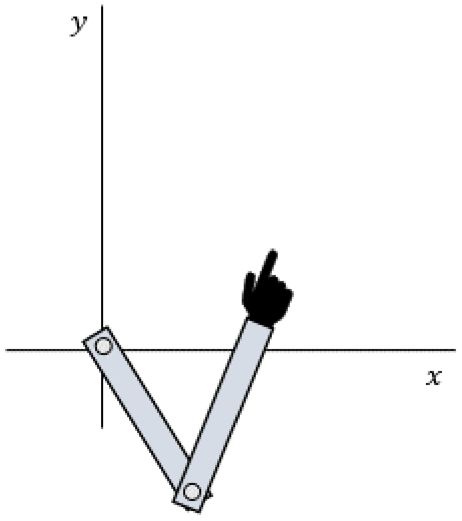
MAT188 - Topic 2 Assignment

Please ensure you upload both MLX and PDF files to Gradescope.

Introduction

For this assignment we will be looking at the two-link robotic arm shown in Figure 1 and the animation below. The first link rotates around the origin and the second link rotates around the end of the first link.



You will be given the lengths of the links L_1 and L_2, as well as their angles relative to the horizontal axis θ_1 and θ_2. Using this information, the goal is to calculate the x,y-coordinates of the hand at the end of the second link as shown in Figure 1c.

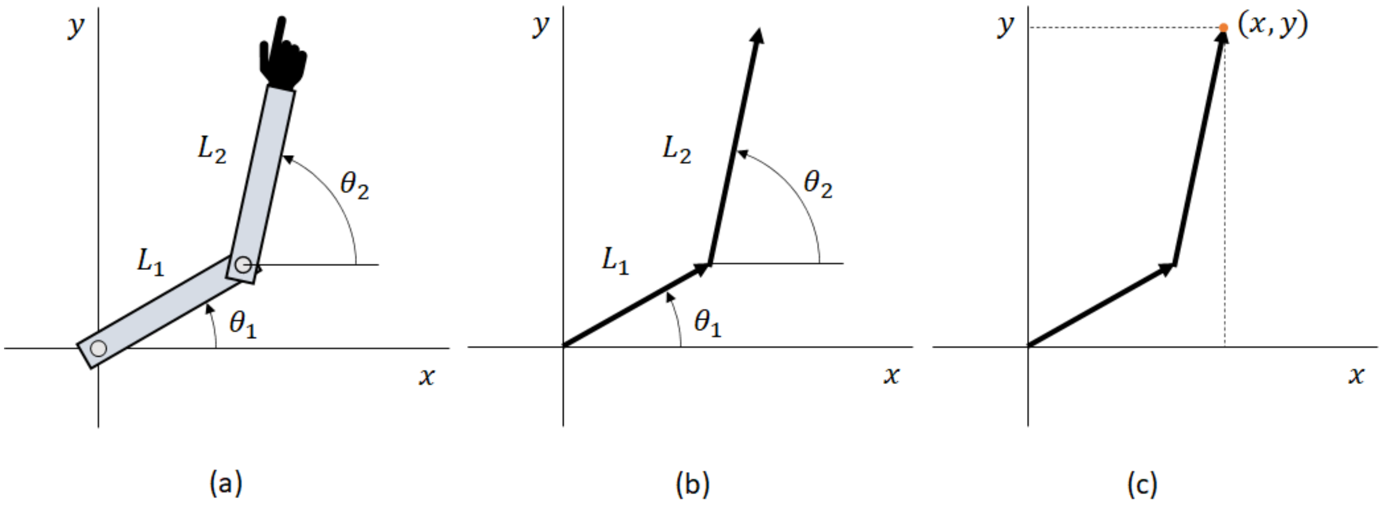


Figure 1: (a) diagram of the two-link robotic arm (b) simplified diagram - each arrow represents one of the links (c) x,y-coordinates of the hand (to be calculated)

This assignment consists of four parts. In each of the parts, you will

- Part 1: load data from a CSV file and visualize the provided data
- Part 2: write a function to calculate the x and y displacement of a single link given its length and angle
- Part 3: compute the x,y-coordinates of the hand using the function from Part 2
- Part 4: use spanning sets to determine if the robotic arm can reach certain locations within the *x*-*y* plane

Part 1: Load and Visualize Data

The first step is to load the data from the CSV file into the workspace and to define variables for the lengths of the two links.

