# Equations Of Motion of Krang

Munzir Zafar

May 7, 2015

In this report we attempt to find the dynamic model of Golem Krang. It is a serial robot with a tree-structure (due to two arms branching out) that is balanced on two wheels as an inverted pendulum. So it is a humanoid insofar as it is bimanual and dynamically stable (i.e. an inverted pendulum). But it has two wheels instead of two legs. Figure 1 shows the frames of references we will be using to determine the transforms and the coordinates on the robot. Also $\dot{x}$ is the heading speed of the robot and $\dot{\psi}$ is the spin speed of the robot. We denote these frames using symbol $R_i$ where $i \in \mathbb{F} = \{0, 1, 2, 3, 4l, 5l, 6l, 7l, 8l, 9l, 10l, 4r, 5r, 6r, 7r, 8r, 9r, 10r\}$. $R_0$ is the world frame fixed in the middle of the two wheels. $R_1, R_2, R_3$ are fixed on the base, spine and torso with their rotations represented by $q_{imu}$, $q_w$ and $q_{torso}$ respectively. Frames $R_{4l}, ...R_{10l}$ are frames fixed on the links left 7-DOF arm with their motion represented by $q_{1l}, ...q_{7l}$. Similarly, frames $R_{4r}, ...R_{10r}$ are frames fixed on the links right 7-DOF arm with their motion represented by $q_{1r}, ...q_{7r}$. All equations in the following text that do not show $r$ or $l$ in the subscript where they are supposed to, will mean that the respective equations are valid for both subscripts.

## 1 Generalized Coordinates

We select twenty-one generalized co-ordinates: $\{q\} = \{\ X_0,\ Y_0,\ \theta_L,\ \theta_R,\ q_{imu},\ q_w,\ q_{torso},\ q_{1l},\ ...,\ q_{7l},\ q_{1r},\ ...,\ q_{7r}\}$. Where:

$X_0, Y_0$ are the position coordintes of $O$ wrt a Newtonian reference frame $xyz$ fixed on ground

$\theta_L, \theta_R$ are the rotation angles of the left and right wheels respectively

Given this Newtonian frame of reference we can define $\psi$ as the heading direction ($x_0$) measured as an angle from the $x$-axis of the Newtonian frame.

## 2 Constraint Equations

Let $\bar{i}_0$, $\bar{j}_0$ and $\bar{k}_0$ be the unit vectors in frame $x_0y_0z_0$ and $\bar{I}$, $\bar{J}$ and $\bar{K}$ be the unit vectors in frame $xyz$.

Under the assumption of no slipping/skidding, we have two constraint equations:

$$\bar{v}_L = R\dot{\theta}_L\bar{i}_0 \tag{1}$$

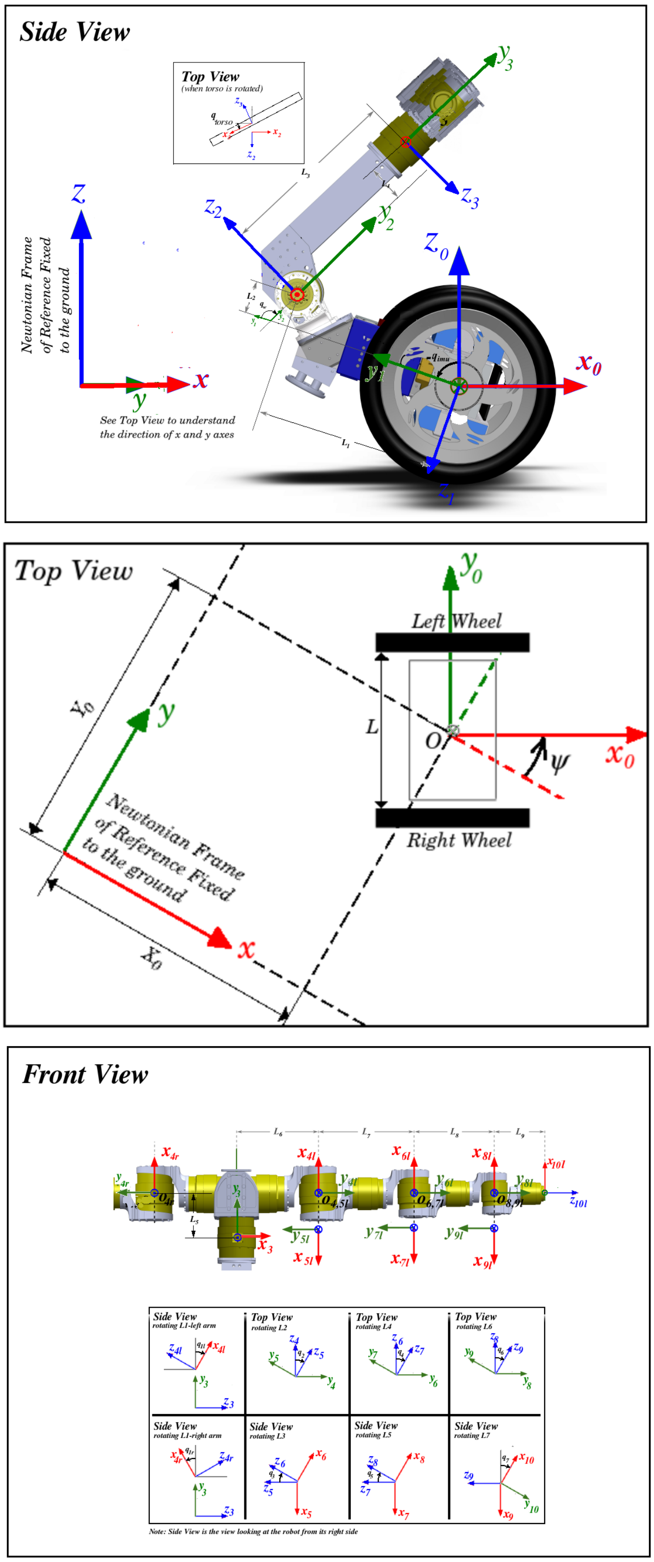


Figure 1: Frames of references on the robot

$$\bar{v}_R = R\dot{\theta}_R\bar{i}_0 \tag{2}$$

Since

$$\begin{aligned}
\bar{v}_L &= \bar{v}_0 + \bar{\omega}_0 \times \bar{r}_{L/O} \\
&= \dot{X}_0\bar{I} + \dot{Y}_0\bar{J} + \dot{\psi}\bar{K} \times (\frac{L}{2}\bar{j}_0) \\
&= \dot{X}_0(cos\psi\bar{i}_0 - sin\psi\bar{j}_0) + \dot{Y}_0(sin\psi\bar{i}_0 + cos\psi\bar{k}_0) - \frac{L}{2}\dot{\psi}\bar{i}_0 \\
&= (\dot{X}_0cos\psi + \dot{Y}_0sin\psi - \frac{L}{2}\dot{\psi})\bar{i}_0 - (\dot{X}_0sin\psi - \dot{Y}_0cos\psi)\bar{j}_0
\end{aligned}$$

And similarly,

$$\bar{v}_R = (\dot{X}_0cos\psi + \dot{Y}_0sin\psi + \frac{L}{2}\dot{\psi})\bar{i}_0 - (\dot{X}_0sin\psi - \dot{Y}_0cos\psi)\bar{j}_0$$

Comparing the coefficients of $\bar{i}_0$ in eqs. 1-2 gives:

$$\dot{X}_0cos\psi + \dot{Y}_0sin\psi - \frac{L}{2}\dot{\psi} = R\dot{\theta}_L \tag{3}$$

$$\dot{X}_0cos\psi + \dot{Y}_0sin\psi + \frac{L}{2}\dot{\psi} = R\dot{\theta}_R \tag{4}$$

Subtraction and addition of the two equations gives us:

$$\dot{\psi} = \frac{R}{L}(\dot{\theta}_R - \dot{\theta}_L) \tag{5}$$

$$\dot{X}_0cos\psi + \dot{Y}_0sin\psi = \frac{R}{2}(\dot{\theta}_L + \dot{\theta}_R) \tag{6}$$

Eq. 5 can be integrated to give $\psi = \frac{R}{L}(\theta_R - \theta_L)$ which can be substituted in eq. 6 to give our first constraint equation relating the first four generalized co-ordinates.

Comparing the coefficients of $\bar{j}_0$ in eqs. 1-2 gives us our second constraint equation:

$$\dot{X}_0sin\psi - \dot{Y}_0cos\psi = 0 \tag{7}$$

Eqs. 6-7 give the two constraint equations relating our generalized velocities as a result of the nonholonomic constrainsts. Twenty-one generalized coordinates with two constraint equations leads to nineteen degrees of freedom.

