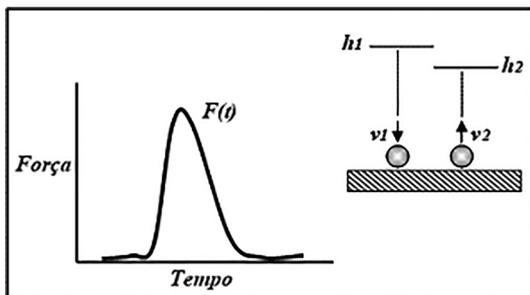
Considerando  $v_2$  positivo e  $v_1$  negativo tem-se

$$I = m(\sqrt{2gh_2} + \sqrt{2gh_1}) \quad (5)$$

O coeficiente de restituição da bola [7] é

$$e = \frac{p_2}{p_1} \quad (6)$$

onde  $p_2$  e  $p_1$  são respectivamente os momentos lineares da bola imediatamente antes e após o choque. Uma discussão



**Figura 1:** Perfil aproximado da força de colisão. O detalhe mostra, para uma queda livre, as velocidades inicial e final durante impacto.

mais detalhada sobre esse coeficiente pode ser vista em [8]. Substituindo a Eq. (4) na Eq. (6) [6]

$$e = \frac{\sqrt{h_2}}{\sqrt{h_1}} \quad (7)$$

A Eq. (5) fica então

$$I = m\sqrt{2g}(e\sqrt{h_1} + \sqrt{h_1}) \quad (8)$$

Esse impulso corresponde à área do gráfico  $F(t)$  na Figura 1, ou seja [9],

$$I = \int F(t)dt \quad (9)$$

Chamando essa área de  $A_F$  e fazendo uso da Eq. (8), já nomeando  $h_1$  de  $h$  tem-se

$$A_F = m\sqrt{2gh}(1 + e) \quad (10)$$

Considerando uma bola caindo de alturas diferentes  $m$  passa a ser constante e  $I$  na Eq. (8) dependerá apenas da altura  $h$ . Portanto

$$I = \alpha\sqrt{h} \quad (11)$$

onde

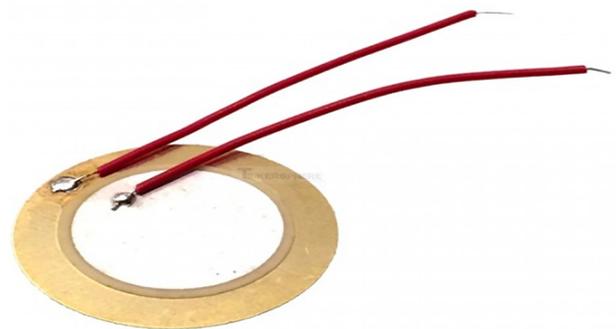
$$\alpha = m\sqrt{2g}(1 + e) \quad (12)$$

## 3. Sensor Piezoelétrico

Dentre os sensores de força o piezoelétrico figura entre os mais simples. Ele responde apenas a forças dinâmicas [10]. Mostrado na Figura 2, na forma de pastilha<sup>3</sup>, uma camada de cristal piezoelétrico é depositada na região central (branca) da lâmina. Essa região gera uma tensão elétrica quando pressionada.

A tensão elétrica gerada é proporcional à força aplicada [11]. Assim ela pode ser escrita como

$$V = \lambda F \quad (13)$$



**Figura 2:** Sensor Piezoelétrico.

<sup>3</sup> Há vários formatos para esse tipo de sensor. Uma diversidade de aplicações podem empregar esse dispositivo.

onde  $\lambda$  é uma constante de proporcionalidade e  $F$  é a força aplicada no sensor.

Esse tipo de sensor não exige manutenção alguma, não requer fonte de alimentação e é sensível a deformações nanométricas. Responde rápido às deformações na sua rede cristalina, permitindo trabalhar com frequências da ordem de dezenas de kHz.

Como praticamente não há atraso no sensor a Eq. (13) diz que os gráficos temporais de  $V$  e  $F$  são proporcionais. Quando a força atinge seu valor máximo a tensão gerada pelo cristal piezoelétrico também será máxima conforme ilustrado na Figura 3.

#### 4. Plataforma Arduino

Arduino, licenciado pela *Creative Commons*, é uma plataforma, de tecnologia aberta, para projetos eletrônicos de pequeno porte. É uma placa programável com um microcontrolador [12, 13] (*Atmega*) e várias portas (canais) [14, 15 e 4] para comunicar com outros dispositivos. Sua estrutura pode ser vista na Figura 4.

