Automated spline trajectory planning for offline robot programming

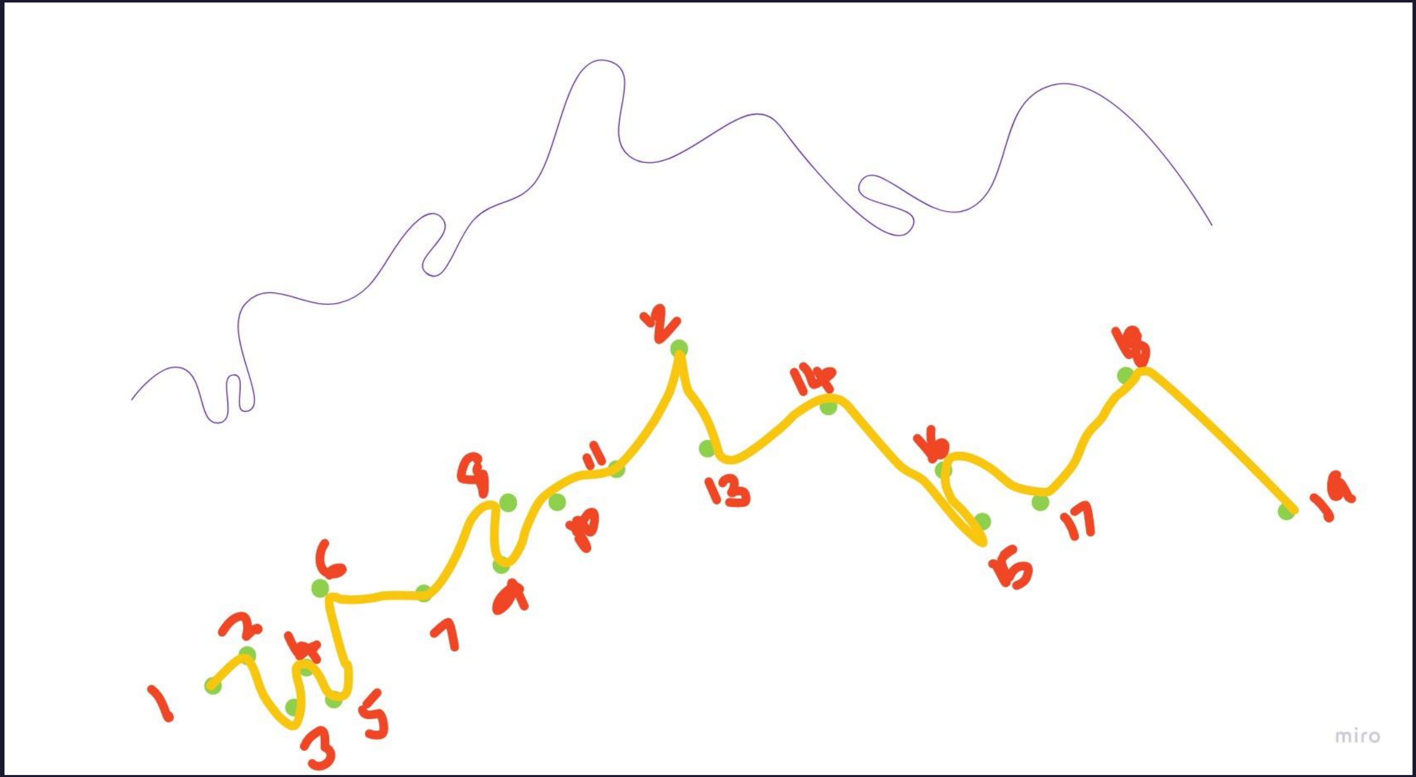
Team:

- Dheeraj Tippani - 226263
- Sravani Dhara - 229753
- Kiran Babu - 229808

Supervisor:
M.Sc.Nadia Schillreff

Examiner:
Prof. Dr. rer. nat. Frank Ortmeier

Connect the dots



Motivation

$F\ (curve) = G(Waypoints)$

- To make a robot move in a desired curved path
- Currently Robot's acceptable inputs
  - Teach the motion to robot manually - Online programming.
  - Convert curved line into points and input into PTP.

