We are implementing Dynamics in Hexa Stewart Platform as well as Zamanov configuration using ReDySim.
The following documents will elaborate the implementation of Hexa Parallel Stewart Platform.

About the manipulator configuration:

The manipulator consists of a base platform and an end-effector platform connected via 6 Revolute-Universal-Spherical (RUS) chain. As the name suggests, the links are in parallel pair. The following pic illustrates the CAD model and actual implementation of the Hexa Parallel platform.

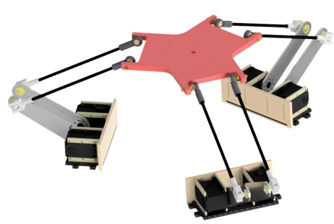
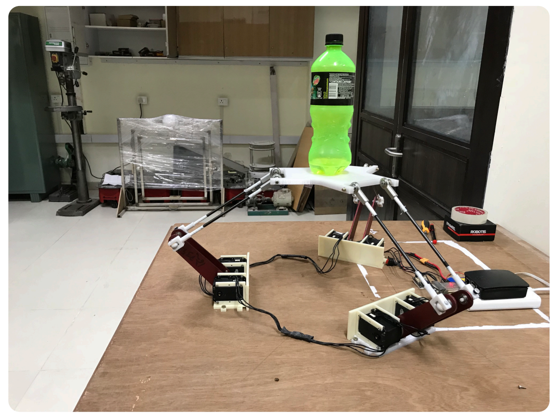


Fig1:Isometric view



The advantages of a Hexa Parallel mechanism are:

1. Bigger workspace - Nearly twice the workspace of other mechanisms.
2. Easy control
3. No constrained joints- Zamanov configuration uses 2 spherical joints in which one always stays twisted.
4. The choice of link lengths is more flexible in contrast with Zamanov configuration which requires the second link to be at least 2-2.5 times the first link.

The disadvantages of this mechanism are:

1. The configuration is very sensitive to design errors- the motor pair axes should be coaxial or it may lead to gain singularities.
2. The configuration has the least quality of workspace in comparison with other configuration.

Implementation in ReDySim:

The Fig.2 shows the files that are required for executing the Inverse Dynamics using ReDySim.

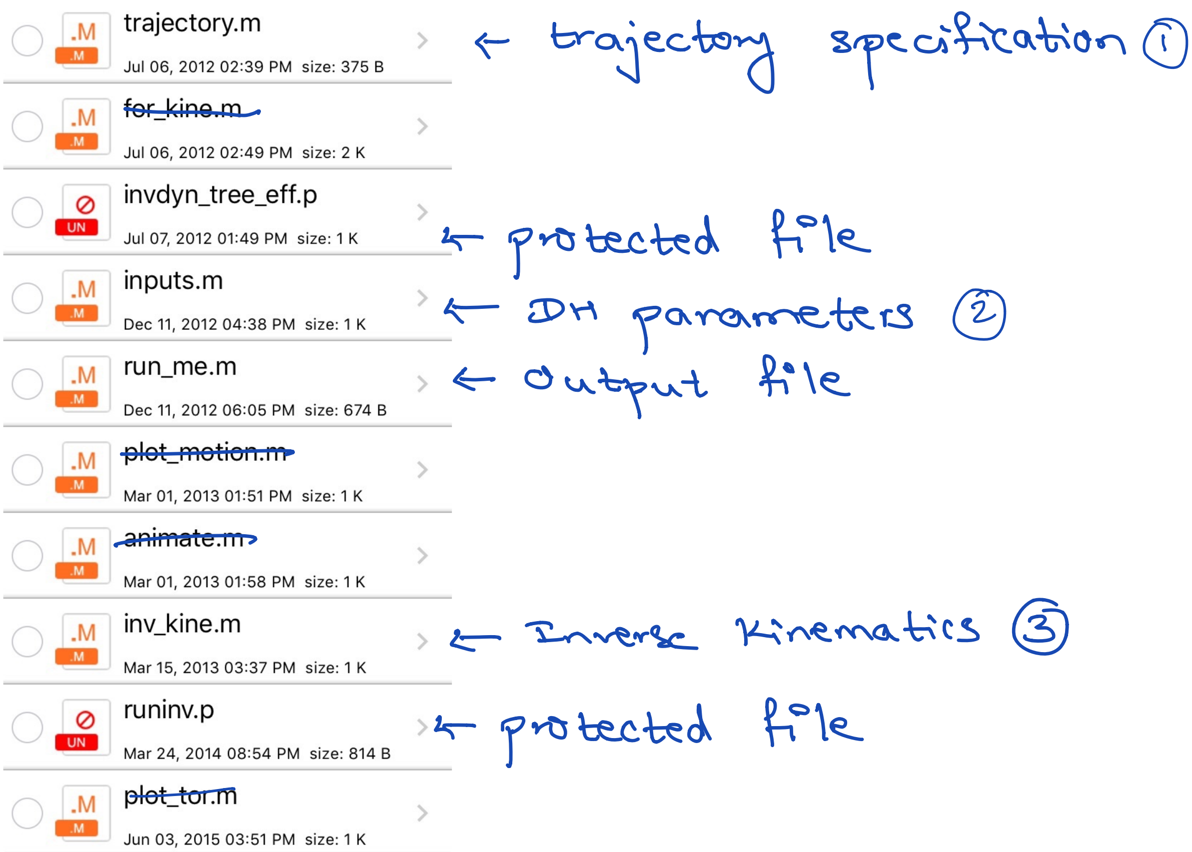
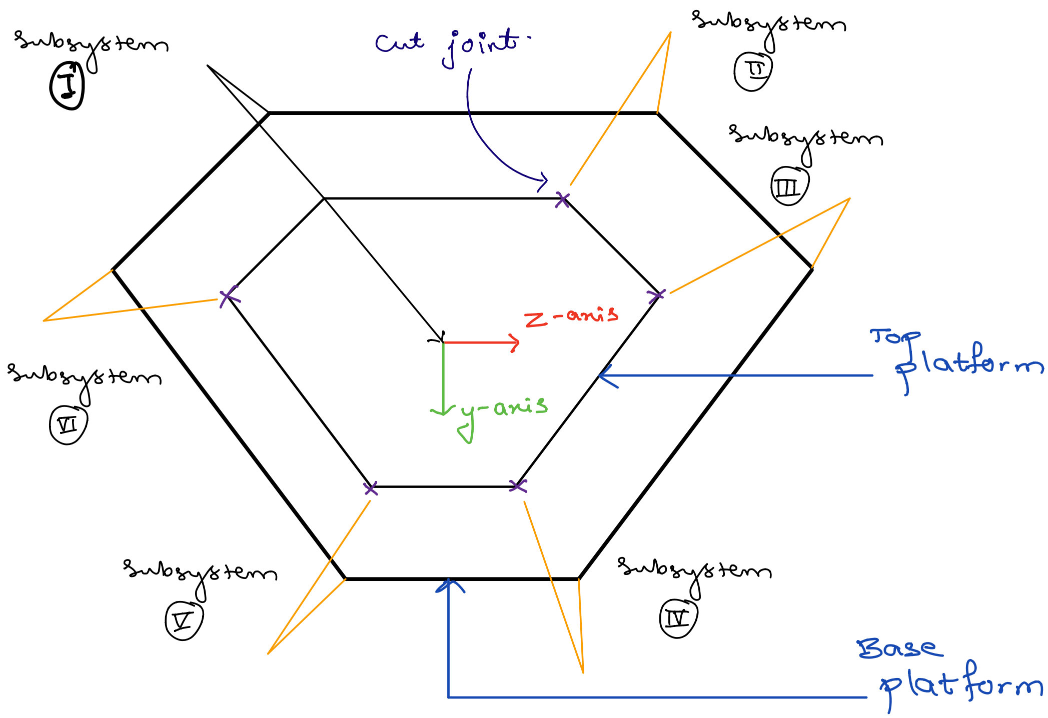
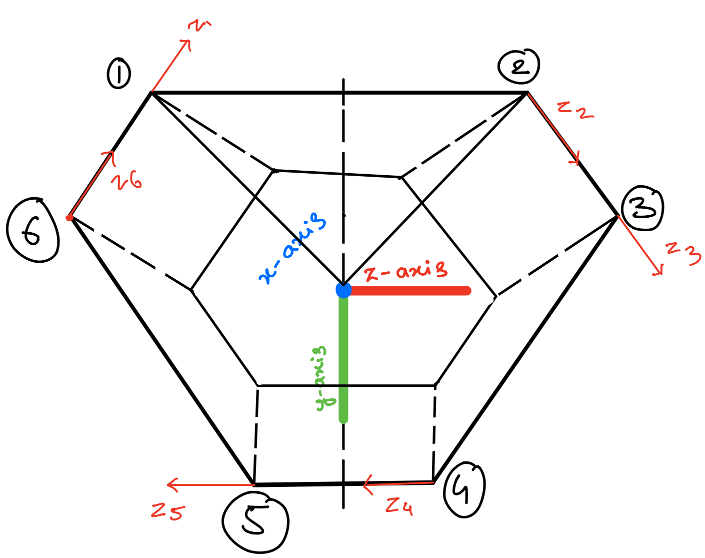