Os dados processados podem ser mostrados no seu console, na forma de uma janela na tela do computador. O diagrama da Figura 5 ilustra as fases de programação e operação desse dispositivo. Para programá-lo basta conectá-lo ao computador via cabo *USB* e executar o am-

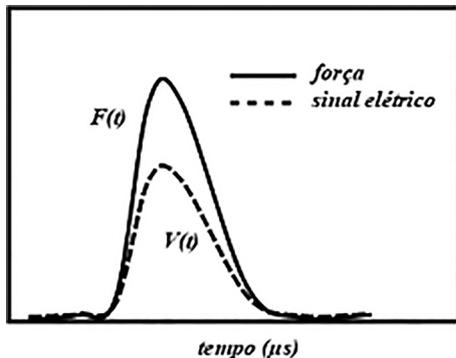


Figura 3: Perfis da força e da tensão elétrica gerada pelo sensor durante a colisão

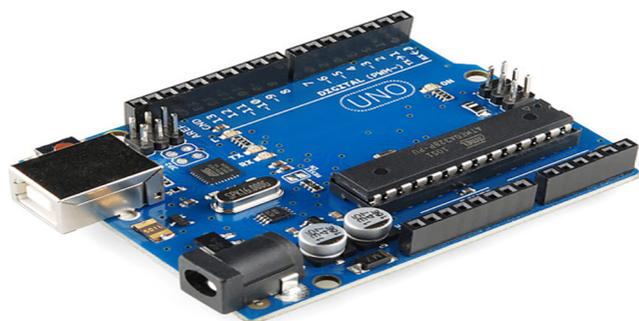


Figura 4: Arduino. As portas de comunicação estão disponíveis na forma de orifícios nas laterais da placa.

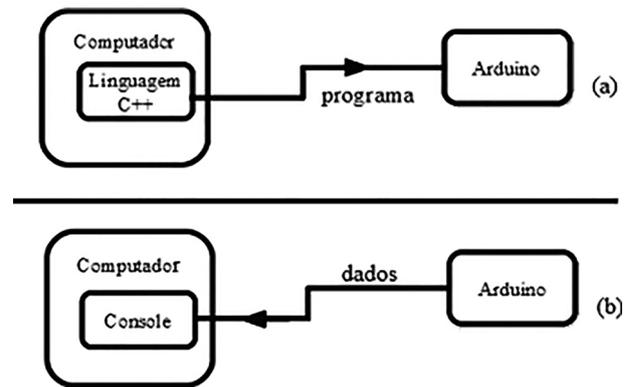


Figura 5: (a) Durante a programação o programa é transferido para o Arduino via cabo USB. (b) Em operação os dados processados podem ser exibidos numa janela de console

biente de desenvolvimento *IDE*<sup>4</sup> *Arduino*, fornecido pelo próprio fabricante. Esse *IDE* emprega a linguagem *Wiring*, baseada em *C++* [16], e é disponível na Internet [3]. Após ser programado o *Arduino* pode ser levado ao local onde irá funcionar, necessitando apenas de alimentação elétrica.

Quando um mecanismo é controlado por microcontrolador ele é dito *sistema embarcado* [14]. O módulo computacional é destinado a monitorar as propriedades do sistema bem como controlar seus mecanismos.

#### 5. Montagem

A Figura 6 ilustra o suporte usado para a queda da bola. Fabricado em metal, a barra horizontal possui altura ajustável, regulada conforme uma régua anexada à barra vertical. Vibrações na base, decorrentes da própria colisão, são transmitidas ao sensor [10], deformando o sinal gerado. Nesse sentido a base deve ser rígida, de massa considerável e estar bem assentada em superfície plana. A base aqui empregada, uma chapa de ferro, possui massa de 1,822 kg e dimensões 8,2 cm x 14,9 cm x 1,9 cm.

O circuito elétrico está esquematizado na Figura 7. A base do sensor é ligada à porta *GND* (terra) do *Arduino* e a região central é ligada à porta *A2* (porta analógica) [14]. Quando essa região é pressionada a tensão elétrica gerada é lida pela porta *A2*, que deve estar aterrada mediante um resistor *R*, conforme ilustra o circuito elétrico. Isso garante na porta uma tensão de referência  $V = 0$ . Caso contrário irá manifestar valores arbitrários de leitura, mesmo sem nenhum sinal do sensor. Ao mesmo tempo esse resistor protege a porta de altas correntes, uma vez que ele atua como um divisor de corrente [17].