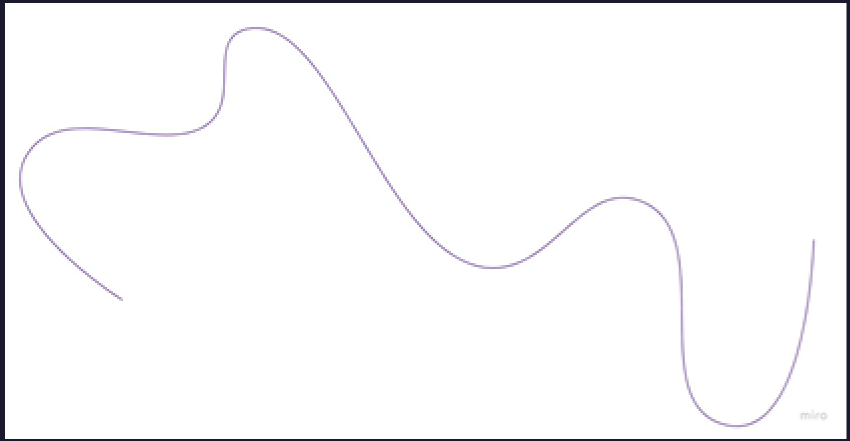


Fig.1 : CAD Curve

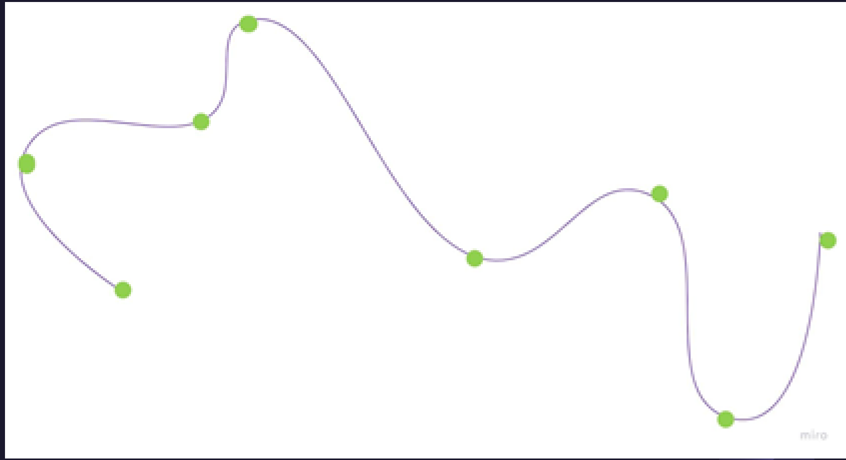


Fig.2 : Way points (Positions of way points such that above CAD curve can be replicated)

Goal

KUKA(waypoints) = curve

- Select "Ideal way points" that lie on curve, input them to the Robot's spline function, such that resulting spline matches our desired curved path.
    - To do this, one needs to know how robot's spline path generation function works.
- Goal: This project tries to achieve a function that can generate a spline path that is same as path generate by Robot's hidden spline function.

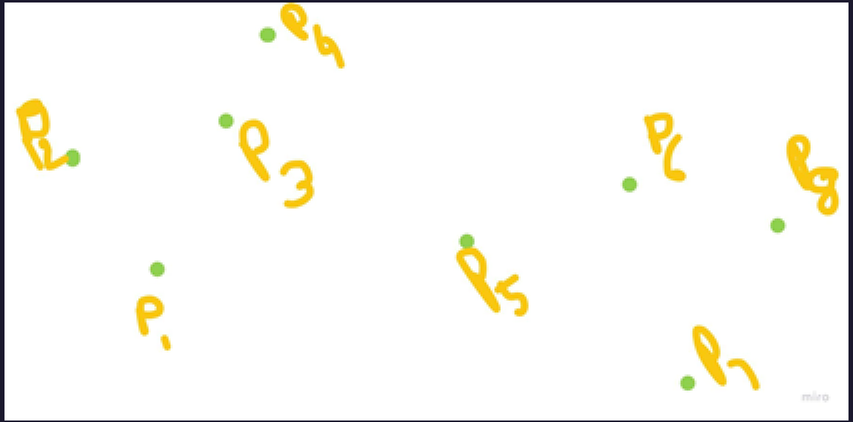


Fig.3 : Input spline points(way points)

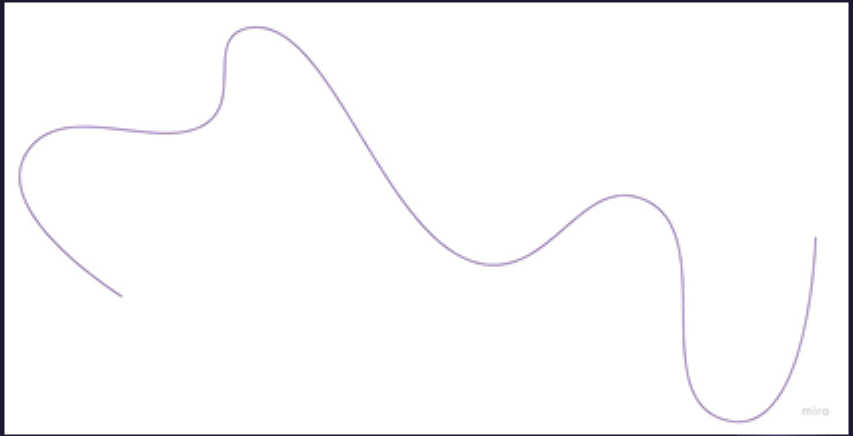


Fig.4 : Generates spline from input way points

Problem : How to find Hidden spline function H.

- H function defines the internal mapping from way points to spline curve.
- If H function is known the way points can be adjusted to create desired curve.

Step to solve the problem:

- Scrutinize the search space of the Hidden spline function.
- Design set of experimental spline trajectories.
- Conduct experiments to collect data for further analysis.
- Analyse the collected data.
- Model a function to mimic KUKA's spline function.

