

## **Robótica Autónoma**

Se pretende desarrollar un caso práctico relacionado con la robótica y la capacidad autónoma de las máquinas, eje fundamental de la Industrial 4.0.

### **Enunciado**

Se trata de simular el movimiento de un robot en un espacio de trabajo simulado con una matriz  $M \times N$  en el que se definen una serie de posibles obstáculos del movimiento del robot. El robot debe probar si es capaz desde un cierto punto origen llegar a un punto destino antes de un cierto tiempo límite de simulación. Si es capaz de llegar diremos que el algoritmo ha tenido éxito. Se debe almacenar la ruta e ir dibujando la trayectoria en pantalla.

El algoritmo de movimiento es un paso aleatorio en las 4 direcciones principales en una unidad de tiempo, siempre que no tope con un obstáculo o los límites del entorno de trabajo, situaciones en las que se contarán 2 unidades de tiempo.

### **Implementación**

`RobotMovil2D.m`: implementa el algoritmo de movimiento mientras no se llegue al destino y no se alcance el tiempo límite de simulación

`SimulacionRobotMovil2D.m`: se define el entorno de trabajo y se lanza la simulación

```

% SCRIPT para simular un Robot Móvil
% Autor: Fundamentos de Informática
% Curso: 2018-19

clc;
clear all;
close all;

disp('Robot Móvil con Algoritmo Aleatorio');
disp('-----');

% El entorno es una matriz MxN donde los obstáculos se simulan con -1
% Se define un entorno
M = 40;
N = 40;
m = zeros(M,N);

% Definición de obstáculos

m(20,10:30) = -1;
m(10:30,20) = -1;

% Definición de origen y destino

% origen
x = input('Introduce coordenada X del origen: ');
y = input('Introduce coordenada Y del origen: ');
origen.x = x;
origen.y = y;

% destino
x = input('Introduce coordenada X del destino: ');
y = input('Introduce coordenada Y del destino: ');
destino.x = x;
destino.y = y;

% Definición del tiempo límite de simulación

tiempo_limite = input('Introduce el tiempo limite de simulación: ');

% Simulaciones con Algoritmo Aleatorio

[exito,ruta] = RobotMovil2D(m,origen,destino,tiempo_limite);

disp('-----');
if (exito == 1)
    disp('El algoritmo tuvo éxito.');
```

```

else
    disp('El algoritmo es malo.');
```

```

end
disp('-----');
disp('Fin de la simulación');
```

```

function [exito,ruta] = RobotMovil2D(m,origen,destino,tiempo_limite)
% RobotMovil2D mueve aleatoriamente un robot en un entorno m con
% obstáculos a -1 desde un origen a un destino decidiendo si se tiene éxito
% en un tiempo_limite

% Autor: Fundamentos de Informática
% Curso: 2018-19

[a,b] = size(m);
% para ver una imagen de la ruta en cada paso
c = m;
ii = find(c==-1);
c(ii) = 1000;
c(origen.x,origen.y) = 100;
c(destino.x,destino.y) = 200;

actual.x = 0; actual.y = 0;
actual = origen;
t = 1;

MuestraImagen(c,origen,destino,actual,t);
saveas(gcf,'robot1','jpeg');

while ~(actual.x == destino.x) && (actual.y == destino.y) && (t < tiempo_limite)
    old = actual;
    d = floor(4*rand(1,1))+1; % aleatorio entre 1 y 4
    if d == 1 % SUR
        actual.x = actual.x + 1;
    end
    if d == 2 % NORTE
        actual.x = actual.x - 1;
    end
    if d == 3 % ESTE
        actual.y = actual.y + 1;
    end
    if d == 4 % OESTE
        actual.y = actual.y - 1;
    end

    if (actual.x <= a && actual.x >= 1 && actual.y <= b && actual.y >= 1) == true
        if m(actual.x,actual.y) == -1
            ruta(t) = old;
            ruta(t+1) = old;
            actual = old;
            t = t + 2;
        else
            % se mueve de forma efectiva
            ruta(t) = actual;
            t = t + 1;
            c(actual.x,actual.y) = mod(t,60); % color cambiante

            MuestraImagen(c,origen,destino,actual,t);

            % almacena en disco
            if mod(t,500) == 0