Percebeu-se, mediante análise em osciloscópio, que o valor de *R* afeta a intensidade do sinal. Essa relação matemática não foi investigada. Para o tipo de sensor aqui empregado encontrou-se boa resposta para  $R = 10\text{ k}\Omega$ .

<sup>4</sup> IDE – Integrated Development Environment

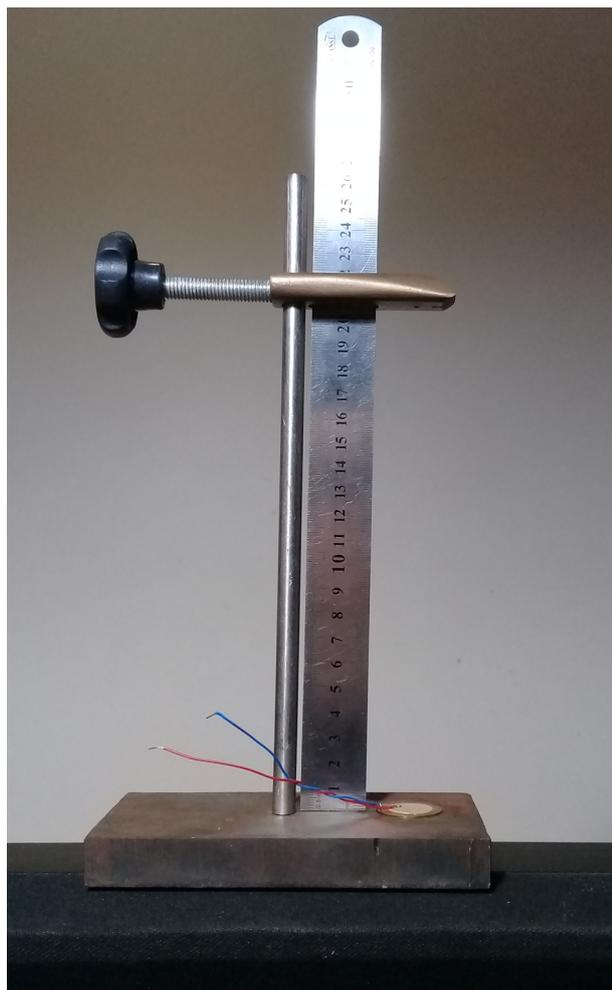


Figura 6: Suporte para a queda da bola.

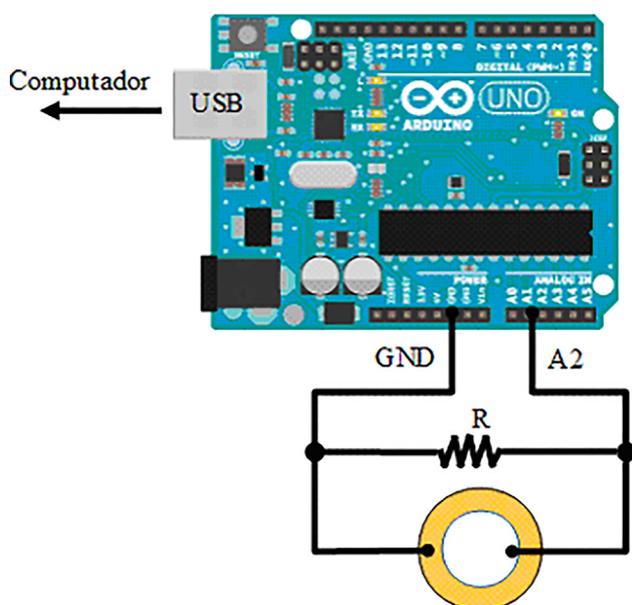


Figura 7: Circuito elétrico

## 6. Calibragem

Em funcionamento o sistema deve gerar uma tabela de forças a partir das tensões elétricas lidas pelo *Arduino*. A força é obtida pela Eq. (13), resultando assim em uma força para cada tensão. Esses dois conjuntos de dados, tensão e força, fornecem dois gráficos e duas áreas de integração, conforme mostrado na Figura 3. Outrossim a Eq. (13) permite calcular a constante que relaciona força e tensão elétrica.

$$\lambda = \frac{A_V}{A_F} \quad (14)$$

onde  $A_V$  é a área sob a curva no gráfico das tensões e  $A_F$  é a área sob a curva no gráfico das forças.