The data is provided to you in `https://raw.githubusercontent.com/dtxe/mat188_datasets/main/robot_trajectory.csv`. This file contains the values of θ_1 and θ_2 at different points in time. The first column contains the values of θ_1 **in degrees** and is labelled `theta1`. The second column contains the values of θ_2 **in degrees** and is labelled `theta2`. Each entry (row) in this table corresponds to the angles at a different point in time.

1.1 Define variables for the lengths of the two links

The lengths of the two links are $L_1 = L_2 = 10\,\mathrm{cm}$. Variables `L1` and `L2` representing the lengths of the links are defined below. You do not need to do anything for this part.

```
L1 = 10; % length of the first link in cm
L2 = 10; % length of the second link in c
```

1.2 Load data from the CSV file provided

Import data from the CSV file provided **as a table** to a variable called `data`. Recall that in the lab exercise, you used `readtable` to import data from a CSV on the internet using its URL.

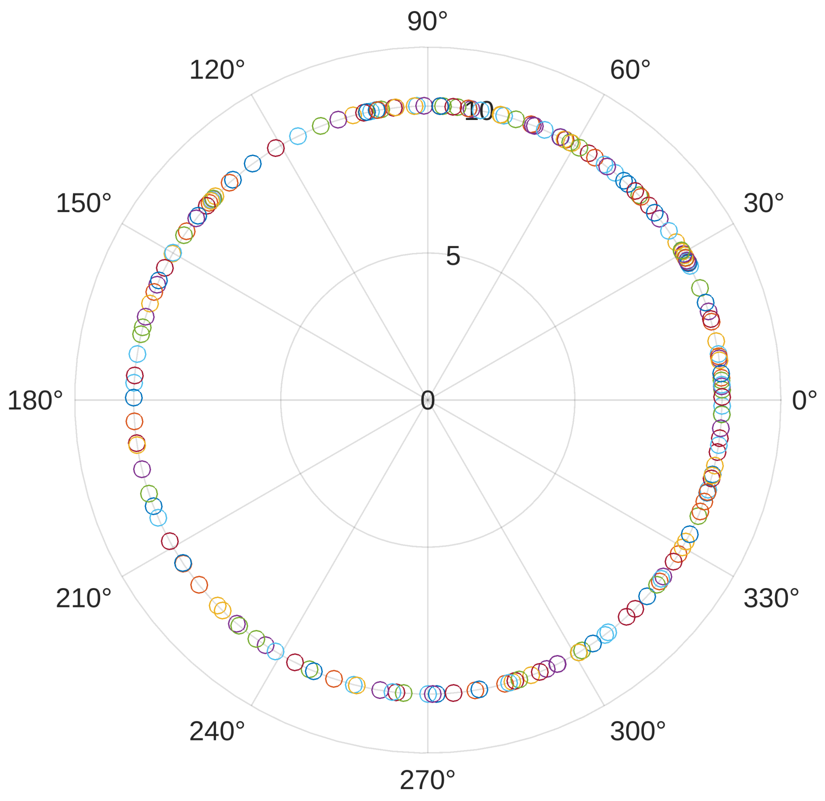
```
data = readtable('https://raw.githubusercontent.com/dtxe/mat188_datasets/main/robot_tra
% show the first few rows to see if we loaded it properly
disp(head(data))
```

```
    theta1    theta2
    ______    ______

    131.55    231.57
    137.97    234.59
    136.21    233.22
    136.89    240.27
    136.77    247.19
    137.28    254.34
    138.74    261.92
    141.28    270.07
```

1.3 Make a polar plot to visualize the trajectory of the first link

```
rad_angle = deg2rad(data.theta1);
polarscatter(rad_angle, L1)
```



Part 2: Forward Kinematics of the Two-Link Robot

Almost there! For this last part, you will compute the x,y-coordinates of the hand at the end of the second link, update your data table, and visualize your results.