## 3 Defining Generalized Velocities

It is easier to derive the dynamic model of the system in terms of the generalized velocities: $\{\dot{q}\} = \{\, \dot{x}, \dot{\psi}, \dot{q}_{imu}, \dot{q}_w, \dot{q}_{torso}, \dot{q}_{1l}, ..., \dot{q}_{7l}, \dot{q}_{1r}, ..., \dot{q}_{7r}\}$. These nineteen velocities can take arbitrary values all of whom will be kinematically admissible. In other words, they represent our nineteen degrees of freedom. Here $\dot{x}$ should be refered to as a quasi-velocity as this velocity has meaning only as a velocity but its corresponding position variable $x$ does not give any physical meaning. Although, the infinitesimal change of position $\delta x = \dot{x}dt$ (sometimes refered to as virtual displacement) has physical meaning.

The older generalized velocities can be calculated from the new ones as follows:

$$\dot{X}_0 = \dot{x} cos\psi \tag{8}$$

$$\dot{Y}_0 = \dot{x} sin\psi \tag{9}$$

$$\dot{\theta}_L = \frac{1}{R}\dot{x} - \frac{L}{2R}\dot{\psi} \tag{10}$$

$$\dot{\theta}_R = \frac{1}{R}\dot{x} + \frac{L}{2R}\dot{\psi} \tag{11}$$

Where the first two relationships are derived by comparing the coefficients in $\dot{X}_0\bar{I} + \dot{Y}_0\bar{J} = \bar{v}_0 = \dot{x}\bar{i}_0 = \dot{x}(cos\psi\bar{I} + sin\psi\bar{J})$. And the next two relationships are derived by substituting the the first two relationships in eqs. 3-4. When these relationships (8-11) are substituted in our constraint equations 6-7, both sides of the equations vanish. Indicating that these twenty degrees of freedom are not bound by any constraint. We will now derive nineteen dynamic equations in terms of our new generalized velocities. Those equations in conjunction with these relationships can solve for all twenty-two generalized coordinates.

We will be using the Kane's formulation. This is done because the presence of quasi-velocity prohibits us from using Lagrange method for dynamic modeling.

# 4 Introduction to Kane's Formulation

$$\sum_k \left[ m_k \bar{a}_{Gk} \cdot (\bar{v}_{Gk})_j + \left( \frac{d\bar{H}_{Gk}}{dt} \right) \cdot (\bar{\omega}_k)_j \right] = \sum_n \bar{F}_n \cdot (\bar{v}_n)_j + \sum_n \bar{M}_m \cdot (\bar{\omega}_m)_j \quad j = 1...K \tag{12}$$

where

- $j$ is the unique number identifying each generalized co-ordinate in the system
- $k$ is the unique number identifying each rigid body in the system
- $n$ is the unique number identifying each external force acting on the system
- $m$ is the unique number identifying each external torque acting on the system
- $m_k$ is the mass of the $k$th body
- $\bar{a}_{Gk}$ is the acceleration of the center of mass of $k$th body
- $\bar{v}_{Gk}$ is the velocity of the center of mass of the $k$th body
- $\bar{H}_{Gk}$ is the angular momentum of body $k$ about its center of mass
- $\bar{\omega}_k$ is the angular velocity of the body $k$
- $F_n$ is the $n$th external force
- $M_m$ is the $m$th external moment
- $\bar{v}_n$ is the velocity of the point at which external Force $F_n$ is acting

$\bar{\omega}_m$ is the angular velocity of the body on which torque is acting relative to the actuator applying the torque

$()_j = \frac{\partial()}{\partial \dot{q}_j}$ the partial derivative of the quantity in brackets () with respect to the generalized velocity $\dot{q}_j$

# 5 Kane's Left Hand Side

The left hand side of the Kane's equation containes a sum whose range is equal to the number of bodies in the system. We have nineteen bodies: Left-wheel ($L$), right wheel ($R$) and the seventeen links in the tree structure of robot each having a frame $R_i$ fixed on it where $i \in \{1, 2, 3, 4l, ...10l, 4r, ...10r\}$. Each term in the sum consists of the acceleration ($\bar{a}_{Gk}$), velocity ($\bar{v}_{Gk}$), angular momentum ($\bar{H}_{Gk}$) of the center of mass and the body's angular velocity ($\bar{\omega}_k$). And then some partial derivatives wrt to the generalized velocities ($(\bar{\omega}_k)_j = \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j}$ and $(\bar{v}_{Gk})_j = \frac{\partial \bar{v}_{Gk}}{\partial \dot{q}_j}$). We will have nineteen equations corresponding to each generalized coordinate $\{\dot{q}\} = \{\ \dot{x},\ \dot{\phi},\ \dot{\psi},\ \dot{q}_{imu},\ \dot{q}_w,\ \dot{q}_{torso},\ \dot{q}_{1l},\ ...,\ \dot{q}_{7l},\ \dot{q}_{1r},\ ...,\ \dot{q}_{7r}\}$.

