

Ứng dụng MFC (Visual C++) trong mô phỏng Robot và hệ Cơ điện tử

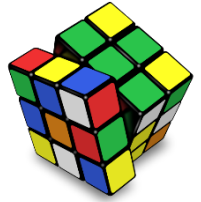


Bài 4: Mô phỏng Robot với đồ họa MFC

PHẠM MINH QUÂN

mquan.ph@gmail.com

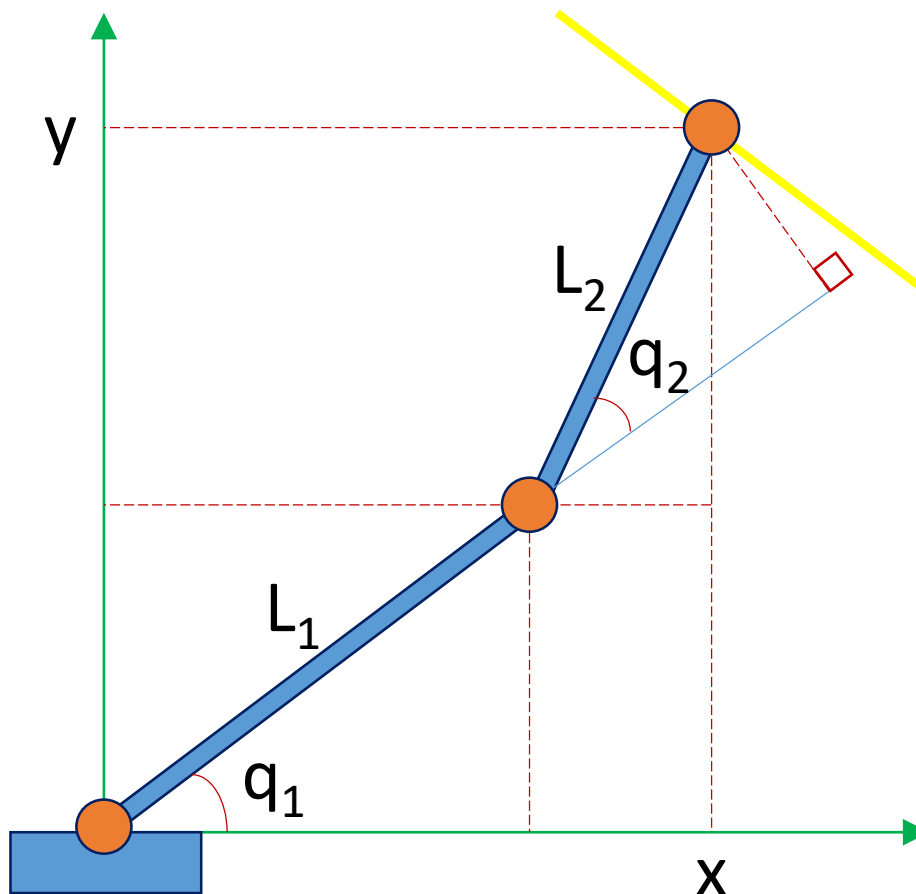
Nội dung



1. Cấu trúc Robot chuỗi 2 khâu đơn giản
2. Thực hành mô phỏng Robot với đồ họa MFC
 - 2.1. *Xây dựng lớp quản lý robot*
 - 2.2. *Xây dựng lớp quản lý quỹ đạo*
 - 2.3. *Mô phỏng Robot trong hộp thoại MFC*
 - 2.4. *Thêm các control của MFC để hoàn thiện ứng dụng*