2.1. Compute the x,y-coordinates and update your table

First, use the angles θ_1 and θ_2 in `data` (the table you loaded in Part 1) and the function you wrote in Part 2 to find the x,y-coordinates of the hand for each entry in `data`. Then, update the table `data` by adding two columns to store the x and y coordinates that you computed. Make sure to display the updated table to demonstrate that you have done this successfully.

Hint: To calculate the location of the hand, you can calculate the displacements of the two links individually (Figure 3a) and then take the sum (Figure 3b).

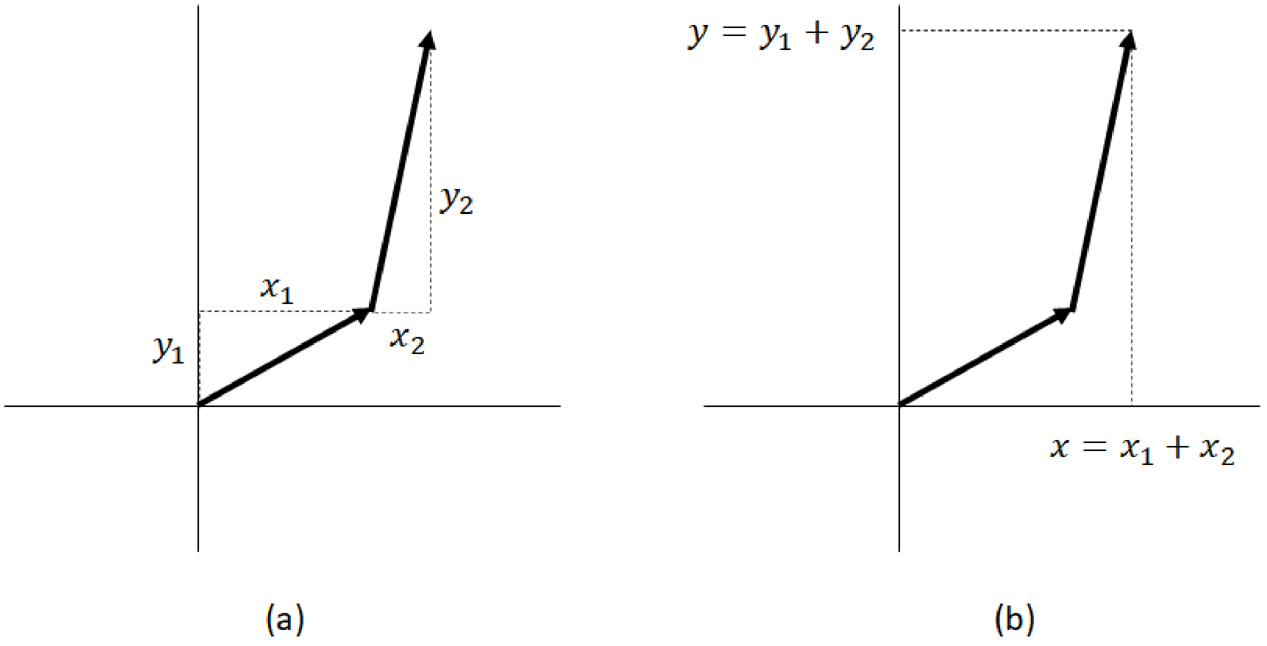


Figure 3: (a) displacements of individual links (b) total displacement of both links