Fig:2

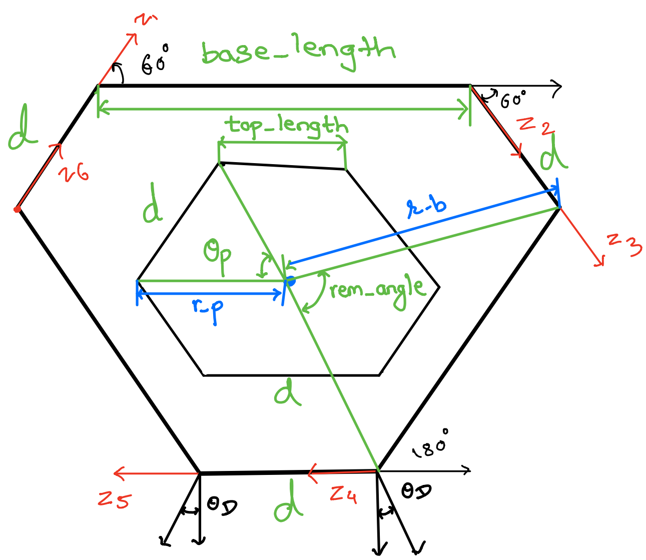
Specifying variables and subsystems.





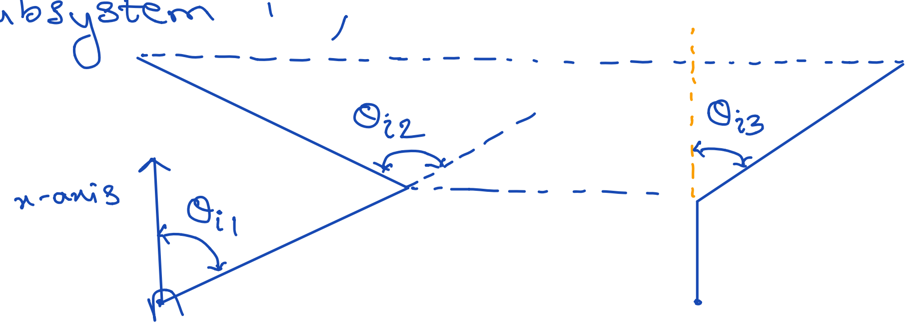
Here $z_1 \rightarrow z_6$ are the axis of individual motors.

The convention for the frames have been critically chosen to define the modified DH parameters for the system.

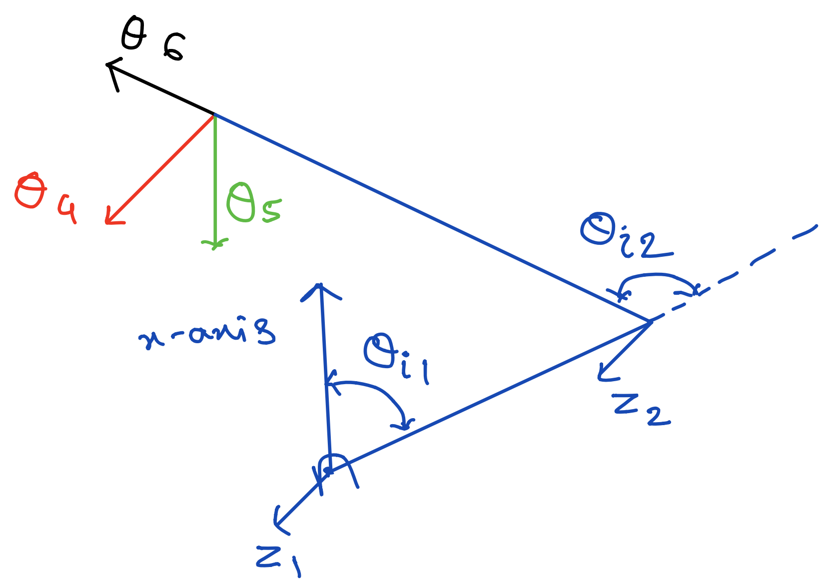


Defining all the "θ" for a subsystem.

In subsystem "i";



Defining spherical angles for subsytem (I).



* If $\theta_{13} = 0$ then $z_4 \parallel z_2 \parallel z_1$

* θ_4 is about z-axis, θ_5 is about y-axis and θ_6 is about x-axis.
  thus the axis of $\theta_6 \times \theta_5$ = axis of θ_4

We define the vectors and matrices used in further calculation:

The vectors written in red are the equivalent vectors defined in ReDySim.

1) q-vector : It is a vector of all the joints (actuated + passive) required to define the entire system.

th- vector

q-vector is a 21 element vector.

$\dot{q}$, $\ddot{q}$ are the time derivatives of this vector. $\left[\dot{q} = \frac{dq}{dt}, \ \ddot{q} = \frac{d^2q}{dt^2}\right]$

2) y-vector : It is a vector of all the parameters required to define the end effector position.

thi - vector

y-vector is a 6 - element vector.

$\dot{y}$, $\ddot{y}$ are the time derivatives of this vector. $\left[\dot{y} = \frac{dy}{dt}, \ \ddot{y} = \frac{d^2y}{dt^2}\right]$

3) Zau - spanning - tree : This is a 21-element vector denoting all the torques in 21 joints before defining the jacobian.

4) Tau-actuated : This is a 6-element vector denoting all the torques in the actuated joints after defining the jacobian.

5) I -matrix : It is the Inertia matrix from the basic equation:

$$I\ddot{q} + C(q,\dot{q}) + N(q) = \tau + J^T\lambda$$

6) C-matrix : It is a matrix representing the coriolis component.

7) N-matrix: It is a vector for gravity component.

8) J-matrix : This is the loop closure jacobian for a closed loop mechanism.

9) G-matrix : This is a jacobian used in explicit loop constraint.

Definition of G-matrix:

Type	position	velocity	acceleration
implicit:	$\phi(\boldsymbol{q}) = 0$	$\boldsymbol{K}\dot{\boldsymbol{q}} = 0$	$\boldsymbol{K}\ddot{\boldsymbol{q}} = \boldsymbol{k}$
explicit:	$\boldsymbol{q} = \gamma(\boldsymbol{y})$	$\dot{\boldsymbol{q}} = \boldsymbol{G}\dot{\boldsymbol{y}}$	$\ddot{\boldsymbol{q}} = \boldsymbol{G}\ddot{\boldsymbol{y}} + \boldsymbol{g}$

Table 1: Loop constraints [5]

Integrating Mimic Joints into Dynamics Algorithms: Exemplified by the Hybrid Recupera Exoskeleton

Shivesh Kumar, Robotics Innovation Center, DFKI GmbH, Bremen, Germany, shivesh.kumar@dfki.de

Marc Simnofske, Robotics Innovation Center, DFKI GmbH, Bremen, Germany, marc.simnofske@dfki.de

Bertold Bongardt, Robotics Innovation Center, DFKI GmbH, Bremen, Germany, bertold.bongardt@dfki.de

Andreas Müller, Institute of Robotics, Johannes Kepler University, Linz, Austria, a.mueller@jku.at

Frank Kirchner, RIC, DFKI GmbH / AG Robotik, Universität Bremen, Bremen, Germany, frank.kirchner@dfki.de

Reference

robot descriptions using the URDF [25] format.