Calibrar o aparelho equivale a encontrar o valor de  $\lambda$ . O cálculo dessa constante faz uso da Eq. (14), onde a Eq. (10) fornece um valor teórico para  $A_F$ . Quanto à área  $A_V$  ela pode ser obtida por integração numérica na tabela de tensões, ou seja,

$$A_V = \sum_{i=0}^{n-1} V(t_i) \cdot (t_{i+1} - t_i) \quad (15)$$

Esses cálculos são realizados pelo *Arduino*. Portanto no processo de calibragem, segundo a Eq. (10), é necessário, antes de soltar a bola, informar a massa  $m$  da bola, a altura  $h$  da queda e o coeficiente de restituição  $e$  da bola.

Uma vez conhecido  $\lambda$ , a Eq. (13) converte o conjunto de tensões  $\{V\}$  no conjunto  $\{F\}$  de forças da seguinte forma

$$\{F\} = \frac{\{V\}}{\lambda} \quad (16)$$

## 7. Programação

Os cálculos executados no programa estão diagramados no fluxograma da Figura 8. A cada calibragem a constante  $\lambda$  é recalculada e gravada na memória *EEPROM* do *Arduino* [3], ali permanecendo, como já mencionado, mesmo após ele desligado. A partir daí, durante as execuções, o programa calcula a força de colisão mediante a Eq. (13), buscando essa constante na memória. Vale comentar que essa memória possui uma vida útil de 100.000 gravações.

Um material suplementar disponível no final desse documento contém a listagem completa do programa desenvolvido. Nesse material a rotina que faz a leitura do sinal denomina-se *loop()*, começando na linha 64. Cuidado especial foi tomado no primeiro comando *for()*. O bloco desse comando rege a taxa de amostragem do sistema. Quanto menos comandos ali houver menor será o tempo entre duas leituras consecutivas. Nesse sentido há apenas duas linhas de comando: uma para ler a tensão e outra para registrar o tempo.

A porta analógica do *Arduino* lê tensões de 0 a 5V numa escala de 0 a 1023. O bloco do segundo comando *for* faz a conversão da escala 0-1023 para a escala 0-5 [12].

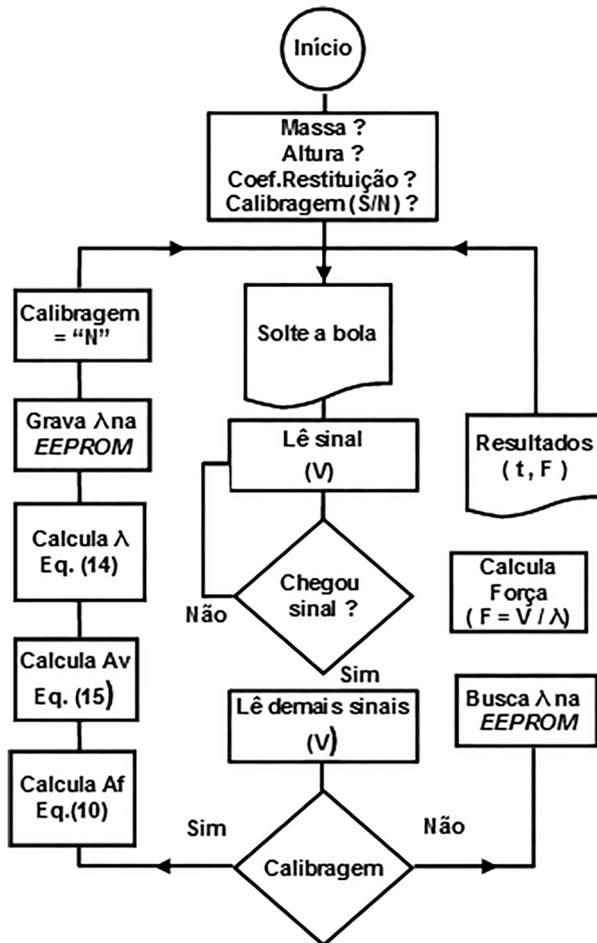


Figura 8: Fluxograma do programa no Arduino.

