HACETTEPE UNIVERSITY

MECHANICAL ENGINEERING DEPARTMENT

OMU 493 COMPUTATIONAL MULTI-BODY DYNAMICS

Term Project

Submitted by

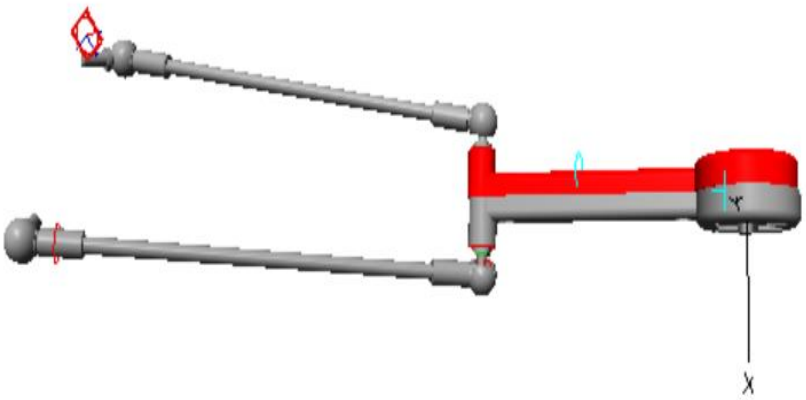
21631211 Cahit Oğuz Saydam

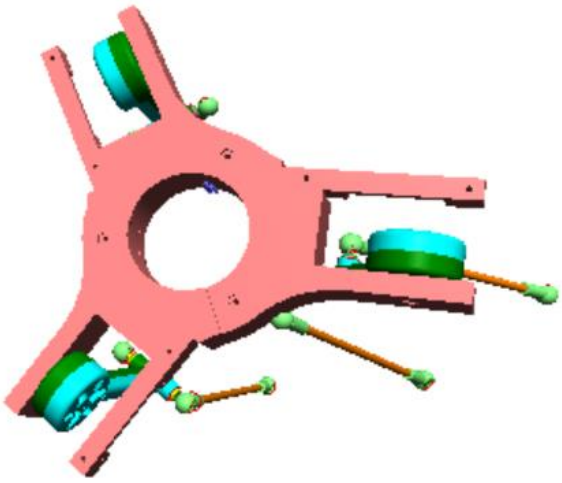
21731334 Fatih Örkmez

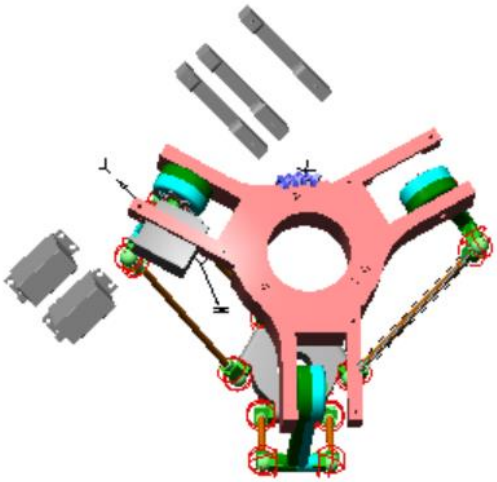
21631076 Serkay Güney

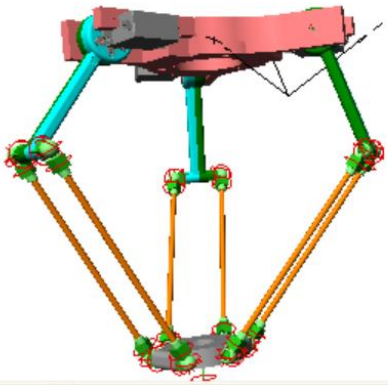
JUNE 2023

DESIGN AND ASSEMBLY PROCESS





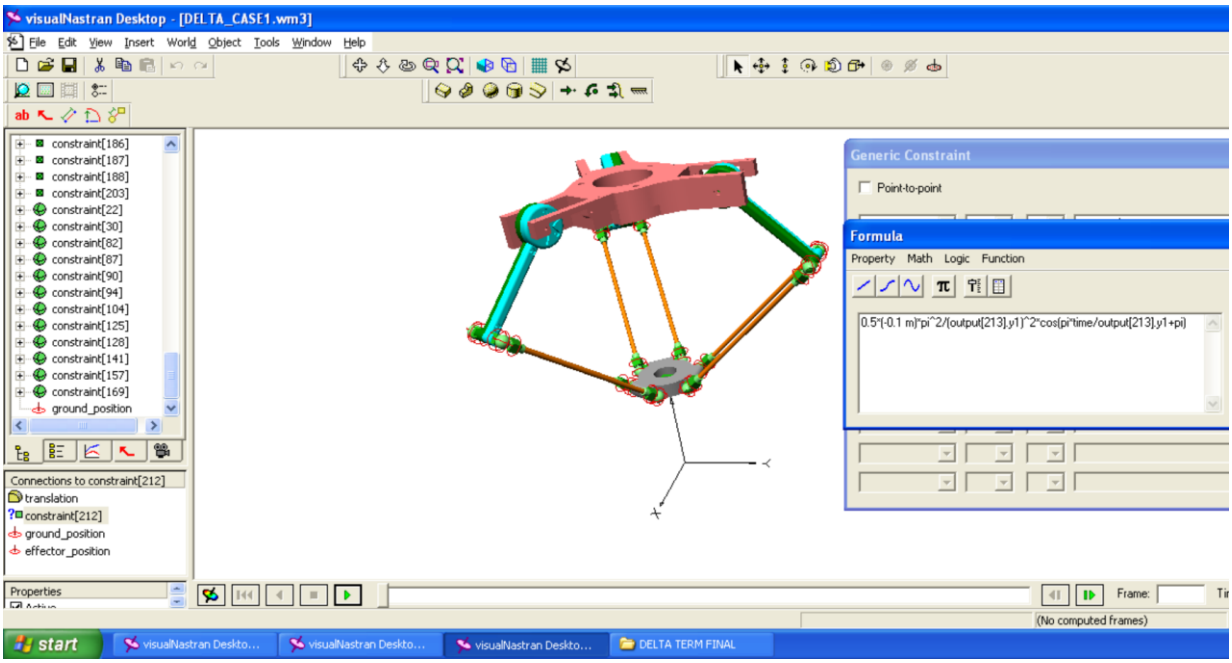




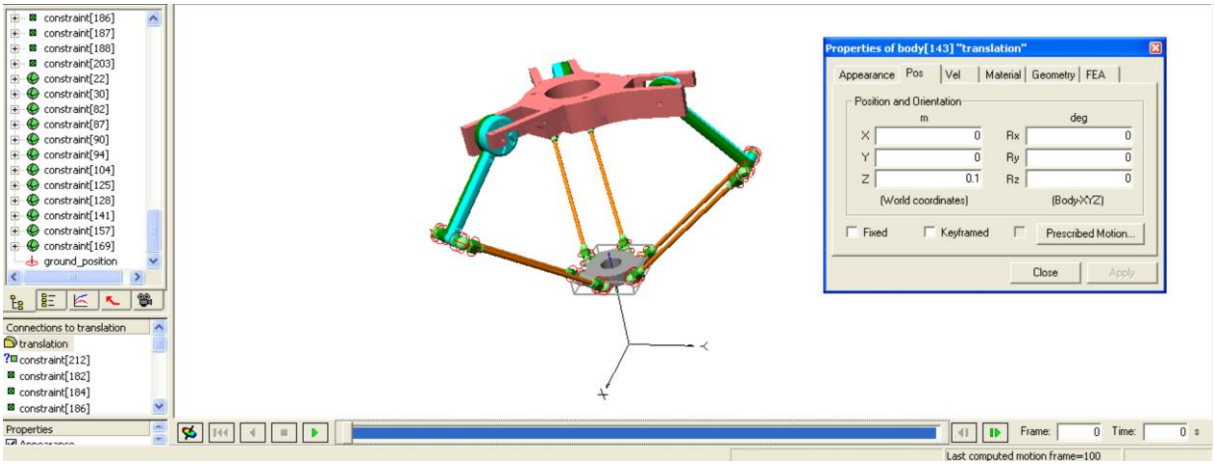
SIMULATION

