Assignment for Research and Development / AI

Aravind B (CB.EN.U4CCE22057)
Computer and Communication engineering
Amrita Vishwa Vidyapeetham Coimbatore

Problem Statement

Find the values of unknown variables in the given parametric equation of a curve:

$$x = (t * \cos(\theta) - e^{M|t|} \cdot \sin(0.3t) \sin(\theta) + X)$$

$$y = (42 + t * \sin(\theta) + e^{M|t|} \cdot \sin(0.3t) \cos(\theta))$$

unknowns are

$$\theta, M, X$$

Given range for unknown params is:

$$0\ deg < \theta < 50\ deg$$

$$-0.05 < M < 0.05$$

$$0 < X < 100$$

parameter 't' has range:

$$6 < t < 60$$

Given is the list of points as data file xy_data.csv that lie on the curve for $6 < t < 60$

This data file contains 1500 data points as x and y values.

Trial - I: Desmos Visualization

Initially Desmos an online graphing calculator is used, to understand behavior of the given parametric equation.

The given parametric equations contain three unknown parameters: θ, M, and X.

The parametric equations were typed in Desmos using insert equation option.

The range of the variable t and the three unknowns θ, M, and X were given.

Using the sliding options of the unknowns the behavior of the parametric curve is analyzed.

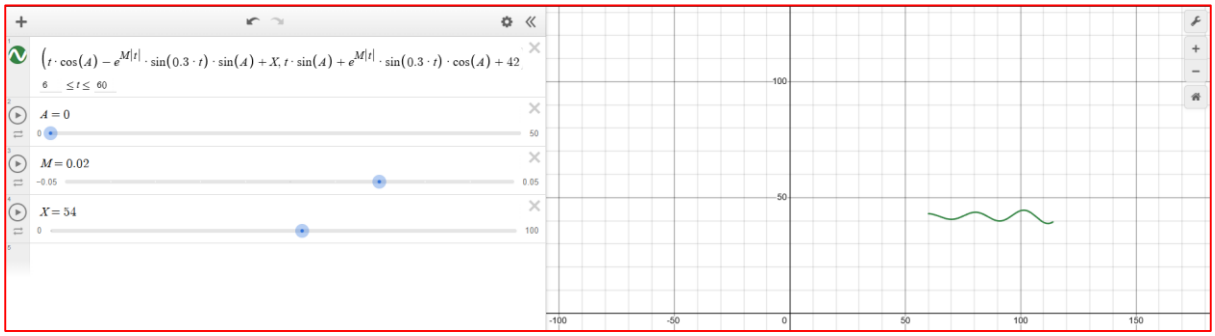
The following observations were made

1. The parameter θ controls the direction of the curve
2. The parameter M affects the magnitude variation resulting in a flat curve for smaller M values
3. The parameter X behaves as a translation constant moving the curve left to right.

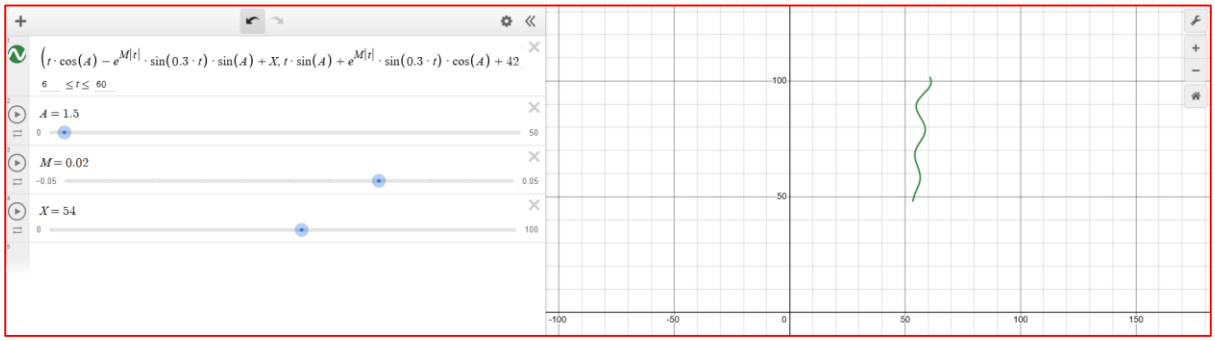
These observations made through Desmos visualization are presented in the following figures

Observation I – The role of parameter θ (by taking the values of M and X as 0.02 and 54)