A rotina do programa que faz a calibragem, ou seja, calcula  $\lambda$ , denomina-se *calculaLambda()* e começa na linha 177. A chamada a essa rotina passa como parâmetros  $m$ ,  $h$  e  $e$  que são respectivamente a massa da bola, a altura da queda e o coeficiente de restituição. As equações contempladas nesses cálculos são acompanhadas pelo ramo esquerdo da Figura 8. A linha que calcula a variável  $\Delta t$  divide cada tempo por  $10^6$  (linha 190), uma vez que eles são registrados em  $\mu s$ .

Uma captura de tela do console, ilustrada na Figura 9, mostra a interação com o usuário. Os dados solicitados ao operador são digitados no campo ao lado do botão *Enviar*, dessa forma, não aparecendo na imagem.

### 8. Medição

O trabalho executou 5 colisões para 8 alturas diferentes que são 4 cm, 5 cm, 6 cm, 7 cm, 8 cm, 9 cm, 10 cm e 11 cm. Em cada colisão a força foi medida a cada 120  $\mu s$ , num total de quarenta medidas. A massa da bola é 9,21 g e o diâmetro 25,1 mm. Confeccionada em borracha e apresentando pouca perda de energia na colisão ela é vendida em máquinas de ficha em lanchonetes.

O coeficiente de restituição  $e$ , Eq. (7), da bola foi estimado medindo, na régua do próprio aparato (Figura 6), as alturas de descida e subida da bola em queda livre. Foi escolhido  $h_1 = 21$  cm e obtido  $h_2 = 17$  cm resultando em  $e = 0,9$ . Esse resultado não é o mesmo para alturas bem maiores. Um método mais preciso para determinar esse coeficiente pode ser obtido mediante [18].

A Figura 10 contém a listagem gerada pelo sistema após uma das colisões. É uma tabela *força x tempo*. Os três primeiros campos, *massa da bola*, *altura da queda*, e *coeficiente de restituição* são informados pelo usuário. O campo *Impulso* é mostrado para fins didáticos, visto que ele equivale à integral da força, conforme indica a Eq. (9). Essa integral é aqui calculada numericamente como

$$I = \sum_{i=0}^{n-1} F(t_i) \cdot (t_{i+1} - t_i) \tag{17}$$

onde  $t_i$  e  $F(t_i)$  correspondem respectivamente às colunas *Tempo* e *Força* na listagem. O fator *Lambda* (calibragem) é buscado na memória EEPROM.

Embora esse estudo considerasse apenas uma determinada bola na análise dos resultados, colisões com outras bolas de tamanhos e massas diferentes também foram efetuadas a título de ilustração. A Figura 11 mostra os gráficos de colisão para bolas diferentes. A tabela *tempo x força* foi copiada do console para o aplicativo *WinPython* [19] que gerou o gráfico. Na legenda  $m$  é a massa da bola e  $h$  a altura da queda. A curva onde  $m = 9,21$  g corresponde à listagem da Figura 10.

### 9. Discussão

O impulso  $I$  em cada colisão foi calculado e apresentado no gráfico da Figura 12. O cálculo segue a Eq. (17) e o resultado é indicado no cabeçalho da listagem de resultados (campo *Impulso* na Figura 10).

Como a massa da bola é constante o valor teórico do impulso é proporcional à altura da queda  $h$ , conforme aponta a Eq. (11). A Eq. (12) permite calcular o valor da constante de proporcionalidade, onde para  $m = 9,21$

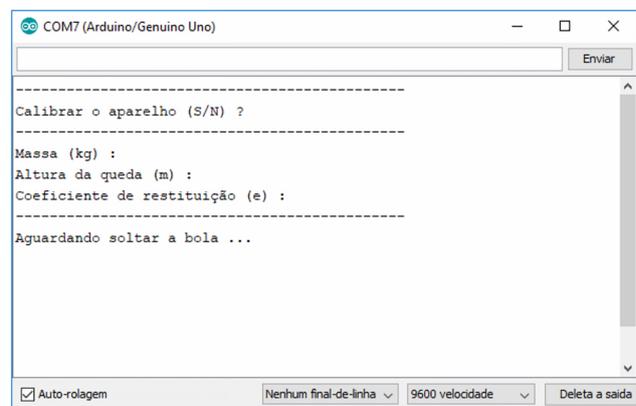


Figura 9: Console a caractere do Arduino.