The presence of closed loops in robots significantly increases the complexity of the kinematics and dynamics problems associated to the multi-body systems (MBS). In particular, planar kinematic loops impose redundant constraints on the system – that either need to be removed manually [5] or demand numerical decomposition techniques which deteriorate the computational performance and numerical accuracy of the solution [18]. In many practical instances of kinematic loops within hybrid robots, the active joint guides the other passive joints inside a loop in a linear manner. The contribution of this paper is an explicit method to compute the forward and inverse dynamics of arbitrary hybrid robots that involve planar kinematic loops which can be described by means of linear mimic joints (for example, parallelograms or chains of parallelograms). Once specified in a URDF file, featuring the mimic joint tag, the associated loop closure functions are detected and transferred into a suitable form to be used in the computations of the analytical the forward and inverse dynamics algorithms. The benefit of this approach is that efficient dynamics algorithms for tree type system with $O(n)$ complexity can be directly used to solve the dynamics. The results are free of numerical errors due to loop closure and free of singularities arising from redundant constraints imposed by the planar kinematic loop. The application of the dynamic modeling is demonstrated for a novel shoulder mechanism which employs a double-parallelogram mechanism at the second shoulder joint.

The organization of the paper is the following: Section 2 provides theoretical preliminaries for modeling robots with closed loops along with an introduction to the concept of loop closure functions. It derives the formulas for the forward and inverse dynamics for these mechanisms. Section 3 presents the concept of mimic joints and the novel method to derive the loop closure functions automatically from the robot description. Section 4 presents the application of this approach to the forward and inverse dynamics computation of a shoulder mechanism developed for a novel, full-body exoskeleton. Section 5 draws the conclusions and presents future work.

2 MODELING RIGID BODY SYSTEMS WITH CLOSED LOOPS

This section briefly introduces the theory of multi-body dynamics subjected to holonomic and scleronomic constraints. It strictly adopts the notation and terminology introduced by Featherstone in [5]. Therefore, consider a rigid body system with N_B bodies, N_J joints, and $N_L = N_J - N_B$ kinematic loops. Assume that a spanning tree is defined and that the joints are enumerated using regular numbering scheme. Let n denote the degree of freedom of the selected spanning tree, computed as $n = \sum_{i=1}^{N_B} n_i$, and let n_c denote the number of loop-closure constraints, computed as $n_c = \sum_{k=N_B+1}^{N_J} n_k^c$. Further, let $\boldsymbol{q}$ indicate the vector of all joints of the spanning tree (of size n) and let $\boldsymbol{y}$ indicate the vector of all independent joints of the spanning tree (of size $n - n_c$).

2.1 Loop Constraints

Loop constraints are algebraic constraints on the motion variables of a multi-body system. Loop constraints can be expressed in an implicit and in an explicit way, they are summarized in Table 1 at position, velocity, and acceleration levels. Here let $\boldsymbol{K} = \frac{\partial \phi}{\partial \boldsymbol{q}}$, $\boldsymbol{k} = -\dot{\boldsymbol{K}}\dot{\boldsymbol{q}}$, $\boldsymbol{G} = \frac{\partial \gamma}{\partial \boldsymbol{y}}$, and $\boldsymbol{g} = \dot{\boldsymbol{G}}\dot{\boldsymbol{y}}$. If both functions ϕ and γ describe the same constraint, $\phi \circ \gamma = \boldsymbol{0}$, $\boldsymbol{KG} = \boldsymbol{0}$, and $\boldsymbol{Kg} = \boldsymbol{k}$ can be deduced. Algorithms to compute variables in Table 1 from the spanning tree are provided in [5] and skipped here for brevity.

2.2 Equations of Motion (EOM)

The equations of motion for the spanning tree of a multi-body system can be written as

$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \ddot{\boldsymbol{q}}) \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are $(n \times 1)$ vectors of joint position, velocity and acceleration variables of the spanning tree, $\boldsymbol{H}(\boldsymbol{q})$ is the $(n \times n)$ mass-inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \ddot{\boldsymbol{q}})$ is a $(n \times 1)$ vector for Coriolis-centrifugal and gravity efforts, and $\boldsymbol{\tau}$ is the $(n \times 1)$ vector of force/torque variables. In case of robots with closed loops, the equivalent spanning tree of the robot system is subjected to loop constraint forces

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \ddot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_a + \boldsymbol{\tau}_c \tag{2}$$

where $\boldsymbol{\tau}_c$ and $\boldsymbol{\tau}_a$ are the constraint and active forces, respectively produced by the loop joints. If the selected loop joint is passive, $\boldsymbol{\tau}_a = 0$ can be substituted in Equation 2. The constraint force $\boldsymbol{\tau}_c$ is usually unknown but its value can either be calculated or eliminated from the equation following the Jourdain's principle [20] of virtual power, i.e., $\boldsymbol{\tau}_c\dot{\boldsymbol{q}} = 0$. Based on the (implicit or explicit) nature of the loop constraints, the equations of motion are developed for the entire system.

2.2.1 EOM with implicit loop constraints. The loop joints impose a set of kinematic constraints on the spanning tree which are briefly introduced in Table 1. Assuming that the position level implicit constraints have been successfully differentiated two times, the acceleration level loop constraints can be collected in a single matrix equation of the form

$$\boldsymbol{K}\ddot{\boldsymbol{q}} = \boldsymbol{k} \tag{3}$$

Explanation of each code of ReDySim

1) trajectory.m

StewartMechanism / stewart_IITJ / force_compliant_stewart / Source

trajectory.m

master | Source | Diff | Histo

force_compliant_stewart / Matlab / ReDySim / trajectory.m

d699c84 2 days ago | Full commit | Blame | Raw | Edit

```
% ReDySim trajectory module. The desired indpendent joint trejectories are
% enterd here
% Contibutors: Dr. Suril Shah and Prof S. K. Saha @IIT Delhi

function [thi dthi ddthi]= trajectory(tim, dof, Tp)
%Enter trejectories here

%Position
thi(1) = 0.25;%+0.01*tim;
thi(2) = 0*sin(tim);
thi(3) = 0*cos(tim);
thi(4) = 0.0;
thi(5) = 0.0;
thi(6) = 0.0;

%Velocities
dthi(1) = 0.;
dthi(2) = 0*cos(tim);
dthi(3) = 0*sin(tim);
dthi(4) = 0;
dthi(5) = 0;
dthi(6) = 0;

%Acceleration
ddthi(1) = 0;
ddthi(2) = -0*sin(tim);
ddthi(3) = -0*cos(tim);
ddthi(4) = 0;
ddthi(5) = 0;
ddthi(6) = 0;

end
```