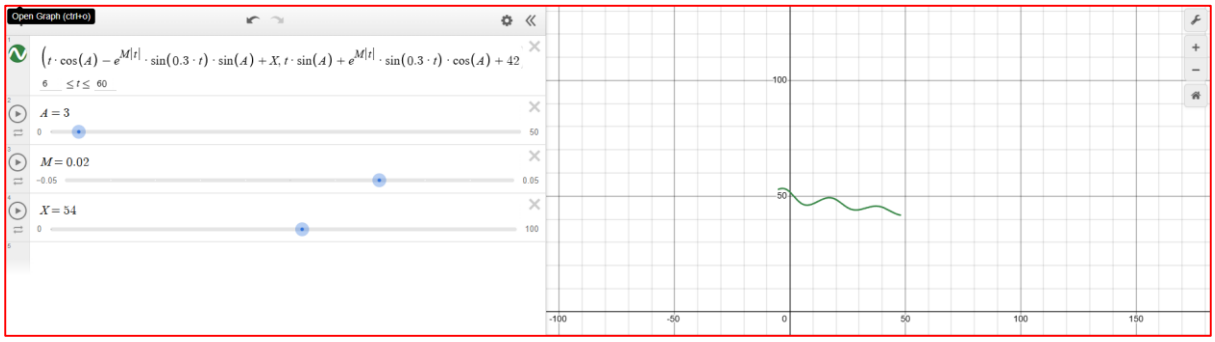
θ = 0



θ = 1.5



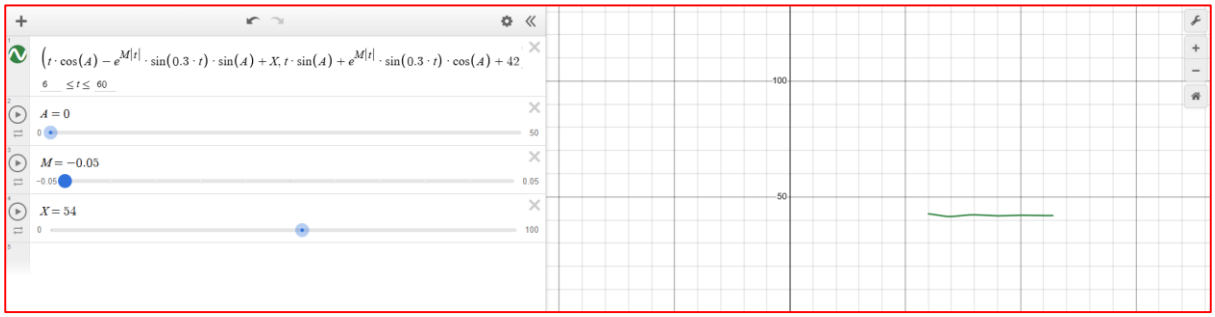
θ = 3



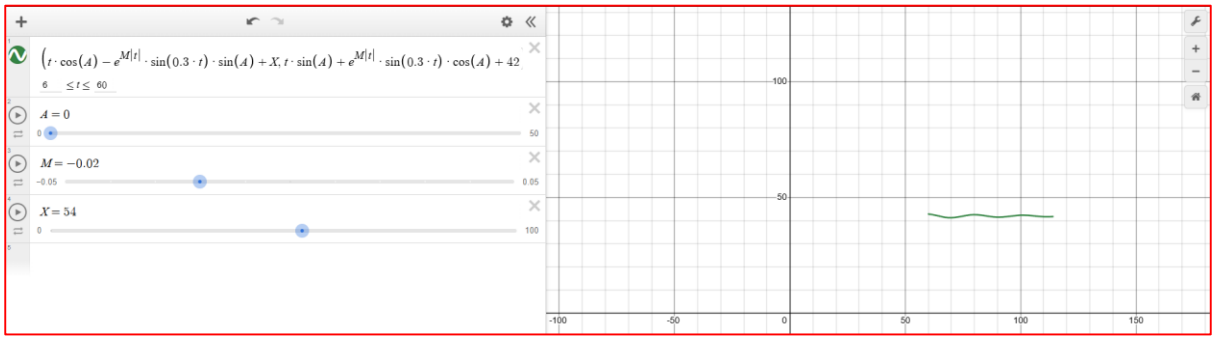
θ = 4.5

Observation II – The role of parameter M (by taking the values of θ and X as 0 and 54)