$g$ ,  $g = 9,81 \text{ m/s}^2$  e  $e = 0,9$  tem-se

$$\alpha = m\sqrt{2g}(1 + e) = 0,0775$$

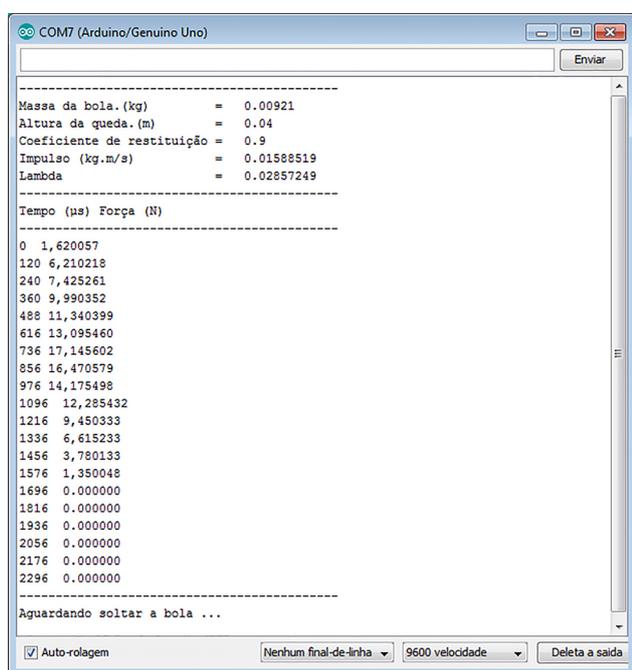
Substituindo esse valor na Eq. (11) chega-se à forma teórica de  $I$  como

$$I = 0,0775h^{0,5} \quad (18)$$

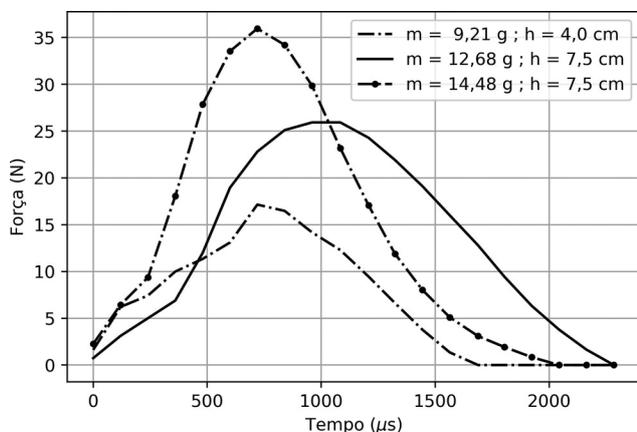
Como dito anteriormente foram efetuadas cinco colisões para cada uma das alturas estipuladas. Uma regressão potencial [20] foi realizada com esses valores e foi obtido

$$I = 0,0765h^{0,486} \quad (19)$$

As duas curvas, teórica e experimental, dadas respectivamente pelas Eq. (18) e (19), estão representadas no



**Figura 10:** Captura da tela do console listando o resultado de uma colisão.



**Figura 11:** Forças de colisão para três bolas distintas em material e tamanho.

gráfico da Figura 12. A comparação entre essas equações manifesta grande semelhança, onde o fator multiplicador e o expoente apresentam erros de 1,29% e 2,8% respectivamente. Essa exatidão é visível também no gráfico, mediante a proximidade entre as duas curvas.

## 10. Conclusão

O experimento permitiu descrever minuciosamente a força de interação durante a colisão elástica de uma bola, gerando uma listagem *tempo x força* com uma resolução de  $120 \mu\text{s}$ . Essa tabela permite o aluno elaborar gráficos bem como fazer análise numérica dos dados verificando as leis da mecânica clássica. O modelo oferece notável exatidão, respaldando o método de calibragem criado. A montagem é simples e os componentes, exceto o computador, são de baixo custo. O sensor piezoelétrico custou por volta de R\$ 1,00 e a plataforma Arduino, amplamente utilizada e com vasta literatura na Internet, custou aproximadamente R\$ 50,00. O suporte de liberação da bola ficou em 100,00, entre material e mão de obra. O algoritmo da calibragem implica conhecimentos de mecânica clássica e cálculo integral. Alunos podem programar a plataforma *Arduino* mediante outros algoritmos, modificando assim o experimento. O modelo pode ser estendido a outros fenômenos na física, bastando trocar o sensor e ajustar o programa de calibragem.