Position: e_x, e_y, e_z, roll, pitch, yaw $= y$

Velocities: $\dot{y}$

Acceleration: $\ddot{y}$

currently the mechanism is in a static position

2) inputs.m

iPad 7:37 PM 40%

inputs.m

```
% ReDySim inputs module. The model parameters are entered in this module
% Contibutors: Dr. Suril Shah and Prof S. K. Saha @IIT Delhi

function [n dof type alp a b bt dx dy dz m g  Icxx Icyy Iczz Icxy Icyz Iczx aj Tp steps]=

%System: Fourbar Mechanism
% INPUTS
%Number of links
n=21;

%Degree of fredom of the system
dof=6;

% Type of mechanism
type=1; % 1 for closed-loop and 0 for open-loop

%Actuated joints of open tree
aj=[1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0]; %enter 1 for actuated joints and 0 otherw


%Time steps and span
Tp=0.3; steps=0.1;

%DH PARAMETERs
[base_length, top_length, half_angle, rem_angle, Base_matrix, l1, L2, theta_p, r_p] = pai
VL = r_p;
rod_radius = 0.003;

rangle_vectclock = [pi/3, -pi/3, -pi/3, pi, pi, pi/3];
alp=[rangle_vectclock(1), 0, pi/2, -pi/2, -pi/2, -pi/2, rangle_vectclock(2), 0, pi/2, rai
a=[0 l1 0 L2 0 0 0 l1 0 0 l1 0 0 l1 0 0 l1 0 0 l1 0];
b=zeros(1,21);
%PARENT ARRAY
bt=[0 1 2 3 4 5 0 7 8 0 10 11 0 13 14 0 16 17 0 19 20];
length_array=[l1, 0, L2, 0, 0, VL, l1, 0, L2, l1, 0, L2, l1, 0, L2, l1, 0, L2, l1, 0, L2
% d - VECTOR FORM ORIGIN TO CG

%%%COG of end effector, about which frame??
dx=[l1/2, 0, L2/2, 0, 0, 0, l1/2, 0, L2/2, l1/2, 0, L2/2, l1/2, 0, L2/2, l1/2, 0, L2/2, 1
dy=zeros(21);
dz=zeros(21);
dz(6) = VL/2;
% MASS AND MOMENT OF INERTIA AND GRAVITY
m=[0.3; 0; 0.6; 0; 0; 1; 0.3; 0; 0.6; 0.3; 0; 0.6; 0.3; 0; 0.6; 0.3; 0; 0.6; 0.3; 0; 0.6
g=[-9.81; 0; 0];
% g=[0; 0; 0];

%Inertia Tensor of the kth link about Center-Of-Mass (COM) in ith frame
%which is rigid attach to the link
Icxx=zeros(n,1);Icyy=zeros(n,1);Iczz=zeros(n,1); % Initialization
Icxy=zeros(n,1);Icyz=zeros(n,1);Iczx=zeros(n,1); % Initialization
for count = 1:21
Icxx(count)=0.5*m(count)*rod_radius*rod_radius;
Icyy(count)=(1/12)*m(count)*length_array(count)*length_array(count);
Iczz(count)= Icyy(count);
end

% End_effector Inertia
Icxx(6)=(1*0.4*0.4)/2;   Icyy(6)= (1*0.4*0.4)/4;   Iczz(6)=(1*0.4*0.4)/4;
```

→ subsystem I → Subsystem II

Modified Denavit-Hartenberg (DH) parameters

The DH parameters were originally proposed by Denavit and Hartenberg (1955) and widely used to define links' configuration of a robotic manipulator consisting of one degree-of-freedom (DOF) joints, i.e., revolute or prismatic. Later, Khalil and Kleinfinger (1986) showed that the DH parameters are powerful tool for serial robots, but, lead to ambiguities in the case of closed and tree structured robots. They presented the modified DH parameter from its original definition. Craig (1991) also used modified DH notation for serial robots. The modified DH parameters are used in Recursive Dynamics Simulator (ReDySim) and illustrated next.

In order to define configuration of a link relative to its neighbours, coordinate frames are attached to the links. The frames $\mathcal{F}_{k-1}$ (O_{k-1}-$X_{k-1}Y_{k-1}Z_{k-1}$) and $\mathcal{F}_k$ (O_k-$X_kY_kZ_k$) are rigidly attached to links (k-1) and k, respectively. The axis Z_{k-1} is along the (k-1)st joint, whereas the origin O_{k-1} is located on a point where common normal to Z_{k-1} and Z_k intersect Z_{k-1}. The axis X_{k-1} is along common normal to Z_{k-1} and Z_k. Moreover, axis Y_{k-1} is chosen to complete a right handed frame. It may be noted that

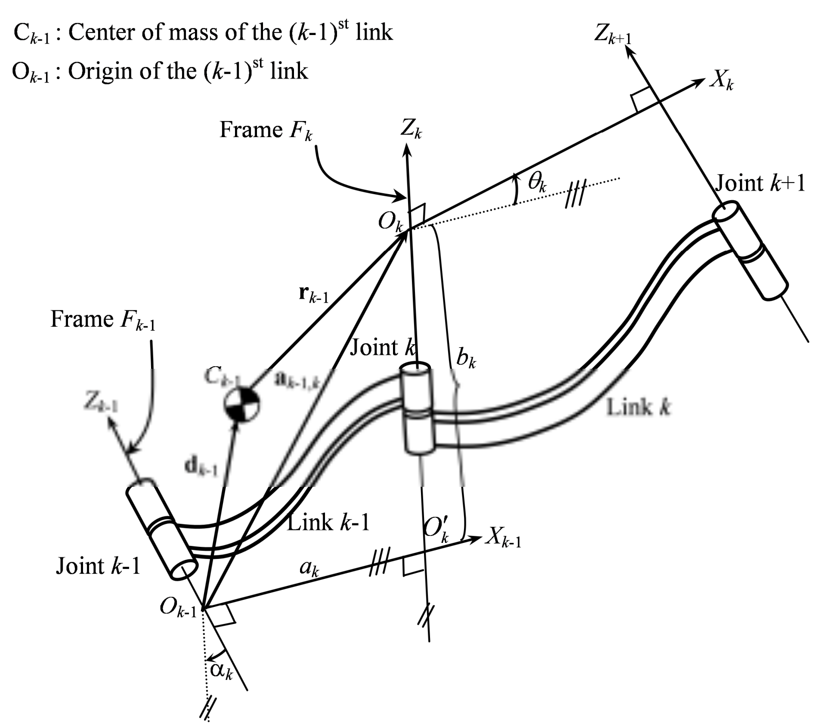


Fig. 1 Frame convention for modified DH parameters

frame $\mathcal{F}_0$, O_0-$X_0Y_0Z_0$, may be arbitrarily chosen, and hence, Z_0 is chosen coincident with Z_1 for the sake of simplification.

Once the link frames have been established, the position and the orientation of the frame k with respect k-1 are specified by the four parameters known as DH parameter. Out of these four parameters only one parameter is variable whereas others are constant. These link parameters in terms of the link frames can be obtained as follows:

Twist angle (α_k) = the angle from $Z_{k\text{-}1}$ to Z_k about$X_{k\text{-}1}$

Link length (a_k) = the distance from $Z_{k\text{-}1}$ to Z_k along $X_{k\text{-}1}$

Joint offset (b_k) = the distance from $X_{k\text{-}1}$ to X_k along Z_k

Joint angle (θ_k) = the angle from $X_{k\text{-}1}$ to X_k about Z_k

← This order was used for deriving the DH-parameters

θ_k or b_k being the joint variable depending on the type of the joint. If the joint is revolute than θ_k is the joint variable, whereas b_k is the joint variable in the case of prismatic joint. Based on the above DH parameters, rotation matrix between the frames k and k-1 can be obtained as follows

$$\mathbf{Q}_k \equiv \begin{bmatrix} C\theta_k & -S\theta_k & 0 \\ S\theta_k C\alpha_k & C\theta_k C\alpha_k & -S\alpha_k \\ S\theta_k S\alpha_k & C\theta_k S\alpha_k & C\alpha_k \end{bmatrix} \tag{1}$$

The representation of the position vector $\mathbf{a}_{k-1,k}$, measured from the origin of link k-1 to the origin of link k, in frame k-1 is given by

$$\left[\mathbf{a}_{k-1,k}\right]_{k-1} \equiv \begin{bmatrix} a_k \\ -b_k S\alpha_k \\ b_k C\alpha_k \end{bmatrix} \tag{2}$$

It is worth noting that the vector $\mathbf{a}_{k-1,k}$ in Eq. (2) is constant in frame k-1, if (k-1)st joint is revolute. Similarly, vector $\mathbf{d}_{k\text{-}1}$ is also constant in frame k-1.