Solution search space

- It is vast but already known characteristics of KUKA spline are
  - Localized with 2 neighbors
  - Interpolation
  - 5-degree polynomial

FRAME	X	Y	Z
P1	0	0	0
P2	100	0	0
P3	102	0	0
P3'	102	**1**	0
P4	104	0	0
P5	204	0	0

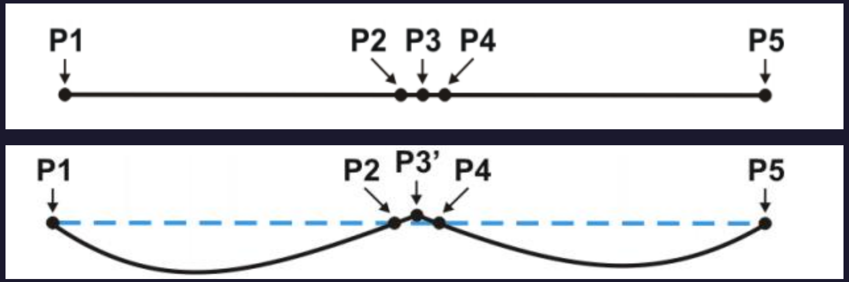


Fig.5 : Depiction of the impact of one point change in the spline for up to 2 neighbor segments. [1]

Experiment setup & Post processing

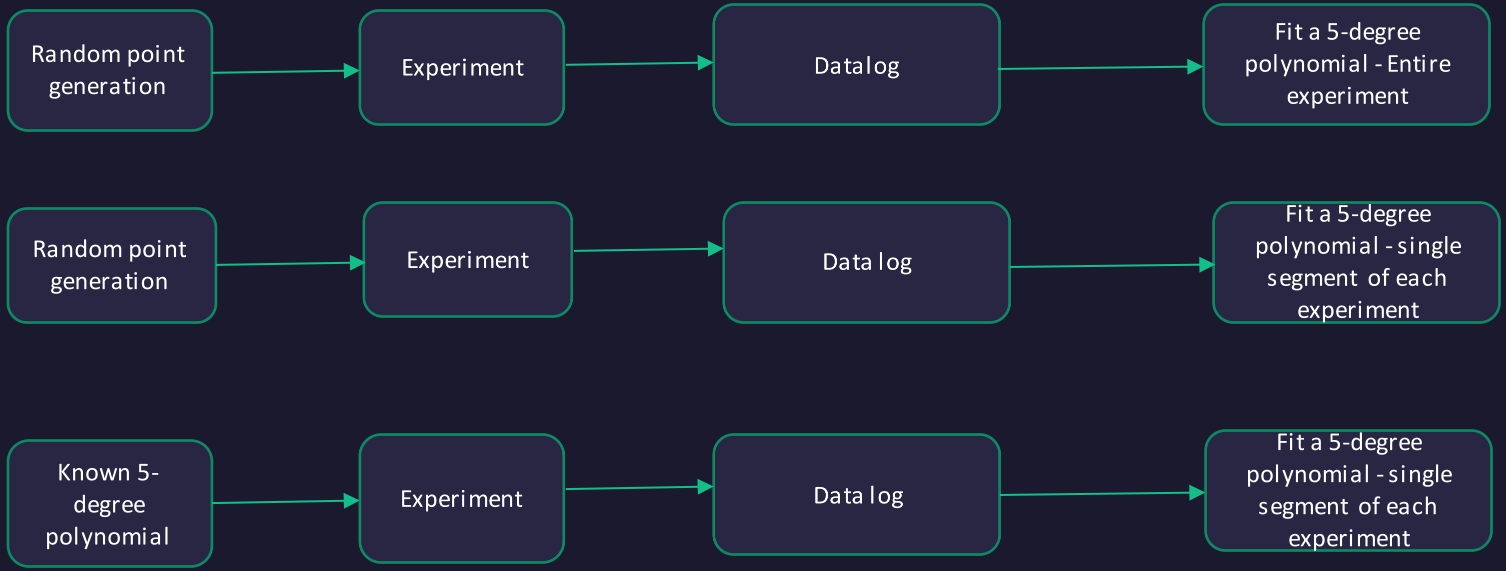
- To analyze the spline trajectories of KUKA robot, we had to instruct the robot to go through some arbitrarily chosen waypoints in a spline motion.
- Waypoints were fed in the robot program in the form of joint configurations.
- The output from the robot is a log of all the joint angle configurations, recorded during the experiment.
- The resultant output is converted back to 3D cartesian points for analysis.