1. Cấu trúc một Robot chuỗi 2 khớp quay

- Động học thuận và động học ngược



□ Động học thuận: $(q_1, q_2) \rightarrow (x, y)$

$$x = L_1 \cos(q_1) + L_2 \cos(q_1 + q_2)$$

$$y = L_1 \sin(q_1) + L_2 \sin(q_1 + q_2)$$

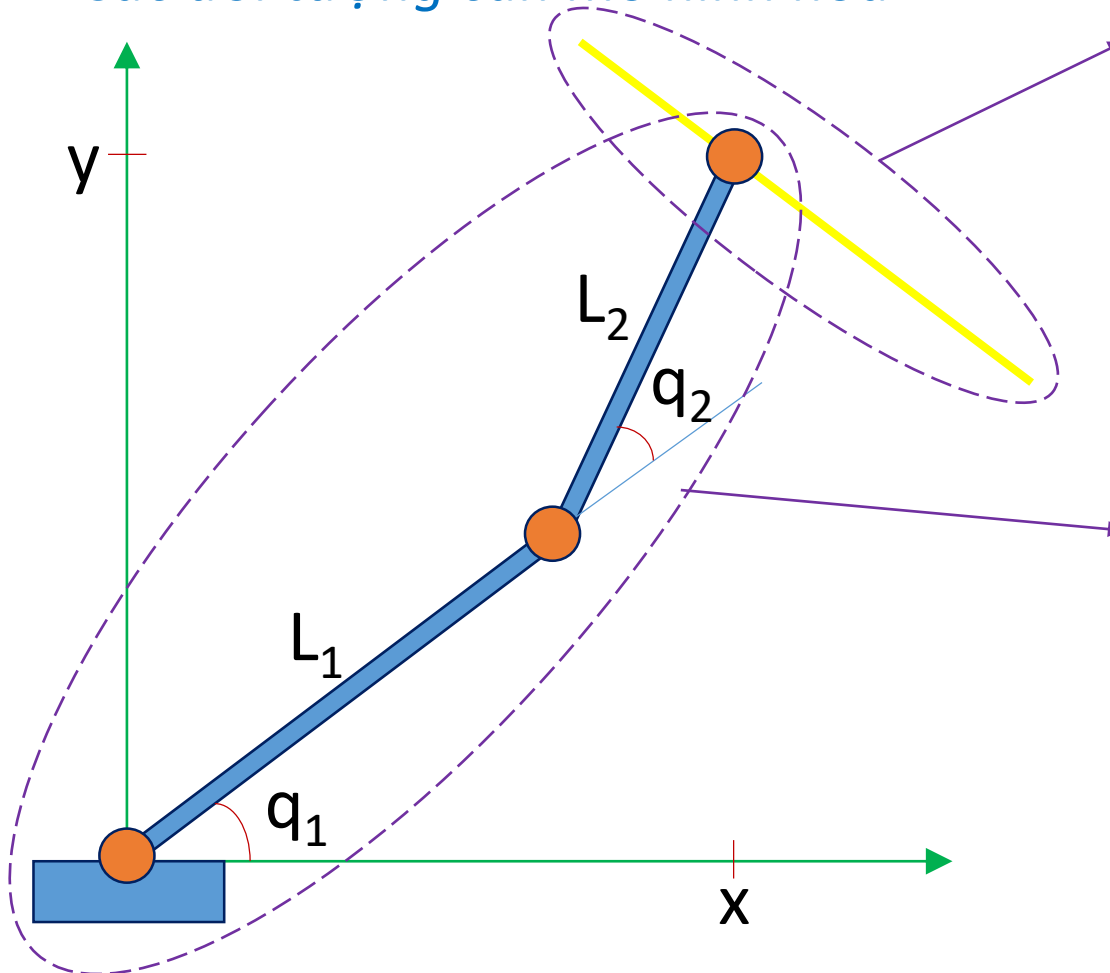
□ Động học ngược: $(x, y) \rightarrow (q_1, q_2)$

$$q_2 = \arccos \left(\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1 L_2} \right)$$

$$q_1 = \operatorname{atan2}(y, x) - \operatorname{atan2}(L_2 \sin q_2, L_1 + L_2 \cos q_2)$$

1. Cấu trúc một Robot chuỗi 2 khớp quay

• Các đối tượng cần mô hình hóa



□ Quỹ đạo:

▪ Thuộc tính:

- Dạng hình học: đoạn thẳng, cung tròn...
- Tọa độ các điểm nút, thông số hình học
- Thông số luật chuyển động ...

▪ Phương thức:

- Nhập, xuất dữ liệu
- Tính toán quỹ đạo
- ...

□ Robot:

▪ Thuộc tính:

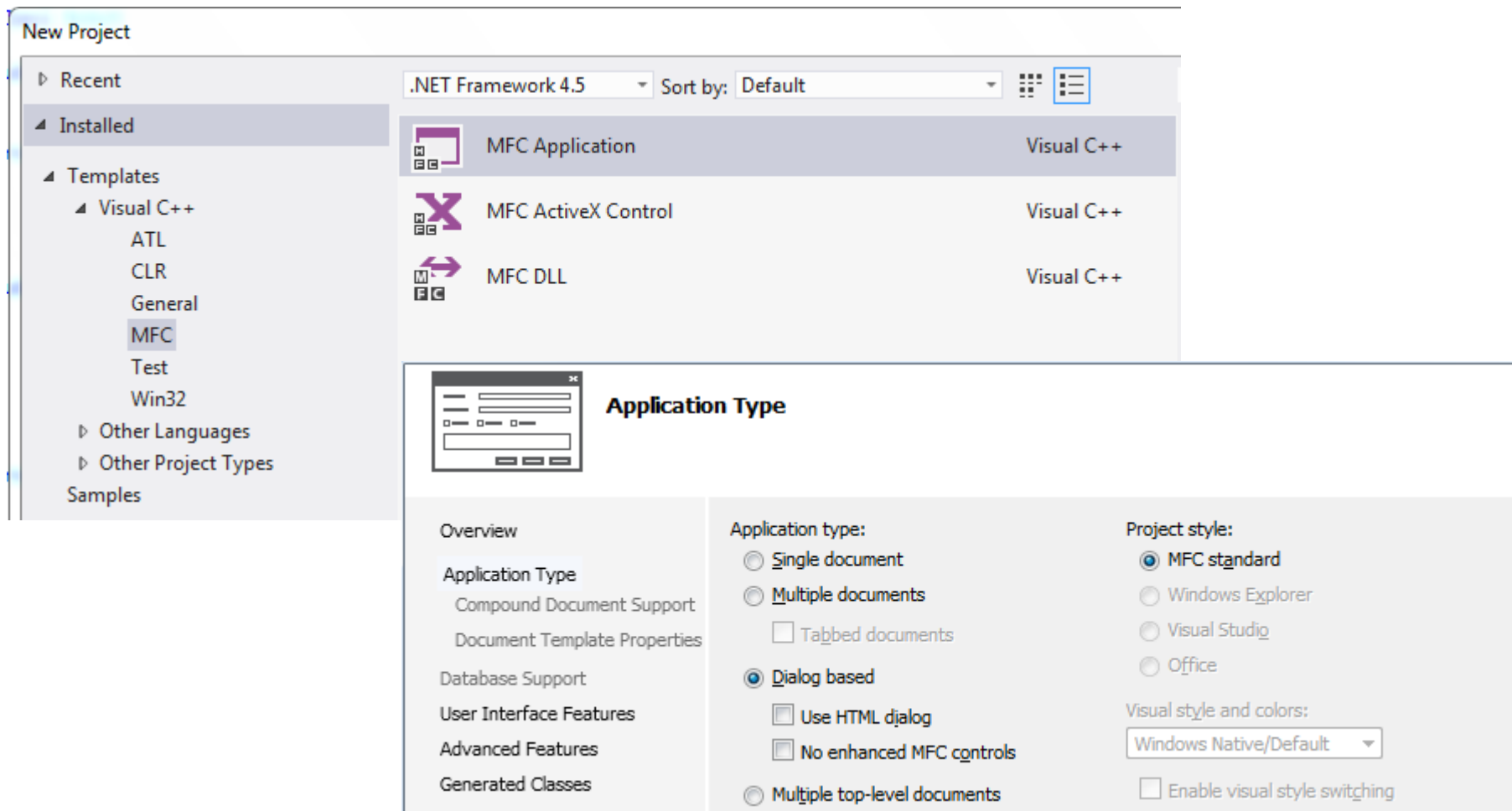
- Độ dài khâu: L_1 , L_2
- Góc khớp: q_1 , q_2
- Tọa độ điểm tác động cuối x , y