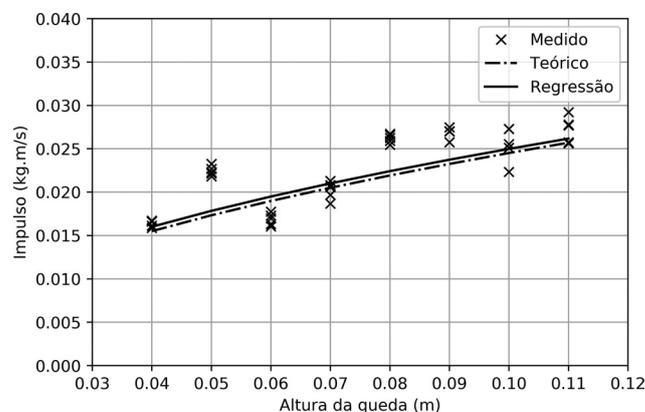
## Material Suplementar

O seguinte material suplementar está disponível online:

Material suplementar - Programa fonte desenvolvido para a plataforma Arduino.

## Referências

- [1] M.P. Norton e D.G. Karczub, *Fundamentals of Noise and Vibrations Analysis for Engineers* (Cambridge University Press, Cambridge, 2003).



**Figura 12:** Impulso recebido pela bola para quedas de 4 cm a 11 cm de altura.

- [2] D. Halliday, R. Resnick e J. Walker, *Fundamentos de Física* (LTC, Rio de Janeiro, 2014), 9ªed.
- [3] <https://www.arduino.cc>, acessado em dez. 2017.
- [4] M.A. Cavalcante, C.R.C. Tavoraro e E. Molisani, *Rev. Bras. Ensino Fís.* **33**, 4503 (2011).
- [5] V.R. Coluci, G. Paulino, D.C. Souza e E.P.R. Vasconcelos, *Rev. Bras. Ensino Fís.* **35**, 1 (2013).
- [6] F.J. Arnold, A.G. Godeny, M.M. Costa, J.F. Viana e R.L. Ximenes, *Rev. Bras. Ensino Fís.* **40**, e2502 (2018).
- [7] F.P. Beer, R.E. Johnston, D.F. Mazurek e E.R. Eisenberg, *Mecânica Vetorial para Engenheiros* (McGraw-Hill, São Paulo, 2001), 5ªed.
- [8] D.P. Meira Filho, J.K.S. Kamassury e R.C.S. Meira, *Rev. Bras. Ensino Fís.* **39**, e4302 (2017).
- [9] J. Santos, B.P.W. Oliveira e O.R. Nelson, *Rev. Bras. Ensino Fís.* **34**, 1 (2012).
- [10] A. Balbinot e V.J. Brusamarello, *Instrumentação e Fundamentos de Medidas* (LTC, Rio de Janeiro, 2011), 2ªed.
- [11] W. B. Lima, *Transdutores de deformação à base de sensores piezoelétricos*. Dissertação de Mestrado, Universidade Federal Fluminense, Rio de Janeiro (2013).
- [12] F. Pereira, *Microcontroladores PIC. Programação em C* (Érica, São Paulo, 2002).
- [13] S. Silveira e M. Girardi, *Rev. Bras. Ensino Fís.* **39**, e4502 (2017)
- [14] S. L. Stevan Junior e R.A. Silva, *Automação e instrumentação industrial com Arduino: teoria e projetos* (Érica, São Paulo, 2015).
- [15] L.A. Dworakowski, A.M. Hartmann, E.M. Kakuno e P.F.T. Dorneles, *Rev. Bras. Ensino Fís.* **38**, e3503 (2016).
- [16] S. Holzner, *Programando em C++: um guia prático para a programação profissional* (Campus, Rio de Janeiro, 1993).
- [17] R.C. Dorf e J.A. Svoboda, *Circuitos Elétricos* (LTC, Rio de Janeiro, 2008), 7ªed.
- [18] M.A. Cavalcante, E. Silva, R. Prado e R. Hagg, *Rev. Bras. Ensino Fís.* **24**, 150 (2002).
- [19] L.E. Borges, *Python Para Desenvolveres* (Novatec Editora, São Paulo, 2014).
- [20] R. Zentgraf, *Estatística Objetiva* (ZTG, Rio de Janeiro, 2001).