```

```

        s = sprintf('robot%d',t);
        saveas(gcf,s,'jpeg');
    end

    end
else
    ruta(t) = old;
    ruta(t+1) = old;
    actual = old;
    t = t + 2;
end
end
if t < tiempo_limite
    exito = 1;
else
    exito = 0;
end
end

function MuestraImagen(c,origen,destino,actual,t)
% muestra la imagen
image(c);
title('Simulación robot móvil. Algoritmo aleatorio.');
```

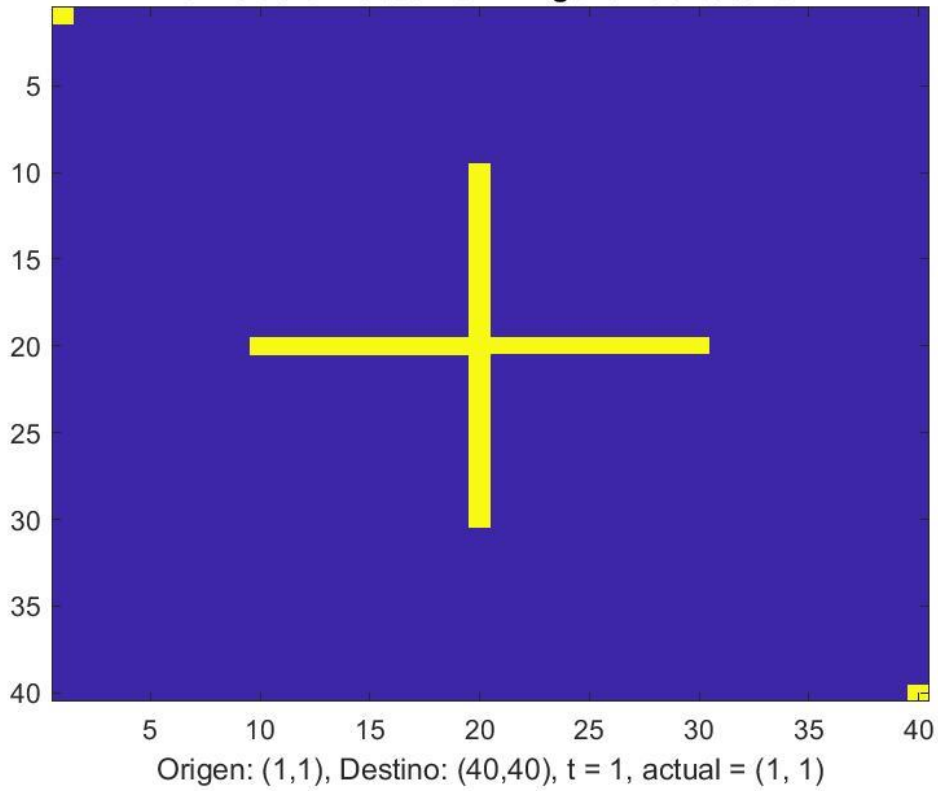
s = sprintf('Origen: (%d,%d), Destino: (%d,%d), t = %d, actual = (%d, %d)', ...  
origen.x,origen.y,destino.x,destino.y,t,actual.x,actual.y);

```

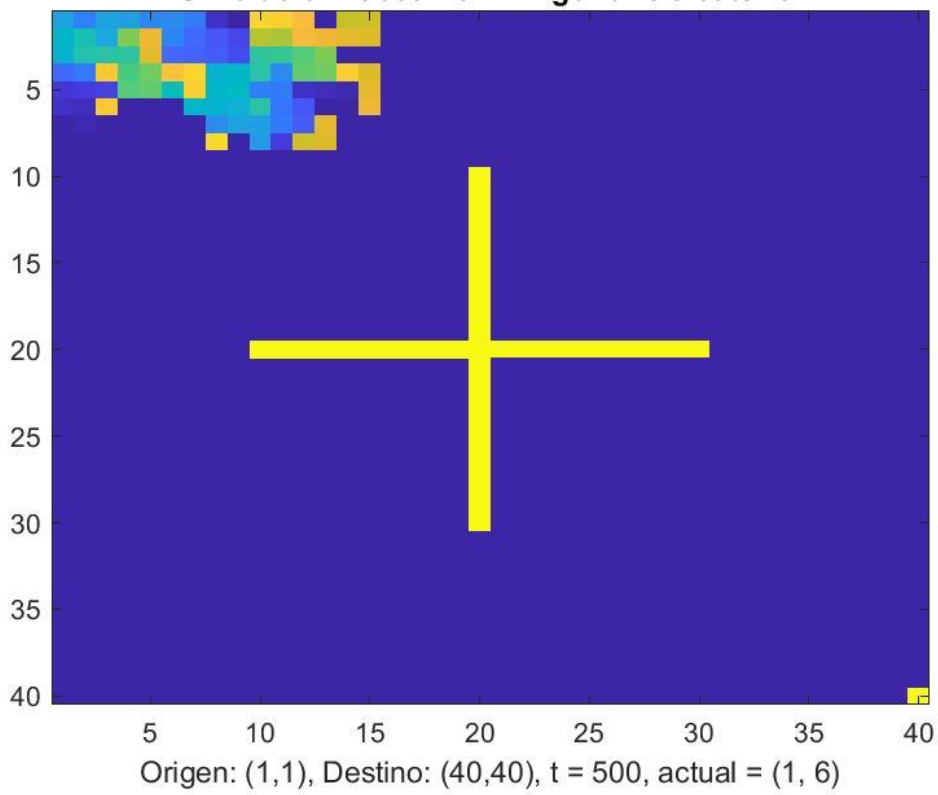
xlabel(s);
pause(0.01);
end