## 5.1 Frame $x_0y_0z_0$

The velocities and accelerations of the frame $x_0y_0z_0$ are needed to evaluate the afore-mentioned quantities related to each body. These velocities are as follows:

$$\bar{\omega}_0 = \dot{\psi}\bar{k}_0 \tag{13}$$

$$\bar{v}_0 = \dot{x}\bar{i}_0 \tag{14}$$

$$\bar{\alpha}_0 = \ddot{\psi}\bar{k}_0 \tag{15}$$

$$\begin{aligned} \bar{a}_0 &= \ddot{x}\bar{i}_0 + \dot{x}\left(\dot{\psi}\bar{k}_0 \times \bar{i}_0\right) \\ &= \ddot{x}\bar{i}_0 + \dot{x}\dot{\psi}\bar{j}_0 \end{aligned} \tag{16}$$

## 5.2 Wheels

### 5.2.1 Left Wheel

This evaluation takes place in the $x_Ly_Lz_L$ frame fixed to the left wheel such that it is parallel to frame $x_0y_0z_0$ when $\theta_L = 0$. So $\bar{i}_0 = cos\theta_L\bar{i}_L + sin\theta_L\bar{k}_L$, $\bar{j}_0 = \bar{j}_L$ and $\bar{k}_0 = -sin\theta_L\bar{i}_L + cos\theta_L\bar{k}_L$. Angular velocity:

$$\begin{aligned} \bar{\omega}_L &= \dot{\psi}\bar{k}_0 + \dot{\theta}_L\bar{j}_0 \\ &= \dot{\psi}\bar{k}_0 + \left(\frac{1}{R}\dot{x} - \frac{L}{2R}\dot{\psi}\right)\bar{j}_0 \\ &= -\dot{\psi}sin\theta_L\bar{i}_L + \left(\frac{1}{R}\dot{x} - \frac{L}{2R}\dot{\psi}\right)\bar{j}_L + \dot{\psi}cos\theta_L\bar{k}_L \end{aligned} \tag{17}$$

The terms that follow are also similarly to be expressed in frame $x_Ly_Lz_L$ but that step is skipped for brevity. Velocity:

$$\bar{v}_{GL} = \bar{v}_0 + \bar{\omega}_0 \times \bar{r}_{L/O}$$

$$
\begin{aligned}
&= \dot{x}\bar{i}_0 + \dot{\psi}\bar{k}_0 \times \frac{L}{2}\bar{j}_0 \\
&= \left(\dot{x} - \frac{L}{2}\dot{\psi}\right)\bar{i}_0
\end{aligned} \tag{18}
$$

Linear acceleration:

$$
\begin{aligned}
\bar{a}_{GL} &= \bar{a}_0 + \bar{\alpha}_0 \times \bar{r}_{L/O} + \bar{\omega}_0 \times \left(\omega_0 \times \bar{r}_{L/O}\right) \\
&= \ddot{x}\bar{i}_0 + \dot{x}\left(\dot{\psi}\bar{k}_0 \times \bar{i}_0\right) + \ddot{\psi}\bar{k}_0 \times \frac{L}{2}\bar{j}_0 + \dot{\psi}\bar{k}_0 \times \left(\dot{\psi}\bar{k}_0 \times \frac{L}{2}\bar{j}_0\right) \\
&= \left(\ddot{x} - \frac{L}{2}\ddot{\psi}\right)\bar{i}_0 + \left(\dot{x}\dot{\psi} - \frac{L}{2}\dot{\psi}^2\right)\bar{j}_0
\end{aligned} \tag{19}
$$

Angular momentum and its derivative:

$$
\begin{aligned}
\bar{H}_{GL} &= I_w\bar{\omega}_L \\
\frac{d\bar{H}_{GL}}{dt} &= \frac{\partial\bar{H}_{GL}}{\partial t} + \bar{\omega}_L \times \bar{H}_{GL}
\end{aligned} \tag{20}
$$

where $I_w = \begin{bmatrix} \mathbf{ZZ}_w & 0 & 0 \\ 0 & \mathbf{YY}_w & 0 \\ 0 & 0 & \mathbf{ZZ}_w \end{bmatrix}$. Due to symmetry the off-diogonal terms in the inertia matrix vanish, and the inertia about $x_L$-axis and $z_L$-axis are both equal (signified by $\mathbf{ZZ}_w$).