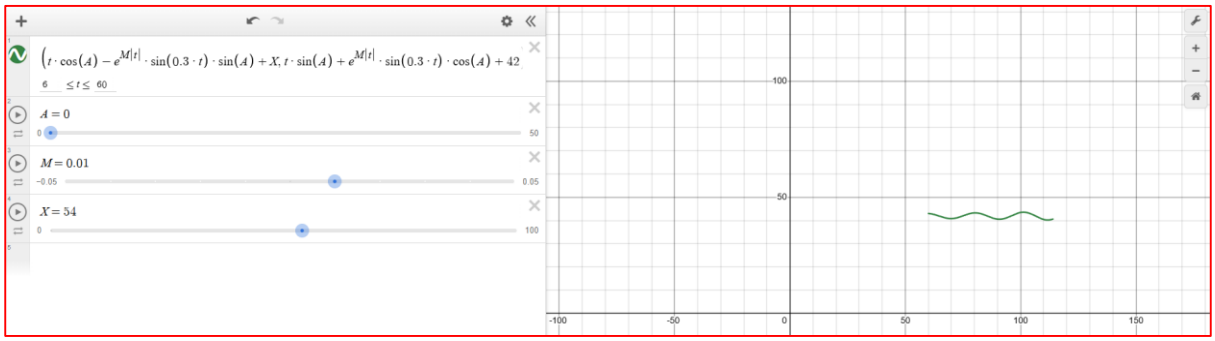
M = -0.05



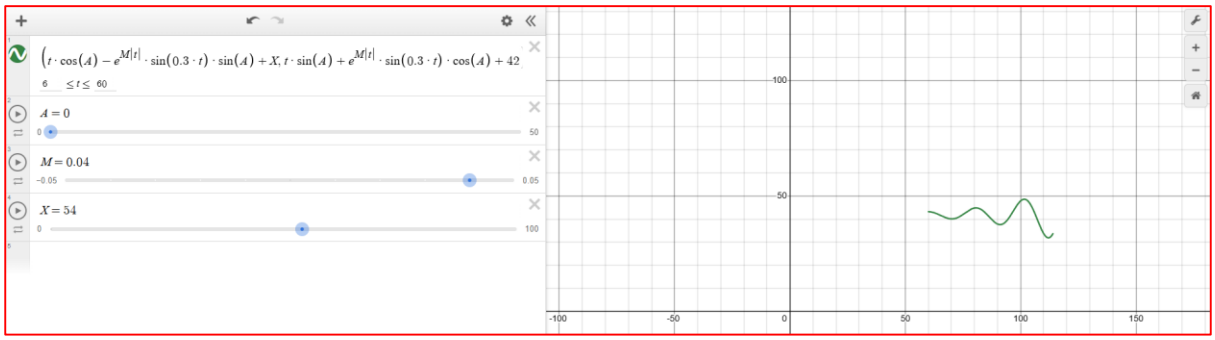
M = -0.02



M = 0.01

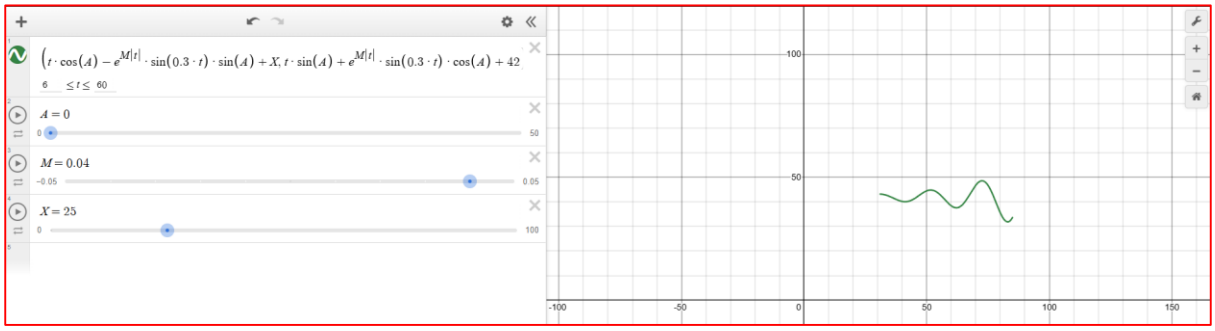


M = 0.04

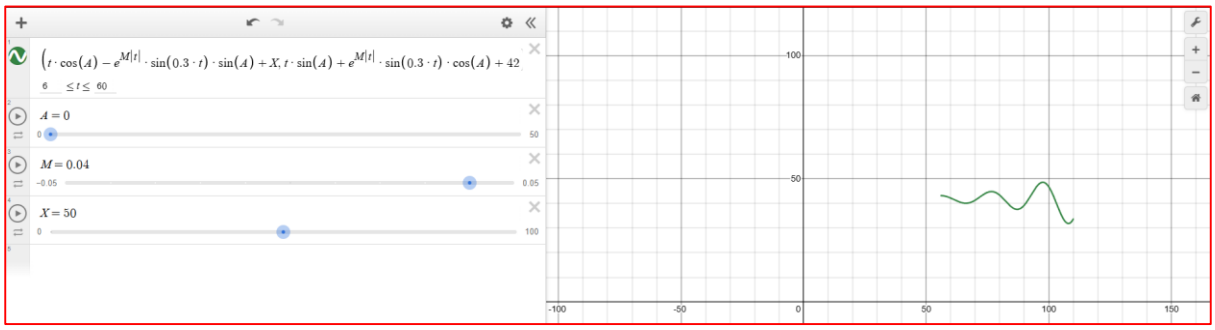


Observation III – The role of parameter X (by taking the values of θ and M as 0 and 0.04)

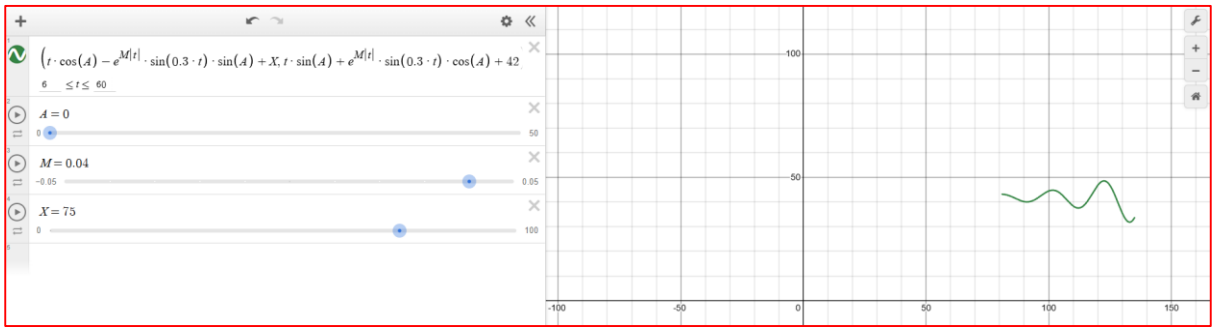
X = 25



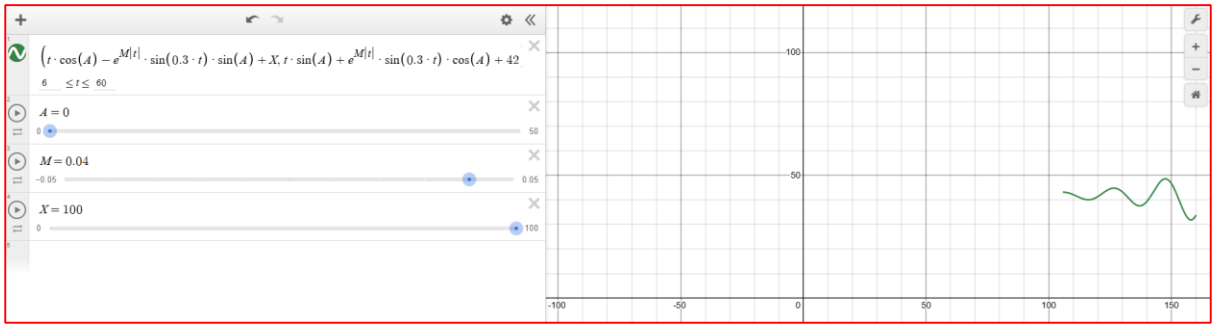
X = 50



X = 75



X = 100



Inference: This helped to understand the behavior of the parametric curve however predicting the unknowns resulting in minimum L1 error is a very cumbersome process.