```

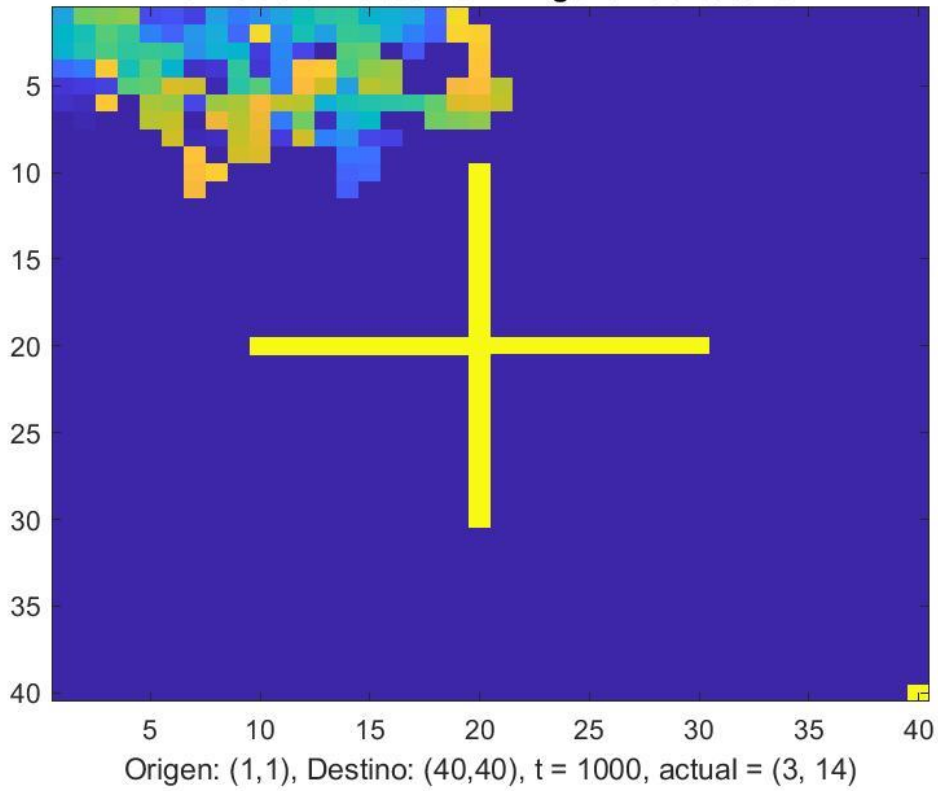
**Simulación robot móvil. Algoritmo aleatorio.**