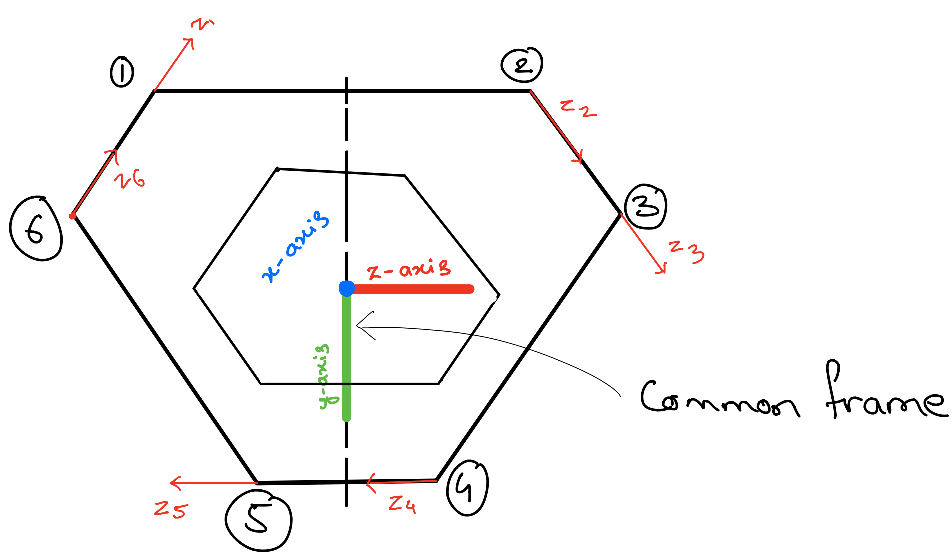
rangle_vectclock = [pi/3, -pi/3, -pi/3, pi, pi, pi/3];

alp=[rangle_vectclock(1), 0, pi/2, -pi/2, -pi/2, -pi/2, rangle_vectclock(2), 0, pi/2, rangle_vectclock(3), 0, pi/2, rangle_vectclock(4), 0, pi/2, rangle_vectclock(5), 0, pi/2, rangle_vectclock(6), 0, pi/2];

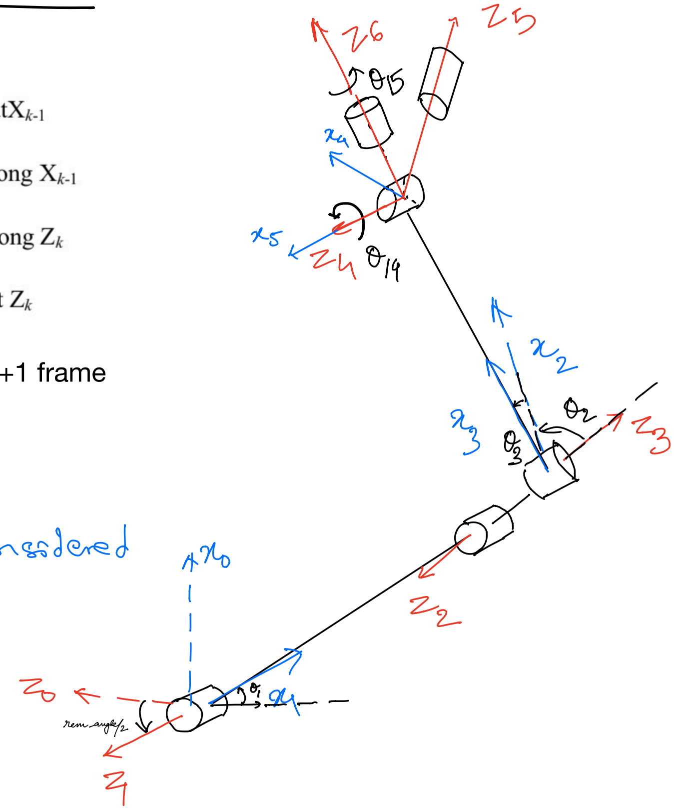
a=[0 l1 0 L2 0 0 0 l1 0 0 l1 0 0 l1 0 0 l1 0 0 l1];

Explaining the angles:

While writing the modified DH parameters, a common frame has been chosen as shown below:



→ DH parameters:



Twist angle (α_k) = the angle from Z_{k-1} to Z_k about X_{k-1}

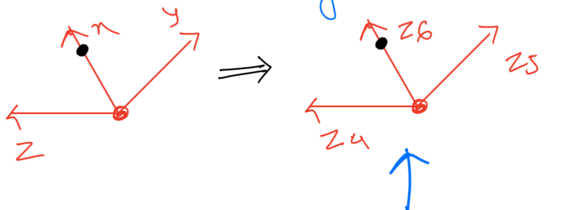
Link length (a_k) = the distance from Z_{k-1} to Z_k along X_{k-1}

Joint offset (b_k) = the distance from X_{k-1} to X_k along Z_k

Joint angle (θ_k) = the angle from X_{k-1} to X_k about Z_k

For reference to traverse from i th frame to i+1 frame

* As the endeffector is considered to be along x-axis,



Such that, $[z_6 \times z_5 = z_4]$

Subsystem ①

$Z_0 \to Z_1$

a	α	d	θ
0	rangle_vectclock(1)	0	θ_1
l_1	0	0	θ_2
0	$\pi/2$	0	θ_3
l_2	$-\pi/2$	0	θ_4
0	$-\pi/2$	0	θ_5
0	$-\pi/2$	0	θ_6

Subsystem II	$z_0 \rightarrow z_2$	0	rangle_vectclock(2)	0	θ_1
		l_1	0	0	θ_2
		0	$\pi/2$	0	θ_3
Subsystem III	$z_0 \rightarrow z_3$	0	rangle_vectclock(3)	0	θ_1
		l_1	0	0	θ_2
		0	$\pi/2$	0	θ_3
Subsystem IV	$z_0 \rightarrow z_4$	0	rangle_vectclock(4)	0	θ_1
		l_1	0	0	θ_2
		0	$\pi/2$	0	θ_3
Subsystem V	$z_0 \rightarrow z_5$	0	rangle_vectclock(5)	0	θ_1
		l_1	0	0	θ_2
		0	$\pi/2$	0	θ_3
Subsystem VI	$z_0 \rightarrow z_6$	0	rangle_vectclock(6)	0	θ_1
		l_1	0	0	θ_2
		0	$\pi/2$	0	θ_3