Trial II: Optimization using Excel

As a next step, Excel spreadsheets is used for optimizing the unknown parameters as well as for calculating the L1 distance between the actual and predicted values. Further, L1 error is also calculated.

Step 1: The given data of x and y are loaded and sorted in ascending order in columns A and B (containing 1500 points).

Step 2: The range of the variable t (6 to 60) is divided into 1500 points with step size of 0.036 in column C.

Step 3: The parametric equation of x is typed in the cell D2 as

=(C2*COS(7)-EXP(0.04*ABS(C2)) *SIN(0.3*C2) *SIN (7) +53)

The formula is copied to D3:D1501

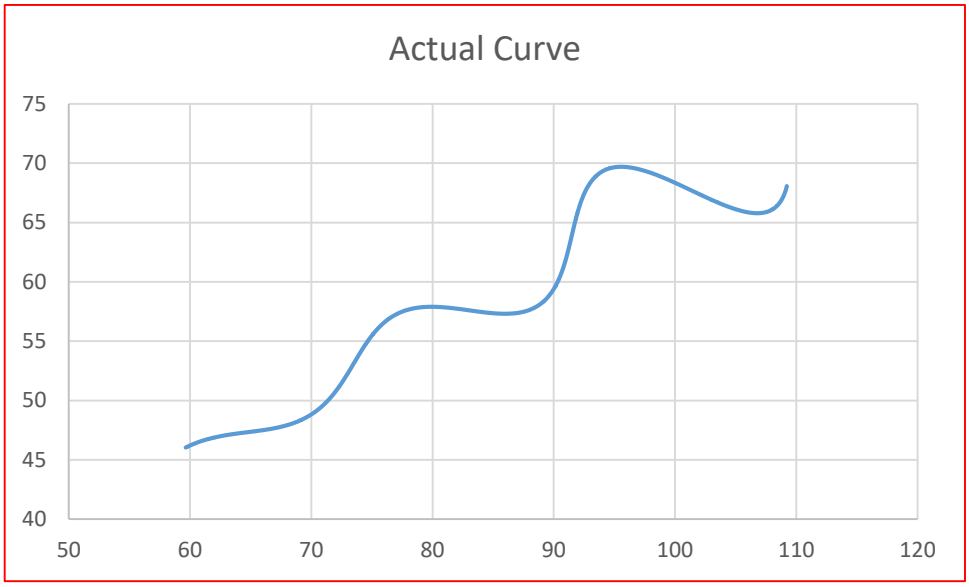
Step 4: The parametric equation of y is typed in the cell E2 as

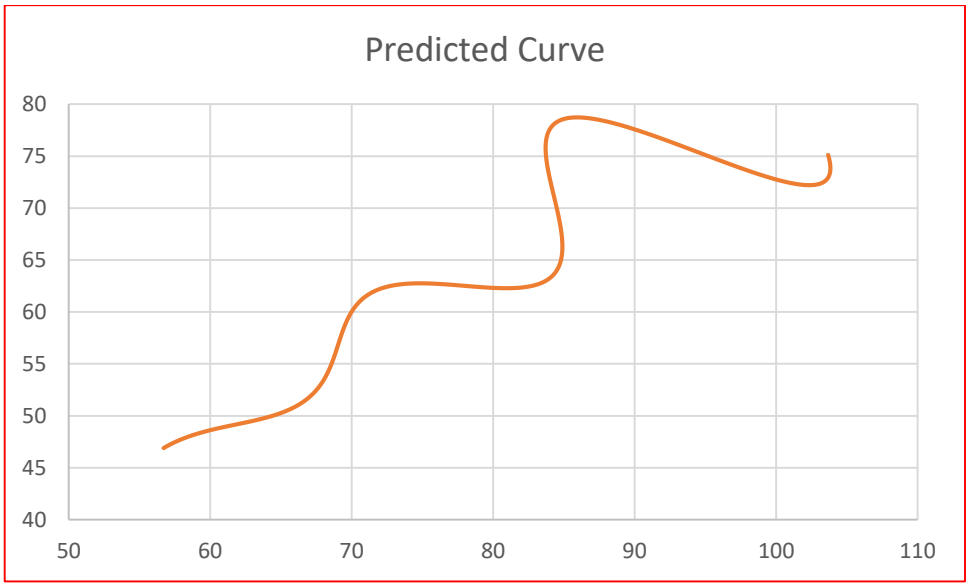
=(C2*SIN(7)+EXP(0.04*ABS(C2)) *SIN(0.3*C2) *COS (7) +42)

The formula is copied to E3:E1501.

For the above steps 3 and 4 the values of θ, M, and X are initially taken as 7, 0.04 and 53. With these values the values of x and y are computed. The actual and predicted curves were plotted as shown below

	A	B	C	D	E	F
1	**Actual_X**	**Actual_y**	**t**	**Pred_x**	**Pred_y**	
2	59.657204	46.032295	6	56.71006216	46.87525282	
3	59.73239	46.07021	6.036	56.73813037	46.89783976	
4	59.79893	46.103054	6.072	56.76629785	46.92031279	
5	59.839375	46.122692	6.108	56.79456474	46.94267173	
6	59.841995	46.123955	6.144	56.82293121	46.96491641	
7	59.84809	46.126892	6.18	56.8513974	46.98704665	
8	59.97834	46.18834	6.216	56.87996344	47.00906232	
9	60.062675	46.226814	6.252	56.90862944	47.03096328	
10	60.113205	46.24939	6.288	56.93739552	47.0527494	
11	60.13468	46.258873	6.324	56.96626176	47.07442058	
12	60.147415	46.26447	6.36	56.99522826	47.09597671	
13	60.1482	46.264812	6.396	57.02429508	47.11741773	
14	60.162483	46.271057	6.432	57.05346229	47.13874354	
15	60.170826	46.274693	6.468	57.08272993	47.15995411	