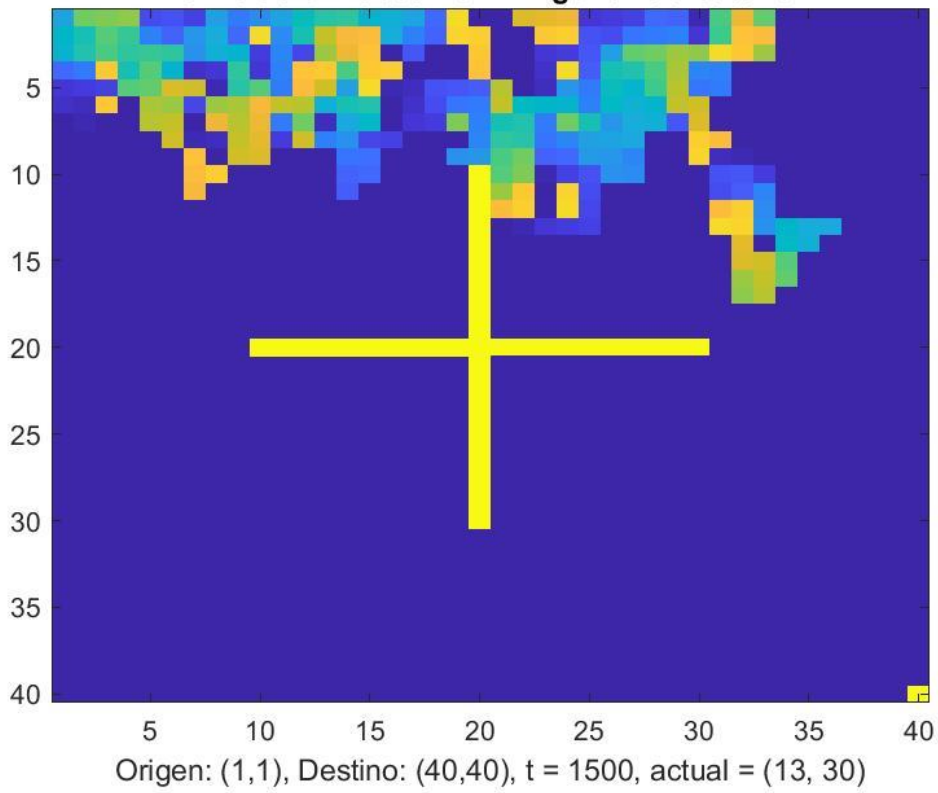
**Simulación robot móvil. Algoritmo aleatorio.**