▪ Phương thức:

- Nhập, xuất dữ liệu
- Tính động học thuận
- Tính động học ngược
- ...

2. Thực hành mô phỏng Robot với đồ họa MFC

❑ Tạo project mới với kiểu ứng dụng MFC, Dialog-based



2. Thực hành mô phỏng Robot với đồ họa MFC

2.1. Xây dựng lớp quản lý Robot

- Tạo lớp mới để quản lý đối tượng Robot //Cách tạo lớp mới như hướng dẫn trong Bài 2
- Khai báo và định nghĩa lớp mới (lớp Robot) như sau

File Robot.h

```
#pragma once
class Robot
{
public:
    Robot(void);
    ~Robot(void);
protected:
    double l1,l2;
    double q1,q2;
    double x,y;

public:
    void Setconfig(double par1, double par2);
    void Setjoints(double par1, double par2);
    void Setpos(double par1, double par2);
    double Getx(); double Gety();
    double Getq1(); double Getq2();
    void DrawRobot(CDC *pDC, int zoom, COLORREF crColor, int x_0, int y_0);
protected:
    void FwdKin();
    void InvKin();
};
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.1. Xây dựng lớp quản lý Robot

- Khai báo và định nghĩa lớp mới (lớp Robot) như sau

File Robot.cpp

```
#include "stdafx.h"
#include "Robot.h"
#include "math.h"

Robot::Robot(void)
{
    l1=0.25;
    l2=0.15;
    q1=0; q2=0;
    FwdKin();
}

Robot::~Robot(void)
{
}

void Robot::FwdKin()
{
    x=l1*cos(q1)+l2*cos(q1+q2);
    y=l1*sin(q1)+l2*sin(q1+q2);
}

void Robot::InvKin()
{
    q2=acos((x*x+y*y-l1*l1-l2*l2)/(2*l1*l2));
    q1=atan2(y,x)-atan2(l2*sin(q2),l1+l2*cos(q2));
}

...

void Robot::Setconfig(double par1, double par2)
{
    l1=par1; l2=par2;
    FwdKin();
}

void Robot::Setjoints(double par1, double par2)
{
    q1=par1; q2=par2;
    FwdKin();
}

void Robot::Setpos(double par1, double par2)
{
    x=par1; y=par2;
    InvKin();
}

double Robot::Getx() { return x; }
double Robot::Gety() { return y; }
double Robot::Getq1() { return q1; }
double Robot::Getq2() { return q2; }

...
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.1. Xây dựng lớp quản lý Robot

- Khai báo và định nghĩa lớp mới (lớp Robot) như sau

File Robot.cpp

```
...
void Robot::DrawRobot(CDC *pDC, int zoom, COLORREF crColor, int x_0, int y_0)
{
    pDC->SetBkMode(TRANSPARENT);
    CPen *pPen=new CPen(PS_SOLID,3,crColor);
    CBrush *pBrush=new CBrush(crColor);
    pDC->SelectObject(pPen);
    pDC->SelectObject(pBrush);

    pDC->MoveTo(x_0,y_0);
    int x_j = x_0+l1*cos(q1)*zoom; int y_j = y_0-l1*sin(q1)*zoom;
    pDC->LineTo(x_j,y_j);
    int x_e = x_0+x*zoom; int y_e = y_0-y*zoom;
    pDC->LineTo(x_e,y_e);

    pDC->Ellipse(x_0-8,y_0-8,x_0+8,y_0+8);
    pDC->Ellipse(x_j-8,y_j-8,x_j+8,y_j+8);
    pDC->Ellipse(x_e-8,y_e-8,x_e+8,y_e+8);
    CRect rect(x_0-20, y_0, x_0+20, y_0+20);
    pDC->FrameRect(rect,pBrush);

    delete pDC->SelectStockObject(WHITE_PEN);
    delete pDC->SelectStockObject(WHITE_BRUSH);
}
```