Analysis - 1

Experiment 1

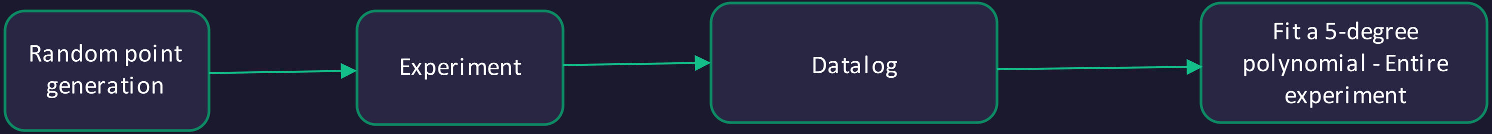
Experiment 2

Experiment 3



Analysis - 1

Experiment 1



- From the data log, We know the exact points in 3D space that the robot has passed through in each experiment.
- We have performed 5th degree polynomial interpolation on the array of each of the co-ordinate axes, i.e, x, y and z with respect to time t.
- To check if a 5th degree polynomial can explain all the complexity in a trajectory, we have generated a plot from the polynomials we have obtained from the polynomial interpolation.
- We realized that the 5th degree polynomial is too 'simple' to capture all the complexity of an entire spline block.
- The first row of plots in the image shown here are the normalized plots of change from start of the spline block to the end of spline block.
- The second row of plots in the image show trajectory generated from 5th degree polynomials, that are in turn generated from the robot data log itself.

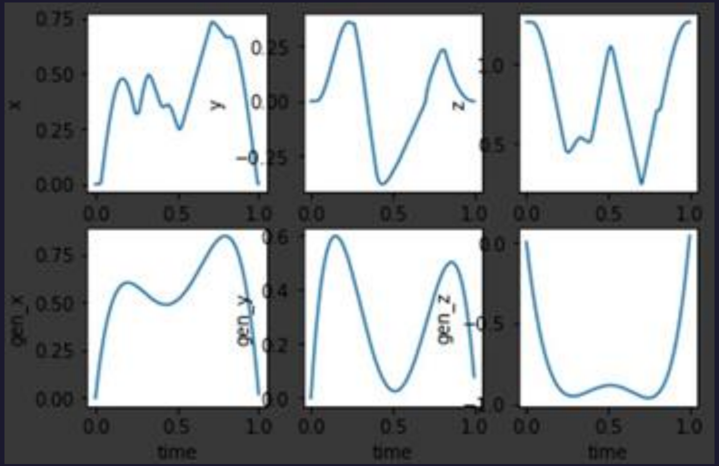
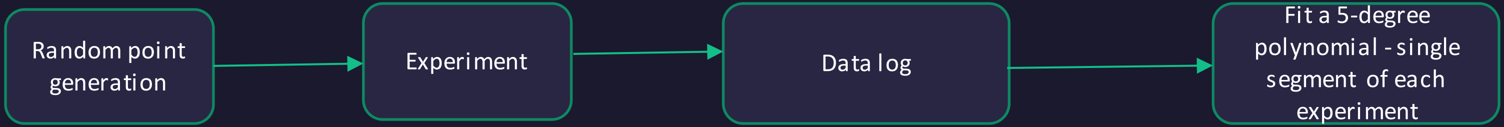


Fig.6 : Experiment: h - p1 - s(p2 p3 p4 p5 p6 p7) - h

Analysis - 1

Experiment 2



- In this set of analyses, we again tried to find if the 5th degree polynomial is good enough to capture all the complexity of the robot's trajectory.
- We have performed similar analysis as in the previous experiment, but this time on only a single segment of all the experiments, i.e, segment P2-P3 that is common in all the experiments.
- Same start and end points yielded in drastically varying spline trajectories due to variations in neighboring points.
- There is no significant difference between the first three and the last two plots, because of the common immediate neighbors in these experiments.

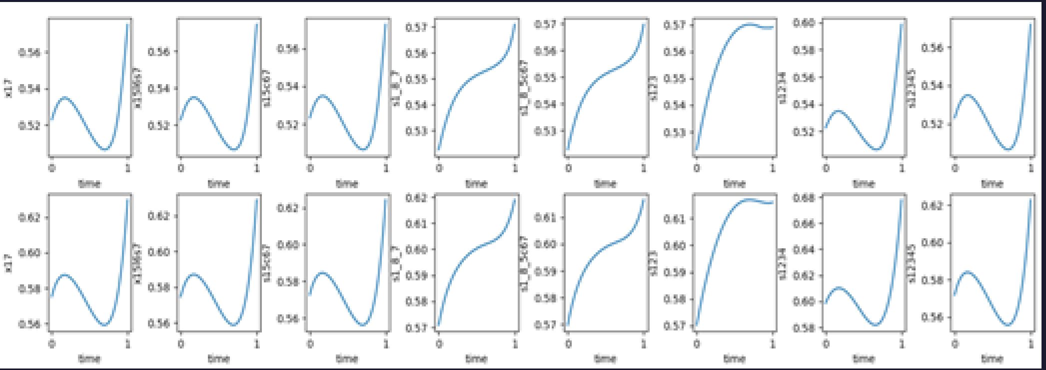


Fig.7 : X-coordinate in segment P_2-P_3 in all experiments

Analysis - 2

Experiment 1

Input : Start angle and end angle → 5-degree spline path → Resultant trajectory → Compare with data log

Experiment 2

Input : Start & end acceleration, velocity → 5-degree spline path → Resultant trajectory → Compare with data log