iPad 4:21 PM 14%

Inverse Kinematics

inv_kine.m

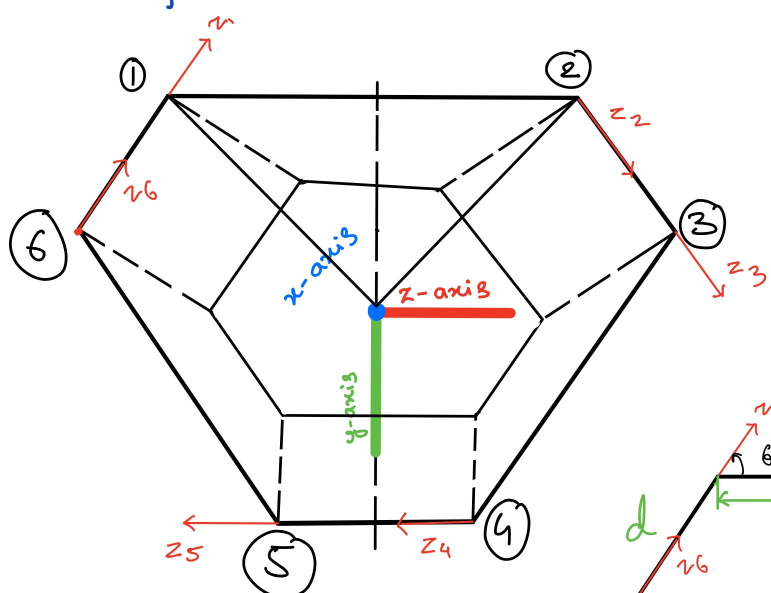
```
function [th, dth, ddth, J] = inv_kine(thi, dthi, ddthi)


[base_length, top_length, half_angle, rem_angle, Base_matrix, l1, L2, theta_p, r_p] = pa
ex = thi(1);
ey = thi(2);
ez = thi(3);
roll = thi(4);
pitch = thi(5);
yaw = thi(6);

[Top_matrix] = end_effector(ex, ey, ez, roll, pitch, yaw, theta_p, r_p);

rangle_vectclock = [pi/3, -pi/3, -pi/3, pi, pi, pi/3];
VL = r_p;


for i = 1:6
    if i ==1
        k = 1;
    elseif i ==2
        k = 7;
    end
    Rx_clock = [1, 0, 0;0, cos(rangle_vectclock(i)), sin(rangle_vectclock(i));0, -sin(ra
    T = Rx_clock*(Top_matrix(:,i)-Base_matrix(:,i));
    th(k+2) = -asin(T(3)/L2);
    l2 = L2*cos(th(k+2));
    th(k+1) = acos((T(1)^2 + T(2)^2 - l1^2 - l2^2)/(2*l1*l2));
    r = sqrt(l1^2 + l2^2 + 2*l1*l2*cos(th(k+1)));
    phi = atan((l1 + l2*cos(th(k+1)))/(l2*sin(th(k+1))));
    th(k) = asin(T(1)/r)-phi;
    k = k+3;
end

%Spherical angles
%theta14 about z axis
end_eff = [ex; ey; ez];
Rx_clock = [1, 0, 0;0, cos(pi/3), sin(pi/3);0, -sin(pi/3), cos(pi/3)];
Rz_clock_12 = [cos(th(2) - th(1)), sin(th(2) - th(1)), 0;-sin(th(2) - th(1)), cos(th(2) -
Ry_clock_3 = [cos(th(3)) 0 -sin(th(3));0 1 0;sin(th(3)) 0 cos(th(3))];

d_end_effector_wrt_vertex1rot = Ry_clock_3*Rz_clock_12*Rx_clock*(end_eff - Top_matrix(:,
th(5) = -asin(d_end_effector_wrt_vertex1rot(3)/VL);
th(4) = pi/2 - asin(d_end_effector_wrt_vertex1rot(1)/(VL*cos(th(5))));
R_theta14 = [cos(th(4)), -sin(th(4)), 0;sin(th(4)), cos(th(4)), 0;0, 0, 1];
R_theta15 = [cos(th(5)) 0 sin(th(5));0 1 0;-sin(th(5)) 0 cos(th(5))];
vertex_2_wrt_vertex1 = Top_matrix(:,2)-Top_matrix(:,1);
vertex_2_wrt_vertex1_rot = Ry_clock_3*Rz_clock_12*Rx_clock*vertex_2_wrt_vertex1;
vertex_2_interim = R_theta15'*R_theta14'*vertex_2_wrt_vertex1_rot;
[top_matrix_regen, end_effector_regen] = end_effector_regeneration(r_p, theta_p);
th(6) = -asin(vertex_2_interim(2)/top_matrix_regen(1,1));

th0 = th;

for corr_i = 1:6
    if corr_i ==1
        corr_k = 1;
    elseif corr_i ==2
        corr_k = 7;
    end
    th(corr_k+2) = -th(corr_k+2);
    th(corr_k) = -th(corr_k);
    corr_k = corr_k+3;
end
th(5) = -pi/2 + th(5);
```

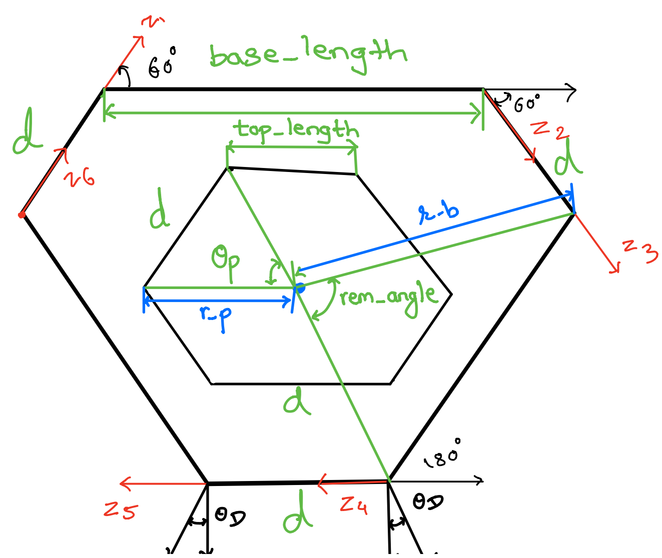
Rx_clock = [1, 0, 0;0, cos(rangle_vectclock(i)), sin(rangle_vectclock(i));0, -sin(rangle_vectclock(i)), cos(rangle_vectclock(i))];

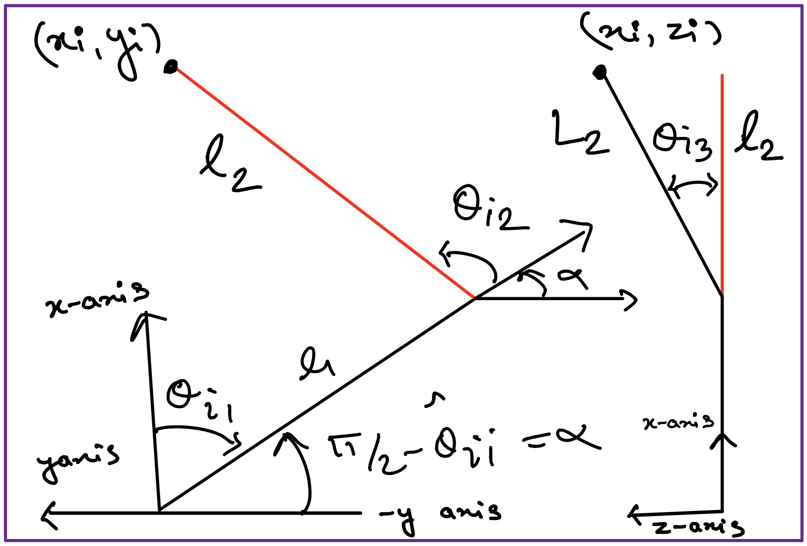
T = Rx_clock*(Top_matrix(:,i)-Base_matrix(:,i))

This expression defines top matrix w.r.t the base matrix in the subsystem's frame.

Now the angles can be found out by considering each subsystem as a 3-link manipulator.







One leg of stewart platform.

$$\boxed{\theta_{i3} = -asin*\left(\frac{z}{L_2}\right)}$$

$$l_2 = L_2 * \cos(\theta_{i3})$$

$$x_i = l_1 \sin(\alpha) + l_2 \sin(\theta_{i2}+\alpha)$$
$$= l_1 \sin(\pi/2 - \theta_{i1}) + l_2 \sin(\pi/2 + \theta_{i2} - \theta_{i1})$$

$$\boxed{x_i = l_1 * \cos(\theta_{i1}) + l_2 * \cos(\theta_{i1} - \theta_{i2})}$$

$$-y_i = l_1 \cos(\alpha) + l_2 \cos(\theta_{i2}+\alpha)$$
$$= l_1 \cos(\pi/2 - \theta_{i1}) + l_2 \cos(\theta_{i2} + \pi/2 - \theta_{i1})$$

$$\boxed{y_i = -[l_1 \sin(\theta_{i1}) + l_2 \sin(\theta_{i1} - \theta_{i2})]}$$

$$x_i^2 + y_i^2 = l_1^2 + l_2^2 + 2 l_1 l_2 [\cos(\theta_{i1} - (\theta_{i1} - \theta_{i2}))]$$
$$= l_1^2 + l_2^2 + 2 l_1 l_2 \cos(\theta_{i2})$$

$$\therefore \boxed{\theta_{i2} = \operatorname{acos}\left[\frac{x_i^2 + y_i^2 - l_1^2 - l_2^2}{2 * l_1 * l_2}\right]}$$

$$x_i = l_1 \cos(\theta_{i1}) + l_2 \cos(\theta_{i1} - \theta_{i2})$$
$$= l_1 \cos(\theta_{i1}) + l_2 [\cos(\theta_{i1}) \cdot \cos(\theta_{i2}) + \sin(\theta_{i2}) \sin(\theta_{i1})]$$

$$x_i = \cos\theta_{i1} [l_1 + l_2 \cos(\theta_{i2})] + l_2 \sin(\theta_{i2}) \cdot \sin(\theta_{i1})$$
$$= m \sin\beta \cdot \cos\theta_{i1} + m \cos\beta \cdot \sin(\theta_{i1})$$
$$= m \sin(\beta + \theta_{i1})$$