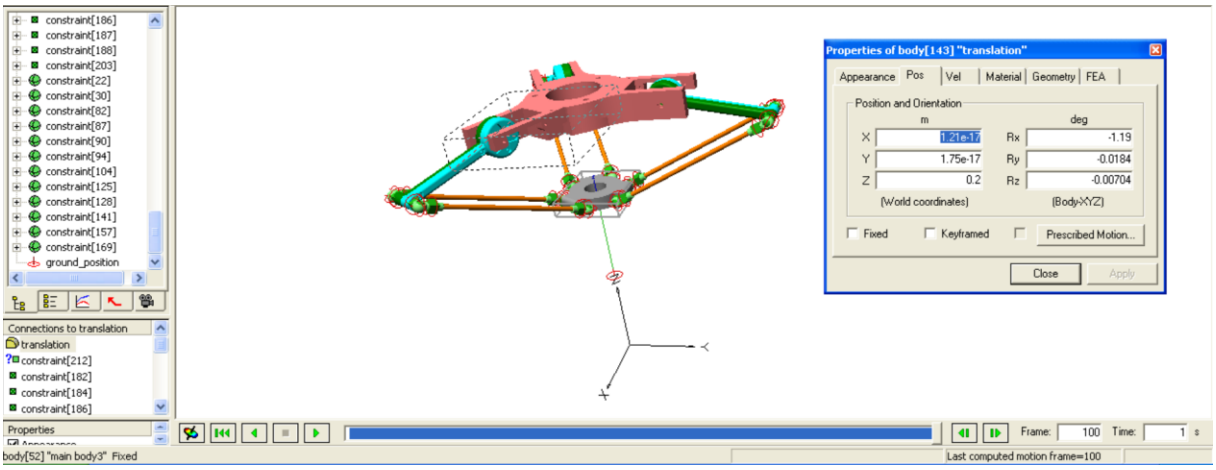
Case 1

Here, we obtained the position change of the effector part.



As a result: we moved the effector part 0.1 m in the z direction. And we track the position change form time 0 to time 1second as we see below:

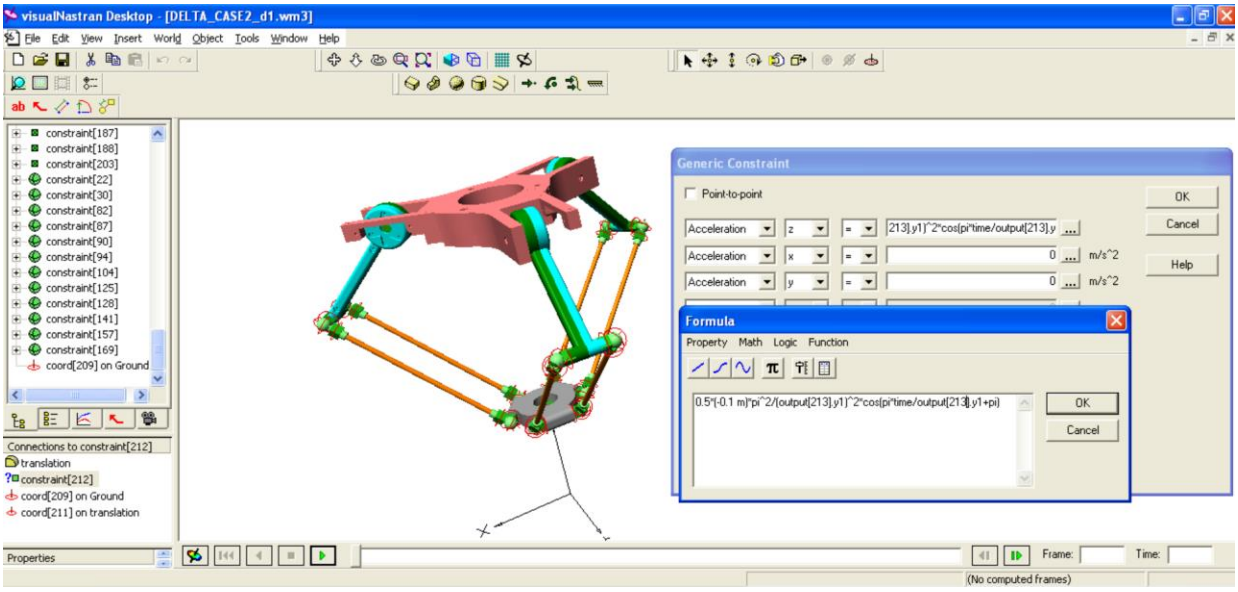




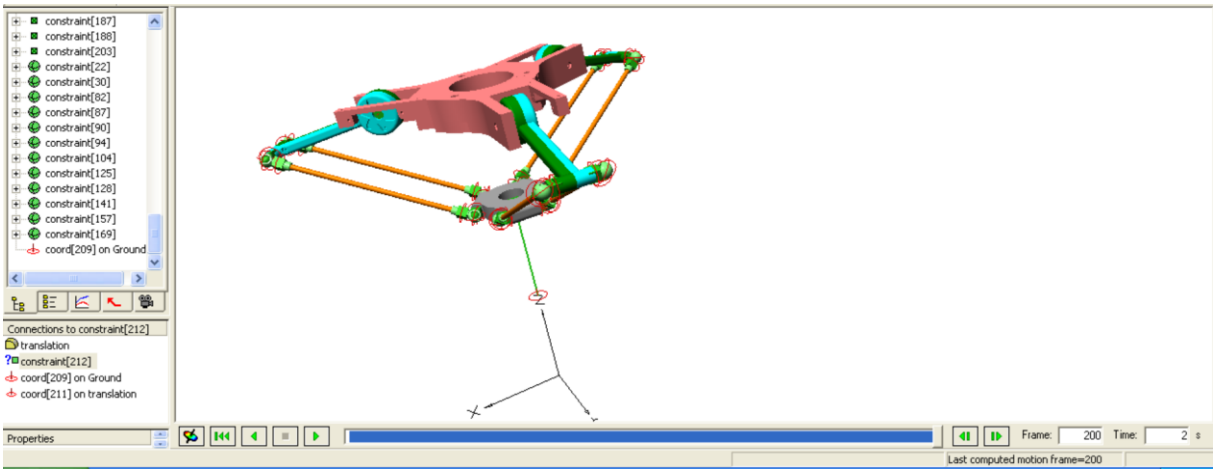
Case 2

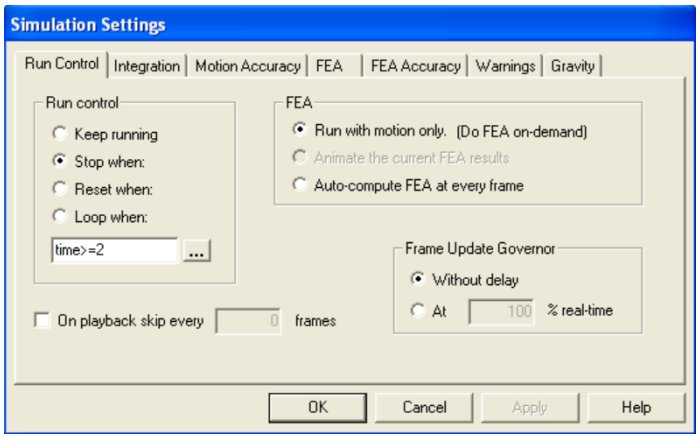
Here the task is to move the effector part to the desired location. By using this function, we define the motion of the effector part more smoothly.