#### 5.2.2 Right Wheel

This evaluation takes place in the $x_R y_R z_R$ frame fixed to the right wheel such that it is parallel to frame $x_0 y_0 z_0$ when $\theta_R = 0$. So $\bar{i}_0 = cos\theta_R\bar{i}_R + sin\theta_R\bar{k}_R$, $\bar{j}_0 = \bar{j}_R$ and $\bar{k}_0 = -sin\theta_R\bar{i}_R + cos\theta_R\bar{k}_R$. Angular velocity:

$$
\begin{aligned}
\bar{\omega}_R &= \dot{\psi}\bar{k}_0 + \dot{\theta}_R\bar{j}_0 \\
&= \dot{\psi}\bar{k}_0 + \left(\frac{1}{R}\dot{x} + \frac{L}{2R}\dot{\psi}\right)\bar{j}_0 \\
&= -\dot{\psi}sin\theta_L\bar{i}_R + \left(\frac{1}{R}\dot{x} + \frac{L}{2R}\dot{\psi}\right)\bar{j}_R + \dot{\psi}cos\theta_L\bar{k}_R
\end{aligned} \tag{21}
$$

The terms that follow are also similarly to be expressed in frame $x_R y_R z_R$ but that step is skipped for brevity. Velocity:

$$
\begin{aligned}
\bar{v}_{GR} &= \bar{v}_0 + \bar{\omega}_0 \times \bar{r}_{R/O} \\
&= \dot{x}\bar{i}_0 + \dot{\psi}\bar{k}_0 \times \left(-\frac{L}{2}\bar{j}_0\right) \\
&= \left(\dot{x} + \frac{L}{2}\dot{\psi}\right)\bar{i}_0
\end{aligned} \tag{22}
$$

Linear acceleration:

$$
\bar{a}_{GR} = \bar{a}_0 + \bar{\alpha}_0 \times \bar{r}_{R/O} + \bar{\omega}_0 \times \left(\omega_0 \times \bar{r}_{R/O}\right)
$$

$$
\begin{aligned}
&= \ddot{x}\bar{i}_0 + \dot{x}\left(\dot{\psi}\bar{k}_0 \times \bar{i}_0\right) + \ddot{\psi}\bar{k}_0 \times \left(-\frac{L}{2}\bar{j}_0\right) + \dot{\psi}\bar{k}_0 \times \left(\dot{\psi}\bar{k}_0 \times \left(-\frac{L}{2}\bar{j}_0\right)\right) \\
&= \left(\ddot{x} + \frac{L}{2}\ddot{\psi}\right)\bar{i}_0 + \left(\dot{x}\dot{\psi} + \frac{L}{2}\dot{\psi}^2\right)\bar{j}_0
\end{aligned} \tag{23}
$$

Angular momentum and its derivative:

$$
\begin{aligned}
\bar{H}_{GR} &= I_w \bar{\omega}_R \\
\frac{d\bar{H}_{GR}}{dt} &= \frac{\partial \bar{H}_{GR}}{\partial t} + \bar{\omega}_R \times \bar{H}_{GR}
\end{aligned} \tag{24}
$$

### 5.2.3 Wheel Contribution to Kane's LHS

The contribution of wheels to LHS of the equation corresponding to generalized speed $\dot{q}_j$ is:

$$
\left(m_w \bar{a}_{GL} \cdot \frac{\partial \bar{v}_{GL}}{\partial \dot{q}_j} + \left(\frac{d\bar{H}_{GL}}{dt}\right) \cdot \frac{\partial \bar{\omega}_L}{\partial \dot{q}_j}\right) + \left(m_w \bar{a}_{GR} \cdot \frac{\partial \bar{v}_{GR}}{\partial \dot{q}_j} + \left(\frac{d\bar{H}_{GR}}{dt}\right) \cdot \frac{\partial \bar{\omega}_R}{\partial \dot{q}_j}\right) \tag{25}
$$

Since:

$$
\frac{\partial \bar{v}_{GL}}{\partial \dot{q}_j} = \begin{cases} cos\theta_L \bar{i}_L + sin\theta_L \bar{k}_L \\ -\frac{L}{2}\left(cos\theta_L \bar{i}_L + sin\theta_L \bar{k}_L\right) \\ 0 \end{cases} \qquad \frac{\partial \bar{v}_{GR}}{\partial \dot{q}_j} = \begin{cases} cos\theta_R \bar{i}_R + sin\theta_R \bar{k}_R & \text{for } \dot{q}_j = \dot{x} \\ \frac{L}{2}\left(cos\theta_R \bar{i}_R + sin\theta_R \bar{k}_R\right) & \text{for } \dot{q}_j = \dot{\psi} \\ 0 & \text{Otherwise} \end{cases} \tag{26}
$$

