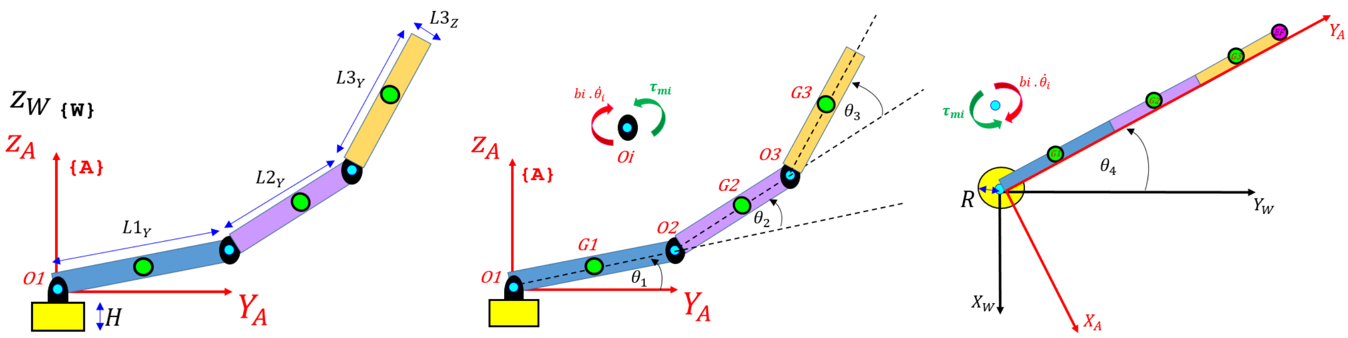
Explore the dynamics of a 4-dof Robotic manipulator

In this example we're going to derive and then implement the equations of motion for a 4-dof robotic manipulator. Specifically we're going to derive the equations of motion using's ***Lagrange's method***. The system that we're going to explore is shown below. At each joint we have:

- τ_m : Actuation torques (eg: by electric motors)
- $b.\dot{\theta}$: Viscous damping torques



The system equation of motion that we'll be deriving has the following general form:

$$M(q,\dot{q}).\ddot{q} \; + \; C(q,\dot{q}).\dot{q} \; + \; K(q).q \; + \; g(q) \; = \; Q(\tau,\dot{q})$$

Background:

In last week's class we practiced applying Lagrange's equation to a Spring Mass Damper (SMD) system. Today we're going to follow exactly the same process as the SMD case, ie:

1. Define Model Parameters
2. Apply the governing physics
3. Apply Lagrange's equation
4. Isolate our expression for M, C, K, g, Q
5. Convert our Analytical expression for M, C, K, g, Q into a Simulink block
6. Simulate of model of this dynamic system

Euler-Lagrange equations:

The Euler-Lagrange formula will be used to derive the equations of motion for our robotic manipulator, and it has the form:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Q_k \quad \text{for } k = 1, 2, \ldots, n$$

where n is the DOF of the system, $\{q_1, q_2, \ldots, q_n\}$ is a set of generalized coordinates, $\{Q_1, Q_2, \ldots, Q_n\}$ is the set of generalized forces associated with those coordinates, and the Lagrangian: ***L = T - V***, is defined as the difference between the kinetic and potential energy of the n- DOF system. The Generalised forces can also be defined in terns of the non conservative forces and torques acting on the multibody system. The formula for the generalised forces acting on the system is:

$$Q_k = \sum_{i=1}^{Nf_{nc}} \left(\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_k} \right) + \sum_{j=1}^{N\tau_{nc}} \left(\vec{\tau}_j \cdot \frac{\partial \vec{\omega}_j}{\partial \dot{q}_k} \right)$$

where:

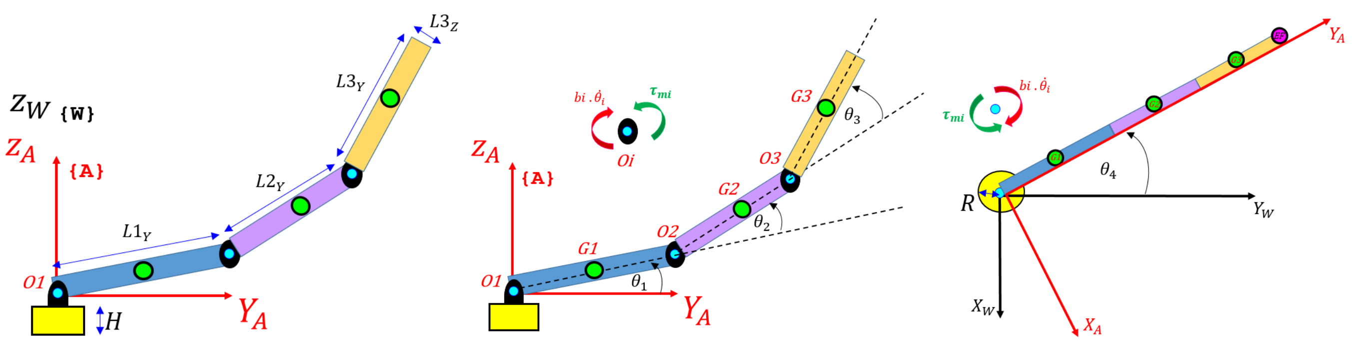
- Q_k : is the generalised force associated with the k^{th} generalised co-ordinate q_k
- Nf_{nc} : is the number of active NON conservative forces
- $N\tau_{nc}$: is the number of active NON conservative TORQUES
- $\vec{v}_i$: is the velocity vector of the point associated with the applied force.
- $\vec{\omega}_i$: is the angular velocity about the point associated with the applied torque.

Note :

- For our system, we will choose $q = \{\theta_1, \theta_2, \theta_3, \theta_4\}$
- This is a 4 degree of freedom system, and as such there are four 2nd order ODEs that can be derived.

`Bradley Horton : 13-Sep-2016, bradley.horton@mathworks.com.au`

STEP_1: Define Model parameters



Here are some model parameters:

```
syms    m2_s      m1_s       m3_s       m4_s  % masses
syms    b1_s      b2_s       b3_s       b4_s  % damping
syms taum1_s   taum2_s    taum3_s    taum4_s  % motor torques
syms     g_s                                  % gravity

syms L1X_s L2X_s L3X_s                        % lengths for LINK #1
syms L1Z_s L2Z_s L3Z_s                        % lengths for LINK #2
syms L1Y_s L2Y_s L3Y_s                        % lengths for LINK #3
syms H4_s  R4_s                               % lengths for TURNTABLE
```

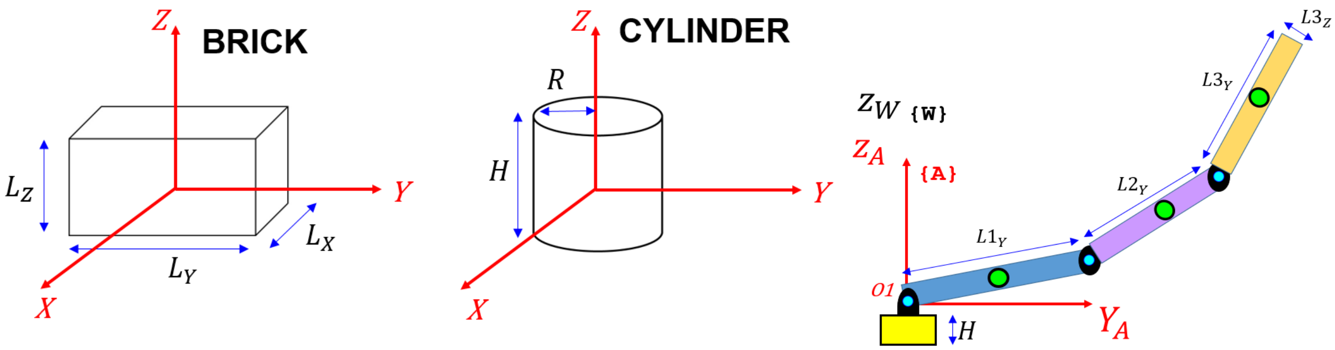
Here are the angles that describe the pose for our machine. These 4 angles are also the generalised co-ordinates that we'll use with Lagrange's equation:

```
syms t theta1(t)  TH1_s
syms   theta2(t)  TH2_s
syms   theta3(t)  TH3_s
```

```
syms   theta4(t)  TH4_s
```

