# Appendix

## Group 9, Redysim based Project 2021

## Results of the Task 2

SN-050, Ankur Phukon
JIST, Assam
SN-053, Chetan Sharma
IIT, Delhi
SN-054, B. Shree Ragaventhra
Amrita University, Tamil Nadu
SN-055, Alok Dwivedi
NIT, Meghalaya

---

# 1. Introduction

Multi-body dynamics find applications in robotics, automobile, aerospace and many other streams for analysis, simulation and control [1]. In this presentation, we are giving the results of the Task 2 which was a four bar linkage. Here we present the graph plotted by the Euler-Lagrange method in Matlab software and the graph plotted using the ReDySim software. Then we superimposed the graph of both methods and coming up with the differences and similarities between the two methods.

# 2. Approach

There are several approaches to solve the dynamic equations of a multi-body system such as Lagrange-Euler Method, Newton-Euler method etc. In this problem, first we have used the Lagrange-Euler method to get dynamics equations of each joint using cut-section method as shown in the figure 1.

Forward Dynamics Formulation: we used cut-section

method and apply Euler Lagrange method

$$\begin{bmatrix} D & -J_B^T \\ J_B & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ f \end{bmatrix} = \begin{bmatrix} \tau - C(q,\dot{q}) + \emptyset \\ \dot{J}_B \dot{q} \end{bmatrix}$$

$$\begin{bmatrix} \ddot{q} \\ f \end{bmatrix} = \begin{bmatrix} D & -J_B^T \\ J_B & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tau - C(q,\dot{q}) + \emptyset \\ \dot{J}_B \dot{q} \end{bmatrix}$$

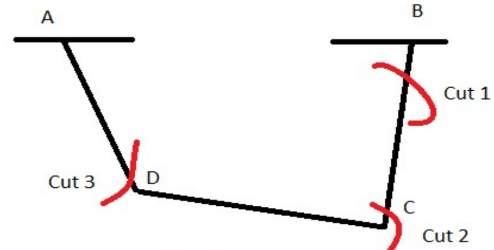



Figure 1

When we do a cut between two joints,

then we have to solve the equations for

two different manipulators. Then the equations becomes:

$$\begin{bmatrix} D_2 & 0 & -J_{C_2}^T \\ 0 & D_1 & J_{C_1}^T \\ J_{C_2} & -J_{C_1} & 0 \end{bmatrix} \begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_1 \\ f \end{bmatrix} = \begin{bmatrix} \tau_2 - C_2(q,\dot{q}) + \emptyset_2 \\ \tau_1 - C_1(q,\dot{q}) + \emptyset_1 \\ \dot{J}_{C_1}\dot{q}_1 - \dot{J}_{C_2}\dot{q}_2 \end{bmatrix}$$

Here D1, D2 are inertia tensors for two for manipulator 1 and 2 respectively.

$$\begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_1 \\ f \end{bmatrix} = \begin{bmatrix} D_2 & 0 & -J_{C_2}^T \\ 0 & D_1 & J_{C_1}^T \\ J_{C_2} & -J_{C_1} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tau_2 - C_2(q,\dot{q}) + \emptyset_2 \\ \tau_1 - C_1(q,\dot{q}) + \emptyset_1 \\ \dot{J}_{C_1}\dot{q}_1 - \dot{J}_{C_2}\dot{q}_2 \end{bmatrix}$$

$C(q,\dot{q}) = \dot{q} * c * q$ ;  $c$ = Christoffel symbols

Inverse Dynamics torque formulation:

$D$ = Inertia matrix, $J_B$ = jacobian of point B(here end effector),

Dynamical equation for close loop system with constrained force at cut point:

$$D\ddot{q} + C(q,\dot{q}) + \emptyset = \tau + J_B^T f\, ; \quad ------(1)$$

Here $\tau_B = J_B^{'} f$, constrained torque.

$$J_B\ddot{q} + \dot{J}_B\dot{q} = 0; \quad ------(2)$$ Constrained equation.

From equation (1), we get:

$$\ddot{q} = D^{-1}(A + J_B^T f)\, ; \quad ----(3) \quad A = \tau - C(q,\dot{q}) - \emptyset\, ;$$

By equation (2) and (3), constrained force:

$$f = -{D_n}^{-1}\left(\dot{J}_B\dot{q} + J_B D^{-1} A\right); \quad ----(4)$$ Here $D_n = J_B D^{-1} {J_B}'$ ;

By putting equation (4) into eqn (1), we got the torque value:

$$\boxed{\tau = {C_\varphi}^{-1}\left(D\ddot{q} + J_B^T {D_n}^{-1} \dot{J}_B \dot{q}\right) + C(q,\dot{q}) + \emptyset; \quad -------(5)}$$

Here $C_\emptyset = (I - J_B^T\, {D_n}^{-1}\, J_B\, D^{-1})$ and $I$ = unity matrix.

