Appendix B.1. The MATLAB codes of the "Path control" block in figure 5.17

1. The MATLAB code function of the "Path control" block in figure 5.17 used for the "Proposed_GA_FLC" when moving on the diamond-shaped path.

```
%------------------------------------------------------------------------------------------------
%------------------------------------------------------------------------------------------------
function [D_error, Theta_error] = fcn(x,y,theta,Waypoints,t)
%  Go to the Waypoint 1
Pi = 3.14;
distanceEps = 0.01;
x_w = Waypoints(1,1);
y_w = Waypoints(1,2);
deltaX = x_w - x;
deltaY = y_w - y;
D_error = sqrt(deltaX^2 + deltaY^2);
Theta_w = atan2(deltaY,deltaX);
Theta_error = Theta_w - theta;
if Theta_error > Pi
    Theta_error = Theta_error - 2*Pi;
end
if Theta_error < -Pi
    Theta_error = Theta_error + 2*Pi;
end
%  Go to the Waypoint 2
if D_error < distanceEps
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
elseif (D_error >= distanceEps) && (t >= 5.98)
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
    %  Go to the Waypoint 3
    if D_error < distanceEps
        i = 3;
        x_w = Waypoints(i,1);
        y_w = Waypoints(i,2);
        deltaX = x_w - x;
        deltaY = y_w - y;
        D_error = sqrt(deltaX^2 + deltaY^2)
        Theta_w = atan2(deltaY,deltaX);
        Theta_error = Theta_w - theta;
        if Theta_error > Pi
            Theta_error = Theta_error- 2*Pi;
        end
        if Theta_error < -Pi
```

```
            Theta_error = Theta_error + 2*Pi;
        end
    elseif (D_error >= distanceEps) && (t >= 12.13)
        i = 3;
        x_w = Waypoints(i,1);
        y_w = Waypoints(i,2);
        deltaX = x_w - x;
        deltaY = y_w - y;
        D_error = sqrt(deltaX^2 + deltaY^2)
        Theta_w = atan2(deltaY,deltaX);
        Theta_error = Theta_w - theta;
        if Theta_error > Pi
            Theta_error = Theta_error- 2*Pi;
        end
        if Theta_error < -Pi
            Theta_error = Theta_error + 2*Pi;
        end
        %  Go to the Waypoint 4 (GOAL)
        if D_error < distanceEps
            i = 4;
            x_w = Waypoints(i,1);
            y_w = Waypoints(i,2);
            deltaX = x_w - x;
            deltaY = y_w - y;
            D_error = sqrt(deltaX^2 + deltaY^2)
            Theta_w = atan2(deltaY,deltaX);
            Theta_error = Theta_w - theta;
            if Theta_error > Pi
                Theta_error = Theta_error- 2*Pi;
            end
            if Theta_error < -Pi
                Theta_error = Theta_error + 2*Pi;
            end
        elseif (D_error >= distanceEps) && (t >= 18.6)
            i = 4;
            x_w = Waypoints(i,1);
            y_w = Waypoints(i,2);
            deltaX = x_w - x;
            deltaY = y_w - y;
            D_error = sqrt(deltaX^2 + deltaY^2)
            Theta_w = atan2(deltaY,deltaX);
            Theta_error = Theta_w - theta;
            if Theta_error > Pi
                Theta_error = Theta_error- 2*Pi;
            end
            if Theta_error < -Pi
                Theta_error = Theta_error + 2*Pi;
            end
        end
    end
end
%-------------------------------------------------------------------------------------------
%---------------------------------------------------------------------------------------------
```