// Vẽ các khâu

// Vẽ các khớp

// Vẽ đế robot

(Lưu ý: trục Y đồ họa hướng xuống dưới, ngược với trục Y làm việc của robot, nên các tọa độ liên quan đến Y cần điều chỉnh bằng dấu "-")

2. Thực hành mô phỏng Robot với đồ họa MFC

2.2. Xây dựng lớp quản lý Quỹ đạo

- Tạo lớp mới để quản lý đối tượng Quỹ đạo
- Khai báo và định nghĩa lớp mới (lớp Trajectory) như sau

File Trajectory.h

```
#pragma once
class Trajectory
{
public:
    Trajectory(void);
    ~Trajectory(void);
protected:
    int type;
    double x0,y0,x1,y1;
    double cex,cey,ra;
    double speed;
    double t, x, y;
public:
    void SetLine(double px0, double py0, double px1, double py1, double spd);
    void SetCircle(double px, double py, double pra, double spd);
    void SetType(int tp);
    void SetTime(double time);
    double Getx(); double Gety();
    void DrawTrajectory(CDC *pDC, int zoom, COLORREF crColor, int x_0, int y_0);
protected:
    void ComputePos();
};
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.2. Xây dựng lớp quản lý Quỹ đạo

- Khai báo và định nghĩa lớp mới (lớp Trajectory) như sau

File Trajectory.cpp

```
#include "stdafx.h"
#include "Trajectory.h"
#include "math.h"

Trajectory::Trajectory(void)
{
    type = 1;
    x0=0.3; y0=-0.1; x1=0; y1=0.2;
    cex=0.2; cey=0.15; ra=0.1;
    speed=1;
    t=0;
    ComputePos();
}

Trajectory::~Trajectory(void)
{}

void Trajectory::SetLine(double px0, double py0,
                        double px1, double py1, double spd)
{
    x0=px0; y0=py0; x1=px1; y1=py1; speed=spd;
}

void Trajectory::SetCircle(double px, double py, double pra, double spd)
{
    cex=px; cey=py; ra=pra; speed=spd;
}

...

void Trajectory::SetType(int tp)
{
    type=tp;
}

void Trajectory::SetTime(double time)
{
    t=time;
    ComputePos();
}

double Trajectory::GetX() { return x; }
double Trajectory::GetY() { return y; }

...
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.2. Xây dựng lớp quản lý Quỹ đạo

- Khai báo và định nghĩa lớp mới (lớp Trajectory) như sau

File Trajectory.cpp

```
...  
  
void Trajectory::ComputePos()  
{  
    float u=(sin(speed*t)+1)/2;  
    switch (type)  
    {  
        case 1:  
            x=x0+u*(x1-x0);  
            y=y0+u*(y1-y0);  
            break;  
        case 2:  
            float phi = u*2*3.14;  
            x=cex+ra*cos(phi);  
            y=cey+ra*sin(phi);  
            break;  
    }  
}
```

...

```
...  
  
void Trajectory::DrawTrajectory(CDC *pDC, int zoom,  
                                COLORREF crColor, int x_0, int y_0)  
{  
    pDC->SetBkMode(TRANSPARENT);  
    CPen *pPen=new CPen(PS_SOLID,1,crColor);  
    pDC->SelectObject(pPen);  
  
    switch (type)  
    {  
        case 1:  
            pDC->MoveTo(x_0+x0*zoom,y_0-y0*zoom);  
            pDC->LineTo(x_0+x1*zoom,y_0-y1*zoom);  
            break;  
        case 2:  
            pDC->MoveTo(x_0+(cex+ra)*zoom,y_0-cey*zoom);  
            pDC->AngleArc(x_0+cex*zoom,y_0-cey*zoom,ra*zoom,0,360);  
            break;  
    }  
  
    delete pDC->SelectStockObject(WHITE_PEN);  
}
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.3. Mô phỏng Robot trong hộp thoại MFC

- “include” file tiêu đề của các lớp mới tạo trong file ...Dlg.h

```
#include "Robot.h"  
#include "Trajectory.h"
```

- Khai báo một số biến mới trong file ...Dlg.h

```
Robot rb1;  
Trajectory trj1;  
  
float t;  
int x_0, y_0;  
int zoom_scale;  
COLORREF Robot_color, Traj_color;
```

- Khai báo và định nghĩa hàm DrawScreen(...)