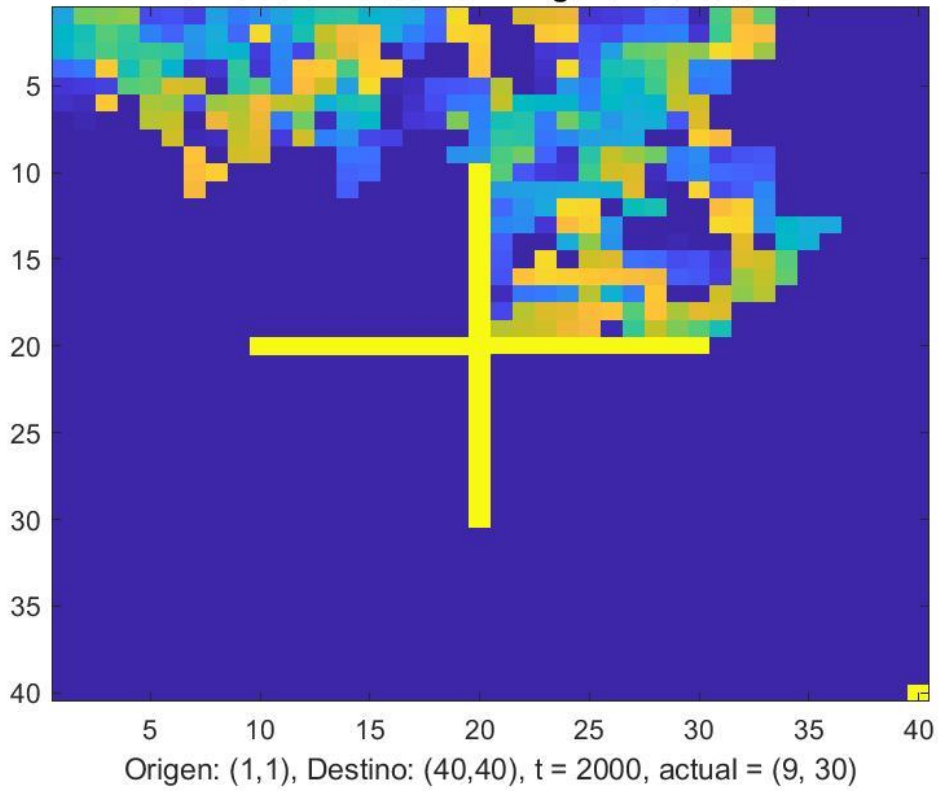
**Simulación robot móvil. Algoritmo aleatorio.**



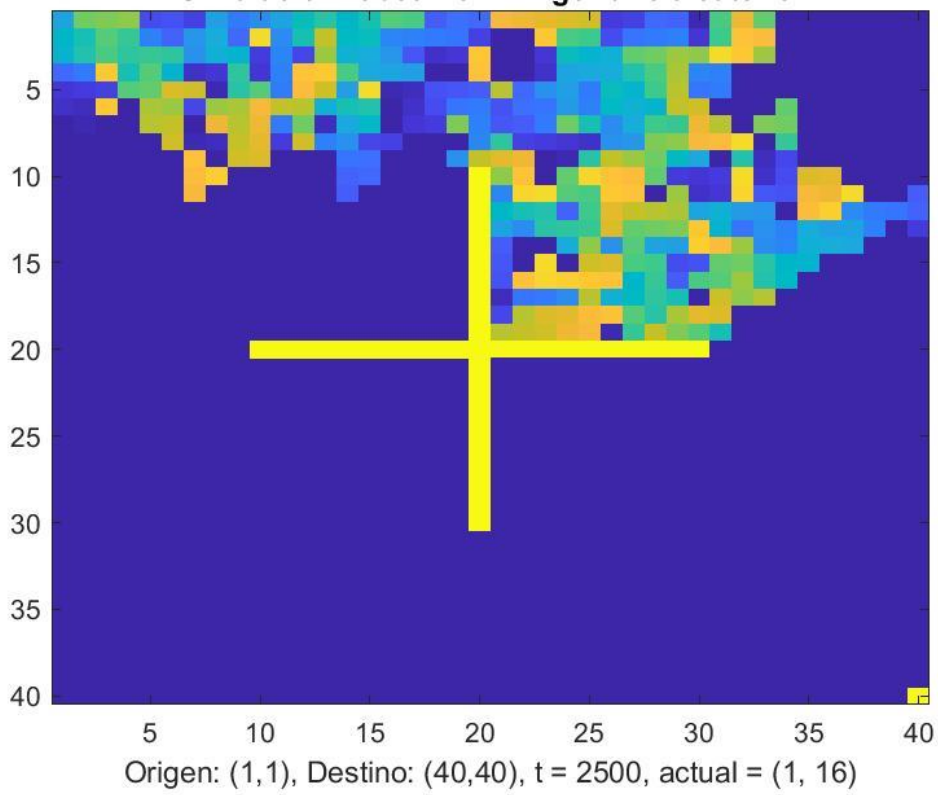
**Simulación robot móvil. Algoritmo aleatorio.**