By comparing the actual and predicted curve it is seen that the optimization of the unknowns is essential

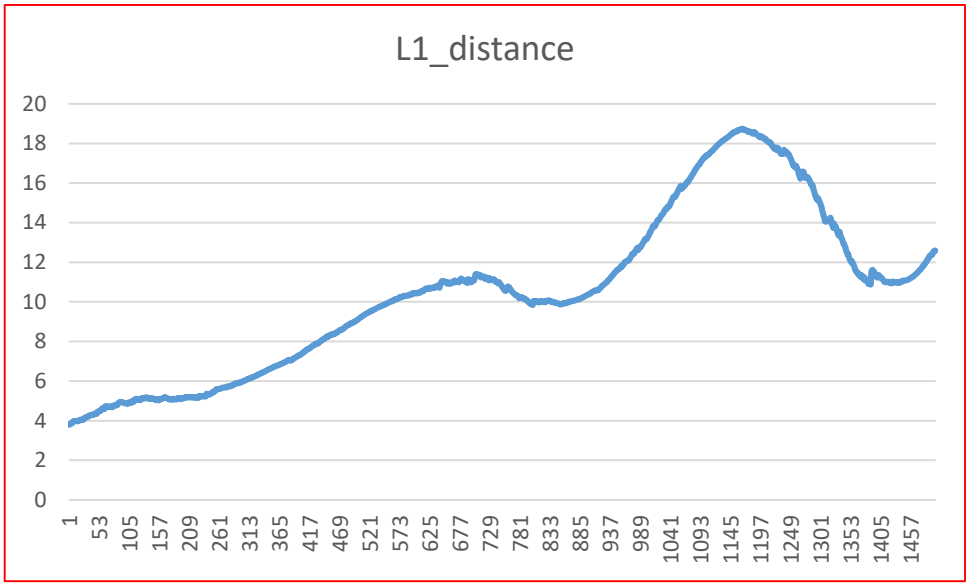
Step 5: In column F2, the L1 distance is calculated using the formula

=ABS(A2-D2)+ABS(B2-E2)

And the same is computed for the range F3:F1501.

The computed L1 values has a minimum of 3.7900 and a maximum of 18.7386

The plot of the same is given below



For a better prediction the L1 distance has to be close to 0.

Step 6: For the computed values of L1 distances the L1 error is computed in cell G2 using the formula

=(1/1500)*SUM(F2:F1501)

resulting in the value of 10.50228.

Excel by default requires the angle values in radian units. Hence in the further computation the angle values are given in radians. By trial and error method the unknown values were suitably changed and an optimal values of the parameters θ, M, and X are obtained as

θ = 0.5235988 radians (30 degrees)