Analysis - 2

- The trajectories are computed using the following information as inputs,[2]
    - Start position
    - Start velocity
    - Start acceleration
    - End position
    - End velocity
    - End acceleration

$$x(t) = a_0 t^0 + a_1 t^1 + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

$$t = 0$$

$$x(t)_0 = a_0 = x_{start}$$

$$V(t)_0 = a_1 = V_{start}$$

$$a(t)_0 = a_2 = astart / 2$$

$$\begin{bmatrix} t^3 & t^4 & t^5 \\ 3t^2 & 4t^3 & 5t^4 \\ 6t & 12t^2 & 20t^3 \end{bmatrix} \times \begin{bmatrix} a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} x_e - x_s - V_s t - \frac{a_s t^2}{2} \\ v_e - V_s - ast \\ a_e - a_s \end{bmatrix}$$

Fig.8: Formulas

Analysis - 2

- Red dots - Algorithm Generated trajectory
- Blue line – Robot Trajectory

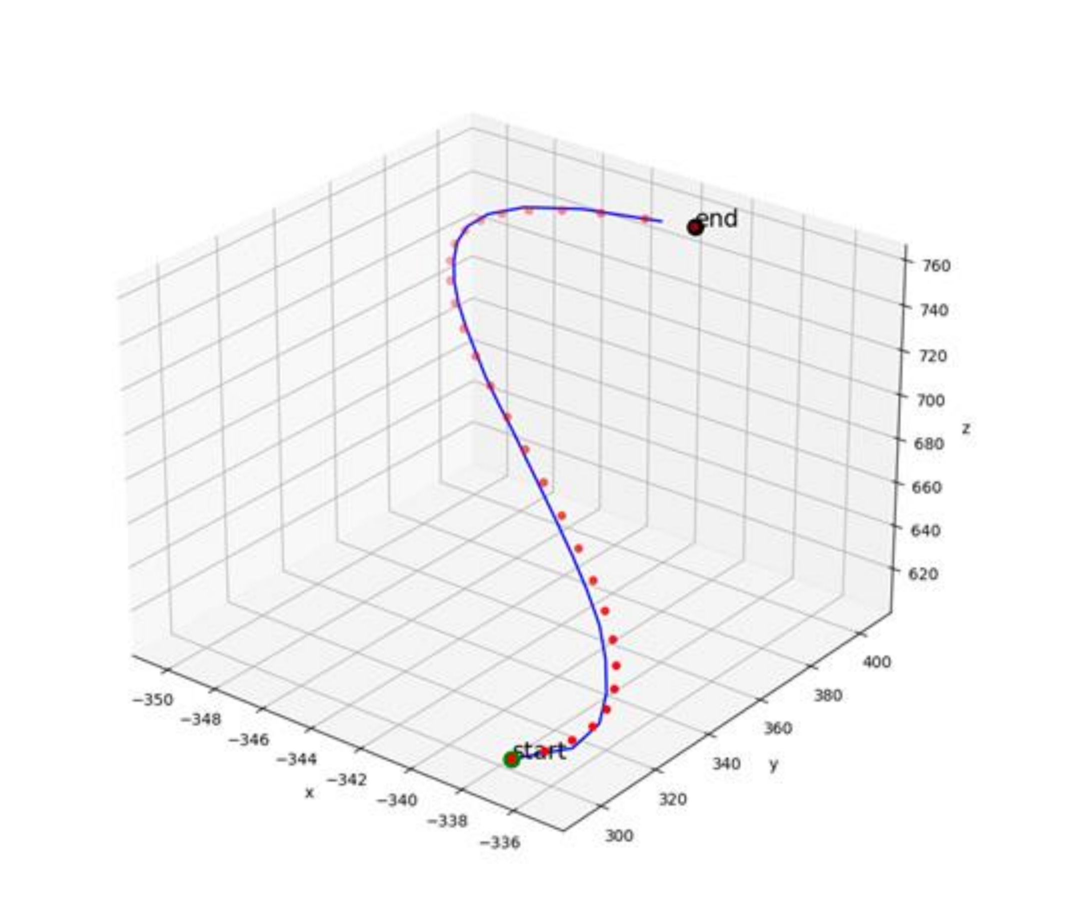


Fig. 9 : The trajectory generated using the quintic path planning algorithm vs the trajectory of the robot.

Analysis – 3[3]

Input : Spline points → First derivative → Second derivative → Generate 5-degree Quintic Beizer

Quintic B´ezier:

$$S(t) = (1-t)^5 P_0 + 5(1-t)^4 t P_1 + 10(1-t)^2 t^3 P_2 + 10(1-t)^3 t^2 P_3 + 5(1-t) t^4 P_4 + t^5 P_5$$

With $0 \le t \le 1$ and for Way points Wi and Wi+1 the spline segment has following co-efficient :

$P_0 = W_i$
$P_1 = (1/5 * T_i) + P_0$
$P_2 = (1/20 * A_i) + 2 * P_1 - P_0$
$P_3 = (1/20 * A_{i+1}) + 2 * P_4 - P_5$
$P_4 = P_5 - (1/5 * T_{i+1})$
$P_5 = W_{i+1}$ where T_i and A_i are Tangent and acceleration at W_i