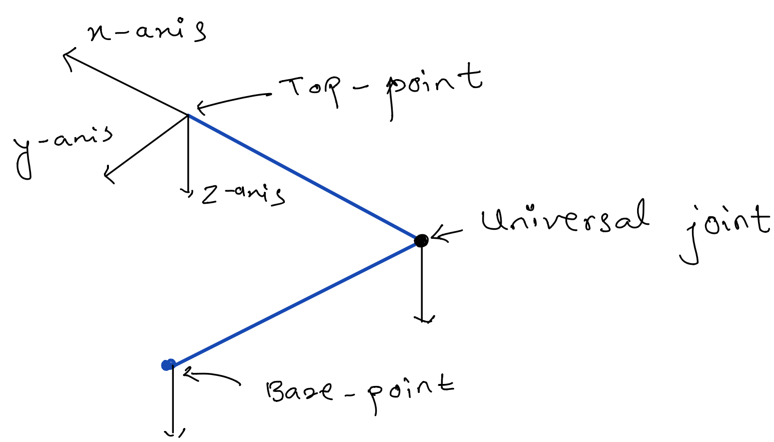
$$m^2 = (l_1 + l_2 \cos\theta_{i2})^2 + (l_2 \sin(\theta_{i2}))^2$$
$$m^2 = l_1^2 + l_2^2 + 2 l_1 l_2 \cos\theta_{i2}$$

$$\therefore m = \sqrt{l_1^2 + l_2^2 + 2 l_1 l_2 \cos\theta_{i2}}$$

$$\beta = \operatorname{atan}\left(\frac{l_1 + l_2\cos(\theta_{i2})}{l_2\sin(\theta_{i2})}\right)$$

$$\therefore \quad \boxed{\theta_{i1} = \operatorname{asin}\left(\frac{x}{m}\right) - \beta}$$

To find the three angles of a spherical joint we write the the end-effector point in a frame attached to second link.



θ_{i4} and θ_{i5} are subsequently calculated by reaching the end_effector point similar to reaching a point on a sphere of radius r_p

θ_{16} is calculated by reaching the top second vertex with respect to the frame attached to the top first vertex. as we already know, θ_{14} and θ_{15}.

$$[\text{Second\_vertex}] = R_{\theta_{14}} * R_{\theta_{15}} * R_{\theta_{16}} * \begin{bmatrix}\text{second-v} \\ \text{wrt first v}\end{bmatrix}$$

J = analytical_jacobian(r_p, theta_p, l1, L2, th);

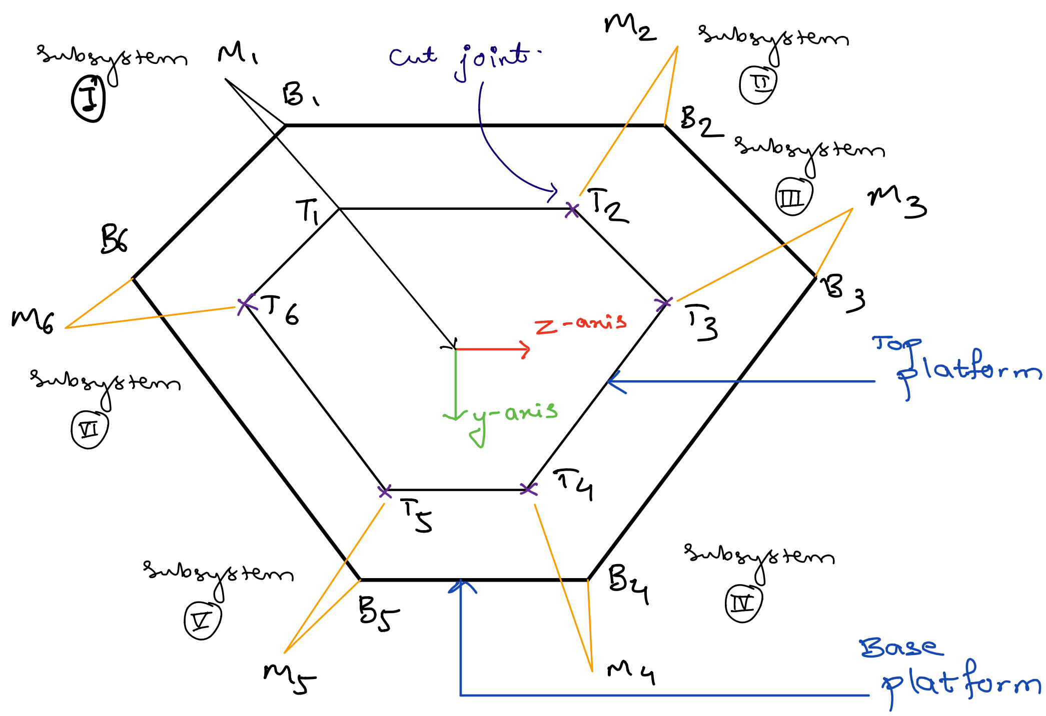
G = G_RedySim(ex, ey, ez, roll, pitch, yaw, base_length, top_length, l1, L2, half_angle, rem_angle, th0, r_p, theta_p, vertex_2_wrt_vertex1_flat);

as defined above, J will be now derived:

Property of the analytical jacobian is:

$$J\dot{q} = 0$$

We will first discuss loop closure at position level.
The following figure shows the cut-joints and all the subsystems.



The loop closure eq^ns will be derived as:

$$\overrightarrow{B1M1} + \overrightarrow{M1T1} + \overrightarrow{T1T2} = \overrightarrow{B2M2} + \overrightarrow{M2T2}$$
$$\overrightarrow{B1M1} + \overrightarrow{M1T1} + \overrightarrow{T1T3} = \overrightarrow{B3M3} + \overrightarrow{M3T3}$$
$$\overrightarrow{B1M1} + \overrightarrow{M1T1} + \overrightarrow{T1T4} = \overrightarrow{B4M4} + \overrightarrow{M4T4}$$
$$\overrightarrow{B1M1} + \overrightarrow{M1T1} + \overrightarrow{T1T5} = \overrightarrow{B5M5} + \overrightarrow{M5T5}$$
$$\overrightarrow{B1M1} + \overrightarrow{M1T1} + \overrightarrow{T1T6} = \overrightarrow{B6M6} + \overrightarrow{M6T6}$$

Differentiating all the above eq^ns we get

$$J_{12}\left[\dot{\theta}_{11}\ \dot{\theta}_{12}\ \dot{\theta}_{13}\ \dot{\theta}_{14}\ \dot{\theta}_{15}\ \dot{\theta}_{16}\right]^T = J_2\left[\dot{\theta}_{21}\ \dot{\theta}_{22}\ \dot{\theta}_{23}\right]$$

$$J_{13}\left[\dot\theta_{11}\ \dot\theta_{12}\ \dot\theta_{13}\ \dot\theta_{14}\ \dot\theta_{15}\ \dot\theta_{16}\right]^T = J_3\left[\dot\theta_{31}\ \dot\theta_{32}\ \dot\theta_{33}\right]^T$$

$$J_{14}\left[\dot\theta_{11}\ \dot\theta_{12}\ \dot\theta_{13}\ \dot\theta_{14}\ \dot\theta_{15}\ \dot\theta_{16}\right]^T = J_4\left[\dot\theta_{41}\ \dot\theta_{42}\ \dot\theta_{43}\right]^T$$

$$J_{15}\left[\dot\theta_{11}\ \dot\theta_{12}\ \dot\theta_{13}\ \dot\theta_{14}\ \dot\theta_{15}\ \dot\theta_{16}\right]^T = J_5\left[\dot\theta_{51}\ \dot\theta_{52}\ \dot\theta_{53}\right]^T$$

$$J_{16}\left[\dot\theta_{11}\ \dot\theta_{12}\ \dot\theta_{13}\ \dot\theta_{14}\ \dot\theta_{15}\ \dot\theta_{16}\right]^T = J_6\left[\dot\theta_{61}\ \dot\theta_{62}\ \dot\theta_{63}\right]^T$$