Part 1

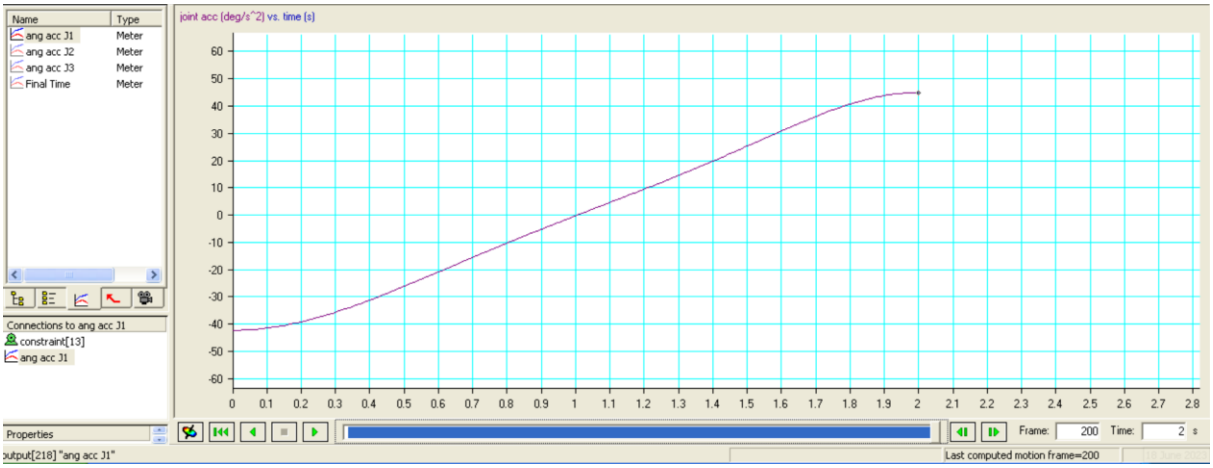


In part 1, to calculate the simple result we define a sinusoidal function in the z direction. We move the effector part only in the z direction. And during this motion we keep the x and y axes acceleration 0.

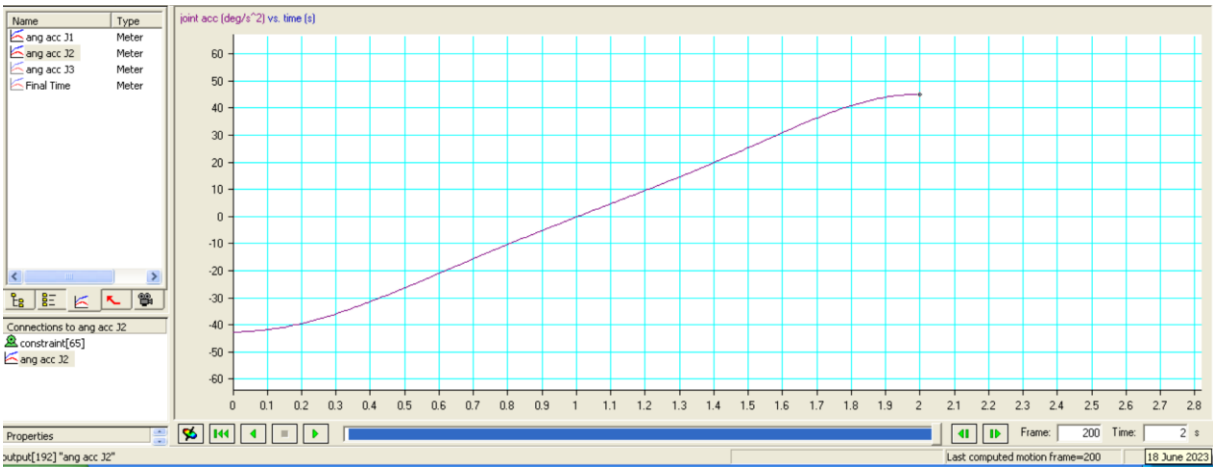




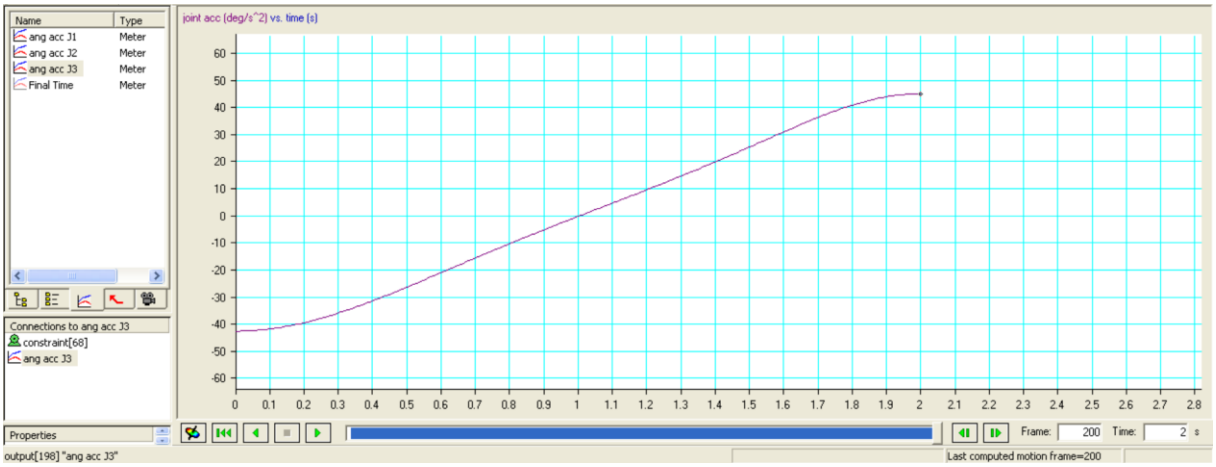
Servo motor 1 angular acceleration:



Servo motor 2 angular acceleration:

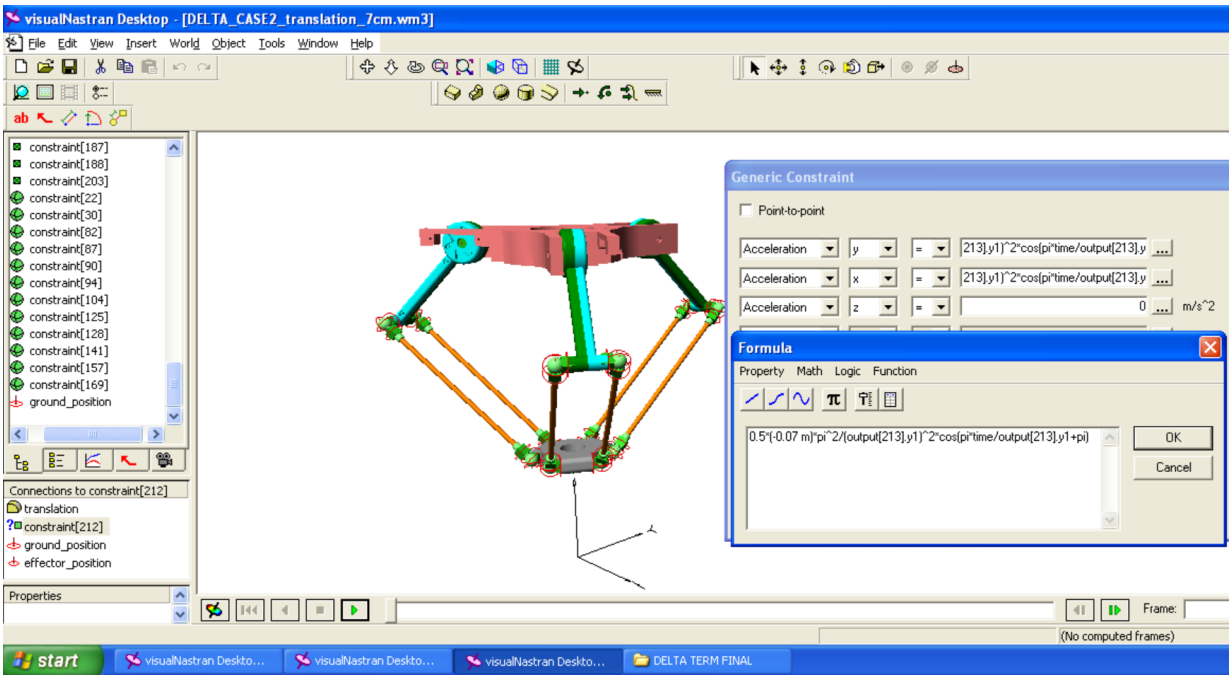


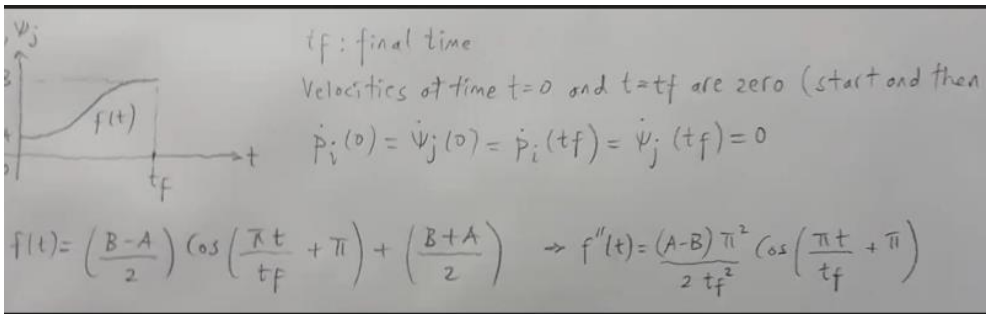
Servo motor 3 angular acceleration:



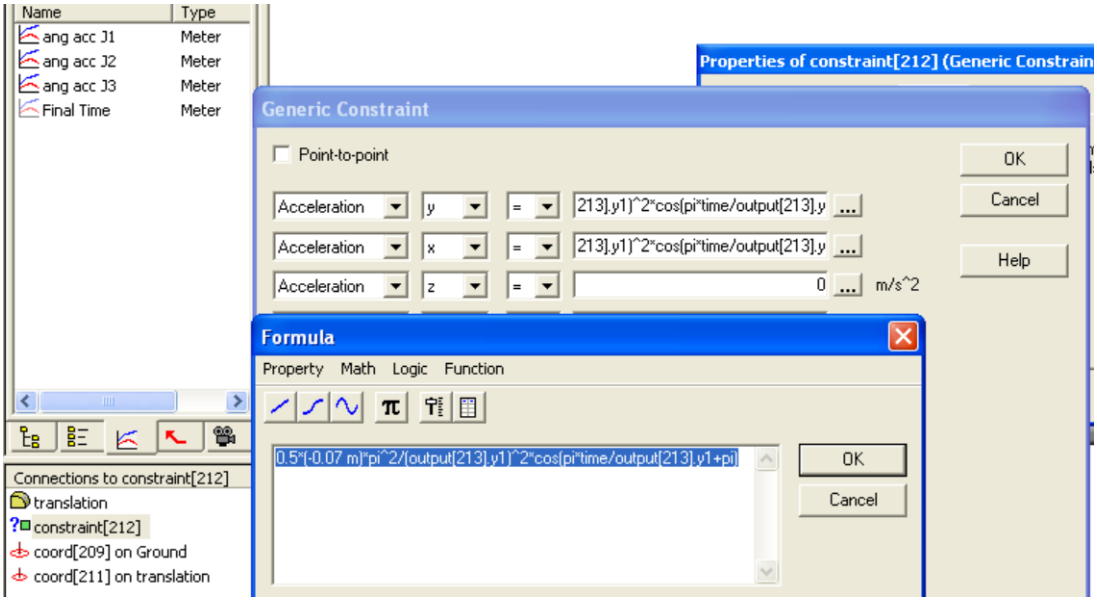
Part 2:

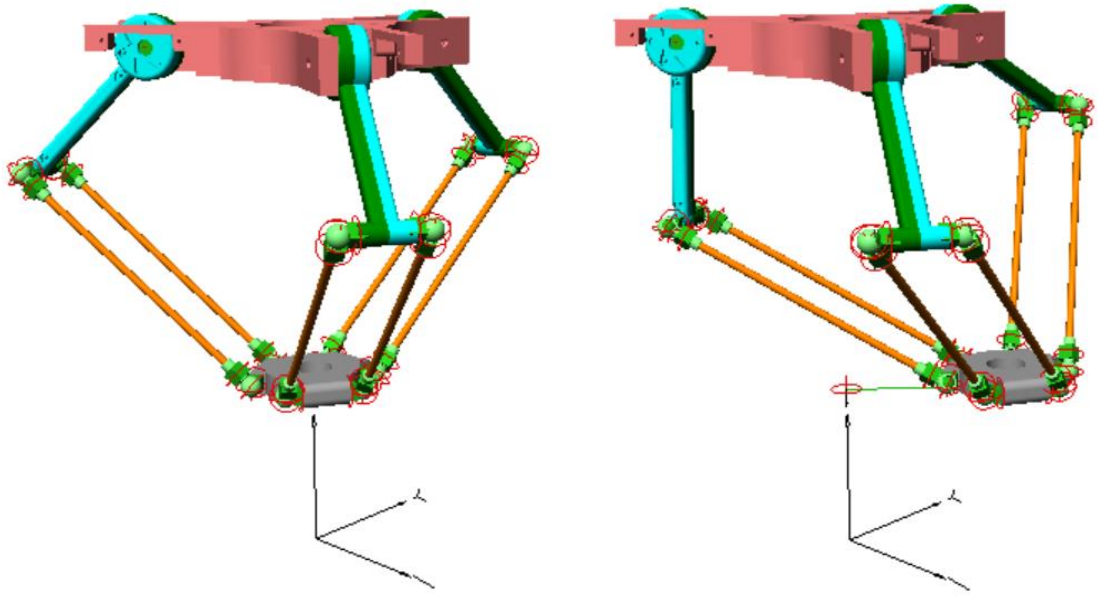
Moving the effector part as 7cm in x direction and 7cm in y direction at the same time. Here Z axis is constant during the motion.