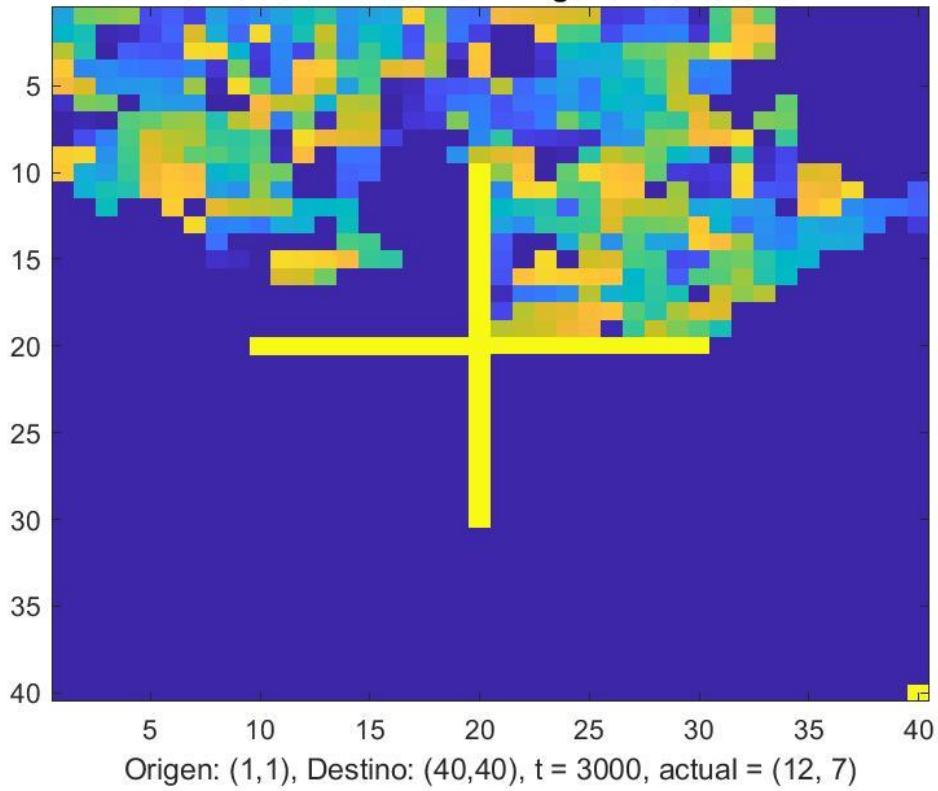
**Simulación robot móvil. Algoritmo aleatorio.**



