**C++ 语 言 程 序 设 计**

实

验

报

告

大作业

姓名： 刘健恒

学号： SZ170320207

班级： 自动化二班

**一 实验项目**

* 1. 编写描述机械臂的类，实现平面3R机械臂的正逆运动学计算
  2. 基于Win32 GUI编程，实现平面3R机械臂的动态运动展示
  3. 大作业说明：
     1. 基本功能实现：20分，程序内设定目标位置后，展示出运动过程与结果
     2. 功能完整性：12分，按钮（开始，归位），参数输入框，信息输出框
     3. C++的使用程度，代码书写：8分
     4. 报告：10分
     5. 项目文件夹打包发送至songshuang@hit.edu.cn （使用回执功能，确保自己的邮件被收到），实验报告打印送至G311.
     6. 发送与提交时间：1月14日前

**二 实验原理**

1. 给出程序与算法的流程图与简要说明

开始 🡪 按钮选择 🡪 角度机械臂变换 🡪 求正解

变换

🡪 目标位置变换

求逆解

🡪 鼠标位置变换

🡪 归位

1. 程序实现与结果（给出源代码，说明关键代码的操作含义，给出运行结果）

**Main.cpp**

#include <windows.h>

#include<iostream>

#include<stdio.h>

#include <cmath>

#include <commctrl.h>

#include"resource.h"

#include"ARM\_SOLVER.h"

#define PI 3.1415926535

#define X 470

#define Y 450

using namespace std;

const char g\_szClassName[] = "myWindowClass";

int x,y,key;

int draw;

int flag;

HWND button1,button2,button3,button4,button5,button6,textBox1,textBox2,textBox3,textBox4,textBox5,textBox6,textBox7;

Arm\_solver arm(0,0,0);

LRESULT CALLBACK WndProc(HWND hwnd, UINT msg, WPARAM wParam, LPARAM lParam)

{

int nTextLength;

TCHAR sztextC[1024];

arm.setLength(100,90,60);

switch(msg)

{

case WM\_CREATE:

SetTimer(hwnd,1,200,NULL);

draw=1;

break;

case WM\_COMMAND:

if((HWND)lParam==button1)//判断消息是否来自按键

{

nTextLength=GetWindowTextLength(textBox1);//读取edit长度

GetWindowText(textBox1,sztextC,nTextLength+1);//读取edit text

arm.tangle1 = atof(sztextC);

arm.radian1 = arm.tangle1\*PI/180;

}

if((HWND)lParam==button2)//判断消息是否来自按键

{

nTextLength=GetWindowTextLength(textBox2);//读取edit长度

GetWindowText(textBox2,sztextC,nTextLength+1);

arm.tangle2 = atof(sztextC);

arm.radian2 = arm.tangle2\*PI/180;

}

if((HWND)lParam==button3)//判断消息是否来自按键

{

nTextLength=GetWindowTextLength(textBox3);//读取edit长度

GetWindowText(textBox3,sztextC,nTextLength+1);

arm.tangle3 = atof(sztextC);

arm.radian3 = arm.tangle3\*PI/180;

}

if((HWND)lParam==button4)//判断消息是否来自按键

{

nTextLength=GetWindowTextLength(textBox4);//读取edit长度

GetWindowText(textBox4,sztextC,nTextLength+1);

arm.xe = atof(sztextC);

nTextLength=GetWindowTextLength(textBox5);//读取edit长度

GetWindowText(textBox5,sztextC,nTextLength+1);

arm.ye = atof(sztextC);

nTextLength=GetWindowTextLength(textBox6);//读取edit长度

GetWindowText(textBox6,sztextC,nTextLength+1);

arm.tangleE = atof(sztextC);

flag = 1;

arm.solver(flag);

}

if((HWND)lParam==button5)//判断消息是否来自按键

{

nTextLength=GetWindowTextLength(textBox4);//读取edit长度

GetWindowText(textBox4,sztextC,nTextLength+1);

arm.xe = atof(sztextC);

nTextLength=GetWindowTextLength(textBox5);//读取edit长度

GetWindowText(textBox5,sztextC,nTextLength+1);

arm.ye = atof(sztextC);

nTextLength=GetWindowTextLength(textBox6);//读取edit长度

GetWindowText(textBox6,sztextC,nTextLength+1);

arm.tangleE = atof(sztextC);

arm.radianE = arm.tangleE\*PI/180;

flag = 2;

arm.solver(flag);

}

if((HWND)lParam==button6)//判断消息是否来自按键

{

arm.radian1 = 0;

arm.radian2 = 0;

arm.radian3 = 0;

}

break;

case WM\_CLOSE:

DestroyWindow(hwnd);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

case WM\_LBUTTONDOWN:

arm.xe=LOWORD(lParam)-X;

arm.ye=Y-HIWORD(lParam);

arm.radianE = atan2(arm.ye,arm.xe);

flag = 1;

arm.solver(flag);

break;

case WM\_TIMER:

if(draw)

{

static float radian1\_ind=0;

static float radian2\_ind=0;

static float radian3\_ind=0;

if (abs(radian1\_ind-arm.radian1)>1e-15)