$$
\frac{\partial \bar{\omega}_L}{\partial \dot{q}_j} = \begin{cases} \frac{1}{R}\bar{j}_L \\ -sin\theta_L \bar{i}_L - \frac{L}{2R}\bar{j}_L + cos\theta_L \bar{k}_L \\ 0 \end{cases} \qquad \frac{\partial \bar{\omega}_R}{\partial \dot{q}_j} = \begin{cases} \frac{1}{R}\bar{j}_R & \text{for } \dot{q}_j = \dot{x} \\ -sin\theta_R \bar{i}_R + \frac{L}{2R}\bar{j}_R + cos\theta_R \bar{k}_R & \text{for } \dot{q}_j = \dot{\psi} \\ 0 & \text{Otherwise} \end{cases} \tag{27}
$$

Therefore the contribution is only there for the first two equations (i.e. for $\dot{q}_j \in \{\dot{x}, \dot{\psi}\}$) and is zero for all other equations. The final expressions are:

$$
\begin{aligned}
\dot{x} &: \quad \left(2m_w + \tfrac{2\mathbf{YY}_w}{R^2}\right)\ddot{x} \\
\dot{\psi} &: \quad \left(\tfrac{m_w L^2}{2} + \tfrac{\mathbf{YY}_w L^2}{2R^2} + 2\mathbf{ZZ}_w\right)\ddot{\psi}
\end{aligned} \tag{28}
$$

## 5.3 Serial Tree-Structure

### 5.3.1 Recusrive Formulae

The angular and linear velocities of the frames on the rest of the robot can be calculated using the recursive formulation because it is a serial chain of links:

$$ {}^j\omega_j = {}^jA_i\, {}^i\omega_i + \dot{q}_j\, {}^je_j \tag{29} $$

$$ {}^j\alpha_j = {}^jA_i\, {}^i\alpha_i + \ddot{q}_j\, {}^je_j + \dot{q}_j\left({}^j\omega_j \times {}^je_j\right) \tag{30} $$

$$ {}^jV_j = {}^jA_i\left({}^iV_i + {}^i\omega_i \times {}^iP_j\right) \tag{31} $$

$$ {}^ja_j = {}^jA_i\left({}^ia_i + {}^i\alpha_i \times {}^iP_j + {}^i\omega_i \times \left({}^i\omega_i \times {}^iP_j\right)\right) \tag{32} $$

where ${}^i\omega_j$, ${}^i\alpha_j$, ${}^ia_j$ and ${}^iV_j$ denote the angular velocity, linear velocity, angular acceleration and linear acceleration repectively of frame $j$ measured with respect to the world frame and represented in frame $i$. ${}^je_j$ denotes the direction of local angular velocity of frame $j$ represented in frame $j$. $i, j \in \mathbb{F}$ identify the frames and $i$ identifies the antecedent frame of $j$. In order to successfully evaluate the expressions for each body we need the following terms:

- $\bar{\omega}_0$ The angular velocity of frame $x_0y_0z_0$ (section 5.1)
- $\bar{v}_0$ The linear velocity of frame $x_0y_0z_0$ (section 5.1)
- $\bar{\alpha}_0$ The angular acceleration of frame $x_0y_0z_0$ (section 5.1)
- $\bar{a}_0$ The linear acceleration of frame $x_0y_0z_0$ (section 5.1)
- ${}^jA_i$ The rotation transform of antecent frame $i$ in the current frame $j$ for all frames in the tree structure (section 5.3.2)
- ${}^iP_j$ The position of the current frame $j$ for all frames in the tree structure wrt the antecedent frame $i$ of each frame (section 5.3.2)
- ${}^je_j$ The direction of local angular velocity of frame $j$ represented in frame $j$ for all frames of the tree structure (section 5.3.3)

### 5.3.2 Transformations

The transformation of frame $R_i$ into frame $R_j$ is represented by the homogeneous transformation matrix ${}^iT_j$ such that.

$$ {}^iT_j = \begin{bmatrix} {}^is_j & {}^in_j & {}^ia_j & {}^iP_j \end{bmatrix} = \begin{bmatrix} & {}^iA_j & & {}^iP_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33} $$

where ${}^is_j$, ${}^in_j$ and ${}^ia_j$ contain the components of the unit vectors along the $x_j$, $y_j$ and $z_j$ axes respectively expressed in frame $R_i$, and where ${}^iP_j$ is the vector representing the coordinates of the origin of frame $R_j$ expressed in frame $R_i$.