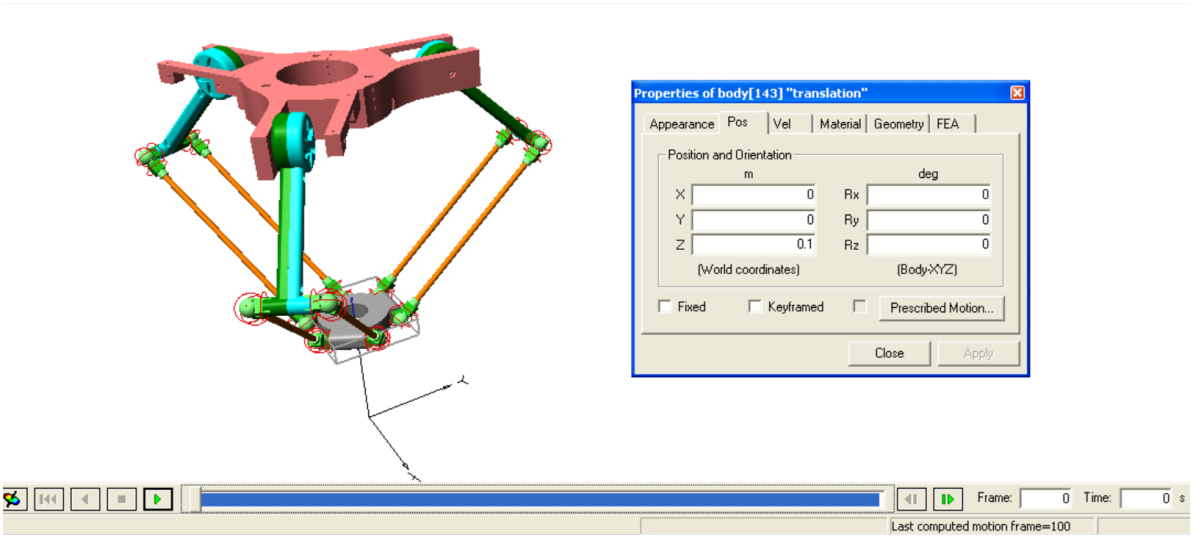


Sinusoidal function is used as we learned in the lecture notes:

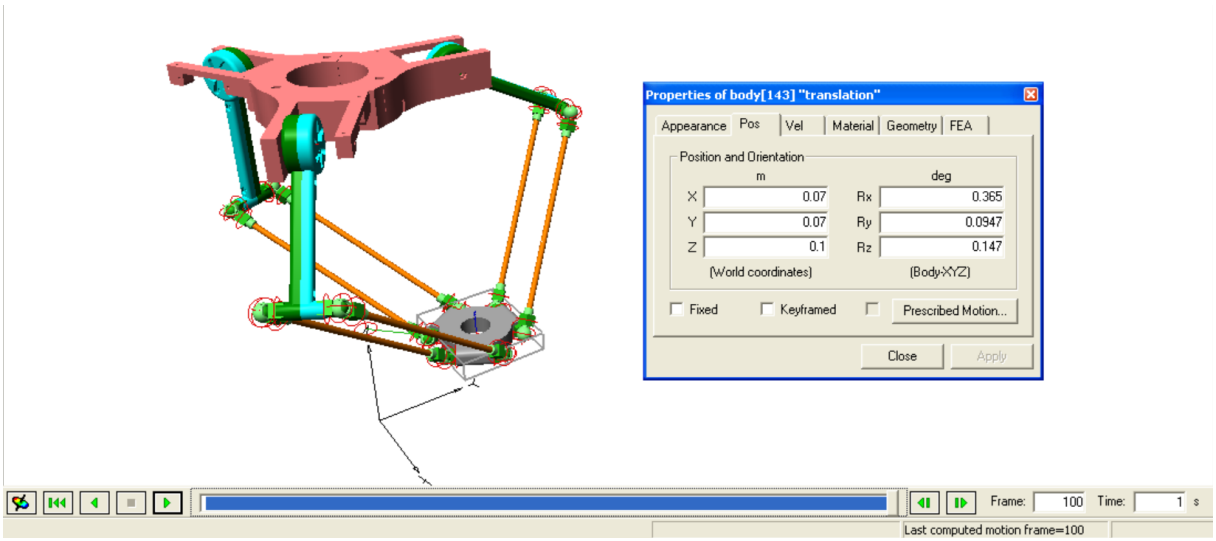

Name
Type
ang acc J1
Meter
ang acc J2
Meter
ang acc J3
Meter
Final Time
Meter
Properties of constraint[212] (Generic Constrain
Generic Constraint
Point-to-point
OK
Cancel
Help
Acceleration
y
=
213].y1)^2*cos(pi*time/output[213].y
Acceleration
x
=
213].y1)^2*cos(pi*time/output[213].y
Acceleration
z
=
0
m/s^2
Formula
Property
Math
Logic
Function
0.5*(-0.07 m)*pi^2/(output[213].y1)^2*cos(pi*time/output[213].y1+pi)
OK
Cancel
Connections to constraint[212]
translation
constraint[212]
coord[209] on Ground
coord[211] on translation



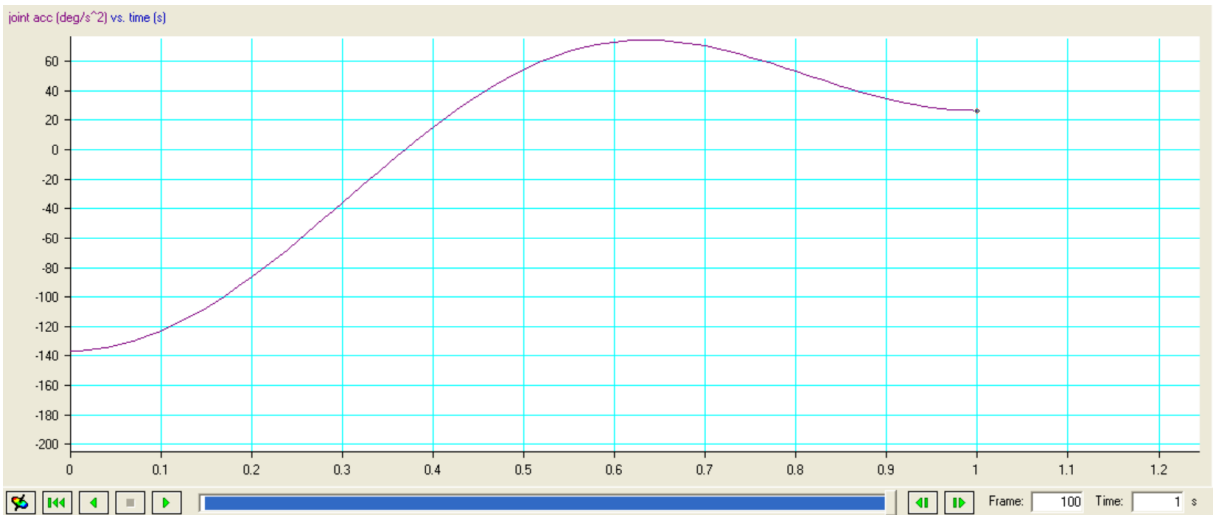
Before the motion:



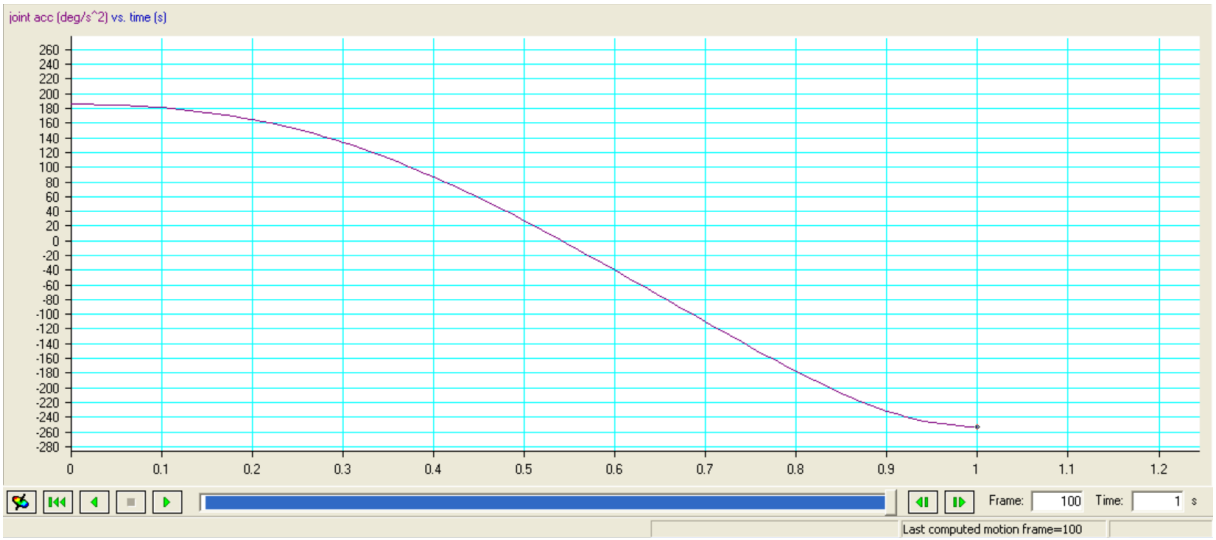
After the motion:



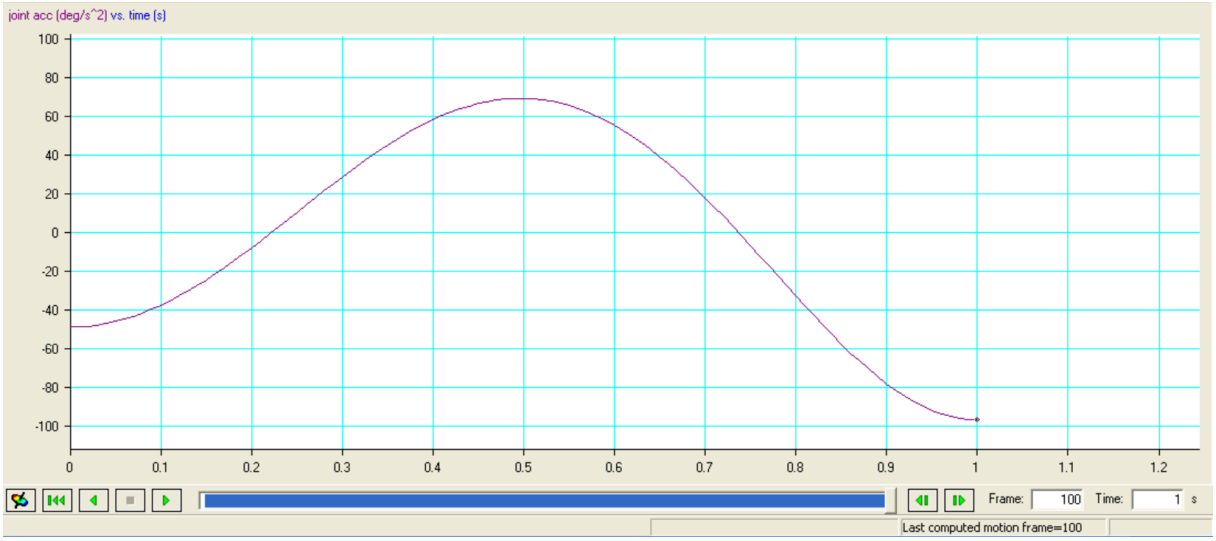
Servo motor 1 angular acceleration:



Servo motor 2 angular acceleration:



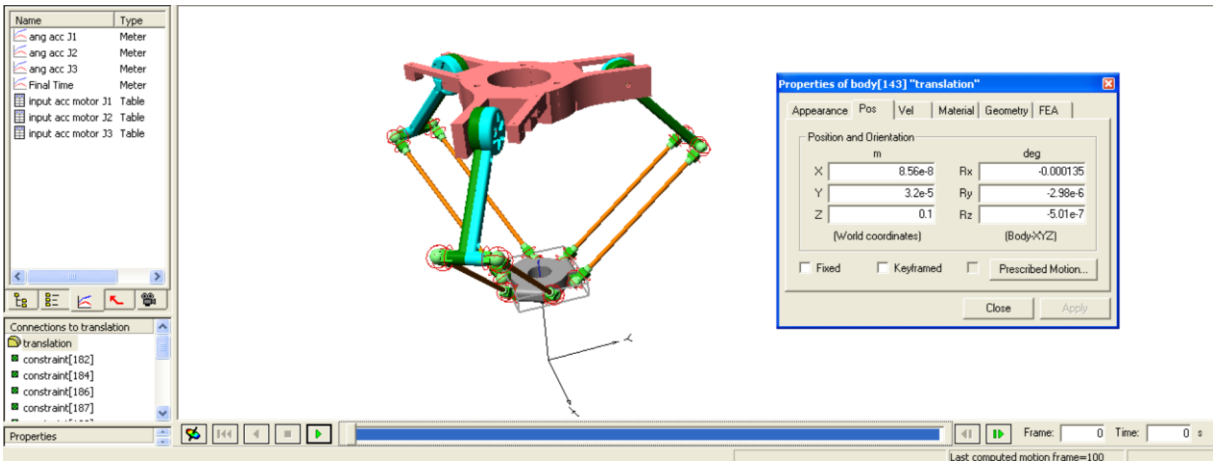
Servo motor 3 angular acceleration:



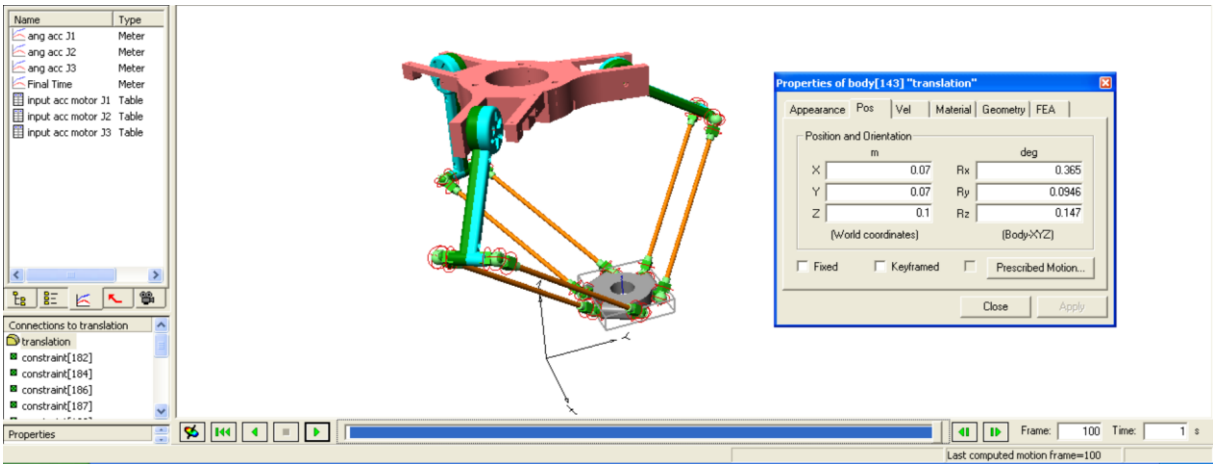
Case 3

Forward kinematics using the angular acceleration data from the case 2 is used here.