2. **The MATLAB code function of the "Path control" block in figure 5.17 used for the "Proposed_GA_FLC" when moving on the zigzag path.**

```
%-----------------------------------------------------------------------------------------
%----------------------------------------------------------------------------------------
function [D_error, Theta_error] = fcn(x,y,theta,Waypoints,t)
%  Go to the Waypoint 1
Pi = 3.14;
distanceEps = 0.01;
x_w = Waypoints(1,1);
y_w = Waypoints(1,2);
deltaX = x_w - x;
deltaY = y_w - y;
D_error = sqrt(deltaX^2 + deltaY^2);
Theta_w = atan2(deltaY,deltaX);
Theta_error = Theta_w - theta;
if Theta_error > Pi
    Theta_error = Theta_error - 2*Pi;
end
if Theta_error < -Pi
    Theta_error = Theta_error + 2*Pi;
end
%  Go to the Waypoint 2
if D_error < distanceEps
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
elseif (D_error >= distanceEps) && (t >= 5.31)
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
    %  Go to the Waypoint 3
    if D_error < distanceEps
        i = 3;
        x_w = Waypoints(i,1);
        y_w = Waypoints(i,2);
        deltaX = x_w - x;
        deltaY = y_w - y;
        D_error = sqrt(deltaX^2 + deltaY^2)
        Theta_w = atan2(deltaY,deltaX);
        Theta_error = Theta_w - theta;
        if Theta_error > Pi
            Theta_error = Theta_error- 2*Pi;
        end
        if Theta_error < -Pi
            Theta_error = Theta_error + 2*Pi;
        end
```

```
    elseif (D_error >= distanceEps) && (t >= 10.9)
        i = 3;
        x_w = Waypoints(i,1);
        y_w = Waypoints(i,2);
        deltaX = x_w - x;
        deltaY = y_w - y;
        D_error = sqrt(deltaX^2 + deltaY^2)
        Theta_w = atan2(deltaY,deltaX);
        Theta_error = Theta_w - theta;
        if Theta_error > Pi
            Theta_error = Theta_error- 2*Pi;
        end
        if Theta_error < -Pi
            Theta_error = Theta_error + 2*Pi;
        end
        %  Go to the Waypoint 4 (GOAL)
        if D_error < distanceEps
            i = 4;
            x_w = Waypoints(i,1);
            y_w = Waypoints(i,2);
            deltaX = x_w - x;
            deltaY = y_w - y;
            D_error = sqrt(deltaX^2 + deltaY^2)
            Theta_w = atan2(deltaY,deltaX);
            Theta_error = Theta_w - theta;
            if Theta_error > Pi
                Theta_error = Theta_error- 2*Pi;
            end
            if Theta_error < -Pi
                Theta_error = Theta_error + 2*Pi;
            end
        elseif (D_error >= distanceEps) && (t >= 17.2)
            i = 4;
            x_w = Waypoints(i,1);
            y_w = Waypoints(i,2);
            deltaX = x_w - x;
            deltaY = y_w - y;
            D_error = sqrt(deltaX^2 + deltaY^2)
            Theta_w = atan2(deltaY,deltaX);
            Theta_error = Theta_w - theta;
            if Theta_error > Pi
                Theta_error = Theta_error- 2*Pi;
            end
            if Theta_error < -Pi
                Theta_error = Theta_error + 2*Pi;
            end
        end
    end
end
%------------------------------------------------------------------------------------------
%------------------------------------------------------------------------------------------
```

3. The MATLAB code function of the "Path control" block in figure 5.17 used for the "Proposed_GA_FLC" when moving on the square path.