The transformation matrix ${}^iT_j$ can be interpreted as: (a) the transformation from frame $R_i$ to frame $R_j$ and (b) the representation of frame $R_j$ with respect to frame $R_i$. Using figure 1, we can write down these transformation matrices for our system as follows:

$$ {}^0T_1 = \begin{bmatrix} 0 & sq_{imu} & -cq_{imu} & 0 \\ -1 & 0 & 0 & 0 \\ 0 & cq_{imu} & sq_{imu} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^1T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_w & sq_w & L_1 \\ 0 & -sq_w & cq_w & -L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^2T_3 = \begin{bmatrix} -cq_{torso} & 0 & sq_{torso} & 0 \\ 0 & 1 & 0 & L_3 \\ -sq_{torso} & 0 & -cq_{torso} & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^3T_{4l} = \begin{bmatrix} 0 & 1 & 0 & L_6 \\ cq_{1l} & 0 & -sq_{1l} & L_5 \\ -sq_{1l} & 0 & -cq_{1l} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^3T_{4r} = \begin{bmatrix} 0 & -1 & 0 & -L_6 \\ cq_{1r} & 0 & -sq_{1r} & L_5 \\ sq_{1r} & 0 & cq_{1r} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^4T_5 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -cq_2 & -sq_2 & 0 \\ 0 & -sq_2 & cq_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^5T_6 = \begin{bmatrix} -cq_3 & 0 & sq_3 & 0 \\ 0 & -1 & 0 & -L_7 \\ sq_3 & 0 & cq_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^6T_7 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -cq_4 & -sq_4 & 0 \\ 0 & -sq_4 & cq_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^7T_8 = \begin{bmatrix} -cq_5 & 0 & sq_5 & 0 \\ 0 & -1 & 0 & -L_8 \\ sq_5 & 0 & cq_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$
{}^{8}T_9 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -cq_6 & -sq_6 & 0 \\ 0 & -sq_6 & cq_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, {}^{9}T_{10} = \begin{bmatrix} -cq_7 & -sq_7 & 0 & 0 \\ 0 & 0 & -1 & -L_9 \\ sq_7 & -cq_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The rotation ${}^{j}A_i$ and the translation ${}^{j}P_i$ that appear in eqs. 29-32 can not be directly deduced from the transformations listed above, as the they all represent ${}^{i}T_j$ (note the position of $i$ and $j$). Rather, we need to use following expressions to deduce our required transformations:

$$
\begin{aligned}
{}^{j}A_i &= {}^{i}A_j^T \\
{}^{j}P_i &= -{}^{i}A_j^T\,{}^{i}P_j
\end{aligned}
$$

### 5.3.3 Local Angular Velocity Directions

The unit vectors along the direction of angular velocities of the frames in the tree-structure each represented in its local frame are as follows (using figure 1).

$$
{}^{0}e_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T, {}^{1}e_1 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T, {}^{2}e_2 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T, {}^{3}e_3 = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T,
$$
$$
{}^{4}e_4 = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T, {}^{5}e_5 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T, {}^{6}e_6 = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T, {}^{7}e_7 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T,
$$
$$
{}^{8}e_8 = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T, {}^{9}e_9 = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T, {}^{10}e_{10} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T
$$

This information can now be used to derive expressions for the velocities and accelerations of the frames.

### 5.3.4 Simplified Kane's Left-Hand Side Term for each body in the Tree-Structure

The inertial forces i.e. the term inside the brackets on the LHS in Kane's formulation can be simplified by expansion and manipulation that results in cancelation of terms leading to a simplified expression. We show the details of this manipulation here. The final expression is the outcome in the end which we will use in our code to find the dynamic model.