Analysis – 3

Input : Spline points → First derivative → Second derivative → Generate 5-degree Quintic Beizer

Tangent Magnitude

- Half of minimum of neighbor vectors

Tangent Direction

- Perpendicular to the angular bisector of neighboring segments

Acceleration magnitude

- Double derivative of cubic spline of neighbor segments

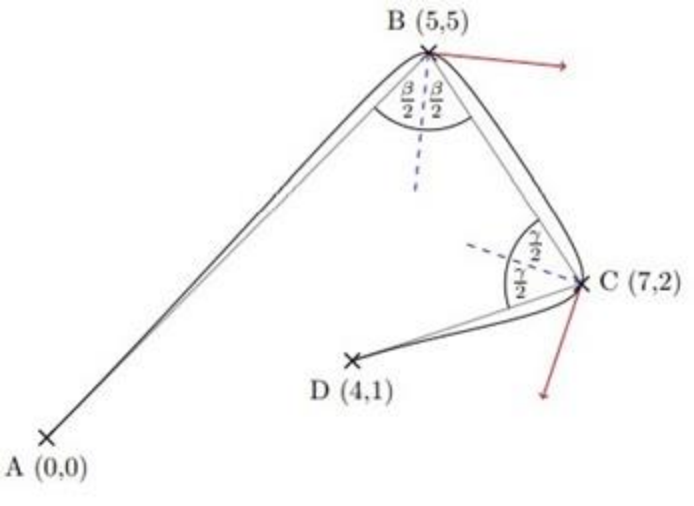


Fig.10 : Tangent Direction, source : Sprunk, Christoph. "Planning Motion Trajectories for Mobile Robots Using Splines." (2008).