```
void DrawScreen(CDC *pDC, int xleft, int ytop, int xright, int ybottom);
```

// File ...Dlg.h

```
void CMFCRobot2DSimDlg::DrawScreen(CDC *pDC, int xleft, int ytop, int xright, int ybottom)  
{  
    CBrush *pBrush=new CBrush(RED);  
    pDC->SelectObject(pBrush);  
    CRect rect(xleft,ytop,xright,ybottom);  
    pDC->FillRect(&rect,pBrush);  
    delete pDC->SelectStockObject(WHITE_BRUSH);  
}
```

// File ...Dlg.cpp

2. Thực hành mô phỏng Robot với đồ họa MFC

2.3. Mô phỏng Robot trong hộp thoại MFC

- Khởi tạo giá trị cho các biến mới

```
// TODO: Add extra initialization here // File ...Dlg.cpp, hàm OnInitDialog()
rb1.Setconfig(0.25,0.15);
trj1.SetType(2);
t = 0;
x_0 = 200;
y_0 = 200;
Robot_color = RGB(255,0,0);
Traj_color = RGB(255,255,0);
zoom_scale = 500;
```

- Thêm hàm OnTimer() cho lớp C...Dlg (như hướng dẫn trong Bài 1)
- Viết nội dung cho hàm OnTimer() như sau

```
void CMFCRobot2DSimDlg::OnTimer(UINT_PTR nIDEvent) // File ...Dlg.cpp
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    DrawScreen(&dc, 20,20,x_0*2-20,y_0*2-20);
    trj1.DrawTrajectory(&dc, zoom_scale, Traj_color, x_0, y_0);
    trj1.SetTime(t);
    rb1.Setpos(trj1.Getx(), trj1.Gety());
    rb1.DrawRobot(&dc, zoom_scale, Robot_color, x_0, y_0);
    t+=0.1;

    CDialogEx::OnTimer(nIDEvent);
}
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.3. Mô phỏng Robot trong hộp thoại MFC

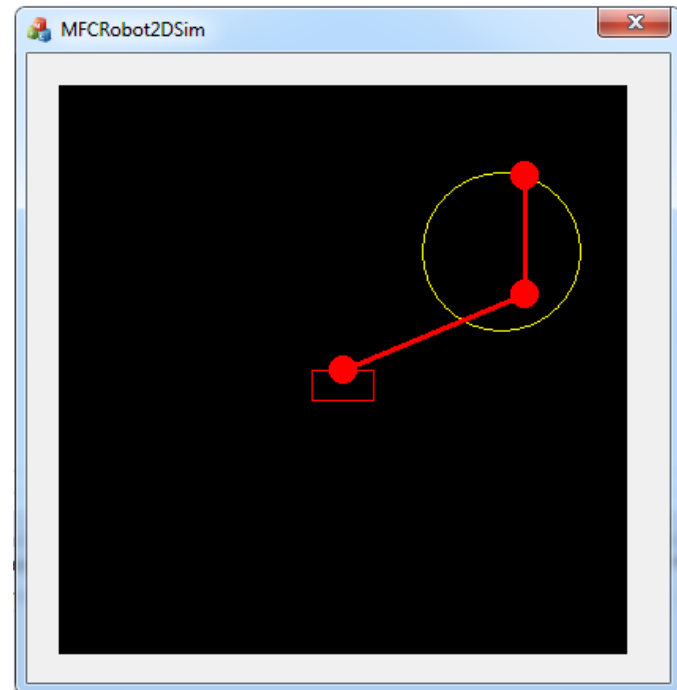
➤ Thiết lập Timer

```
| SetTimer(1,100,NULL); // File ...Dlg.cpp, hàm OnInitDialog()
```



Kết quả chạy:

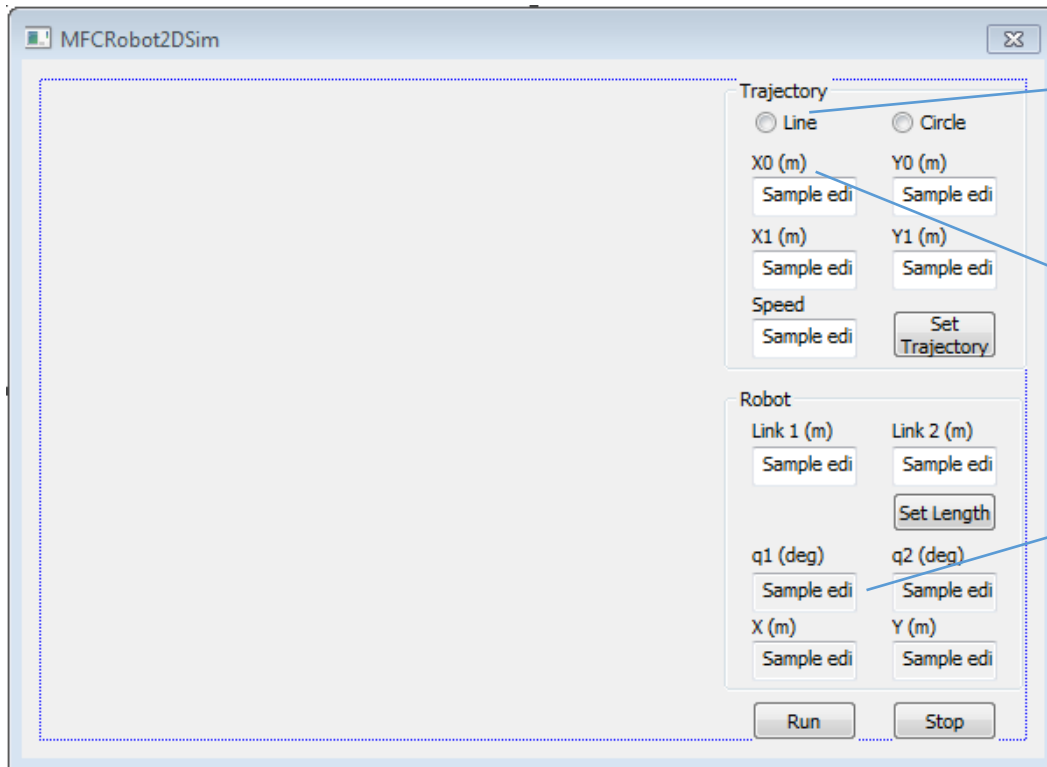
*Điểm tác động cuối của Robot
di chuyển theo quỹ đạo*