In the second approach, the software 'ReDySim' uses the recursive dynamics for solving inverse and forward dynamics equations. We have to input the Denavit-Hartenberg parameters, moment of inertia terms, centre of gravity terms, and time for simulation. After doing the simulation, we can animate our system and plot the results of joint angular displacement, angular velocities, angular acceleration of each link.

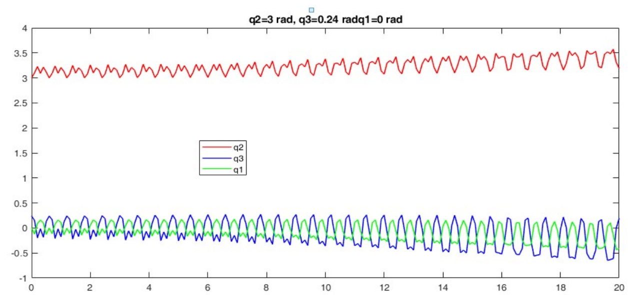
# 3. Results

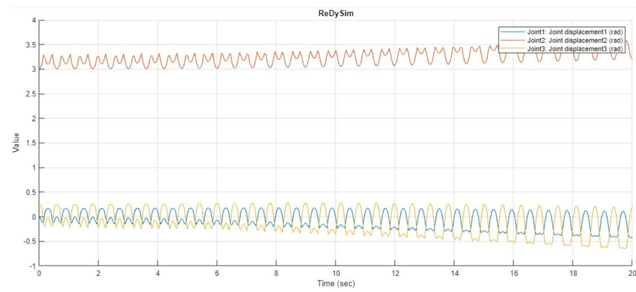
## Forward dynamics:

Results of free-fall simulation:

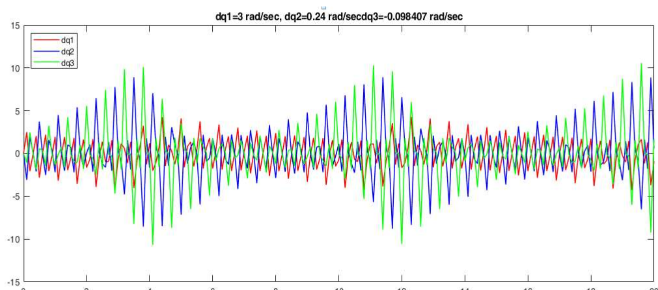
Matlab

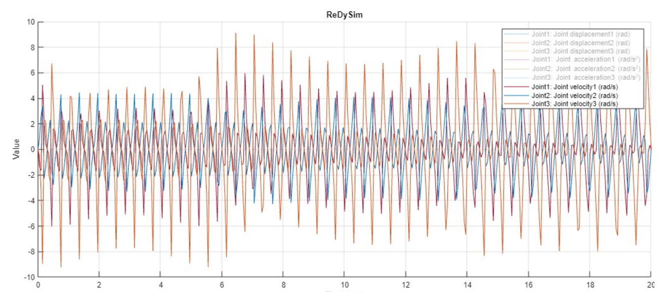
ReDySim



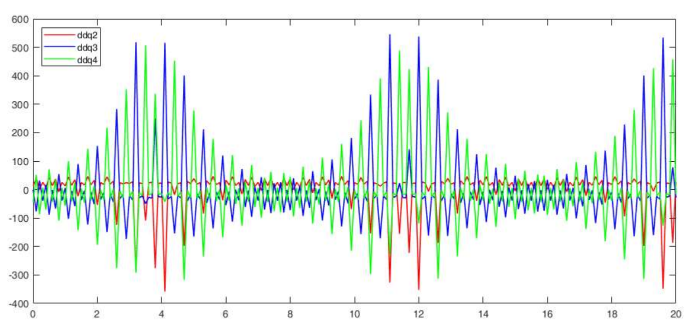



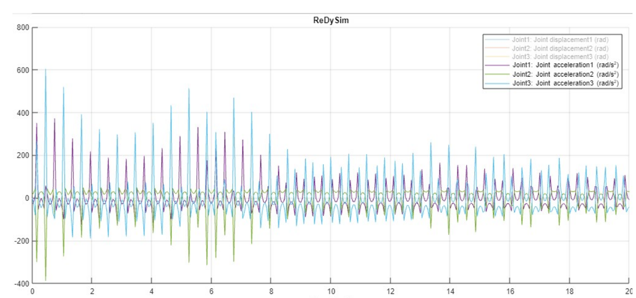

Position Velocity




Angular Velocity



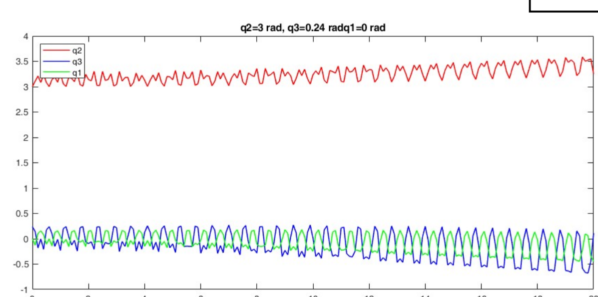

Angular Acceleration

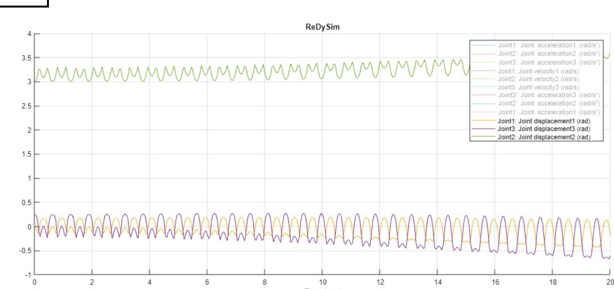
## For forced simulation:

Matlab

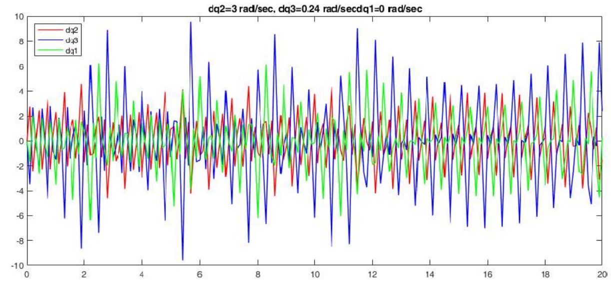
Redysim

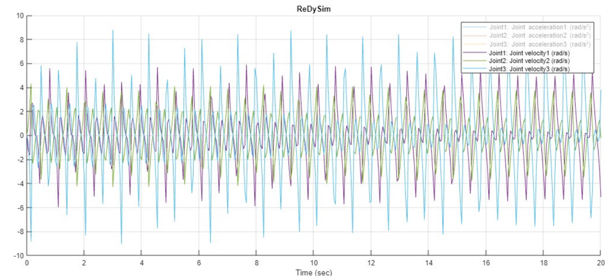
Position Velocity



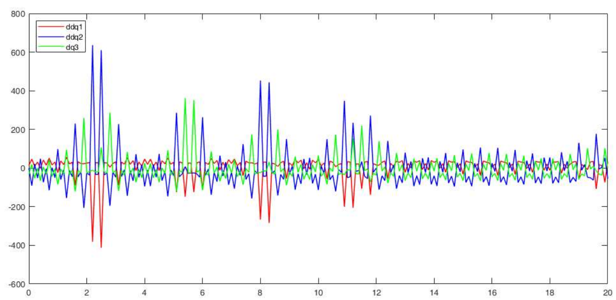



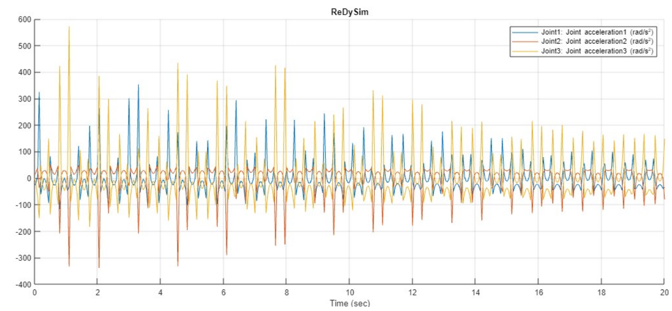
Angular Velocity




Angular Acceleration

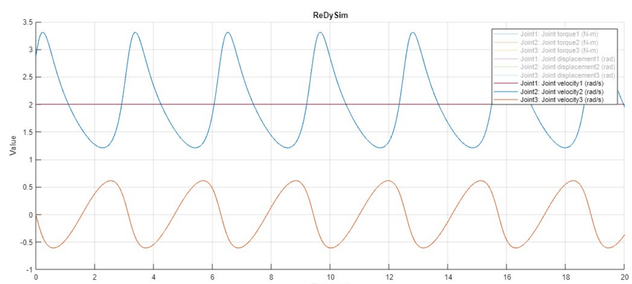


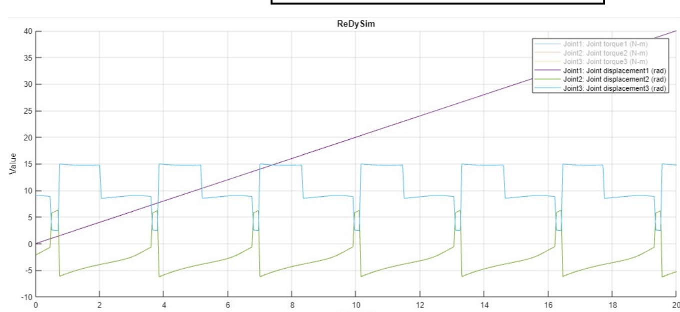

# Inverse Dynamics Results:

## Redysim:

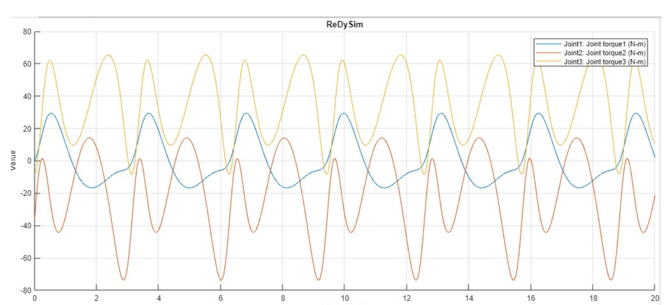
Joint Velocity




Angular Displacement



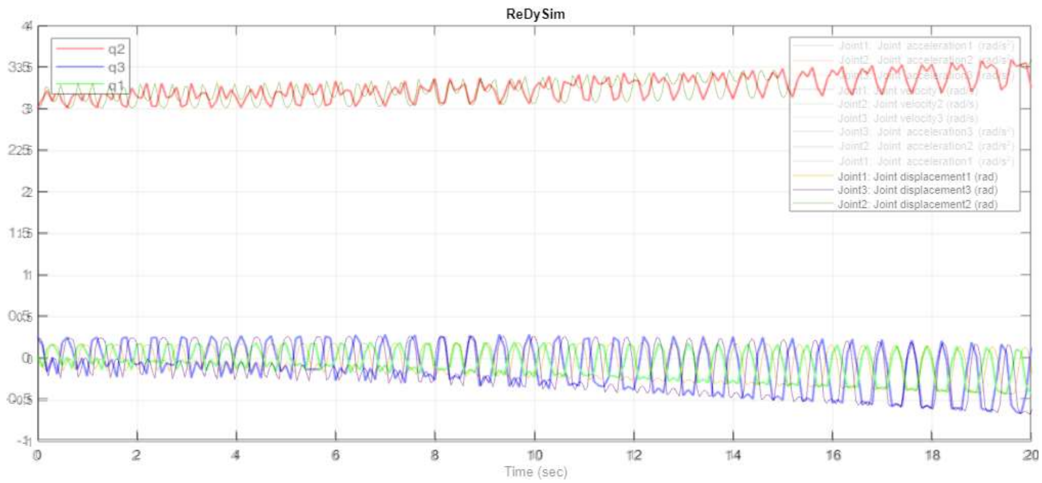

Joint Torque



## Superimposed Results:

Forced system for torque = 2 N-m

Angular Displacement




# 4. Conclusion

The result of both Lagrange-Euler method and Recursive Dynamics are almost similar but in former method, the velocity and acceleration of the system, we got some variation in the graphs prepared by Lagrange-Euler method than that of prepared by the 'ReDySim'.

We found nearly similar graph for angular position in both methods. For inverse dynamics, we have formulated the equations but the graphs were not plotted because of having some issue with the trajectory selection.

Recursive Dynamics provides almost similar results than traditional methods of solving the dynamics of the multi-body systems. The advantage of using the recursive dynamics is that it is based on tree type Decoupled Natural Orthogonal Complement(DeNOC)[1] which makes the solver recursive in nature.

# 5. References

[1] Shah, S et. al(2012). Recursive Dynamics simulator(ReDySim): A multibody dynamics solver. *Theoretical and applied mechanics letters 2, 063011(2012)*

[2] Mark W. Spong et. al(1st Edition). Robot modelling and control, *John Willey and Sons, Inc.*