Results

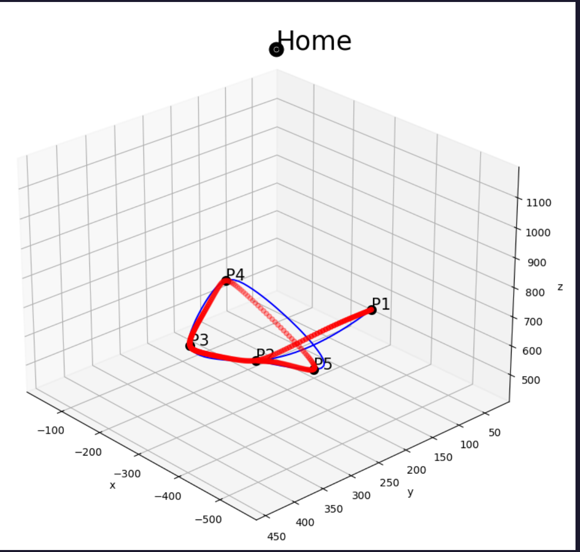


Fig.11 :Experiment.1: home(H)1234567H

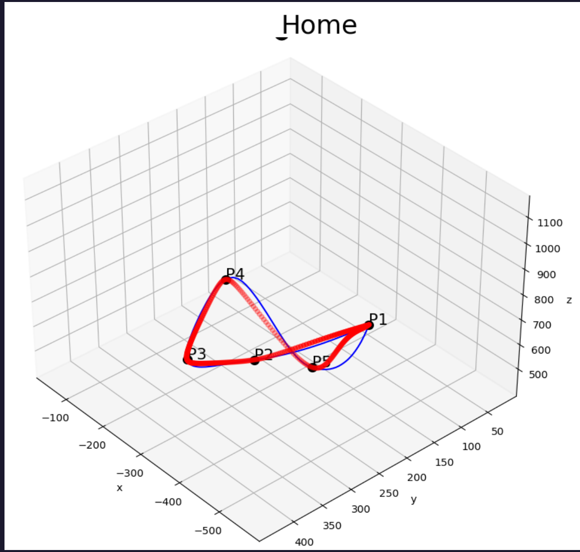


Fig.12 : Experiment.2: H123451H

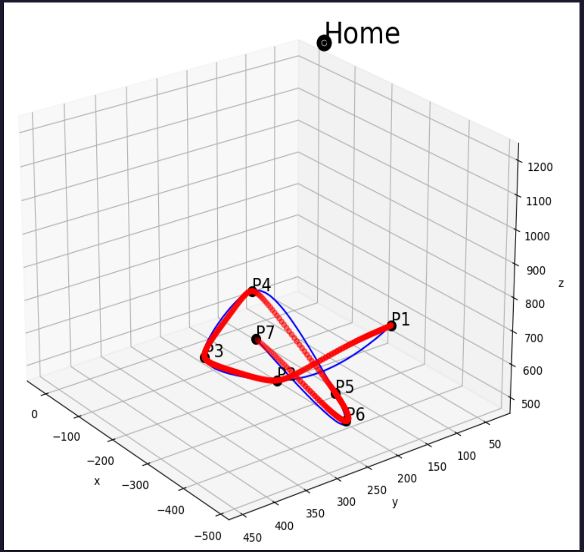


Fig.13 : Experiment.3: H123452H

Optimization Variables

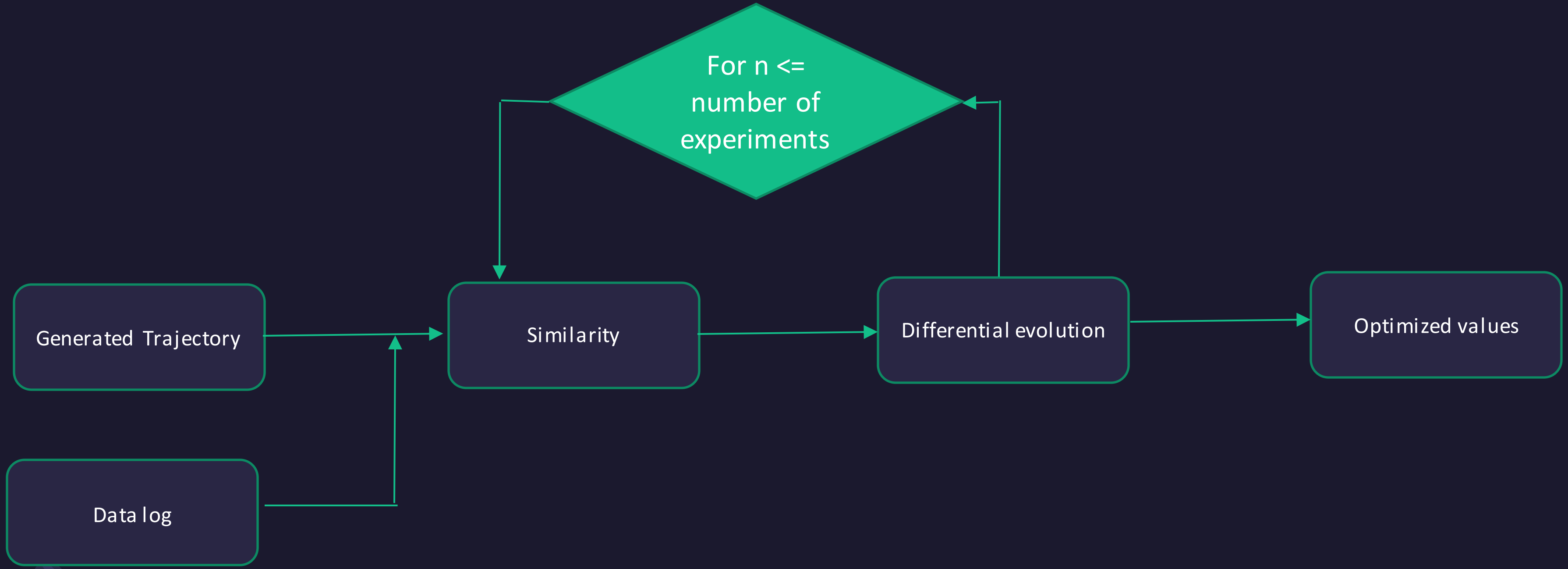
Tangent Magnitude

Acceleration magnitude variable 1

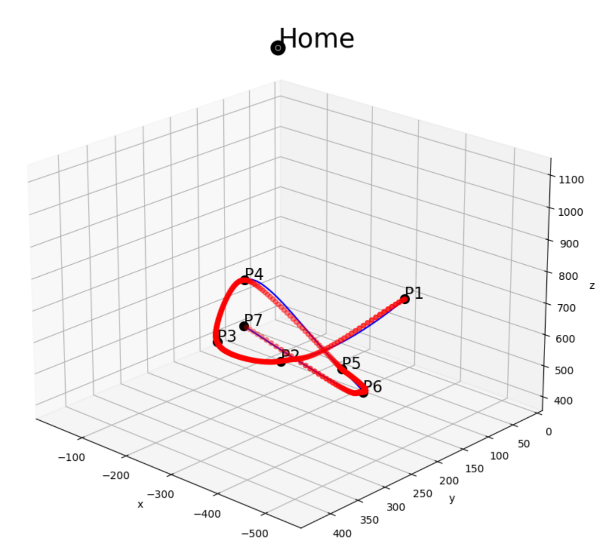
Acceleration magnitude variable 2

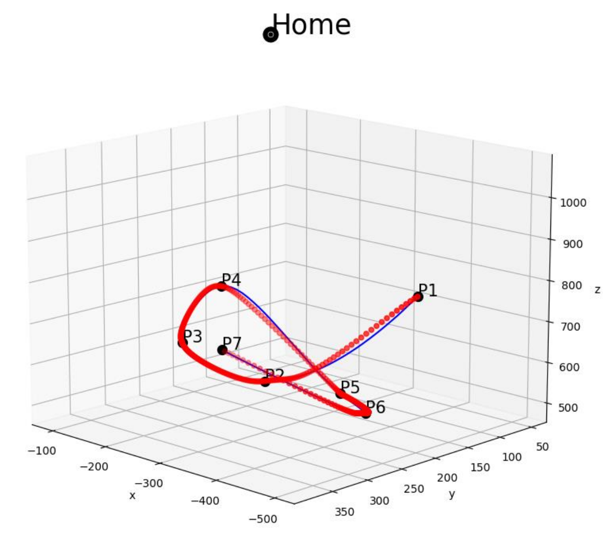
Acceleration magnitude end points

Optimization



Optimization results





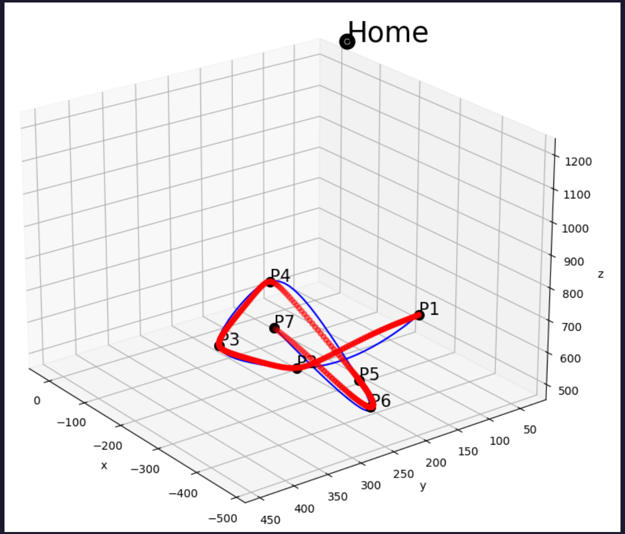


Fig.14 : Trajectory with overall optimized values.
Optimized DTW(Dynamic time warping) distance Score:1836.71

Fig.15 : Trajectory with specific optimized value, optimized for specifically for the experiment.
Optimized DTW distance Score:1822.45

Fig.16 : Trajectory with unoptimized values, DTW distance score :3359.55

Min DTW distance of unoptimized trajectory	Min DTW distance of overall experiments optimized trajectory	Difference	% difference between unoptimised trajecotry and Overall optimised trajectiory
3359.556999	1836.716063	1522.840936	82.91106973
3000.818554	1528.982396	1471.836158	96.26246592
2605.36705	1649.883389	955.4836609	57.9121935
2502.233389	1781.190768	721.0426213	40.48093188
2901.605697	1558.262149	1343.343548	86.207802
2466.975886	1716.378627	750.5972585	43.73144985
3048.87614	1630.656229	1418.219911	86.97234195
3032.369057	1605.065772	1427.303285	88.92490953
3513.337202	1943.838491	1569.498711	80.74223851
3365.414375	1846.840547	1518.573828	82.2254975
2737.791898	1463.700662	1274.091236	87.04588777
2994.561541	1817.795799	1176.765742	64.73585989