Defining component INERTIAs:

The manipulator is made up of 3 rectangular prisms - which form the 3 links of the robotic "arm". The robot also has a 4th link which is the cylindrical base or "turntable". The inertias for these fundamental shapes can be computed about a local body frame positioned at their center of mass (CoM):



So let's define the INERTIA matrices for these components about their body fixed **center of mass** frames. First let's define some formulas for the inertias of these fundamental shapes:

```
I_brick = @(Lx,Ly,Lz,m)...
    ( [m*(Ly^2 + Lz^2)/12,  0,                    0; ...
                        0,  m*(Lx^2 + Lz^2)/12,   0; ...
                        0,  0,                    m*(Lx^2 + Ly^2)/12;] );

I_cyl = @(H,R,m)...
    ( [m*(H^2 + 3*R^2)/12,  0,                    0; ...
                        0,  m*(H^2 + 3*R^2)/12,   0; ...
                        0,  0,                    m*(R^2)/2;] );
```

So the INERTIA matrices for our 4 bodies are:

```
I1_s = I_brick(L1X_s, L1Y_s, L1Z_s, m1_s);
I2_s = I_brick(L2X_s, L2Y_s, L2Z_s, m2_s);
I3_s = I_brick(L3X_s, L3Y_s, L3Z_s, m3_s);
I4_s = I_cyl(H4_s,  R4_s, m4_s);
```

As a concrete example, here's what the inertia matrix for Link 1, looks like:

```
I1_s
```

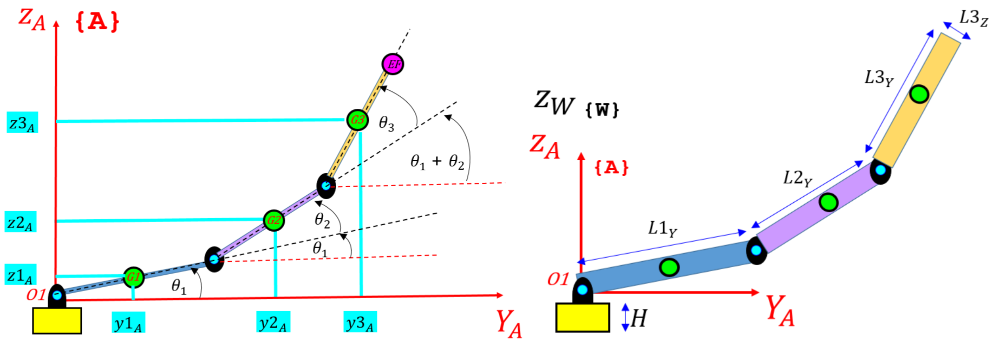
I1_s =

$$\begin{pmatrix} \frac{m_{1,s}\,(\mathrm{L1Y}_s^{\,2}+\mathrm{L1Z}_s^{\,2})}{12} & 0 & 0 \\ 0 & \frac{m_{1,s}\,(\mathrm{L1X}_s^{\,2}+\mathrm{L1Z}_s^{\,2})}{12} & 0 \\ 0 & 0 & \frac{m_{1,s}\,(\mathrm{L1X}_s^{\,2}+\mathrm{L1Y}_s^{\,2})}{12} \end{pmatrix}$$

STEP_2: Apply the governing physics - PART 1 of 5

Define Center of mass positions and TRANSLATIONAL velocities for ARM links:

In the diagrams below, we've defined the inertial(fixed) World co-ordinate frame to be at the pivot point between links 1 and 4 - this {W}-frame is fixed in space ... it does NOT move. We've also defined a frame called the {A-frame} - the {A}-frame shares the same origin as the {W}-frame, however the {A}-frame may rotate about the common ${}^{W}Z$ and ${}^{A}Z$ axes.



First, let's define the x,y,z position of the **centre of mass** (CoM) for each link - and we'll define these positions in terms of the {A}-Frame. And after we've done this, we'll convert the CoM positions to our inertial {W}-frame, and then we can differentiate to get translational velocities:

Consider LINK_1: - *where is the CoM relative to the {A}-Frame*

```
G1      = sym([0;0;0]);
G1(2,1) = (L1Y_s/2)*cos(theta1(t));
G1(3,1) = (L1Y_s/2)*sin(theta1(t));
```

Consider LINK_2: - *where is the CoM relative to the {A}-Frame*

```
alpha   = theta1(t) + theta2(t);
G2      = sym([0;0;0]);
G2(2,1) = L1Y_s*cos(theta1(t)) + (L2Y_s/2)*cos(alpha);
G2(3,1) = L1Y_s*sin(theta1(t)) + (L2Y_s/2)*sin(alpha);
```

Consider LINK_3: - *where is the CoM relative to the {A}-Frame*

```
beta           = theta1(t) + theta2(t) + theta3(t);
G3      = sym([0;0;0]);
G3(2,1) = L1Y_s*cos(theta1(t)) + L2Y_s*cos(alpha) + (L3Y_s/2)*cos(beta);
G3(3,1) = L1Y_s*sin(theta1(t)) + L2Y_s*sin(alpha) + (L3Y_s/2)*sin(beta);
```

As a concrete example, here's the {x,y,z} co-ordinate of the centre of mass for LINK-3, expressed in components of the {A}-frame:

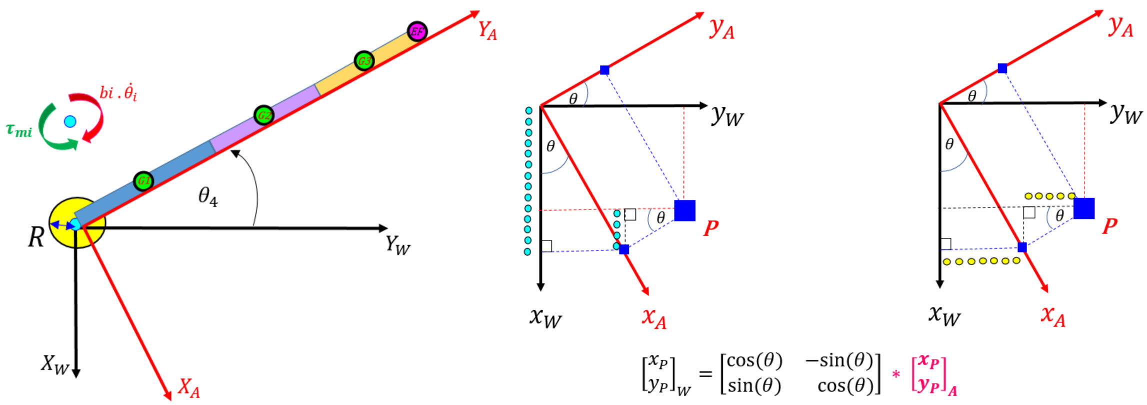
```
G3
```

G3 =

$$\begin{pmatrix} 0 \\ \frac{\mathrm{L3Y}_s \cos\left(\theta_1(t)+\theta_2(t)+\theta_3(t)\right)}{2} + \mathrm{L2Y}_s \cos\left(\theta_1(t)+\theta_2(t)\right) + \mathrm{L1Y}_s \cos\left(\theta_1(t)\right) \\ \mathrm{L1Y}_s \sin\left(\theta_1(t)\right) + \frac{\mathrm{L3Y}_s \sin\left(\theta_1(t)+\theta_2(t)+\theta_3(t)\right)}{2} + \mathrm{L2Y}_s \sin\left(\theta_1(t)+\theta_2(t)\right) \end{pmatrix}$$