2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Thêm các điều khiển vào dialog và thiết lập đặc tính (caption, ID...) cho phù hợp



// Chú ý một số đặc tính sau đây:

- Thiết lập đặc tính Group của Radio Button đầu tiên thành true

Misc	
(Name)	IDC_RADIO_LINE (Radio-
Group	True
ID	IDC_RADIO_LINE

- Đặt ID cho các Static Text X0, Y0, X1, Y1 để gọi tới sau này

Group	True
ID	IDC_STATIC_X0
Tabstop	False

- Thiết lập đặc tính Read Only của các Edit Box Q1, Q2, X, Y thành true

Behavior	
Accept Files	False
Disabled	False
Help ID	False
Multiline	False
No Hide Selection	False
OEM Convert	False
Password	False
Read Only	True
Visible	True

// Có thể dùng các công cụ để căn chỉnh kích thước, vị trí control:



2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Thêm các biến để quản lý các control

File ...Dlg.h

```
int radio_traj;  
double edit_par1;  
double edit_par2;  
double edit_par3;  
double edit_par4;  
double edit_spd;  
double edit_L1;  
double edit_L2;
```

File ...Dlg.cpp

```
CMFCRobot2DSimDlg::CMFCRobot2DSimDlg(CWnd* pParent /*=NULL*/)  
: CDialogEx(CMFCRobot2DSimDlg::IDD, pParent)  
, radio_traj(0)  
, edit_par1(0)  
, edit_par2(0)  
, edit_par3(0)  
, edit_par4(0)  
, edit_spd(0)  
, edit_L1(0)  
, edit_L2(0)  
{  
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);  
}  
  
void CMFCRobot2DSimDlg::DoDataExchange(CDataExchange* pDX)  
{  
    CDialogEx::DoDataExchange(pDX);  
    DDX_Radio(pDX, IDC_RADIO_LINE, radio_traj);  
    DDX_Text(pDX, IDC_EDIT1, edit_par1);  
    DDX_Text(pDX, IDC_EDIT2, edit_par2);  
    DDX_Text(pDX, IDC_EDIT3, edit_par3);  
    DDX_Text(pDX, IDC_EDIT4, edit_par4);  
    DDX_Text(pDX, IDC_EDIT_SPEED, edit_spd);  
    DDX_Text(pDX, IDC_EDIT_L1, edit_L1);  
    DDX_Text(pDX, IDC_EDIT_L2, edit_L2);  
}
```


2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Bổ sung vào lớp Robot hàm sau

```
| void Getconfig(double &par1, double &par2); // File Robot.h  
  
| void Robot::Getconfig(double &par1, double &par2) // File Robot.cpp  
| {  
|     par1=l1; par2=l2;  
| }
```

- Bổ sung vào lớp Trajectory các hàm sau

```
| void GetLine(double &px0, double &py0, double &px1, double &py1, double &spd); // File Trajectory.h  
| void GetCircle(double &px, double &py, double &pra, double &spd);  
  
| void Trajectory::GetLine(double &px0, double &py0, double &px1, double &py1, double &spd) // File Trajectory.cpp  
| {  
|     px0=x0; py0=y0; px1=x1; py1=y1; spd=speed;  
| }  
| void Trajectory::GetCircle(double &px, double &py, double &pra, double &spd)  
| {  
|     px=cex; py=cey; pra=ra; spd=speed;  
| }
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Khai báo thêm biến trong lớp C...Dlg

```
LPCTSTR NumberFormat;
```

- Khai báo và định nghĩa thêm các hàm trong lớp C...Dlg

File ...Dlg.h

```
void DrawAll();  
void UpdateTrajectoryMode(int radio_value);  
void DisplayRobotStatus(LPCTSTR NFormat);
```

File ...Dlg.cpp

```
void CMFCRobot2DSimDlg::DrawAll()  
{  
    CClientDC dc(this);  
    DrawScreen(&dc, 20,20,x_0*2-20,y_0*2-20);  
    trj1.DrawTrajectory(&dc, zoom_scale, Traj_color, x_0,y_0);  
    trj1.SetTime(t);  
    rb1.Setpos(trj1.Getx(),trj1.Gety());  
    rb1.DrawRobot(&dc, zoom_scale, Robot_color, x_0,y_0);  
}
```

...