2994.917367	1826.240003	1168.677364	63.99363511
1538.02076	661.6721785	876.3485819	**132.4445262**
1519.321947	853.5466808	665.7752666	78.0010375
116.827006	116.827006	0	0
124.8729228	124.8729228	0	0
109.3931614	109.3931614	0	0

Experiment	Control Variable 1	Control Variable 2	Control Variable 3	Control Variable 4
home(H)1234567H	0.51625016	0.21824477	0.09832954	-4.70750856
H123451H	0.49150283	0.29449145	0.0030767	-3.01724061
H123452H	0.93627083	0.84603186	0.97434401	-2.76241202
H123453H	0.43832923	0.06750762	0.0241364	-0.99963715
H123456H	0.65109629	0.38521984	0.3804515	-4.61705169
H123457H	0.40909474	0.01349362	0	-1.3055848
H123458H	0.69497262	0.4831442	0.40740118	-4.15549924
H123459H	0.6555418	0.5402687	0.28669777	-4.48028927
H1231567H	0.51458768	0.14121976	0.16821533	-4.74459839
H1234567H	0.51486911	0.19911942	0.11412941	-4.71831165
H1237567H	0.51137583	0.12729807	0.04581003	-4.33867638
H1238567H	0.42161375	0	0.17204282	-5.1465091
H1238567H	0.4292206	0	0.11627169	-4.94535119
H134H	1	0	1	-1.31121945
H123H	1	0	1	-0.9397716
H12H	0.31531583	0.82849286	0.4007474	1.40770912
H23H	0.13927194	0.19206982	0.25749711	-2.44304374
H34H	0.56966978	0.74041517	0.72877876	8.43227581
Globally optimized variable set	0.4898907210	0.2540456360	0.0038133780	4.5758553200

References

1. https://www.oir.caltech.edu/twiki_oir/pub/Palomar/ZTF/KUKARoboticArmMaterial/KUKA_SunriseOS_111_SI_en.pdf
2. Takahashi, Arata, et al. "Local path planning and motion control for agv in positioning." *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems'.(IROS'89)'The Autonomous Mobile Robots and Its Applications*. IEEE, 1989.
3. Sprunk, Christoph. "Planning motion trajectories for mobile robots using splines." (2008).

Thank you