{

radian1\_ind+=(arm.radian1-radian1\_ind)/3;

}

if (abs(radian2\_ind-arm.radian2)>1e-15)

{

radian2\_ind+=(arm.radian2-radian2\_ind)/3;

}

if (abs(radian3\_ind-arm.radian3)>1e-15)

{

radian3\_ind+=(arm.radian3-radian3\_ind)/3;

}

HDC hdc=GetDC(hwnd);

HBRUSH hBrush;

RECT rect;

SetRect(&rect, 200, 0,800,800);

hBrush = CreateSolidBrush(RGB(112, 112, 112));

FillRect(hdc, &rect, hBrush);

hBrush = NULL;

MoveToEx(hdc,X,Y,NULL);

arm.location\_x = X;

arm.location\_y = Y;

hBrush = CreateSolidBrush(RGB(200,200,200));

SelectObject(hdc, hBrush);

HPEN pen = CreatePen(PS\_SOLID,5,RGB(200,200,200));

HPEN old = (HPEN) SelectObject(hdc,pen);

arm.tangleDraw(hdc,arm.location\_x,arm.location\_y,radian1\_ind,arm.length1);

hBrush = CreateSolidBrush(RGB(164,220,236));

SelectObject(hdc, hBrush);

pen = CreatePen(PS\_SOLID,5,RGB(164,220,236));

old = (HPEN) SelectObject(hdc,pen);

arm.tangleDraw(hdc,arm.location\_x,arm.location\_y,radian2\_ind,arm.length2);

hBrush = CreateSolidBrush(RGB(255,255,255));

SelectObject(hdc,hBrush);

pen = CreatePen(PS\_SOLID,5,RGB(255,255,255));

old = (HPEN) SelectObject(hdc,pen);

arm.tangleDraw(hdc,arm.location\_x,arm.location\_y,radian3\_ind,arm.length3);

ReleaseDC(hwnd,hdc);

}

break;

default:

return DefWindowProc(hwnd, msg, wParam, lParam);

}

return 0;

}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,

LPSTR lpCmdLine, int nCmdShow)

{

WNDCLASSEX wc;//注冊我们的窗口类的代码

HWND hwnd;//创建窗口

MSG Msg;

wc.cbSize = sizeof(WNDCLASSEX);//结构体的大小

wc.style = 0;

wc.lpfnWndProc = WndProc;

wc.cbClsExtra = 0;

wc.cbWndExtra = 0;

wc.hInstance = hInstance;//应用程序实例的句柄

wc.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wc.hCursor = LoadCursor(NULL, IDC\_ARROW);

wc.hbrBackground = (HBRUSH)(COLOR\_WINDOW+1);

wc.lpszMenuName = NULL;

wc.lpszClassName = g\_szClassName;//类的名字

wc.hIconSm = LoadIcon(NULL, IDI\_APPLICATION);

if(!RegisterClassEx(&wc))

{

MessageBox(NULL, "Window Registration Failed!", "Error!",

MB\_ICONEXCLAMATION | MB\_OK);

return 0;

}

hwnd = CreateWindowEx(

WS\_EX\_CLIENTEDGE,

g\_szClassName, "3R armMove",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT, 800, 800, NULL, NULL, hInstance, NULL);

if(hwnd == NULL)

{

MessageBox(NULL, "Window Creation Failed!", "Error!",

MB\_ICONEXCLAMATION | MB\_OK);

return 0;

}

ShowWindow(hwnd, nCmdShow);

UpdateWindow(hwnd);

button1=CreateWindow("BUTTON","关节角1(角度)",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|BS\_PUSHBUTTON,10,50,100,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建按键

button2=CreateWindow("BUTTON","关节角2(角度)",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|BS\_PUSHBUTTON,10,140,100,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建按键

button3=CreateWindow("BUTTON","关节角3(角度)",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|BS\_PUSHBUTTON,10,230,100,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建按键

button4=CreateWindow("BUTTON","目标(x,y,φ)-解一",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|BS\_PUSHBUTTON,10,320,150,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建按键

button5=CreateWindow("BUTTON","目标(x,y,φ)-解二",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|BS\_PUSHBUTTON,10,360,150,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建按键

button6=CreateWindow("BUTTON","归位",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|BS\_PUSHBUTTON,10,400,50,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建按键

textBox1=CreateWindow("EDIT","0",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,10,10,100,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框 textBox2=CreateWindow("EDIT","0",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,10,100,100,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框 textBox3=CreateWindow("EDIT","0",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,10,190,100,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框 textBox4=CreateWindow("EDIT","0",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,10,280,30,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框 textBox5=CreateWindow("EDIT","0",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,60,280,30,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框 textBox6=CreateWindow("EDIT","0",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,110,280,30,30,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框

textBox7=CreateWindow("EDIT","可实现功能:\t1.输入机械臂角度进行变换\t2.输入目标坐标及角度进行变换\t3.机械臂运动到鼠标点击位置",WS\_VISIBLE|WS\_CHILD|WS\_BORDER|ES\_MULTILINE|ES\_WANTRETURN,10,440,140,100,hwnd,NULL,(HINSTANCE) GetWindowLong(hwnd, GWL\_HINSTANCE),NULL);//创建文本框

while(GetMessage(&Msg, NULL, 0, 0) > 0)