```
void CMFCRobot2DSimDlg::DisplayRobotStatus(LPCTSTR NFormat)  
{  
    CString str;  
    str.Format(NFormat,rb1.Getx());  
    GetDlgItem(IDC_EDIT_X)->SetWindowText(str);  
    str.Format(NFormat,rb1.Gety());  
    GetDlgItem(IDC_EDIT_Y)->SetWindowText(str);  
    str.Format(NFormat,rb1.Getq1()*180/3.14159);  
    GetDlgItem(IDC_EDIT_Q1)->SetWindowText(str);  
    str.Format(NFormat,rb1.Getq2()*180/3.14159);  
    GetDlgItem(IDC_EDIT_Q2)->SetWindowText(str);  
}
```

...

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Khai báo và định nghĩa thêm các hàm trong lớp C...Dlg

...

```
void CMFCRobot2DSimDlg::UpdateTrajectoryMode(int radio_value)
{
    trj1.SetType(radio_value+1);
    switch (radio_value)
    {
    case 0:
        SetDlgItemText(IDC_STATIC_X0, _T("X0 (m)"));
        SetDlgItemText(IDC_STATIC_Y0, _T("Y0 (m)"));
        SetDlgItemText(IDC_STATIC_X1, _T("X1 (m)"));
        GetDlgItem(IDC_STATIC_Y1)->ShowWindow(TRUE);
        GetDlgItem(IDC_EDIT4)->ShowWindow(TRUE);
        trj1.GetLine(edit_par1, edit_par2, edit_par3, edit_par4, edit_spd);
        break;
    case 1:
        SetDlgItemText(IDC_STATIC_X0, _T("X_center(m)"));
        SetDlgItemText(IDC_STATIC_Y0, _T("Y_center(m)"));
        SetDlgItemText(IDC_STATIC_X1, _T("Radius (m)"));
        GetDlgItem(IDC_STATIC_Y1)->ShowWindow(FALSE);
        GetDlgItem(IDC_EDIT4)->ShowWindow(FALSE);
        trj1.GetCircle(edit_par1, edit_par2, edit_par3, edit_spd);
        break;
    }
    UpdateData(FALSE);
}
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

➤ Cập nhật hàm OnInitDialog()

```
// TODO: Add extra initialization here
UpdateTrajectoryMode(radius_traj);
rb1.Getconfig(edit_L1, edit_L2);
UpdateData(FALSE);

t = 0;
trj1.SetTime(t);
rb1.Setpos(trj1.Getx(), trj1.Gety());
NumberFormat = _T("%.2f");
DisplayRobotStatus(NumberFormat);

Robot_color = RGB(255,0,0);
Traj_color = RGB(255,255,0);
zoom_scale = 500;
x_0 = 200;
y_0 = 200;
```

➤ Cập nhật hàm OnTimer()

```
void CMFCRobot2DSimDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    DrawAll();
    DisplayRobotStatus(NumberFormat);
    t+=0.1;

    CDialogEx::OnTimer(nIDEvent);
}
```

➤ Cập nhật hàm OnPaint()

```
else
{
    DrawAll();
    CDialogEx::OnPaint();
}
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Click đúp vào các control trên giao diện dialog để chương trình tự động thêm các hàm bắt sự kiện sau:

File ...Dlg.h

```
afx_msg void OnBnClickedRadioLine();  
afx_msg void OnBnClickedRadioCircle();  
afx_msg void OnBnClickedButtonSetTraj();  
afx_msg void OnBnClickedButtonSetLength();  
afx_msg void OnBnClickedButtonRun();  
afx_msg void OnBnClickedButtonStop();
```

File ...Dlg.cpp

```
BEGIN_MESSAGE_MAP(CMFCRobot2DSimDlg, CDialogEx)  
    ON_WM_SYSCOMMAND()  
    ON_WM_PAINT()  
    ON_WM_QUERYDRAGICON()  
    ON_WM_TIMER()  
    ON_BN_CLICKED(IDC_RADIO_LINE, &CMFCRobot2DSimDlg::OnBnClickedRadioLine)  
    ON_BN_CLICKED(IDC_RADIO_CIRCLE, &CMFCRobot2DSimDlg::OnBnClickedRadioCircle)  
    ON_BN_CLICKED(IDC_BUTTON_SET_TRAJ, &CMFCRobot2DSimDlg::OnBnClickedButtonSetTraj)  
    ON_BN_CLICKED(IDC_BUTTON_SET_LENGTH, &CMFCRobot2DSimDlg::OnBnClickedButtonSetLength)  
    ON_BN_CLICKED(IDC_BUTTON_RUN, &CMFCRobot2DSimDlg::OnBnClickedButtonRun)  
    ON_BN_CLICKED(IDC_BUTTON_STOP, &CMFCRobot2DSimDlg::OnBnClickedButtonStop)  
END_MESSAGE_MAP()
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

- Viết code cho các hàm vừa thêm

File ...Dlg.cpp

```
void CMFCRobot2DSimDlg::OnBnClickedRadioLine()
{
    // TODO: Add your control notification handler code here
    radio_traj = 0;
    UpdateTrajectoryMode(radio_traj);
    t=0;
    DrawAll();
}

void CMFCRobot2DSimDlg::OnBnClickedRadioCircle()
{
    // TODO: Add your control notification handler code here
    radio_traj = 1;
    UpdateTrajectoryMode(radio_traj);
    t=0;
    DrawAll();
}
```

...