The above 5 eq^{n}s can be rearranged in matrix form to get a form

$$J\dot{q} = 0$$

$$15\ \Big\downarrow \underbrace{\begin{bmatrix} -J_{12_{3\times6}} & J_{2_{3\times3}} & & & & \\ -J_{13_{3\times6}} & & J_{3_{3\times3}} & & & \\ -J_{14_{3\times6}} & & & J_{4_{3\times3}} & & \\ -J_{15_{3\times6}} & & & & J_{5_{3\times3}} & \\ -J_{16_{3\times6}} & & & & & J_{6_{3\times3}} \end{bmatrix}}_{15\times21} \underbrace{\begin{bmatrix} \theta_{11} \\ \downarrow \\ \theta_{16} \\ \theta_{21} \\ \downarrow \\ \theta_{23} \\ \vdots \\ \theta_{61} \\ \downarrow \\ \theta_{63} \end{bmatrix}}_{21\times1} = 0$$

$$\longrightarrow 21$$

G-matrix :

In explicit loop closure "G" is defined as :

$$q = q(y)$$

$$\dot{q} = G\dot{y}$$

$$\ddot{q} = G\ddot{y} + \dot{G}\dot{y}$$

1) $\theta_{i3} = -\mathrm{asin} * \left(\frac{z}{L_2}\right)$

$\therefore \quad -\sin(\theta_3) = \frac{z}{L_2}$

$\therefore \quad -\cos(\theta_3) \cdot \dot{\theta}_3 = \frac{\dot{z}}{L_2}$

$$\dot{z} = \begin{bmatrix} \mathrm{diff}(z, ex) \\ \mathrm{diff}(z, ey) \\ \mathrm{diff}(z, ez) \\ \mathrm{diff}(z, roll) \\ \mathrm{diff}(z, pitch) \\ \mathrm{diff}(z, yaw) \end{bmatrix}$$

$\therefore \quad \boxed{\dot{\theta}_3 = \underbrace{\left[\frac{-1}{L_2 \cdot \cos(\theta_3)} * \dot{z}\right]}_{1\times 6} \underbrace{\dot{y}}_{6\times 1}}$

2) $\theta_{i2} = \mathrm{acos}\left[\frac{x_i^2 + y_i^2 - l_1^2 - l_2^2}{2 * l_1 * l_2}\right]$

Here, $l_2 = L_2 * \cos\theta_3$

$$2 * l_1 * l_2 \cos(\theta_2) = x^2 + y^2 - l_1^2 - l_2^2 \quad \text{— (I)}$$

<u>differentiating eqn (I)</u>

$$2l_1\dot{l}_2\cos\theta_2 - 2l_1l_2\sin\theta_2\cdot\dot{\theta}_2 = 2x\dot{x} + 2y\dot{y} - 2l_2\cdot\dot{l}_2$$

$$\boxed{\dot{\theta}_2 = \underset{1\times 6}{\left[(2x\dot{x} + 2y\dot{y} - 2l_2\cdot\dot{l}_2 - 2l_1\dot{l}_2\cos\theta_2)/(-2l_1l_2\sin\theta_2)\right]}\ \underset{6\times 1}{\dot{y}}}$$

$$\dot{l}_2 = \frac{d(L_2 * \cos\theta_3)}{dt} = L_2 * -\sin\theta_3\cdot\dot{\theta}_3$$

3) $\theta_{i1} = a\sin\left(\frac{x}{m}\right) - \beta$

$$m = \sqrt{l_1^2 + l_2^2 + 2l_1l_2\cos\theta_{i2}} \quad \text{— (II)}$$

$\therefore m\sin(\theta_1 + \beta) = x$

$\therefore \dot{m}\sin(\theta_1+\beta) + m\cos(\theta_1+\beta)(\dot{\theta}_1 + \dot{\beta}) = \dot{x}$

from eq^n (I) and eq^n (II)

$m^2 = x^2 + y^2$

$\therefore \dot{m} = (2x\cdot\dot{x} + 2y\cdot\dot{y})/2*m$

$$\beta = \operatorname{atan}\left(\frac{l_1 + l_2\cos(\theta_{i2})}{l_2\sin(\theta_{i2})}\right)$$

$$\tan\beta = \frac{l_1 + l_2\cos\theta_2}{l_2\sin\theta_2}$$

$$= \frac{l_1}{l_2\sin\theta_2} + \cot\theta_2$$

$$\sec^2\beta\cdot\dot\beta = \frac{-l_1\left(\dot l_2\sin\theta_2 + l_2\cos\theta_2\cdot\dot\theta_2\right)}{l_2^2\cdot\sin^2\theta_2\cdot} - \frac{1}{\sin^2\theta_2}\cdot\dot\theta_2$$

$$\dot m\sin(\theta_1+\beta) + m\cos(\theta_1+\beta)(\dot\theta_1+\dot\beta) = \dot x$$

$$\boxed{\dot\theta_1 = \left[\frac{\dot x - \dot m\sin(\theta_1+\beta)}{m\cos(\theta_1+\beta)} - \dot\beta\right]_{1\times 6}\cdot \dot y_{\,6\times 1}}$$

4) $$\theta_{i5} = -\operatorname{asin}\left(\frac{ee\_wrt\_v1(3)}{VL}\right) \quad \text{— III}$$

here, ee_wrt_v1 is the end effector vector with respect to top-vertex 1.

Rearranging and differentiating eq^n III

$$\boxed{\dot{\theta}_{15} = \left[\frac{diff(ee\_wrt\_v1(3), t)}{VL * \cos\theta_{15}}\right]_{1\times 6} \dot{y}_{6\times 1}}$$

5) $\theta_{14} = \frac{\pi}{2} - asin\left(\frac{ee\_wrt\_v1(1)}{VL * \cos(\theta_{15})}\right)$

$$\cos\theta_{14} = \frac{-ee\_wrt\_v1(1)}{VL * \cos(\theta_{15})} \quad — \text{IV}$$

$$\dot{\theta}_{14} = \frac{-[diff(d\_ende)\cos\theta_5 + \sin\theta_5 * d\_ende]}{VL * \cos^2\theta_5}$$

$$\boxed{\dot{\theta}_{14} = -\left[\frac{diff(ee\_wrt\_v1(1), t)\cdot\cos\theta_{15} + \sin\theta_{15} * ee\_wrt\_v1(1)}{VL * \cos^2\theta_{15}}\right]_{1\times 6} \dot{y}_{6\times 1}}$$

6) Similarly $\dot{\theta}_{16}$ is derived such that

$$\boxed{\dot{\theta}_{16} = [U]_{1\times 6}\, \dot{y}_{6\times 6}}$$

Thus, the derived explicit loop closure jacobian is of dimension $\underline{21 \times 6}$.

$$\underset{21\times 1}{\dot{q}} = \underset{21\times 6}{G}\ \underset{6\times 1}{\dot{y}}$$

This vector is used to confirm the analytical jacobian.

$$\ddot{q} = G\ddot{y} + \dot{G}\dot{y}$$

$$\dot{G} = \mathrm{diff}(G, t)$$

This matrix is easily derived in MATLAB's symbolic environment.

now we have all the outputs required for the inverse dynamics implementation.