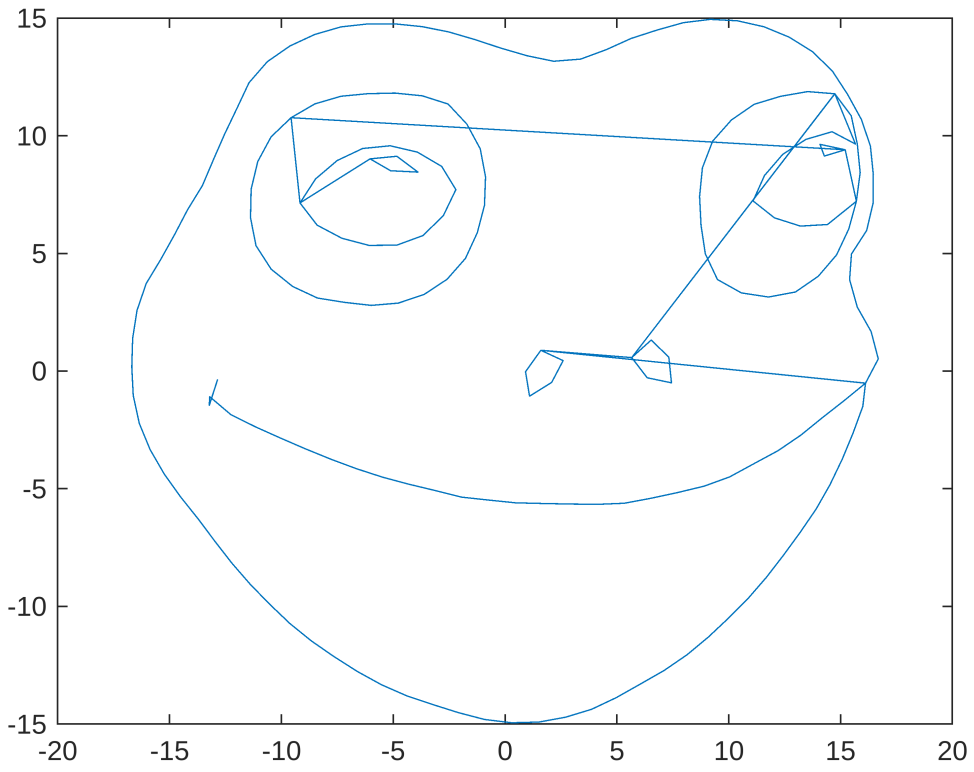
Please write your code below.

```
% ---- YOUR CODE HERE ----
[x1, y1] = poltocart(L1, deg2rad(data.theta1));
[x2, y2] = poltocart(L2, deg2rad(data.theta2));
x = x1 + x2;
y = y1 + y2;
```

2.2. Plot the trajectory

Plot your x and y coordinates to draw a picture.

```
% ---- YOUR CODE HERE ----
plot(x, y)
```



Part 3: Linear Combinations and Spanning Sets

3.1 Basis Vectors and Linear Combinations

Write a function that takes in two 3D vectors and returns the following:

- true, if the two vectors are linearly independent
- false, if the two vectors are linearly dependent

```
% your code here
% Function written at the back
```

3.2 Case Palletizer Robot

Atlas Robots is an engineering firm that happens to employ recent UofT graduates with fond memories of MAT188, CIV100 and APS111/112. Atlas was contracted to improve the production workflow for one of their clients who wanted to automate case pallet stacking and storage. The final recommendation was the KUKA case palletizer robot.

ATLAS ROBOTS
RECOGEMOS EL PALET CON PALAS MECÁNICAS
LO QUE PERMITE ASEGURAR SU COGIDA Y POSICIÓN

Image caption: A KUKA case palletizer robot with mobile track (image source: atlas_robots).

As observed in the gif image above, the robot has several actuators(*) and is mounted on a horizontal mobile track that extends its range of movement to different workstations. Note that the sequence of actuators is not important. The focus is rather on the combination of movements required to effect a desired motion.