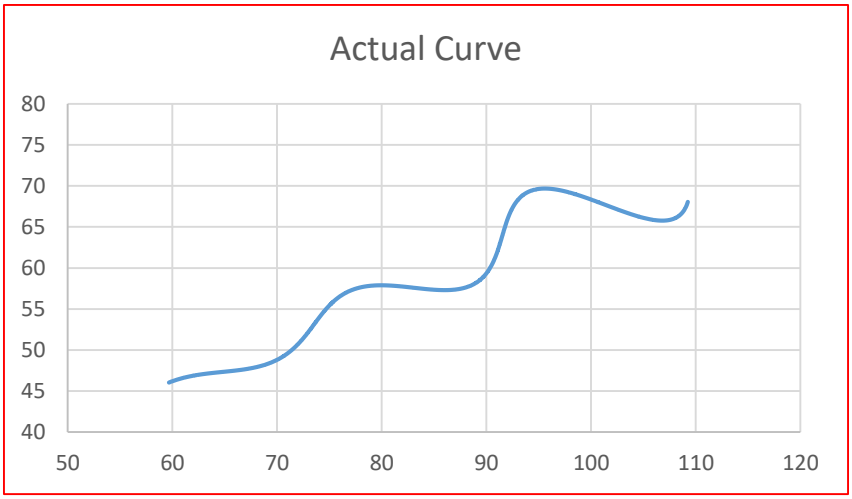
M = 0.0307

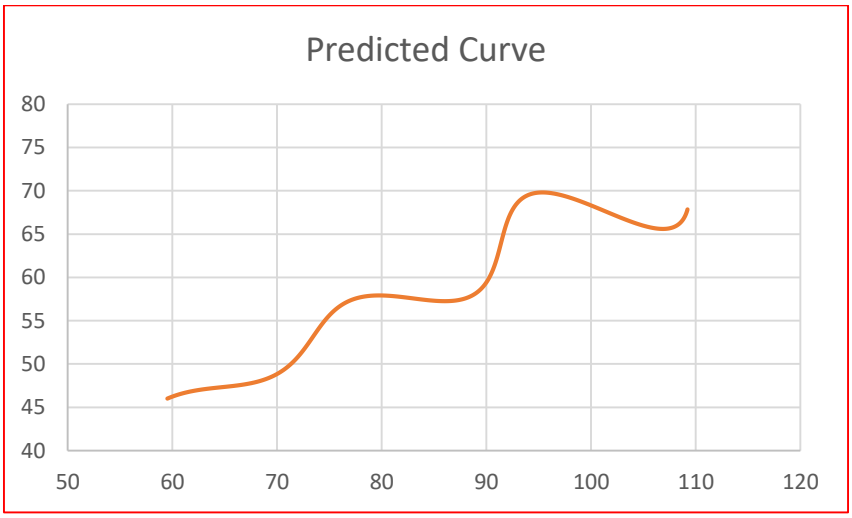
$X = 54.9$

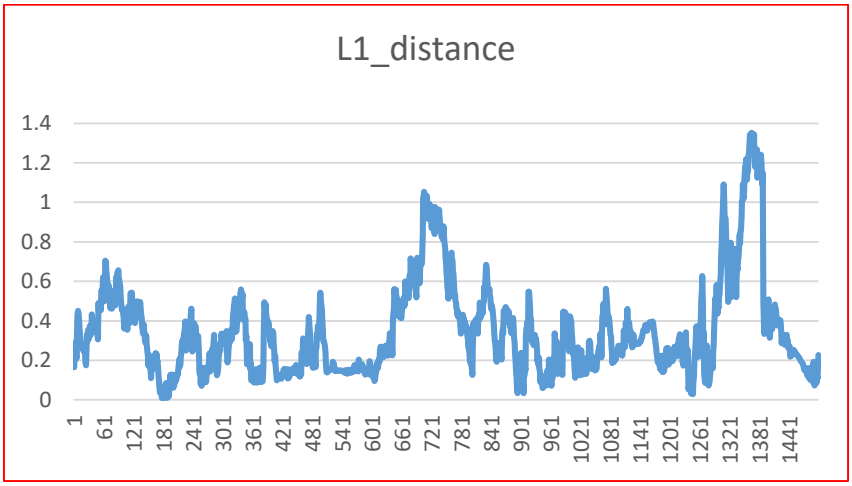
With these optimized values the L1 distance between the actual and predicted values drastically reduced with a minimum and maximum values as 0.00886 and 1.3512 and the L1 error as 0.349711

The corresponding Excel sheet and plots of the same are given below

	A	B	C	D	E	F	G	H
1	**Actual_X**	**Actual_y**	**t**	**Pred_x**	**Pred_y**	**L1_distance**	**L1_error**	
2	59.657204	46.032295	6	59.51074512	46.01395513	0.164798749	0.349711	
3	59.73239	46.07021	6.036	59.54278551	46.03045956	0.229354937		
4	59.79893	46.103054	6.072	59.57489677	46.04684121	0.280246014		
5	59.839375	46.122692	6.108	59.60707896	46.06310002	0.291888014		
6	59.841995	46.123955	6.144	59.63933212	46.07923591	0.247381965		
7	59.84809	46.126892	6.18	59.67165628	46.09524883	0.208076888		
8	59.97834	46.18834	6.216	59.70405145	46.11113875	0.351489801		
9	60.062675	46.226814	6.252	59.73651766	46.12690563	0.426065713		
10	60.113205	46.24939	6.288	59.7690549	46.14254948	0.450990626		







Inference: Though using Excel helps in finding the values of unknowns resulting in minimum L1 error, however the unknown values were obtained using manual tuning which are not based on any optimization methods.

Trial III: Optimization using Python programming

As a next step, to overcome the short comings of using Excel spreadsheet a python code is written with suitable optimization method *viz* Limited-memory Broyden–Fletcher–Goldfarb–Shanno with bounds popularly known as L-BFGS-B method

The developed python code:

```
Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.optimize import minimize
```

```
Loading the actual data from the csv file
df = pd.read_csv(r'E:\xy_data.csv')
x_data = df.iloc[:, 0].values
y_data = df.iloc[:, 1].values
n_points = len(x_data)

print("Data loaded:")
print(f"Points: {n_points}, X: [{x_data.min():.1f}, {x_data.max():.1f}], Y:
[{y_data.min():.1f}, {y_data.max():.1f}]")
```

Data loaded:
Points: 1500, X: [59.7, 109.2], Y: [46.0, 69.7]

```
Defing the parametric equations and L1 computation
def parametric_eqn(t, theta, M, X):
    x = t * np.cos(theta) - np.exp(M * np.abs(t)) * np.sin(0.3 * t) *
np.sin(theta) + X
    y = 42 + t * np.sin(theta) + np.exp(M * np.abs(t)) * np.sin(0.3 * t) *
np.cos(theta)
    return x, y


def l1_error(params):
    theta, M, X = params
    t = np.linspace(6, 60, 1500)
    x_curve, y_curve = parametric_eqn(t, theta, M, X)

    total_error = 0
    for i in range(n_points):
        distances = np.abs(x_curve - x_data[i]) + np.abs(y_curve - y_data[i])
        total_error += np.min(distances)
    return total_error / n_points
```

```
Initialization of the unknowns
theta_initial = np.deg2rad(25)
M_initial = 0.0
X_initial = 50.0

initial_params = [theta_initial, M_initial, X_initial]
initial_error = l1_error(initial_params)

print("Initial setup:")
print(f"θ: {np.rad2deg(theta_initial):.1f} degrees, M: {M_initial}, X:
{X_initial}")
print(f"Initial L1 Error: {initial_error:.4f}")
```

