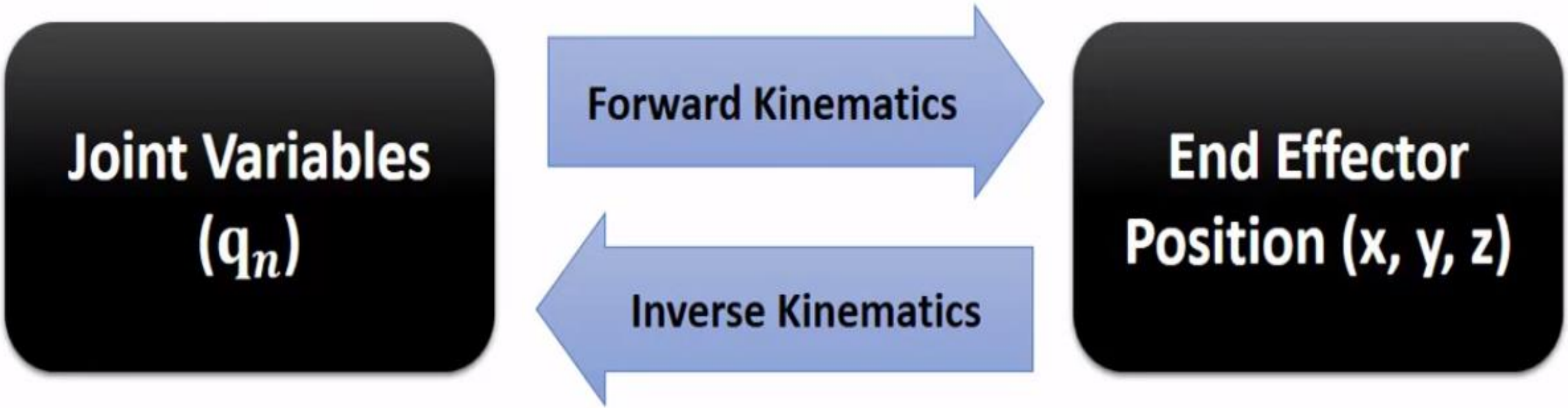
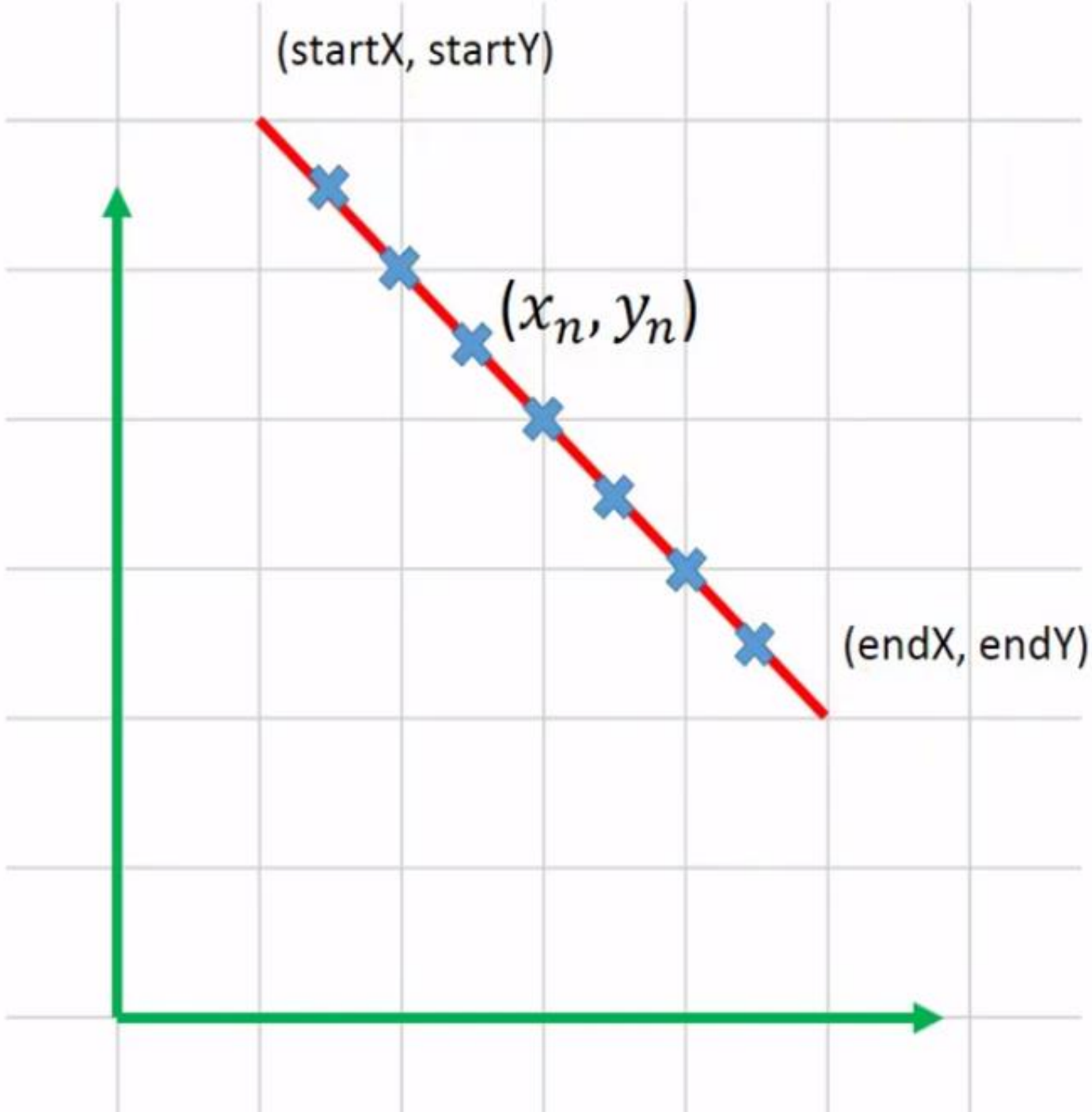
Kinematics