Convert co-ordinates into the {W}-Frame:

We can now convert these {A}-Frame co-ordinates into their corresponding {W}-frame co-ordinates using the following transformation ${}^{W}r = {}^{W}R_A . {}^{A}r$:




```
wRa = [ cos(theta4(t)),  -sin(theta4(t)),   0;
        sin(theta4(t)),   cos(theta4(t)),   0;
                     0,                0,   1 ]
```

wRa =

$$\begin{pmatrix} \cos\left(\theta_4(t)\right) & -\sin\left(\theta_4(t)\right) & 0 \\ \sin\left(\theta_4(t)\right) & \cos\left(\theta_4(t)\right) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

So our {W}-Frame ***position*** co-ordinates for the link center of masses are:

```
WF_pos_G1 = wRa * G1;
WF_pos_G2 = wRa * G2;
WF_pos_G3 = wRa * G3;
```

As a concrete example, here's the {x,y,z} co-ordinate of the centre of mass for LINK-3, expressed in the {W}-frame:

```
WF_pos_G3
```

WF_pos_G3 =

$$\begin{pmatrix} -\sin(\theta_4(t))\,\sigma_1 \\ \cos(\theta_4(t))\,\sigma_1 \\ \mathrm{L1Y}_s \sin(\theta_1(t)) + \frac{\mathrm{L3Y}_s \sin(\theta_1(t) + \theta_2(t) + \theta_3(t))}{2} + \mathrm{L2Y}_s \sin(\theta_1(t) + \theta_2(t)) \end{pmatrix}$$

where

$$\sigma_1 = \frac{\mathrm{L3Y}_s \cos(\theta_1(t) + \theta_2(t) + \theta_3(t))}{2} + \mathrm{L2Y}_s \cos(\theta_1(t) + \theta_2(t)) + \mathrm{L1Y}_s \cos(\theta_1(t))$$

Now calculate our ***translational velocities*** for each arm link and express in terms of the {W}-Frame:

```
WF_tran_vel_G1 = diff( WF_pos_G1, t);
WF_tran_vel_G2 = diff( WF_pos_G2, t);
WF_tran_vel_G3 = diff( WF_pos_G3, t);
```

As a concrete example, here's what the translational velocity vector $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$ looks like for the center of mass of Link #3, expressed in components of the INERTIAL {W}-frame:

```
WF_tran_vel_G3
```

```
WF_tran_vel_G3 =
```

$$\begin{pmatrix} \sin(\theta_4(t))\,\sigma_1 - \cos(\theta_4(t))\,\sigma_2 \frac{\partial}{\partial t}\,\theta_4(t) \\ -\cos(\theta_4(t))\,\sigma_1 - \sin(\theta_4(t))\,\sigma_2 \frac{\partial}{\partial t}\,\theta_4(t) \\ \mathrm{L2Y}_s \cos(\theta_1(t) + \theta_2(t))\,\sigma_4 + \mathrm{L1Y}_s \cos(\theta_1(t)) \frac{\partial}{\partial t}\,\theta_1(t) + \frac{\mathrm{L3Y}_s\,\sigma_5\,\sigma_3}{2} \end{pmatrix}$$

where

$$\sigma_1 = \mathrm{L2Y}_s \sin(\theta_1(t) + \theta_2(t))\,\sigma_4 + \mathrm{L1Y}_s \sin(\theta_1(t)) \frac{\partial}{\partial t}\,\theta_1(t) + \frac{\mathrm{L3Y}_s \sin(\theta_1(t) + \theta_2(t) + \theta_3(t))\,\sigma_3}{2}$$

$$\sigma_2 = \frac{\mathrm{L3Y}_s\,\sigma_5}{2} + \mathrm{L2Y}_s \cos(\theta_1(t) + \theta_2(t)) + \mathrm{L1Y}_s \cos(\theta_1(t))$$

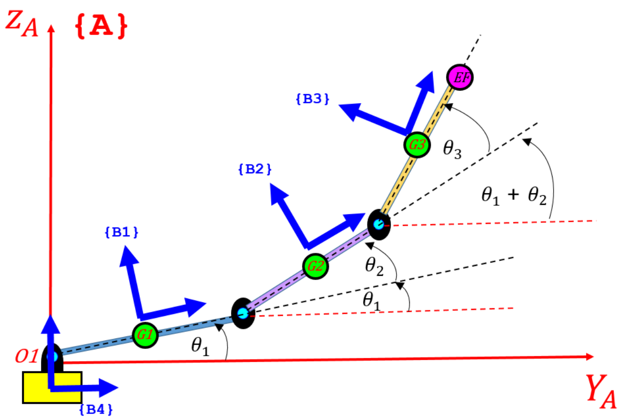
$$\sigma_3 = \frac{\partial}{\partial t}\,\theta_1(t) + \frac{\partial}{\partial t}\,\theta_2(t) + \frac{\partial}{\partial t}\,\theta_3(t)$$

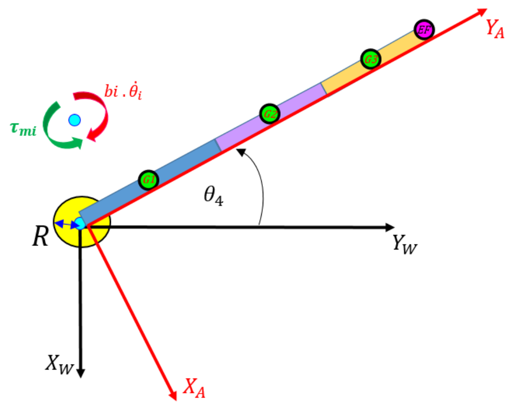
$$\sigma_4 = \frac{\partial}{\partial t}\,\theta_1(t) + \frac{\partial}{\partial t}\,\theta_2(t)$$

$$\sigma_5 = \cos(\theta_1(t) + \theta_2(t) + \theta_3(t))$$

STEP_2: Apply the governing physics - PART 2 of 5

Define ROTATIONAL velocities for all bodies:





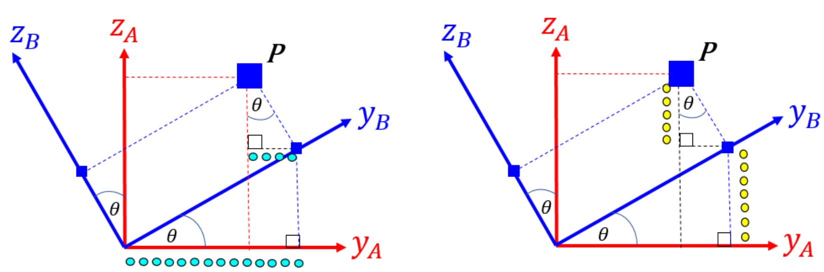
Next we're going to define the rotational velocities for our 4 bodies. And we're going to express these rotational velocities in components of the local centre of mass **BODY fixed frames attached to each body** - this is a natural approach because a body fixed frame means our inertias stay constant relative to that frame. The angular velocities that we'll define are:

- ${}^{B1}\omega_1$: is the angular velocity of LINK 1, expressed in components of the {B1}-Frame
- ${}^{B2}\omega_2$: is the angular velocity of LINK 2, expressed in components of the {B2}-Frame
- ${}^{B3}\omega_3$: is the angular velocity of LINK 3, expressed in components of the {B3}-Frame
- ${}^{B4}\omega_4$: is the angular velocity of TURNTABLE, expressed in components of the {B4}-Frame

Note also, we can use the following transformation matrix, to convert an {A}-Frame vector into it's {B}-Frame components - we'll use this to convert our TURNTABLE angular velocity into components of the LINK body fixed frames. WHY? - the total angular velocity of an individual arm link is a vector sum which will include the contribution of the turntable angular velocity. So for for LINKS {1,2,3}, we'll specify the total angular velocity as:

- ${}^{\mathrm{Bi}}\omega_i = {}^{\mathrm{Bi}}R_A . {}^{A}\omega_A + {}^{\mathrm{Bi}}\omega_{i|A}$

where: ${}^{\mathrm{Bi}}\omega_{i|A}$ is the angular velocity of LINK-i releative to the {A}-frame.



$$\begin{bmatrix} y_P \\ z_P \end{bmatrix}_A = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} * \begin{bmatrix} y_P \\ z_P \end{bmatrix}_B$$

$$\begin{bmatrix} y_P \\ z_P \end{bmatrix}_B = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} * \begin{bmatrix} y_P \\ z_P \end{bmatrix}_A$$

NOTE: in the code below, we've created "function handle" variables to parameterise the rotation matrix ${}^{B}R_{A}$ *.*

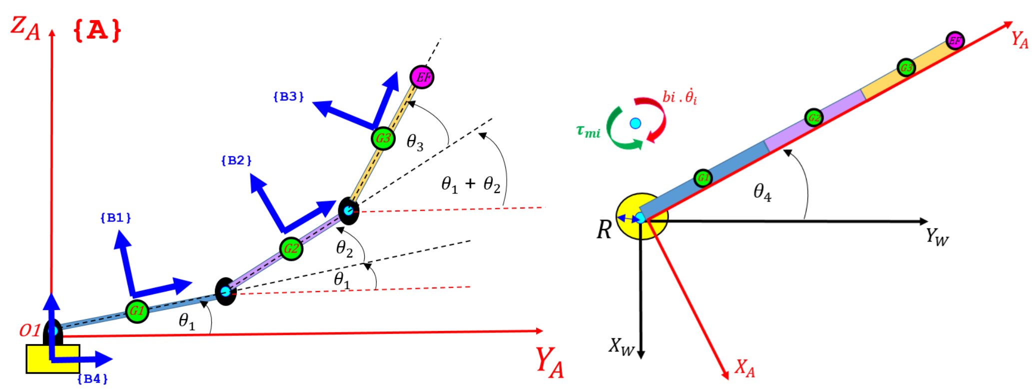
```
bRa = @(th) [       1,          0,         0;
                    0,     cos(th),   sin(th);
                    0,    -sin(th),   cos(th); ];
```

OK, so let's do it. First let's define the angular velocity of the turntable (ie Link #4) ... which is the same thing as our {A}-frame angular velocity:

```
BF_rot_vel_B4 = [0;0; diff(theta4(t))];
AF_w          = [0;0; diff(theta4(t))];
```

Next, let's define the angular velocity for LINKS {1,2,3}. Note that in addition to the local link rotation angles, we also have the turntable rotation angle. So our link angular velocities are:

- ${}^{\mathrm{Bi}}\omega_i = {}^{\mathrm{Bi}}R_A . {}^{A}\omega_A + {}^{\mathrm{Bi}}\omega_{i|A}$




```
th1           = theta1(t)                         ;
th12          = theta1(t) + theta2(t)             ;
th123         = theta1(t) + theta2(t) + theta3(t);

BF_rot_vel_B1 = bRa(th1   )*AF_w  +  [diff(th1)    ; 0; 0];
BF_rot_vel_B2 = bRa(th12  )*AF_w  +  [diff(th12)   ; 0; 0];
BF_rot_vel_B3 = bRa(th123)*AF_w   +  [diff(th123)  ; 0; 0];
```

As a concrete example, here's what the angular velocity vector ${}^{B2}\omega_2 = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$, looks like, for Link 2 and expressed in components of the body fixed {B2}-frame:

```
BF_rot_vel_B2
```

```
BF_rot_vel_B2 =
```

$$\begin{pmatrix} \frac{\partial}{\partial t}\,\theta_1(t) + \frac{\partial}{\partial t}\,\theta_2(t) \\ \sin\big(\theta_1(t) + \theta_2(t)\big)\,\frac{\partial}{\partial t}\,\theta_4(t) \\ \cos\big(\theta_1(t) + \theta_2(t)\big)\,\frac{\partial}{\partial t}\,\theta_4(t) \end{pmatrix}$$

STEP_2: Apply the governing physics - PART 3 of 5

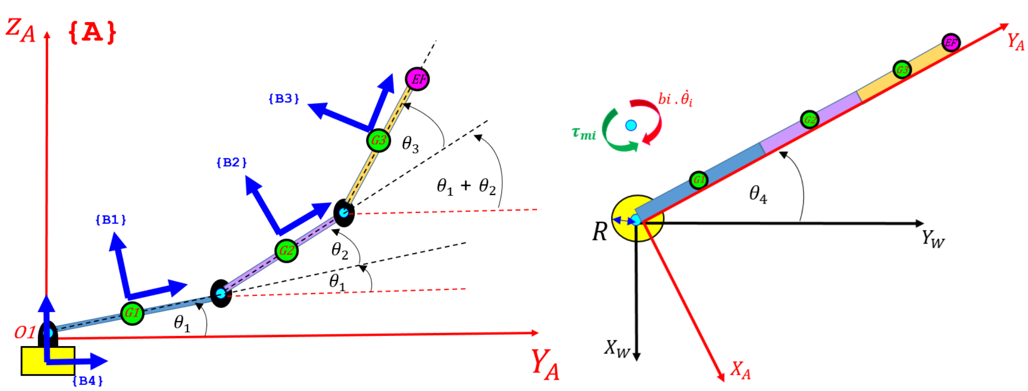
Define the system KINETIC energy:

Next we'll use our previously derived expressions for body velocities and use them to define the "system" Kinetic Energy for our machine. Because we've attached our BODY frames to the **centre of masses** of every link, the kinetic energy for the i^{th} link can be stated as:

$$\mathrm{KE}_i = \frac{1}{2} v_{\mathrm{cm}_i}^T . m_i . v_{\mathrm{cm}_i} \quad + \quad \frac{1}{2} . \omega_{B_i}^T . \left[I_{\mathrm{cm}_i}\right] . \omega_{B_i}$$

The total system kinetic energy is then just the sum of the "*N*" individual link kinetic energies, ie:

$$\mathrm{KE}_{\mathrm{system}} = \sum_{i=1}^{N} \mathrm{KE}_i$$



Our TRANSLATIONAL kinetic energy is:

```
KE_trans = 0.5*m1_s * (WF_tran_vel_G1.') *  WF_tran_vel_G1   + ...
           0.5*m2_s * (WF_tran_vel_G2.') *  WF_tran_vel_G2   + ...
           0.5*m3_s * (WF_tran_vel_G3.') *  WF_tran_vel_G3       ;
```

Our ROTATIONAL kinetic energy is:

```
KE_rot   = 0.5 * (BF_rot_vel_B1.') * I1_s *  BF_rot_vel_B1  + ...
           0.5 * (BF_rot_vel_B2.') * I2_s *  BF_rot_vel_B2  + ...
           0.5 * (BF_rot_vel_B3.') * I3_s *  BF_rot_vel_B3  + ...
           0.5 * (BF_rot_vel_B4.') * I4_s *  BF_rot_vel_B4;
```

So our total system kinetic energy is:

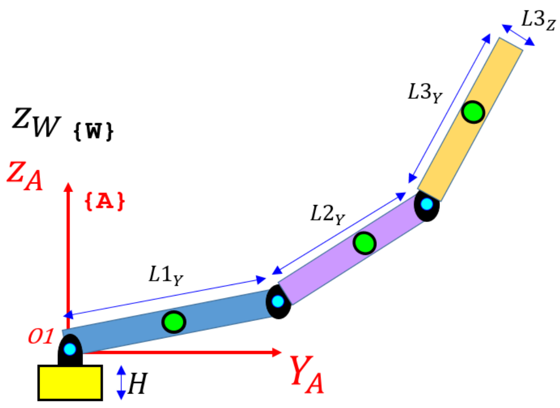
```
KE = KE_trans + KE_rot;
```

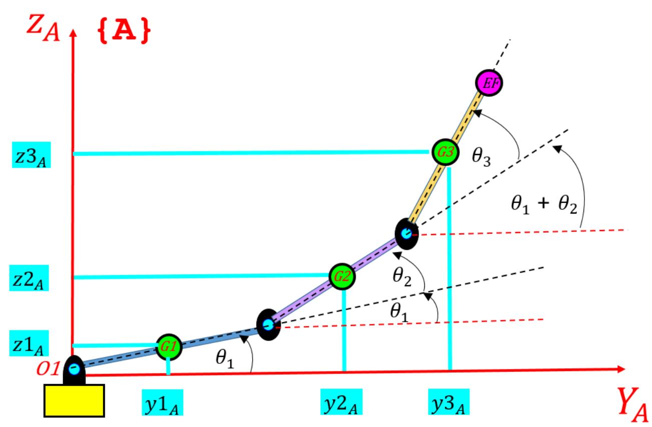
STEP_2: Apply the governing physics - PART 4 of 5

Define the system POTENTIAL energy:

Similarly we can define our "system" Potential Energy. The machine operates within the presence of a constant gravitational field and there is the capacity to do work based on the height above ground of each LINK, ie:

- $PE_{\text{system}} = \sum_{i=1}^{N} PE_i$ where: $PE_i = m_i \,.\, g.\, h_i$





Our system Potential energy is:

```
PE = g_s*( m1_s*WF_pos_G1(3)  + m2_s*WF_pos_G2(3)  + m3_s*WF_pos_G3(3)  );
```

STEP_2: Apply the governing physics - PART 5 of 5

Define the system Lagrangian:

Next define our system Lagrangian:

```
L_ORIGINAL = KE - PE;
```

IFFF you really wanted to see what the terms inside the Lagrangian looked like (are you sure?) ... then we could echo them:

```
the_list_of_terms_making_up_L = children(L_ORIGINAL);
the_list_of_terms_making_up_L(:)
```

```
ans =
```

$$\frac{m_{3,s}\left(\sigma_{11}+\sigma_2+\frac{\mathrm{L3Y}_s\cos(\sigma_{15})\,\sigma_{12}}{2}\right)^2}{2}$$

$$\frac{m_{2,s}\left(\frac{\sigma_{11}}{2}+\sigma_2\right)^2}{2}$$

$$\frac{m_{2,s}\left(\cos(\theta_4(t))\left(\frac{\sigma_{14}}{2}+\sigma_{13}\right)+\sin(\theta_4(t))\,\sigma_{10}\frac{\partial}{\partial t}\,\theta_4(t)\right)^2}{2}$$

$$\frac{m_{2,s}\left(\sin(\theta_4(t))\left(\frac{\sigma_{14}}{2}+\sigma_{13}\right)-\cos(\theta_4(t))\,\sigma_{10}\frac{\partial}{\partial t}\,\theta_4(t)\right)^2}{2}$$

$$\frac{m_{3,s}\left(\cos(\theta_4(t))\,\sigma_8+\sin(\theta_4(t))\,\sigma_9\frac{\partial}{\partial t}\,\theta_4(t)\right)^2}{2}$$

$$\frac{m_{3,s}\left(\sin(\theta_4(t))\,\sigma_8-\cos(\theta_4(t))\,\sigma_9\frac{\partial}{\partial t}\,\theta_4(t)\right)^2}{2}$$

$$\frac{m_{1,s}\left(\frac{\mathrm{L1Y}_s\cos(\theta_1(t))\cos(\theta_4(t))\frac{\partial}{\partial t}\,\theta_4(t)}{2}-\frac{\mathrm{L1Y}_s\sin(\theta_1(t))\sin(\theta_4(t))\frac{\partial}{\partial t}\,\theta_1(t)}{2}\right)^2}{2}$$

STEP_3: Apply Lagrange's equation - PART 1 of 3

To derive the equations of motion for our machine, "all" we need to do is a series of derivative calculations according to Lagrange's equation:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Q_k$$

Where:

- *L = KE - PE*, is our system lagrangian
- *q* = $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ are the generalised co-ordinates
- Q_k are the generalised forces (Torques) for our system

First we'll create some variables that define our generalised co-ordinates. I'm going to use 2 sets of these generalised co-ordinates:

- the **ACTUAL** set of symbols are our "proper" set of symbols
- the **HOLDER** set are for easier expression manipulation (these are purely for convenience)

```
actual_list_SYM_pos = formula( [ theta1,      theta2,      theta3   ,  theta4]  );
holder_list_SYM_pos = [              TH1_s,       TH2_s        TH3_s,     TH4_s];
```

OK: let's create a Lagrangian object using the class <**`bh_lagr4manips_CLS`**>

```
lag_OBJ = bh_lagr4manips_CLS( KE, PE, actual_list_SYM_pos, holder_list_SYM_pos);
```

And let's compute the system's equations of motion:

```
lag_OBJ = lag_OBJ.calc_eom()
```

```
lag_OBJ = 
  bh_lagr4manips_CLS with properties:

                 T_KE: [1×1 sym]
                 V_PE: [1×1 sym]
                    L: [1×1 sym]
              Qk_list: [4×1 sym]
                  EOM: [1×4 bh_eom_CLS]
                N_dof: 4
    actual_list_SYM_pos: [4×1 sym]
    actual_list_SYM_vel: [4×1 sym]
    actual_list_SYM_acc: [4×1 sym]
    holder_list_SYM_pos: [4×1 sym]
    holder_list_SYM_vel: [4×1 sym]
    holder_list_SYM_acc: [4×1 sym]
```

So what do the equations of motion actually look like ? - they are long equations involving many terms

```
lag_OBJ.show_eom()
```

```
#######################################################
### q = theta1(t)
###
```

```
### LHS of EOM is:
###
      (L1Y_s^2*m1_s*diff(theta1(t), t, t))/3 + L1Y_s^2*m2_s*diff(theta1(t), t, t) + L1Y_s^2*m3_s*diff(
###
### RHS of EOM is:
      Q1_s
#######################################################
### q = theta2(t)
###
### LHS of EOM is:
###
      (L2Y_s^2*m2_s*diff(theta1(t), t, t))/3 + (L2Y_s^2*m2_s*diff(theta2(t), t, t))/3 + L2Y_s^2*m3_s*d
###
### RHS of EOM is:
      Q2_s
#######################################################
### q = theta3(t)
###
### LHS of EOM is:
###
      (L3Y_s^2*m3_s*diff(theta1(t), t, t))/3 + (L3Y_s^2*m3_s*diff(theta2(t), t, t))/3 + (L3Y_s^2*m3_s*
###
### RHS of EOM is:
      Q3_s
#######################################################
### q = theta4(t)
###
### LHS of EOM is:
###
      (L1X_s^2*m1_s*diff(theta4(t), t, t))/12 + (L2X_s^2*m2_s*diff(theta4(t), t, t))/12 + (L3X_s^2*m3_
###
### RHS of EOM is:
      Q4_s
```

If you want to look at an individual equation of motion (eg: the LHS of the θ_4 equation), then you could do this:

```
lag_OBJ.get_eom(4, 'actual', 'LHS')
```

```
ans =
```

$$\frac{\mathrm{L1X}_s^2\, m_{1,s}\,\sigma_1}{12} + \frac{\mathrm{L2X}_s^2\, m_{2,s}\,\sigma_1}{12} + \frac{\mathrm{L3X}_s^2\, m_{3,s}\,\sigma_1}{12} + \frac{\mathrm{L1Y}_s^2\, m_{1,s}\,\sigma_1}{6} + \frac{\mathrm{L1Y}_s^2\, m_{2,s}\,\sigma_1}{2} + \frac{\mathrm{L1Y}_s^2\, m_{3,s}\,\sigma_1}{2} + \frac{\mathrm{L2Y}_s^2\, m_{2,s}\,\sigma_1}{6} +$$

where

$$\sigma_1 = \frac{\partial^2}{\partial t^2}\,\theta_4(t)$$

$$\sigma_2 = \sin\left(2\,\theta_1(t)\right)$$

$$\sigma_3 = \cos\left(2\,\theta_1(t)\right)$$

$$\sigma_4 = 2\,\theta_1(t) + 2\,\theta_2(t) + 2\,\theta_3(t)$$

$$\sigma_5 = 2\,\theta_1(t) + 2\,\theta_2(t) + \theta_3(t)$$

$$\sigma_6 = 2\,\theta_1(t) + \theta_2(t) + \theta_3(t)$$

$$\sigma_7 = 2\,\theta_1(t) + 2\,\theta_2(t)$$

$$\sigma_8 = 2\,\theta_1(t) + \theta_2(t)$$

$$\sigma_9 = \theta_2(t) + \theta_3(t)$$

In a moment we'll show how to "collect" the terms in these EOMs and present them in a format that looks like this:

