

Robot Navigation Project - Organize Sequence Data

This dataset was obtained from the UCI Machine Learning Repository.

You can find more information on this dataset here:

<https://archive.ics.uci.edu/ml/datasets/Wall-Following+Robot+Navigation+Data>

Load the unprocessed sensor data. This mat file contains 3 variables.

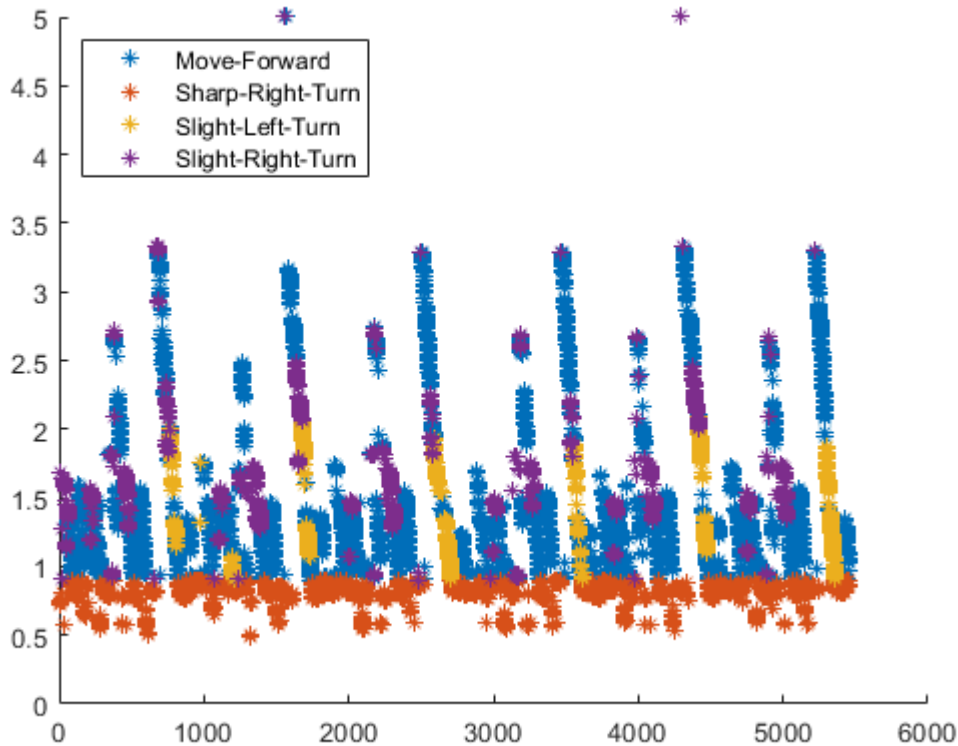
- label1: The action the robot should take at this time step. At any point, the robot can either move forward, turn slightly to the left, turn slightly to the right, or turn sharply to the right
- sensor1: Sensor reading from the front of the robot
- sensor2: Sensor reading from the left of the robot

```
load robotData.mat
```

Display data

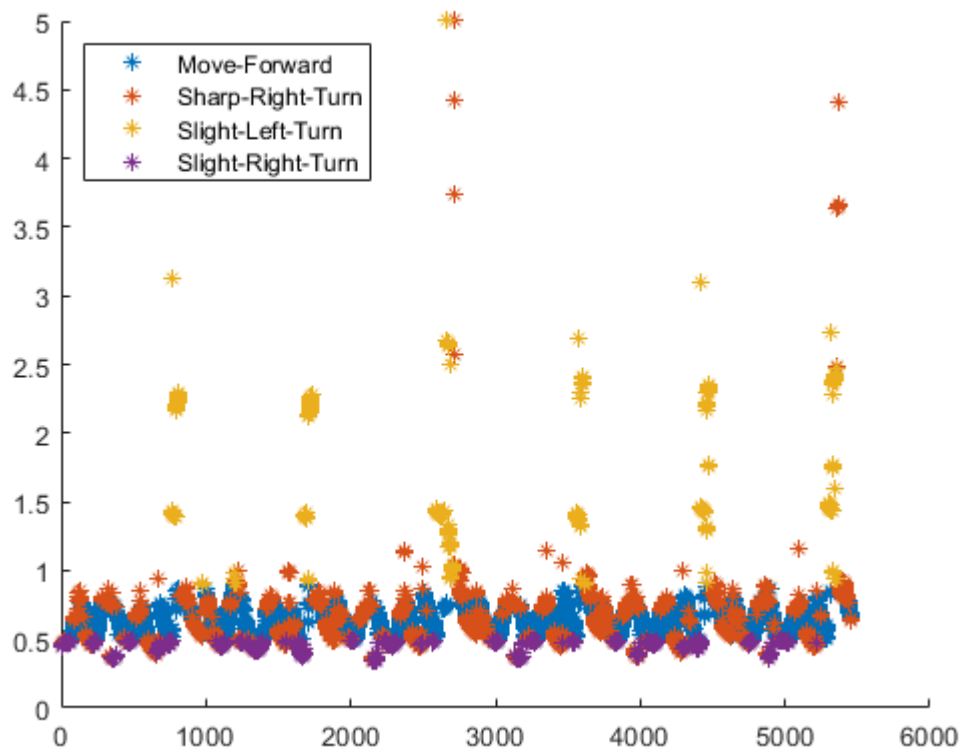
Display the first feature of the sequence data. The function `displaySequence` located at the bottom of the script. This function takes two inputs, the sensor data and the class labels.

```
displaySequence(sensor1,label1)
```



Display the second feature.

```
displaySequence(sensor2,label)
```



Create matrix of sequence data

Concatenate the two vectors of sensor data into one matrix.

```
X = [sensor1 sensor2];
```

Transpose the matrix so that the columns are the time steps.

```
X = X'
```

```
X = 2x5456
    1.6870    1.6870    1.6870    1.6870    1.6870    1.6860    1.6840    1.6800 ...
    0.4450    0.4490    0.4490    0.4490    0.4490    0.4460    0.4510    0.4530
```

Transpose sequence labels

Each time step of the sequence data has a label. The labels should be stored in a row vector, but currently it is a column. Take the transpose.

```
Y = label';
```

view the class distribution of the data.

```
summary(label)
```

Move-Forward	2205
Sharp-Right-Turn	2097
Slight-Left-Turn	328
Slight-Right-Turn	826

Separate training and testing data

The last 1000 time steps of the sequence are used to test network.

```
n = 1000;
XTrain = X(:,1:end-n);
YTrain = Y(1:end-n);

XTest = X(:,end-n+1:end);
YTest = Y(end-n+1:end);
```

This function displays the sequence data colored by its corresponding label.

```
function displaySequence(X,Y)
    figure
    classNames = categories(Y);
    for j = 1:length(classNames)
        label = classNames(j);
        idx = find(Y == label);
        hold on
        plot(idx,X(idx), '*')
    end
    hold off
    legend(classNames, 'Location', 'northwest')
end
```