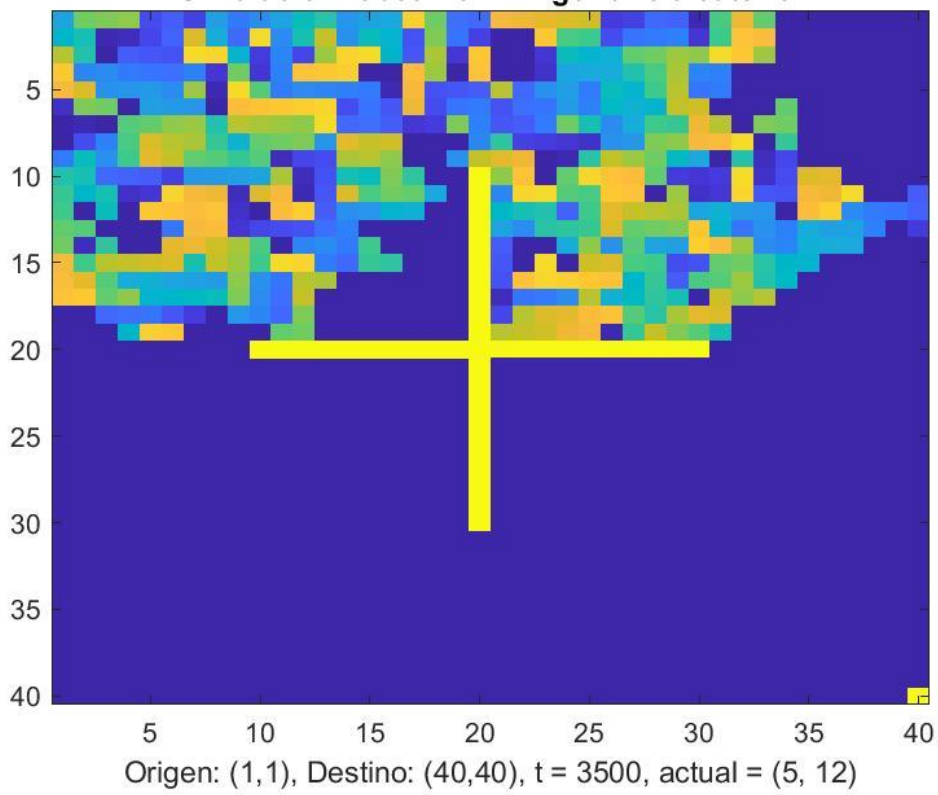
**Simulación robot móvil. Algoritmo aleatorio.**



**Simulación robot móvil. Algoritmo aleatorio.**

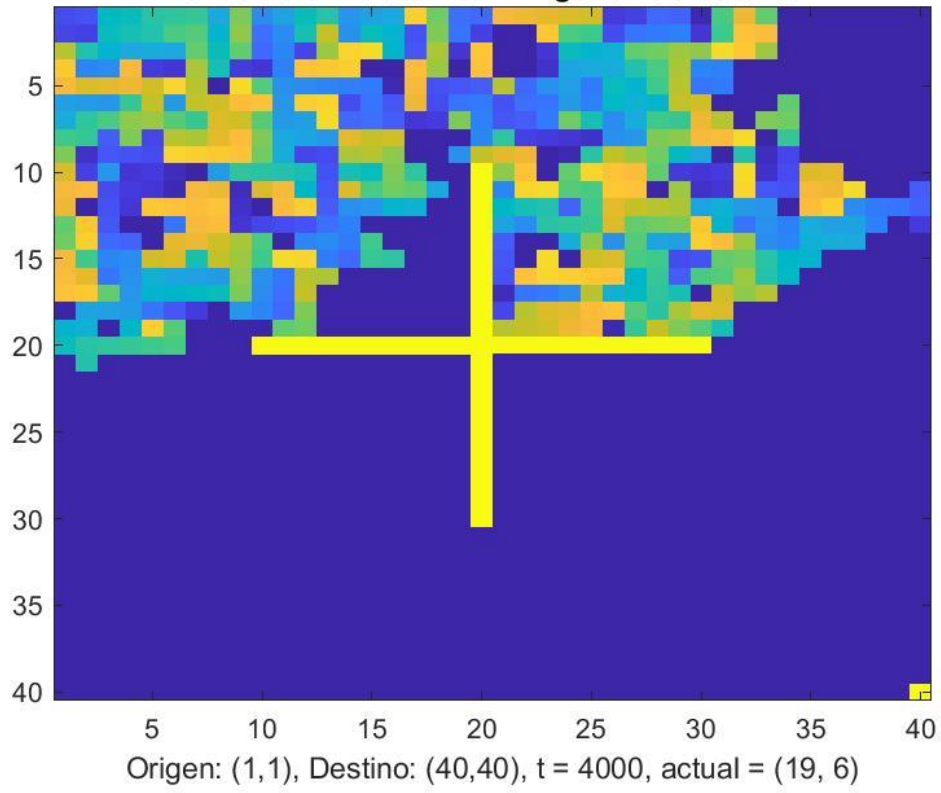


**Simulación robot móvil. Algoritmo aleatorio.**





### Simulación robot móvil. Algoritmo aleatorio.



*Published with MATLAB® R2018b*