- $M(q,\dot{q}).\ddot{q} \;+\; C(q,\dot{q}).\dot{q} \;+\; K(q).q \;+\; g(q) = Q(\tau,\dot{q})$

STEP_3: Apply Lagrange's equation - PART 2 of 3

Define the Generalised forces:

Next we need to calculate our Generalised forces. Recall the formula for the generalised forces acting on the system:

$$Q_k = \sum_{i=1}^{Nf_{nc}} \left(\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_k} \right) + \sum_{j=1}^{N\tau_{nc}} \left(\vec{\tau}_j \cdot \frac{\partial \vec{\omega}_j}{\partial \dot{q}_k} \right)$$

where:

- Q_k : is the generalised force associated with the k^{th} generalised co-ordinate q_k
- Nf_{nc} : is the number of active NON conservative forces
- $N\tau_{nc}$: is the number of active NON conservative TORQUES
- $\vec{v}_i$: is the velocity vector of the point associated with the applied force.

- $\vec{\omega}_i$: is the angular velocity about the point associated with the applied torque.

Recall that we have already calculated the angular velocties of our LINKS relative to the {W}-frame. And we have expressed these angular velocities in their components of the local BODY fixed frames, ie:

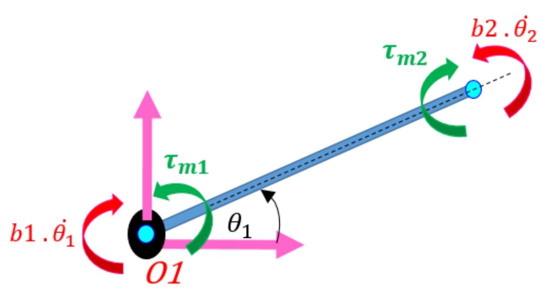
- ${}^{B1}\omega_1 \rightarrow$ **`BF_rot_vel_B1`**
- ${}^{B2}\omega_2 \rightarrow$ **`BF_rot_vel_B2`**
- ${}^{B3}\omega_3 \rightarrow$ **`BF_rot_vel_B3`**
- ${}^{B4}\omega_4 \rightarrow$ **`BF_rot_vel_B4`**

Define some additional angular velocities which we'll use when defining the viscous damping torques:

```
th1dot     = formula( diff(theta1, t)  );
th2dot     = formula( diff(theta2, t)  );
th3dot     = formula( diff(theta3, t)  );
th4dot     = formula( diff(theta4, t)  );
```

For each link, I'm going to define a matrix, whos columns represent the vectors of the NON conservative torques acting on that body, and another matrix whose columns represent the angular velocities associated with these torques.

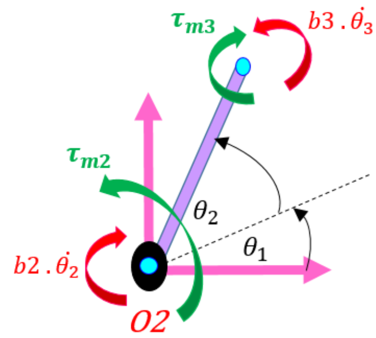
So for **LINK #1**, we have: ${}^{B1}\omega_1 \rightarrow$ **`BF_rot_vel_B1`**




```
the_tau_mat_LINK_1 = [ ...
       (taum1_s),   (-b1_s*th1dot),   (-taum2_s),      (b2_s*th2dot);
               0,                0,            0,                  0;
               0,                0,            0,                  0;    ];

the_w_mat_LINK_1 = [ ...
    BF_rot_vel_B1,    BF_rot_vel_B1,   BF_rot_vel_B1,    BF_rot_vel_B1 ];
```

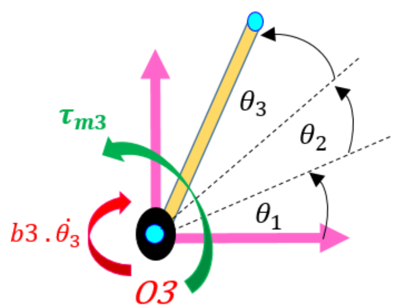
And similarly for **LINK #2**: ${}^{B2}\omega_2 \rightarrow$ **`BF_rot_vel_B2`**




```
the_tau_mat_LINK_2 = [ ...
       (taum2_s),    (-b2_s*th2dot),    (-taum3_s),     (b3_s*th3dot);
               0,                 0,              0,                0;
               0,                 0,              0,                0;   ];

the_w_mat_LINK_2 = [ ...
    BF_rot_vel_B2,     BF_rot_vel_B2,    BF_rot_vel_B2,     BF_rot_vel_B2 ];
```

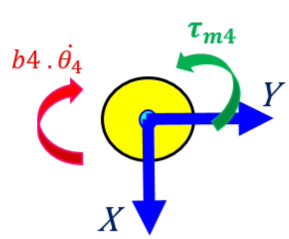
And similarly for **LINK #3**: ${}^{B3}\omega_3 \rightarrow$ **`BF_rot_vel_B3`**

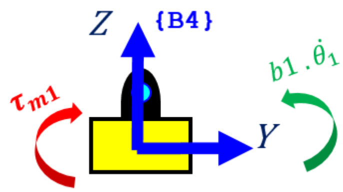



```
the_tau_mat_LINK_3 = [ ...
     (taum3_s),     (-b3_s*th3dot);
             0,                   0;
             0,                   0;   ];

the_w_mat_LINK_3 = [ ...
     BF_rot_vel_B3,         BF_rot_vel_B3 ];
```

And similarly for the turntable (ie: **LINK #4**): ${}^{B4}\omega_4 \rightarrow$ **`BF_rot_vel_B4`**






```
the_tau_mat_LINK_4 = ...
   [            0,                   0,       (b1_s*th1dot - taum1_s);
                0,                   0,                             0;
        (taum4_s),      (-b4_s*th4dot),                             0;    ];

the_w_mat_LINK_4 = [ ...
```

```
    BF_rot_vel_B4,      BF_rot_vel_B4,               BF_rot_vel_B4   ];
```

Next, let's concatenate these into single matrices:

```
the_tau_mat_actual = [the_tau_mat_LINK_1, the_tau_mat_LINK_2, the_tau_mat_LINK_3,  the_tau_mat
the_w_mat_actual   = [  the_w_mat_LINK_1,   the_w_mat_LINK_2,   the_w_mat_LINK_3,    the_w_mat
```

Now let's compute Q_k:

$$Q_k = \sum_{i=1}^{Nf_{nc}} \left(\vec{F}_i \cdot \frac{\partial \vec{v}_i}{\partial \dot{q}_k} \right) + \sum_{j=1}^{N\tau_{nc}} \left(\vec{\tau}_j \cdot \frac{\partial \vec{\omega}_j}{\partial \dot{q}_k} \right)$$

To do this we'll create a generalised force object using the class <**`bh_genF4manips_CLS`**>

```
genF_OBJ = bh_genF4manips_CLS( the_tau_mat_actual, ...
                               the_w_mat_actual, ...
                               actual_list_SYM_pos, ...
                               holder_list_SYM_pos);
```

