Polynomial_3_4_5_Generalisation_v3

Plot the 3 4 5 polynomial

Init

```
close all
```

P5_position = [0 0 0 10 -15 6]; P5_velocity = [0 0 30 -60 30 0]; P5_acceleration = [0 60 -180 120 0 0]; P5_jerk = [60 -360 360 0 0 0]; P5_position = [0 0 0 p3 p4 p5];

```
P5_acceleration = [0 6*p3 12*p4 20*p5 0 0];
P5_jerk =         [6*p3 24*p4 60*p5 0 0 0];
```

polynomial	**t^0**	**t^1**	**t^2**	**t^3**	**t^4**	**t^5**
position	0	0	0	10	-15	6
velocity	0	0	30	-60	30	0
acceleration	0	60	-180	120	0	0
jerk	60	-360	360	0	0	0

Code avec taskCycleTime as a parameter

```
% Use CEILING   ANY_NUM --> Return the next highest whole number. in PLC
NumberOfPoints = ceil(t*1000/taskCycleTime_ms)
```

Some vectors to store results

```
% Define Matrix fo size 1 ; NumberOfPoints + 1
position_yAxis     = [1;NumberOfPoints+1];
velocity_yAxis     = [1;NumberOfPoints+1];
acceleration_yAxis = [1;NumberOfPoints+1];
jerk_yAxis         = [1;NumberOfPoints+1];
```

T_axis with increment of 2 ms

```
t_xAxis = 0:taskCycleTime_ms/1000:t;
```

Compute motion profile of position

```
for iLoop = 1 : (NumberOfPoints + 1 )
    position_yAxis(iLoop) = (t_xAxis(iLoop)^3 * P5_position(4) + t_xAxis(iLoop)^4 * P5_position(5) + t_xAxis(iLoop)^5 * P5_position(6));
end
figure
plot(t_xAxis,position_yAxis);
title('Position')
```