Initial setup:
θ: 25.0 degrees, M: 0.0, X: 50.0
Initial L1 Error: 2.2321

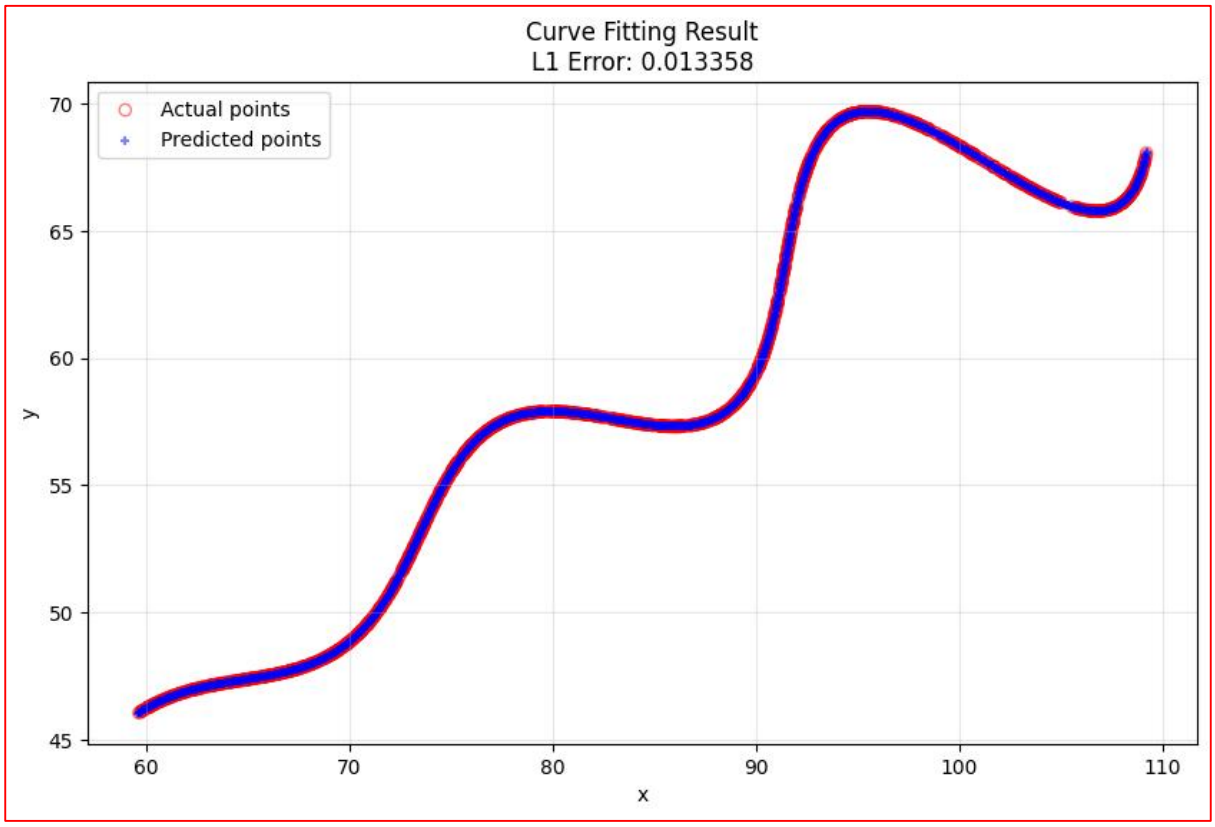
```
Optimization for the bound values of the unknowns
bounds = [(0, np.deg2rad(50)), (-0.05, 0.05), (0, 100)]

print("Optimizing...")
result = minimize(
    l1_error,
    initial_params,
    method='L-BFGS-B',
    bounds=bounds,
    options={'maxiter': 500}
)

theta_opt, M_opt, X_opt = result.x
final_error = result.fun

print("Optimization complete!")
print(f"Final L1 Error: {final_error:.6f}")
```

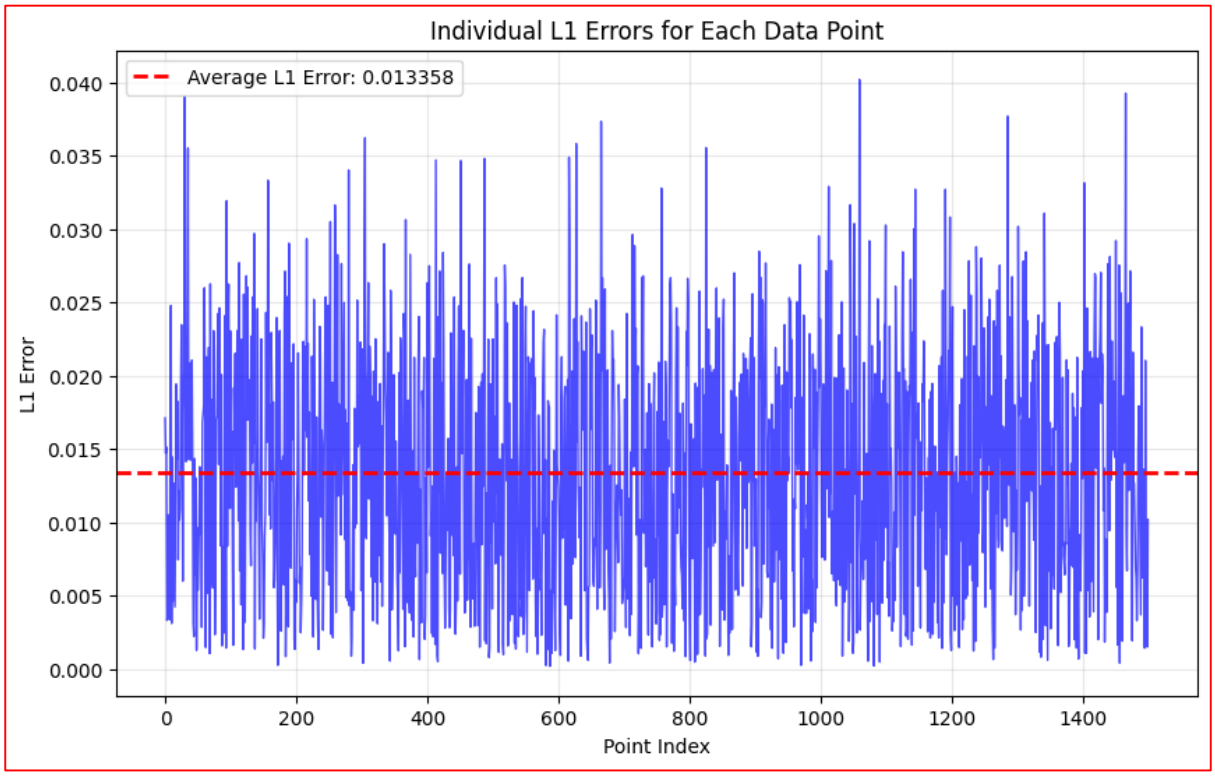
Optimizing...
Optimization complete!
Final L1 Error: 0.013358

```
Printing of the optimized parameters
print("FINAL OPTIMAL PARAMETERS")
print(f"theta = {np.rad2deg(theta_opt):.8f} degrees")
print(f"M     = {M_opt:.8f}")
print(f"X     = {X_opt:.8f}")
print(f"L1 Error = {final_error:.8f}")
```