And now calculate our system's generalised forces:

```
genF_OBJ = genF_OBJ.calc_genF();
```

What do the $Q_k's$ look like?

```
the_Qk_vec = genF_OBJ.get_Qk('all', 'holder')
```

```
the_Qk_vec =
```

$$\begin{pmatrix} \mathrm{taum}_{1,s} - \mathrm{TH}_{1,s,D}\, b_{1,s} \\ \mathrm{taum}_{2,s} - \mathrm{TH}_{2,s,D}\, b_{2,s} \\ \mathrm{taum}_{3,s} - \mathrm{TH}_{3,s,D}\, b_{3,s} \\ \mathrm{taum}_{4,s} - \mathrm{TH}_{4,s,D}\, b_{4,s} \end{pmatrix}$$

STEP_3: Apply Lagrange's equation - PART 3 of 3

Absorb the Generalised Forces: We can now insert these Generalised forces into the "Lagrangian" object that we created earlier, and then recalculate the equations of motion:

```
lag_OBJ = lag_OBJ.calc_eom(genF_OBJ);
```

Now echo our equations:

```
lag_OBJ.show_eom( )
```

```
#######################################################
### q = theta1(t)
###
### LHS of EOM is:
###
        (L1Y_s^2*m1_s*diff(theta1(t), t, t))/3 + L1Y_s^2*m2_s*diff(theta1(t), t, t) + L1Y_s^2*m3_s*diff(
###
```

```
### RHS of EOM is:
        tauml_s - b1_s*diff(theta1(t), t)
#######################################################
### q = theta2(t)
###
### LHS of EOM is:
###
        (L2Y_s^2*m2_s*diff(theta1(t), t, t))/3 + (L2Y_s^2*m2_s*diff(theta2(t), t, t))/3 + L2Y_s^2*m3_s*d
###
### RHS of EOM is:
        taum2_s - b2_s*diff(theta2(t), t)
#######################################################
### q = theta3(t)
###
### LHS of EOM is:
###
        (L3Y_s^2*m3_s*diff(theta1(t), t, t))/3 + (L3Y_s^2*m3_s*diff(theta2(t), t, t))/3 + (L3Y_s^2*m3_s*
###
### RHS of EOM is:
        taum3_s - b3_s*diff(theta3(t), t)
#######################################################
### q = theta4(t)
###
### LHS of EOM is:
###
        (L1X_s^2*m1_s*diff(theta4(t), t, t))/12 + (L2X_s^2*m2_s*diff(theta4(t), t, t))/12 + (L3X_s^2*m3_
###
### RHS of EOM is:
        taum4_s - b4_s*diff(theta4(t), t)
```

STEP_4: Isolate the term of interest M,C,K,G

We can express our system equations of motion in the following form:

$$M(q,\dot{q}).\ddot{q} \ + C(q,\dot{q}).\dot{q} \ + K(q).q \ + g(q) = Q(\tau,\dot{q})$$

```
lag_OBJ = lag_OBJ.create_MCKGQ();
```

Retrieve the MCKGQ struct:

```
res_T = lag_OBJ.get_MCKGQ()
```

```
res_T = 
  bh_MCKGQ_CLS with properties:

          M: [4×4 sym]
          C: [4×4 sym]
          K: [4×4 sym]
          G: [4×1 sym]
          Q: [4×1 sym]
    ACC_col: [4×1 sym]
    VEL_col: [4×1 sym]
    POS_col: [4×1 sym]
   acc_eoms: [4×1 sym]
```

```
fh_BOUNDARY = @(txt)fprintf('\n %s \n Here is the %s matrix: \n',repmat('#',1,75),txt);
```

Here's **M**:

```
fh_BOUNDARY('M');  res_T.M
```

```
################################################################################
Here is the M matrix:
ans =
```

$$\left(\begin{array}{c} \frac{\mathrm{L1Y}_s^2\, m_{1,s}}{3} + \mathrm{L1Y}_s^2\, m_{2,s} + \mathrm{L1Y}_s^2\, m_{3,s} + \sigma_{11} + \mathrm{L2Y}_s^2\, m_{3,s} + \sigma_{10} + \frac{\mathrm{L1Z}_s^2\, m_{1,s}}{12} + \sigma_9 + \sigma_8 + \sigma_{14} + 2\,\mathrm{L1Y}_s\,\mathrm{L2Y}_s\, m \\ \sigma_6 \\ \sigma_1 \\ 0 \end{array}\right.$$

where

$$\sigma_1 = \frac{m_{3,s}\,\left(4\,\mathrm{L3Y}_s^2 + \mathrm{L3Z}_s^2 + 6\,\mathrm{L1Y}_s\,\mathrm{L3Y}_s\cos\left(\mathrm{TH}_{2,s} + \mathrm{TH}_{3,s}\right) + 6\,\mathrm{L2Y}_s\,\mathrm{L3Y}_s\cos\left(\mathrm{TH}_{3,s}\right)\right)}{12}$$

$$\sigma_2 = \frac{m_{3,s}\,\left(4\,\mathrm{L3Y}_s^2 + 6\,\mathrm{L2Y}_s\cos\left(\mathrm{TH}_{3,s}\right)\mathrm{L3Y}_s + \mathrm{L3Z}_s^2\right)}{12}$$

$$\sigma_3 = \cos\left(2\,\mathrm{TH}_{1,s} + 2\,\mathrm{TH}_{2,s}\right)$$

$$\sigma_4 = \cos\left(2\,\mathrm{TH}_{1,s} + 2\,\mathrm{TH}_{2,s} + 2\,\mathrm{TH}_{3,s}\right)$$

$$\sigma_5 = \cos\left(2\,\mathrm{TH}_{1,s} + \mathrm{TH}_{2,s}\right)$$

$$\sigma_6 = \sigma_{11} + \mathrm{L2Y}_s^2\, m_{3,s} + \sigma_{10} + \sigma_9 + \sigma_8 + \frac{\sigma_{14}}{2} + \sigma_{13} + \sigma_7 + \frac{\sigma_{12}}{2}$$

$$\sigma_7 = \mathrm{L2Y}_s\,\mathrm{L3Y}_s\, m_{3,s}\cos\left(\mathrm{TH}_{3,s}\right)$$

$$\sigma_8 = \frac{\mathrm{L3Z}_s^2\, m_{3,s}}{12}$$

Here's **C**:

```
fh_BOUNDARY('C');  res_T.C
```

```
################################################################################
Here is the C matrix:
ans =
```

$$\left(\begin{array}{c} -\sigma_3 - \sigma_2 - \sigma_4 - 2\,\mathrm{L1Y}_s\,\mathrm{L2Y}_s\,\mathrm{TH}_{2,s,D}\,m_{3,s}\sin\left(\mathrm{TH}_{2,s}\right) - \sigma_1 \\ \frac{\mathrm{L1Y}_s\,\mathrm{L3Y}_s\,\mathrm{TH}_{1,s,D}\,m_{3,s}\sin\left(\mathrm{TH}_{2,s}+\mathrm{TH}_{3,s}\right)}{2} + \frac{\mathrm{L1Y}_s\,\mathrm{L2Y}_s\,\mathrm{TH}_{1,s,D}\,m_{2,s}\sin\left(\mathrm{TH}_{2,s}\right)}{2} + \mathrm{L1Y}_s\,\mathrm{L2Y}_s\,\mathrm{TH}_{1,s,D}\,m_{3,s}\sin(\mathrm{T} \\ \frac{\mathrm{L3Y}_s\,m_{3,s}\,\left(\mathrm{L1Y}_s\,\mathrm{TH}_{1,s,D}\sin\left(\mathrm{TH}_{2,s}+\mathrm{TH}_{3,s}\right) + \mathrm{L2Y}_s\,\mathrm{TH}_{1,s,D}\sin\left(\mathrm{TH}_{3,s}\right) + 2\,\mathrm{L2Y}_s\,\mathrm{TH}_{2,s,D}\sin\left(\mathrm{TH}_{3,s}\right)\right.}{2} \\ -\frac{\sigma_5}{12} \end{array} \right.$$