```
%-----------------------------------------------------------------------------------------------
%---------------------------------------------------------------------------------------------
function [D_error, Theta_error] = fcn(x,y,theta,Waypoints,t)
%  Go to the Waypoint 1
Pi = 3.14;
distanceEps = 0.01;
x_w = Waypoints(1,1);
y_w = Waypoints(1,2);
deltaX = x_w - x;
deltaY = y_w - y;
D_error = sqrt(deltaX^2 + deltaY^2);
Theta_w = atan2(deltaY,deltaX);
Theta_error = Theta_w - theta;
if Theta_error > Pi
    Theta_error = Theta_error - 2*Pi;
end
if Theta_error < -Pi
    Theta_error = Theta_error + 2*Pi;
end
%  Go to the Waypoint 2
if D_error < distanceEps
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
elseif (D_error >= distanceEps) && (t >= 5.31)
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
    %  Go to the Waypoint 3
    if D_error < distanceEps
        i = 3;
        x_w = Waypoints(i,1);
        y_w = Waypoints(i,2);
        deltaX = x_w - x;
        deltaY = y_w - y;
        D_error = sqrt(deltaX^2 + deltaY^2)
        Theta_w = atan2(deltaY,deltaX);
        Theta_error = Theta_w - theta;
        if Theta_error > Pi
            Theta_error = Theta_error- 2*Pi;
        end
        if Theta_error < -Pi
            Theta_error = Theta_error + 2*Pi;
        end
    elseif (D_error >= distanceEps) && (t >= 10.91)
```

```
        i = 3;
        x_w = Waypoints(i,1);
        y_w = Waypoints(i,2);
        deltaX = x_w - x;
        deltaY = y_w - y;
        D_error = sqrt(deltaX^2 + deltaY^2)
        Theta_w = atan2(deltaY,deltaX);
        Theta_error = Theta_w - theta;
        if Theta_error > Pi
            Theta_error = Theta_error- 2*Pi;
        end
        if Theta_error < -Pi
            Theta_error = Theta_error + 2*Pi;
        end
        %  Go to the Waypoint 4 (GOAL)
        if D_error < distanceEps
            i = 4;
            x_w = Waypoints(i,1);
            y_w = Waypoints(i,2);
            deltaX = x_w - x;
            deltaY = y_w - y;
            D_error = sqrt(deltaX^2 + deltaY^2)
            Theta_w = atan2(deltaY,deltaX);
            Theta_error = Theta_w - theta;
            if Theta_error > Pi
                Theta_error = Theta_error- 2*Pi;
            end
            if Theta_error < -Pi
                Theta_error = Theta_error + 2*Pi;
            end
        elseif (D_error >= distanceEps) && (t >= 16.48)
            i = 4;
            x_w = Waypoints(i,1);
            y_w = Waypoints(i,2);
            deltaX = x_w - x;
            deltaY = y_w - y;
            D_error = sqrt(deltaX^2 + deltaY^2)
            Theta_w = atan2(deltaY,deltaX);
            Theta_error = Theta_w - theta;
            if Theta_error > Pi
                Theta_error = Theta_error- 2*Pi;
            end
            if Theta_error < -Pi
                Theta_error = Theta_error + 2*Pi;
            end
        end
    end
end
%-------------------------------------------------------------------------------------------
%-------------------------------------------------------------------------------------------
```

4. The MATLAB code function of the “Path control” block in figure 5.17 used for the “Proposed_GA_FLC” when moving on the Sharp turn path.

```
%-------------------------------------------------------------------------------
%-----------------------------------------------------------------------------
function [D_error, Theta_error] = fcn(x,y,theta,Waypoints,t)
%  Go to the Waypoint 1
Pi = 3.14;
distanceEps = 0.01;
x_w = Waypoints(1,1);
y_w = Waypoints(1,2);
deltaX = x_w - x;
deltaY = y_w - y;
D_error = sqrt(deltaX^2 + deltaY^2);
Theta_w = atan2(deltaY,deltaX);
Theta_error = Theta_w - theta;
if Theta_error > Pi
    Theta_error = Theta_error - 2*Pi;
end
if Theta_error < -Pi
    Theta_error = Theta_error + 2*Pi;
end
%  Go to the Waypoint 2
if D_error < distanceEps
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
elseif (D_error >= distanceEps) && (t >= 5.31)
    i = 2;
    x_w = Waypoints(i,1);
    y_w = Waypoints(i,2);
    deltaX = x_w - x;
    deltaY = y_w - y;
    D_error = sqrt(deltaX^2 + deltaY^2)
    Theta_w = atan2(deltaY,deltaX);
    Theta_error = Theta_w - theta;
    if Theta_error > Pi
        Theta_error = Theta_error- 2*Pi;
    end
    if Theta_error < -Pi
        Theta_error = Theta_error + 2*Pi;
    end
    end
end
%-----------------------------------------------------------------------------
%-----------------------------------------------------------------------------
```