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

➤ Viết code cho các hàm vừa thêm

File ...Dlg.cpp

```
...
void CMFCRobot2DSimDlg::OnBnClickedButtonSetTraj()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    switch (radio_traj)
    {
        case 0:
            trj1.SetLine(edit_par1, edit_par2, edit_par3, edit_par4, edit_spd);
            break;
        case 1:
            trj1.SetCircle(edit_par1, edit_par2, edit_par3, edit_spd);
            break;
    }
    t=0;
    DrawAll();
}

void CMFCRobot2DSimDlg::OnBnClickedButtonSetLength()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    rb1.Setconfig(edit_L1, edit_L2);
    t=0;
    DrawAll();
}

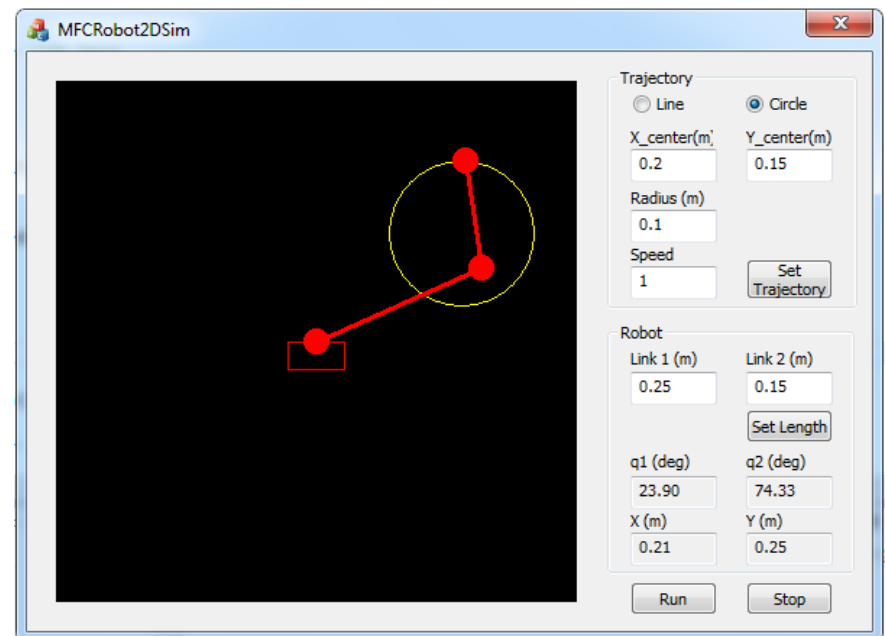
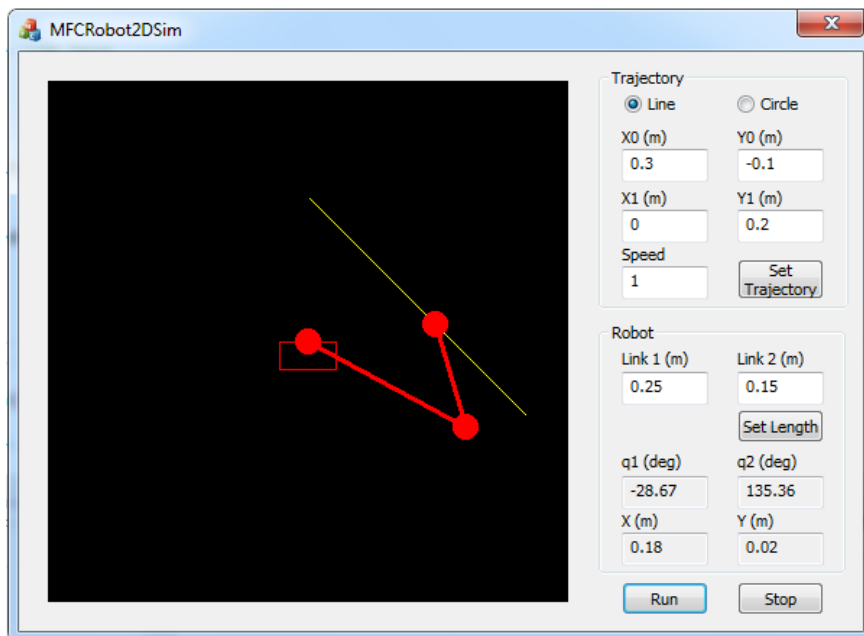
...
void CMFCRobot2DSimDlg::OnBnClickedButtonRun()
{
    // TODO: Add your control notification handler code here
    SetTimer(1,100,NULL);
}

void CMFCRobot2DSimDlg::OnBnClickedButtonStop()
{
    // TODO: Add your control notification handler code here
    KillTimer(1);
}
```

2. Thực hành mô phỏng Robot với đồ họa MFC

2.4. Thêm các control của MFC để hoàn thiện ứng dụng

➡ Kết quả chạy:



Hết Bài 4