where

$$\sigma_1 = \mathrm{L2Y}_s\,\mathrm{L3Y}_s\,\mathrm{TH}_{3,s,D}\,m_{3,s}\sin\left(\mathrm{TH}_{3,s}\right)$$

$$\sigma_2 = \mathrm{L1Y}_s\,\mathrm{L3Y}_s\,\mathrm{TH}_{3,s,D}\,m_{3,s}\sin\left(\mathrm{TH}_{2,s}+\mathrm{TH}_{3,s}\right)$$

$$\sigma_3 = \mathrm{L1Y}_s\,\mathrm{L3Y}_s\,\mathrm{TH}_{2,s,D}\,m_{3,s}\sin\left(\mathrm{TH}_{2,s}+\mathrm{TH}_{3,s}\right)$$

$$\sigma_4 = \mathrm{L1Y}_s\,\mathrm{L2Y}_s\,\mathrm{TH}_{2,s,D}\,m_{2,s}\sin\left(\mathrm{TH}_{2,s}\right)$$

$$\sigma_5 = \mathrm{TH}_{4,s,D}\,\left(\sigma_{12} - \sigma_{13} + 4\,\mathrm{L1Y}_s^2\,m_{1,s}\sin\left(2\,\mathrm{TH}_{1,s}\right) + 12\,\mathrm{L1Y}_s^2\,m_{2,s}\sin\left(2\,\mathrm{TH}_{1,s}\right) + 12\,\mathrm{L1Y}_s^2\,m_{3,s}\sin(2\,\mathrm{TH}_{1,}\right.$$

$$\sigma_6 = \mathrm{TH}_{4,s,D}\,\left(\sigma_{12} - \sigma_{13} + \sigma_{10} + \sigma_9 - \sigma_{14} + 6\,\mathrm{L1Y}_s\,\mathrm{L2Y}_s\,m_{2,s}\sin\left(\mathrm{TH}_{2,s}\right) + 12\,\mathrm{L1Y}_s\,\mathrm{L2Y}_s\,m_{3,s}\sin\left(\mathrm{TH}_{2,s}\right) + 6\,\mathrm{L}\right.$$

$$\sigma_7 = \mathrm{TH}_{4,s,D}\,m_{3,s}\,\left(4\,\mathrm{L3Y}_s^2\,\sigma_{17} - \mathrm{L3Z}_s^2\,\sigma_{17} + 6\,\mathrm{L1Y}_s\,\mathrm{L3Y}_s\,\sigma_{15} + 6\,\mathrm{L1Y}_s\,\mathrm{L3Y}_s\sin\left(\mathrm{TH}_{2,s}+\mathrm{TH}_{3,s}\right) + 6\,\mathrm{L2Y}_s\,\mathrm{L3Y}\right.$$

$$\sigma_8 = \sin\left(2\,\mathrm{TH}_{1,s} + \mathrm{TH}_{2,s}\right)$$

$$\sigma_9 = 12\,\mathrm{L2Y}_s^2\,m_{3,s}\,\sigma_{18}$$

Here's **K**:

```
fh_BOUNDARY('K');  res_T.K
```

```
 ##########################################################################
 Here is the K matrix:
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Here's **G**:

```
fh_BOUNDARY('G');  res_T.G
```

```
 ###########################################################################
 Here is the G matrix:
ans =
```

$$
\left(\begin{array}{c}
\dfrac{g_s\,\left(\mathrm{L2Y}_s\,m_{2,s}\cos\left(\mathrm{TH}_{1,s}+\mathrm{TH}_{2,s}\right)+2\,\mathrm{L2Y}_s\,m_{3,s}\cos\left(\mathrm{TH}_{1,s}+\mathrm{TH}_{2,s}\right)+\mathrm{L1Y}_s\,m_{1,s}\cos\left(\mathrm{TH}_{1,s}\right)+2\,\mathrm{L1Y}_s\,m_{2,s}\cos\right.}{2} \\
\sigma_1+\dfrac{\mathrm{L2Y}_s\,g_s\,m_{2,s}\cos\left(\mathrm{TH}_{1,s}+\mathrm{TH}_{2,s}\right)}{2}+\mathrm{L2Y}_s\,g_s\,m_{3,s}\cos\left(\mathrm{TH}_{1,s}+\right. \\
\sigma_1 \\
0
\end{array}\right.
$$

where

$$\sigma_1=\frac{\mathrm{L3Y}_s\,g_s\,m_{3,s}\,\sigma_2}{2}$$

$$\sigma_2=\cos\left(\mathrm{TH}_{1,s}+\mathrm{TH}_{2,s}+\mathrm{TH}_{3,s}\right)$$

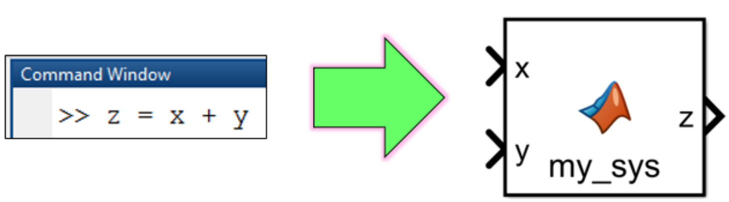
Here's **Q**:

```
fh_BOUNDARY('Q');  res_T.Q
```

```
 ###########################################################################
 Here is the Q matrix:
ans =
```

$$
\left(\begin{array}{c}
\mathrm{taum}_{1,s}-\mathrm{TH}_{1,s,D}\,b_{1,s} \\
\mathrm{taum}_{2,s}-\mathrm{TH}_{2,s,D}\,b_{2,s} \\
\mathrm{taum}_{3,s}-\mathrm{TH}_{3,s,D}\,b_{3,s} \\
\mathrm{taum}_{4,s}-\mathrm{TH}_{4,s,D}\,b_{4,s}
\end{array}\right)
$$

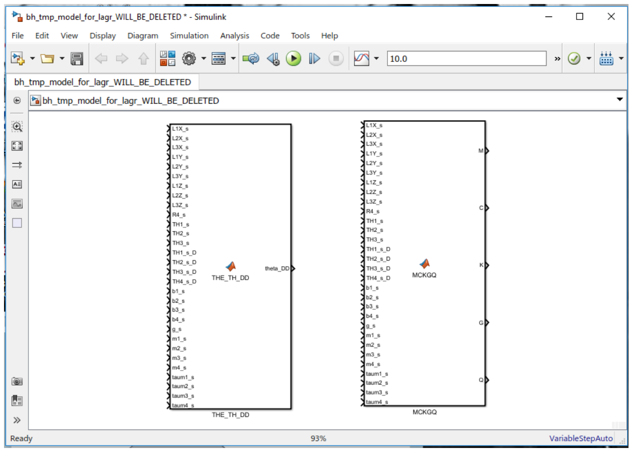
STEP_5: Convert symbolic expression into a block diagram model



To use/solve these derived equations of motion we'll create a MATLAB Function block that can be used inside Simulink:

```
lag_OBJ.create_MLF_blocks()
```

```
Warning: File 'bh_tmp_model_for_lagr_WILL_BE_DELETED' not found.
Warning: The model name 'bh_tmp_model_for_lagr_WILL_BE_DELETED' is shadowing another name in
the MATLAB workspace or path. Type "which -all bh_tmp_model_for_lagr_WILL_BE_DELETED" at the
command line to find the other uses of this name. You should change the name of the model to
avoid problems.
```



The "MCKGQ" block can then be pasted into a Simulink model for you to use in a simulation of the robot.

STEP_6: Simulate the model of the dynamic system

Why not use the DEMO_SELECTOR app to look at 2 such examples:

- In Example 1 we validate the derived model against a Simscape Multibody model
- In Example 2 we have a "complete" control system that makes our model write *"Hello"*