{

TranslateMessage(&Msg);

DispatchMessage(&Msg);

}

return Msg.wParam;

}

**ARM\_SOLVER.h**

#ifndef ARM\_SOLVER\_H

#define ARM\_SOLVER\_H

#include <windows.h>

#include<iostream>

#include<stdio.h>

#include <cmath>

#include <commctrl.h>

#include"resource.h"

#define PI 3.1415926535

#define X 470

#define Y 450

class Arm\_solver

{

private:

float cosB = 0;

float sinB = 0;

float xc = 0;

float yc = 0;

float radianA;

float radianV;

public:

float tangle1=0,tangle2=0,tangle3=0,radian1=0,radian2=0,radian3=0;

float xe = 0;

float ye = 0;

float tangleE = 0;

float radianE = 0;

float location\_x = X;

float location\_y = Y;

float length1 = 100;

float length2 = 90;

float length3 = 60;

Arm\_solver(float xe\_t,float ye\_t,float tangleE\_t);

void solver(int flag);

void setLength(float length1\_t,float length2\_t,float length3\_t);

void tangleDraw(HDC hdc, float& location\_x, float& location\_y, float radian, float length\_temp);

};

#endif

**ARM\_SOLVER.cpp**

#include <windows.h>

#include<iostream>

#include<stdio.h>

#include <cmath>

#include <commctrl.h>

#include"resource.h"

#include"ARM\_SOLVER.h"

#define PI 3.1415926535

Arm\_solver::Arm\_solver(float xe\_t,float ye\_t,float tangleE\_t)

{

xe = xe\_t;

ye =ye\_t;

tangleE = tangleE\_t;

radianE = tangleE\*PI/180;

}

void Arm\_solver::solver(int flag)

{

if(flag == 1)

{

xc = xe - length3\*cos(radianE);

yc = ye - length3\*sin(radianE);

cosB =(pow(length1,2)+pow(length2,2)-pow(xc,2)-pow(yc,2))/(2\*length1\*length2);

sinB = sqrt(1-pow(cosB,2));

if((length1+length2) >= sqrt(pow(xc,2)+pow(yc,2)))

{

radian2 = PI - atan2(sinB,cosB);

radianA = atan2(yc,xc);

radianV = atan2(length2\*sin(radian2), length1 + length2\*cos(radian2));

radian1 = radianA - radianV;

radian3 = radianE - radian1 - radian2;

radian2 += radian1;

radian3 += radian2;

}

else

{

MessageBox(NULL,"无解！","提示",MB\_OK);

}

}

if(flag == 2)

{

xc = xe - length3\*cos(radianE);

yc = ye - length3\*sin(radianE);

cosB =(pow(length1,2)+pow(length2,2)-pow(xc,2)-pow(yc,2))/(2\*length1\*length2);

sinB = -sqrt(1-pow(cosB,2));

if((length1+length2) >= sqrt(pow(xc,2)+pow(yc,2)))

{

radian2 = PI - atan2(sinB,cosB);

radianA = atan2(yc,xc);

radianV = atan2(length2\*sin(radian2), length1 + length2\*cos(radian2));

radian1 = radianA - radianV;

radian3 = radianE - radian1 - radian2;

radian2 += radian1;

radian3 += radian2;

}

else

{

MessageBox(NULL,"无解！","提示",MB\_OK);

}

}

}

void Arm\_solver::setLength(float length1\_t,float length2\_t,float length3\_t)

{

length1 = length1\_t;

length2 = length2\_t;

length3 = length3\_t;

}

void Arm\_solver::tangleDraw(HDC hdc, float& location\_x, float& location\_y, float radian, float length\_temp)

{

Ellipse(hdc, location\_x-10,location\_y-10,location\_x+10,location\_y+10);

location\_x += 10\*cos(radian+PI/2);

location\_y -= 10\*sin(radian+PI/2);

LineTo(hdc,location\_x,location\_y);

location\_x += length\_temp\*cos(radian);

location\_y -= length\_temp\*sin(radian);

LineTo(hdc,location\_x,location\_y);

location\_x += 20\*cos(radian-PI/2);

location\_y -= 20\*sin(radian-PI/2);

LineTo(hdc,location\_x,location\_y);

location\_x += length\_temp\*cos(radian-PI);

location\_y -= length\_temp\*sin(radian-PI);

LineTo(hdc,location\_x,location\_y);

location\_x += 10\*cos(radian-3\*PI/2);

location\_y -= 10\*sin(radian-3\*PI/2);

LineTo(hdc,location\_x,location\_y);

location\_x += length\_temp\*cos(radian);

location\_y -= length\_temp\*sin(radian);

MoveToEx(hdc,location\_x,location\_y,NULL);

}

**三 实验总结与建议**

（总结实验实施过程，说明实验过程中遇到的问题与解决方案；提出实验环节的建议）

本次大作业学习了win32编程，并综合运用了学习的C++知识，进行融会贯通，受益匪浅。