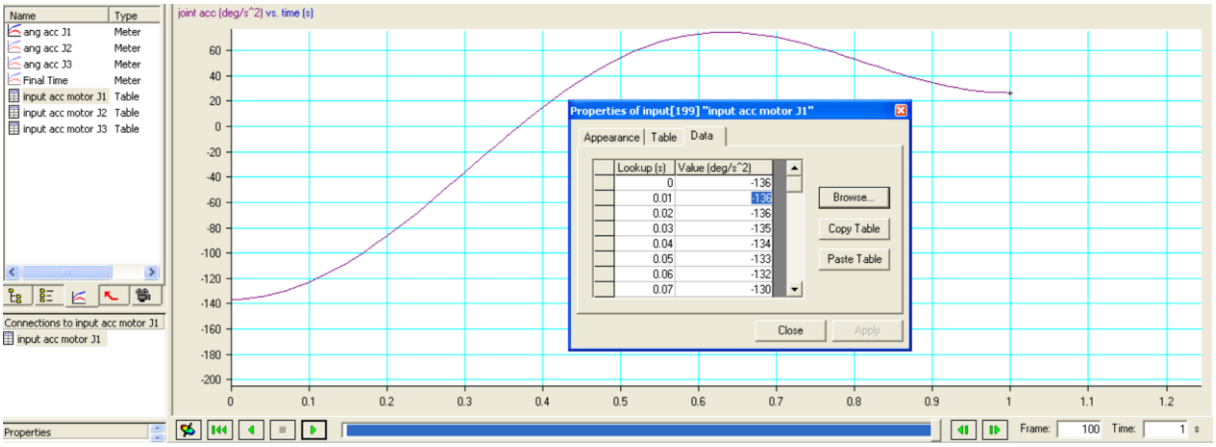
At the time of 0 second:



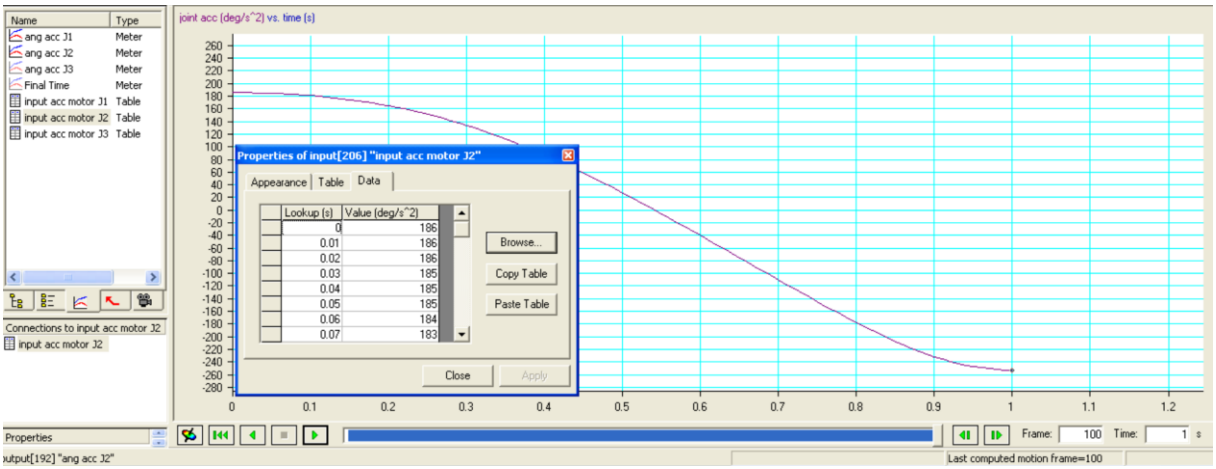
At the time of 1 second: Here position is changed as we expected.



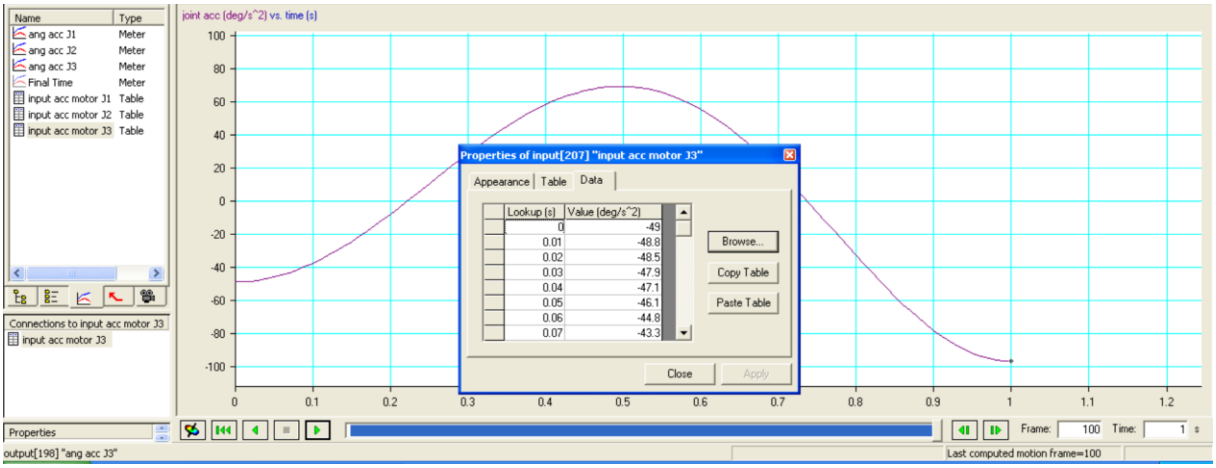
Servo motor 1:



Servo motor 2:

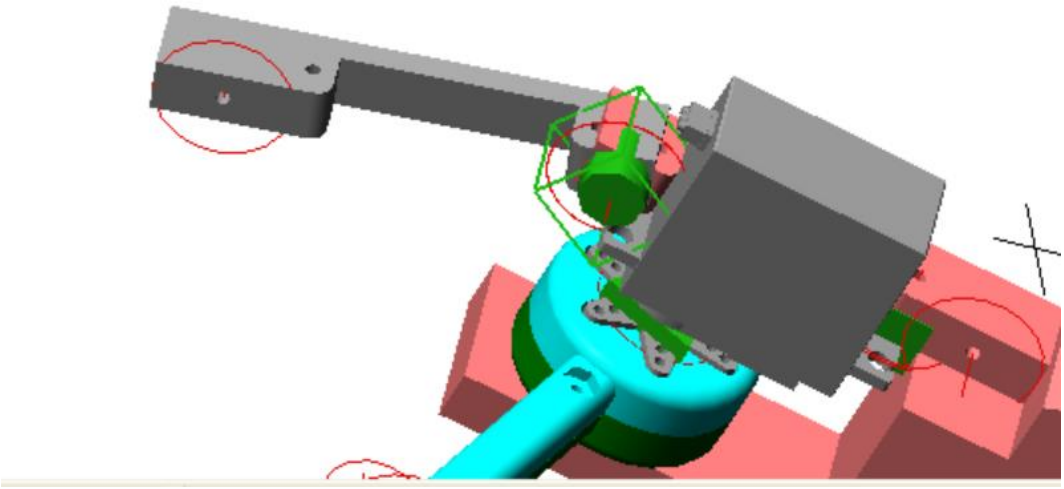


Servo motor 3:

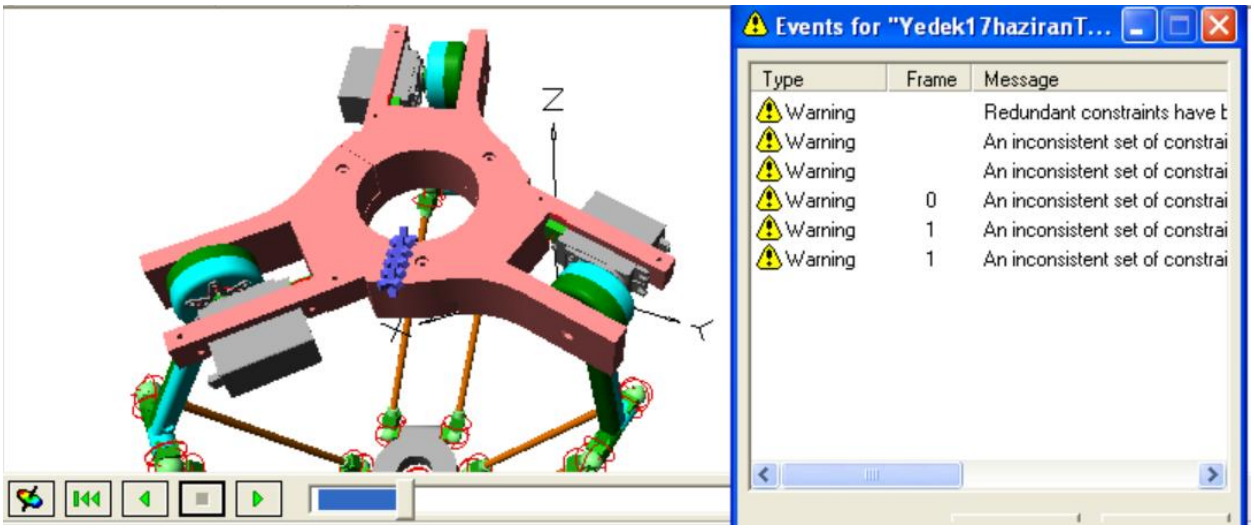


ERRORS

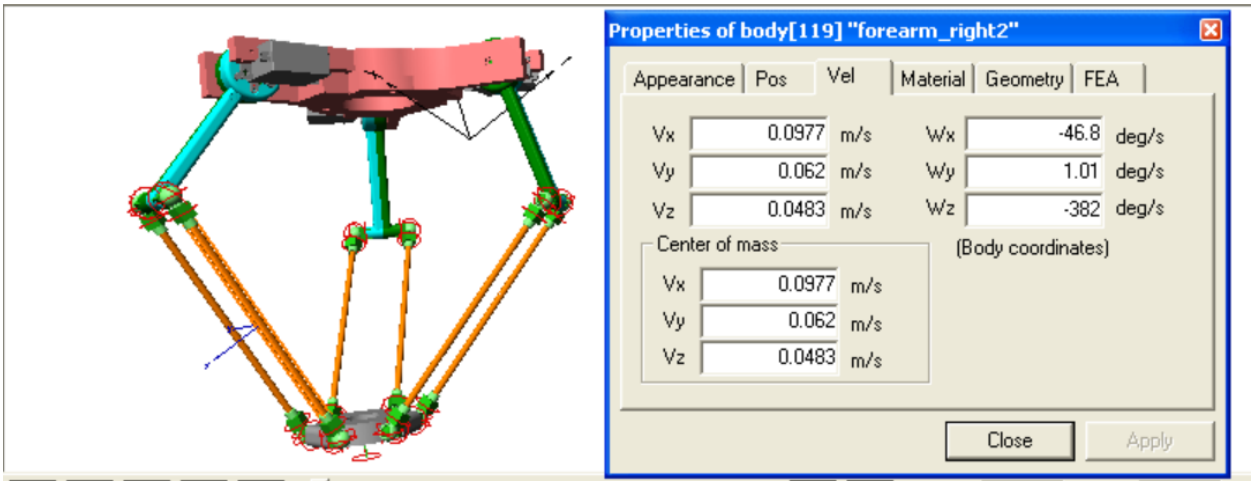
In the designing process of the system during the assembly process, many errors occurred from Visual Nastran.



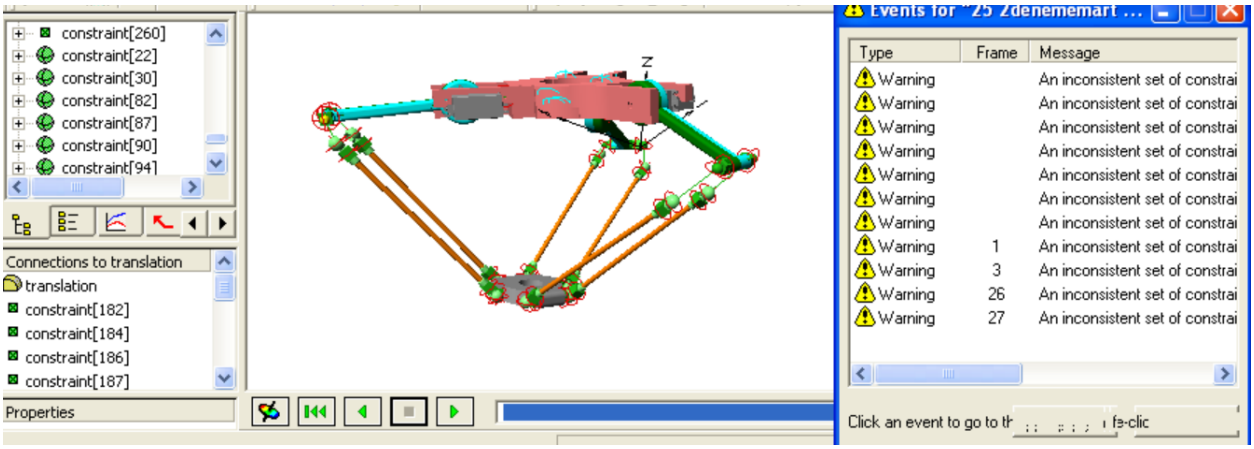
In this figure of error is a constraint connection between different parts. Adding axes and correct degrees of freedom fixed this error.



Another error, we got inconsistent constraints warning by assigning restraints to the system. We solved this problem by fixing base parts in position properties.



In this figure, the end-effector motion equations that we defined with prescribed motion were at zero when the velocity of the end effector was not zero. To correct this situation, we canceled the prescribed motion and the positions and velocities of its parts reached 0 after simulation.



Due to the Inconsistent Constraints warnings, we received in the last figure, we thought of using the inverse kinematic equation to give system acceleration.