

Departamento da Indústria

Curso técnico em Eletrotécnica - P1

Lógica e linguagem de Programação

Relatório:

Bola de Ping-Pong

Alunos

Leonidas Leite

Marcos Da Silva

Fortaleza, junho de 2019

Apresentação

O seguinte código em linguagem C tem o objetivo de descrever o movimento de uma bola de PingPong, através da analise do gráfico da trajetória da bola e do gráfico de uma plataforma.

O gráfico possui os eixos relacionados a freqüência da bola pelo tempo, e conforme as três características físicas da bola, apresentarão variações na freqüência que podem ser analisadas quanto ao rendimento do material da bola em teste.

Relatório

No sistema do PingPong a bolinha *(no qual as características físicas/cinemáticas dessa bolinha é escolhida pela pessoa que vai roda o programa)* vai cai de um certa altura *(escolhida pela pessoa que vai roda o programa)* e bate em uma plataforma que se mexe em um movimento harmônico simples *(no qual a pessoa no programa também escolhe até que altura vai a plataforma e qual a velocidade angula desse movimento).*

No programa a primeira coisa a ser definida é as características física/cinemáticas da bolinha então a pessoa que vai roda o programa vai ter que escolhe ‘a’ se a bolinha se comporta elasticamente, ‘b’ se a bolinha se comporta elasticamente mas com amortecimento ou ‘c’ se a bolinha se comporta de forma inelástica, caso a bolinha se comporta elasticamente mas com amortecimento também é necessário escolhe o fator de amortecimento, caso seja elástica fator de amortecimento = 1 caso seja inelástica fator de amortecimento = 0.

Em seguidaa pessoa que vai roda o programa vai ter que escolhe a altura de qual a bolinha vai cai em metros, a altura máxima da plataforma em metros e a velocidade angular da plataforma em rad/s. Após todos os parâmetros fornecidos o programa vai fazer os cálculos e fornece uma planilha no excel com valores de tempo, das alturas da plataforma em função do tempo e com as alturas da bolinha em função do tempo.

Todos os valores de entradas pela a pessoa que vai roda o programa são testados e casos valores inadequados sejam colocados o programa terá que ser rodado novamente.

A plataforma se mexer num movimento harmônico simples então para calcular sua altura foi utilizada a seguinte equação :

**Altura plataforma = \*Cos( w\*Δt)**

A bolinha se mexer num movimento acelerado com aceleração igual a gravidade que foi considerada = 10m/s² então para calcular sua altura foi utilizada a seguintes equações:

**Altura2bolinha = + velocidade da bolinha1\*Δt +/-**

**velocidade da bolinha2= +/-gravidade\*Δt + velocidade da bolinha1**

o valor de + ou – da equação depende se a bolinha esta descendo ou subindo e tem uma variável dentro do programa para identificar isso.

Para calcular a conservação de energia da bolinha toda vida que ela colidia com o coma plataforma seguia as seguintes equações

energia cinetica2 = fator de amortecimento\* energia cinética1

**=**

Com essas formulas desenvolvidas foi variando o tempo com intervalos de 0,01 segundo e a cada 100 variações era guardado em vetores os valores do tempo, de altura da plataforma e de altura da bolinha no final dessas 100 variações.

Esse procedimento é realizado por 100 vezes assim é possível acompanhar a altura da bolinha e da por 100 segundos.

Por estar trabalhando com uma precisão de apenas 0,01 do tempo pode haver momentos em que a bolinha se ultrapassa a altura da plataforma ou sua altura máxima, nesses casos o programa identifica e iguala o valor da altura aos valores que foram ultrapassados.

Outra aproximação que foi realizada foi de zera a velocidade da bolinha a bate na plataforma se sua velocidade fosse menor ou igual a 3m/s, pois ela já não iria atingi altura relevantes então nesse caso o programa identificar e a parti desse momento a bolinha acompanha a plataforma.

No final é criado uma planilha chamada gráficos e todos os valores de altura e tempo armazenados nos vetores são colocados na planilha para que possa ser feito os gráficos.

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <locale.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char\*\* argv) {

setlocale(LC\_ALL, "Portuguese");

float hpmax,hbmax,hpc=0,hbc,v=0,v1,w,g=10,cose,k;

int a=0,b=0,x=0,i=0,l,tc=0;

float hp[101],t[101],hb[101];

char tb;

printf ("defina como se a bola se comporta : \n a = elastica \n b = elastica amortecido \n c = inelastica \n");

scanf("%c",&tb);

if(tb == 'a' || tb == 'b' || tb=='c'){

switch(tb){

case'a':

k=1;

break;

case'b':

printf ("defina o amortercimento da bola:\n ");

scanf("%f",&k);

break;

case'c':

k=0;

break;

}

printf ("Entre com valor de altura da bolinha: \n");

scanf ("%f",&hbmax);

printf ("Entre com valor de altura da plataforma: \n");

scanf ("%f", &hpmax);

printf ("Entre com valor da velocidade angular da plataforma em rad/s: \n");

scanf ("%f",&w);

if(hbmax<=0||hpmax<0|| hpmax>hbmax||k<0||k>1)

{

printf ("escolha não possivel rode o programa novamente");

}

else{

hbc=hbmax;

hp[i]=hpc;

hb[i]=hbc;

t[i]=tc;

while(tc!=100){

while(b!=100){

if(a==0){

b++;

hbc=(v\*0.01-(g\*0.0001)/2)+hbc;

v=-g\*0.01 + v;

cose = cos(w\*(tc+0.01\*b));

hpc=(hpmax/2)\*(1-cose);

if(hpc>=hbc){

hbc=hpc;

a=1;

v=(-1)\*v\*(sqrt(k));

if (v<=3){

hbc=hpc;

x=1;

}

}

}else{

b++;

hbc=(((v\*0.01))-(g\*0.0001)/2)+hbc;

v=-g\*0.01 + v;

cose = cos(w\*(tc+0.01\*b));

hpc=(hpmax/2)\*(1-cose);

if(v<=0){

v=0;

a=0;

if(hbc>hbmax){

hbc=hbmax;

}

}

if(x==1){

hbc=hpc;

}

}

}

b=0;

i++;

tc++;

hp[i]=hpc;

hb[i]=hbc;

t[i]=tc;

}

char url[]="graficos.xls";

//criando ponteiro

FILE \*arq;

// testando condição p abrir arquivo

arq = fopen(url, "w");

if(arq == NULL)

printf("Erro, nao foi possivel abrir o arquivo\n");

else{

// Escrevendo no arquivo

fprintf(arq, "\n\n",l);

fprintf(arq, "\t Tempo \t\t\ Altura da plataforma \t\t Altura da bolinha \n",l );

for(i=0;i<=100;i++) {

fprintf(arq, "\t %.2f \t\t %.2f\t\t %f \n",t[i], hp[i], hb[i]);

}

}

//fechando o arquivo

fclose(arq);

}

}

else{

printf ("escolha não possivel rode o programa novamente");

}

return 0;

}