- The branch of classical mechanics that describes the motion of objects without consideration of the forces that cause it



Steps..

- Slice the shape into a set of paths.
- Divide each path into a set of points.
- Apply inverse kinematics for each point.
- Apply the calculated joint angles to motors.



$$dx = \frac{endX - startX}{numOfSteps}$$

$$dy = \frac{endY - startY}{numOfSteps}$$

$$x_n = startX + n \times dx$$

$$y_n = startY + n \times dy$$

$$n \in [0, numOfSteps]$$

path_K.m | explain_line.m | explain_circle.m | explain_path.m | getLine.m | getCurve.m | path_fun.m

```
clear;
clc;

roboArm.L = [8, 8];
roboArm.offset = [0, 0];

plot(0,0,'b*');
axis([-17 17 -17 17]);
hold on;
grid on;
ind = 1;

delay = 0.1;

startX = 0;
startY = 5;
endX = 5;
endY = 10;

numOfSteps = 30;
dx = (endX - startX)/numOfSteps;
dy = (endY - startY)/numOfSteps;

for n = 0:numOfSteps
    theta = getikine(roboArm, [startX+n*dx, startY+n*dy],'right');
```

```
13 -    delay = 0.1;
14
15 -    startX = 0;
16 -    startY = 5;
17 -    endX = 5;
18 -    endY = 10;
19
20 -    numOfSteps = 30;
21 -    dx = (endX - startX)/numOfSteps;
22 -    dy = (endY - startY)/numOfSteps;
23
24 -    for n = 0:numOfSteps
25 -        theta = getikine(roboArm, [startX+n*dx, startY+n*dy],'right');
26 -        P = getfkine(roboArm, theta - roboArm.offset);
27 -        locationX(ind) = P(1);
28 -        locationY(ind) = P(2);
29 -        ind = ind + 1;
30 -        hold off;
31 -        plot(locationX,locationY,'b');
32 -        hold on;
33 -        grid on;
34 -        plotRobot(roboArm, theta);
35 -        axis([-17 17 -17 17]);
36 -        pause(delay);
37 -    end
```