FINAL OPTIMAL PARAMETERS
theta = 29.99949062 degrees
M = 0.03000126
X = 54.99860335
L1 Error = 0.01335823

```
Plotting of the results
t_plot = np.linspace(6, 60, 1500)
x_curve, y_curve = parametric_eqn(t_plot, theta_opt, M_opt, X_opt)

individual_errors = []
for i in range(n_points):
    distances = np.abs(x_curve - x_data[i]) + np.abs(y_curve - y_data[i])
    individual_errors.append(np.min(distances))

plt.figure(figsize=(10, 6))
plt.plot(range(n_points), individual_errors, 'b-', alpha=0.7, linewidth=1)
plt.axhline(y=final_error, color='red', linestyle='--', linewidth=2,
label=f'Average L1 Error: {final_error:.6f}')
plt.xlabel('Point Index')
plt.ylabel('L1 Error')
plt.title('Individual L1 Errors for Each Data Point')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()

total_l1 = sum(individual_errors)
print(f"Total L1 Error (sum of all points): {total_l1:.8f}")
print(f"Average L1 Error (total / {n_points}): {final_error:.8f}")

plt.figure(figsize=(10, 6))
plt.scatter(x_data, y_data, s=35, marker='o', alpha=0.5, label='Actual
points',facecolors='none', edgecolors='red')
plt.scatter(x_curve, y_curve, s=15, marker='+', alpha=0.5, label='Predicted
points', color='b')
plt.xlabel('x')
plt.ylabel('y')
plt.title(f'Curve Fitting Result\nL1 Error: {final_error:.6f}')
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```


Curve Fitting Result
L1 Error: 0.013358
Actual points
Predicted points
70
65
60
55
50
45
y
60
70
80
90
100
110
x


Individual L1 Errors for Each Data Point
Average L1 Error: 0.013358
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
L1 Error
0
200
400
600
800
1000
1200
1400
Point Index

Inference:

The python code with suitable optimization procedure gives the optimized values of the three unknowns as

θ = 29.99949062 degrees or 0.52358988524 radians

M = 0.03000126

$X = 54.99860335$

L1 Error = 0.01335823

It is to be mentioned that the incremental value for the variable t was taken as a constant value while using Excel spreadsheet approach. However in the python code the choice of t is is also optimized so as to give the minimum error between the actual and predicted values. The following Excel Spreadsheet with optimized t values also gives same L1 error.

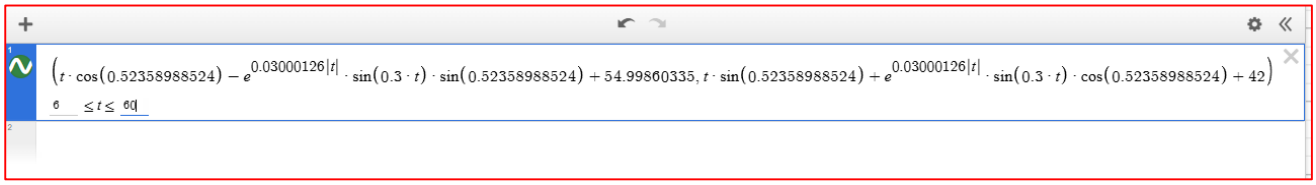
	A	B	C	D	E	F	G	H
1	Actual_X	Actual_y	t	Pred_x	Pred_y	L1_distance	L1_error	
2	59.657204	46.032295	6.036024	59.64390633	46.02616795	0.019424718	0.013358227	
3	59.73239	46.07021	6.144096	59.74054917	46.07492163	0.012870803		
4	59.79893	46.103054	6.216144	59.80533132	46.10681164	0.010158956		
5	59.839375	46.122692	6.252168	59.83782852	46.12257282	0.001665661		
6	59.841995	46.123955	6.252168	59.83782852	46.12257282	0.005548661		
7	59.84809	46.126892	6.252168	59.83782852	46.12257282	0.014580661		
8	59.97834	46.18834	6.396264	59.96852473	46.18439225	0.013763011		

The final Equation for the given problem is:

LaTeX format:

\left(t\cdot\cos\left(0.52358988524\right)-e^{0.03000126\left|t\right|}\cdot\sin\left(0.3\cdot t\right)\cdot\sin\left(0.52358988524\right)+54.99860335,\ t\cdot\sin\left(0.52358988524\right)+e^{0.03000126\left|t\right|}\cdot\sin\left(0.3\cdot t\right)\cdot\cos\left(0.52358988524\right)+42\right)

Desmos screenshot:



Excel format:

X_value=(C2*COS(0.52358988524)-EXP(0.03000126*ABS(C2))*SIN(0.3*C2)*SIN(0.52358988524)+54.99860335),
Y_value=(C2*SIN(0.52358988524)+EXP(0.03000126*ABS(C2))*SIN(0.3*C2)*COS(0.52358988524)+42)