(* *An actuator is a device that requires a source of energy e.g., electrical, hydraulic pressure or pneumatic pressure to move part of a mechanism*)

Now consider that some modifications were made to the KUKA robot such that the

- when actuator 1 is activated, the pallet moves along a line described by the vector $\vec{a} = \begin{bmatrix} 0.75 \\ 0 \\ 0.25 \end{bmatrix} t$
- when actuator 2 is activated, the pallet moves along a line described by the vector $\vec{b} = \begin{bmatrix} 0.45 \\ 0.45 \\ 0.45 \end{bmatrix} t$
- when actuator 3 is activated, the pallet moves along a line described by the vector $\vec{c} = \begin{bmatrix} 0.75 \\ 0 \\ 0 \end{bmatrix} t$
- when actuator 4 is activated, the pallet moves along a line described by the vector $\vec{d} = \begin{bmatrix} 0.75 \\ 0.25 \\ 0 \end{bmatrix} t$

for arbitrary parameter t.

Find basis vectors that span the robot's resulting range of motion if:

- all the actuators are working?

```
% your code here
a = [0.75; 0; 0.25];
b = [0.45; 0.45; 0.45];
c = [0.75; 0; 0];
d = [0.75; 0.25; 0];
A1 = [a b c d];
getbasis(A1);
```

```
    0.7500
         0
    0.2500

    0.4500
    0.4500
    0.4500

    0.7500
         0
         0

are the basis vectors
```

- only actuator 1 and 4 are working?

```
% your code here
A2 = [a d];
getbasis(A2);
```

```
    0.7500
         0
    0.2500

    0.7500
    0.2500
         0

are the basis vectors
```

- only actuator 2 and 3 are working?

```
% your code here
A3 = [b c];
getbasis(A3);
```

```
    0.4500
    0.4500
    0.4500

    0.7500
         0
         0
```

```
are the basis vectors
```

Are there one or more actuators which, if failed, would still allow the robot to retain its range of motion?

```
% your answer here
checkredundent(A1);
```

```
Any actuators failing will not allow the robot to retain its range of motion
```

Functions

Here you will put all functions you are asked to write in this assignment. In total, there are four functions required.

Part 2.1

Inputs:

- `theta` - a column vector (`N x 1`) containing the angles of the link in radians
- `rho` - a scalar, the length of the link

Outputs:

- `x` - a column vector (`N x 1`) containing x displacements of the link
- `y` - a column vector (`N x 1`) containing y displacements of the link

Part 3.1

Inputs:

- `vec1` - Any 3D vector.
- `vec2` - Any other 3D vector.

Output:

- `islind` - A logical output that is `true` if the vectors are linearly independent, and `false` if they are linearly dependent.

Put all functions in the code block below

```
function [islind] = checklindep(vec1, vec2)
% ---- YOUR CODE HERE ---- %
    if rank(rref([vec1 vec2])) == 2
        islind = true;
    else
        islind = false;
    end
```

```
end

function [x, y] = poltocart(rho, theta)
    x = rho * cos(theta);
    y = rho * sin(theta);
end

function getbasis(A)
    Ar = rref(A);
    existbasis = false;
    for col = 1 : width(Ar)
        if ismember(Ar(:, col), [[1; 0; 0], [0; 1; 0], [0; 0; 1]])
            disp(A(:, col))
            existbasis = true;
        end
    end
    if existbasis == true
        disp("are the basis vectors")
    else
        disp("There is no basis for the setup")
    end
end

function checkredundent(A)
    allnotlindep = true;
    for i = 1 : 4 - 1
        for j = i + 1 : 4
            vec1 = A(:, i);
            vec2 = A(:, j);
            if checklindep(vec1, vec2) == false
                disp("One of", vec1, vec2, "failing will still allow the robot to retai
                allnotlindep = false;
            end
        end
    end
    if allnotlindep == true
        disp("Any actuators failing will not allow the robot to retain its range of mot
    end
end
```