$$
\bar{v}_{Gk} = \bar{v}_k + \bar{\omega}_k \times \bar{S}_k \tag{34}
$$
$$
\bar{a}_{Gk} = \bar{a}_k + \bar{\alpha}_k \times \bar{S}_k + \bar{\omega}_k \times (\bar{\omega}_k \times \bar{S}_k) \tag{35}
$$
$$
\bar{H}_{Gk} = \mathbf{J}_{Gk}\bar{\omega}_k \tag{36}
$$
$$
\begin{aligned}
\frac{d\bar{H}_{Gk}}{dt} &= \mathbf{J}_{Gk}\bar{\alpha}_k + \bar{\omega}_k \times \mathbf{J}_{Gk}\bar{\omega}_k \\
&= (\mathbf{J}_k + m_k\mathbf{S}_k^\times\mathbf{S}_k^\times)\bar{\alpha}_k + \bar{\omega}_k \times (\mathbf{J}_k + m_k\mathbf{S}_k^\times\mathbf{S}_k^\times)\bar{\omega}_k \\
&= \mathbf{J}_k\bar{\alpha}_k + \bar{\omega}_k \times \mathbf{J}_k\bar{\omega}_k + m_k\mathbf{S}_k^\times\mathbf{S}_k^\times\bar{\alpha}_k + \bar{\omega}_k \times (m_k\mathbf{S}_k^\times\mathbf{S}_k^\times\bar{\omega}_k)
\end{aligned} \tag{37}
$$
$$
\begin{aligned}
m_k\bar{a}_{Gk} \cdot (\bar{v}_{Gk})_j &= m_k\bar{a}_{Gk} \cdot \left(\bar{v}_k + \bar{\omega}_k \times \bar{S}_k\right)_j \\
&= m_k\bar{a}_{Gk} \cdot (\bar{v}_k)_j + m_k\bar{a}_{Gk} \cdot \left(\bar{\omega}_k \times \bar{S}_k\right)_j \\
&= m_k\bar{a}_{Gk} \cdot (\bar{v}_k)_j - m_k\left(\bar{a}_k + \bar{\alpha}_k \times \bar{S}_k + \bar{\omega}_k \times (\bar{\omega}_k \times \bar{S}_k)\right) \cdot \left(\bar{S}_k \times \bar{\omega}_k\right)_j \\
&= m_k\bar{a}_{Gk} \cdot (\bar{v}_k)_j - m_k\left(\bar{a}_k - \bar{S}_k \times \bar{\alpha}_k - \bar{\omega}_k \times (\bar{S}_k \times \bar{\omega}_k)\right) \cdot \left(\bar{S}_k \times (\bar{\omega}_k)_j\right) \\
&= m_k\bar{a}_{Gk} \cdot (\bar{v}_k)_j - m_k\left(\bar{a}_k - \mathbf{S}_k^\times\bar{\alpha}_k - \omega_k^\times\mathbf{S}_k^\times\bar{\omega}_k\right)^T \mathbf{S}_k^\times\,(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk} \cdot (\bar{v}_k)_j - m_k\left(\mathbf{S}_k^{\times T}\bar{a}_k - \mathbf{S}_k^{\times T}\mathbf{S}_k^\times\bar{\alpha}_k - \mathbf{S}_k^{\times T}\omega_k^\times\mathbf{S}_k^\times\bar{\omega}_k\right)^T (\bar{\omega}_k)_j
\end{aligned}
$$

$$
\begin{aligned}
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + m_k\left(\mathbf{S}_k^{\times}\bar{a}_k - \mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\alpha}_k - \mathbf{S}_k^{\times}\omega_k^{\times}\mathbf{S}_k^{\times}\bar{\omega}_k\right)^T(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + m_k\left(\bar{S}_k\times\bar{a}_k - \bar{S}_k\times\bar{S}_k\times\bar{\alpha}_k - \bar{S}_k\times(\bar{\omega}_k\times(\bar{S}_k\times\bar{\omega}_k))\right)\cdot(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + m_k\left(\bar{S}_k\times\bar{a}_k - \bar{S}_k\times\bar{S}_k\times\bar{\alpha}_k + \bar{\omega}_k\times((\bar{S}_k\times\bar{\omega}_k)\times\bar{S}_k) + (\bar{S}_k\times\bar{\omega}_k)\times(\bar{S}_k\times\bar{\omega}_k)\right)\cdot(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + m_k\left(\bar{S}_k\times\bar{a}_k - \bar{S}_k\times\bar{S}_k\times\bar{\alpha}_k - \bar{\omega}_k\times(\bar{S}_k\times(\bar{S}_k\times\bar{\omega}_k)) + 0\right)\cdot(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + \left(m_k\bar{S}_k\times\bar{a}_k - m_k\mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\alpha}_k - \bar{\omega}_k\times(m_k\mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\omega}_k)\right)\cdot(\bar{\omega}_k)_j
\end{aligned}
\tag{38}
$$

$$
\begin{aligned}
m_k\bar{a}_{Gk}\cdot(\bar{v}_{Gk})_j + &\left(\frac{d\bar{H}_{Gk}}{dt}\right)\cdot(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + \left(m_k\bar{S}_k\times\bar{a}_k - m_k\mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\alpha}_k - \bar{\omega}_k\times(m_k\mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\omega}_k)\right)\cdot(\bar{\omega}_k)_j \\
&\quad + \left(\mathbf{J}_k\bar{\alpha}_k + \bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k + m_k\mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\alpha}_k + \bar{\omega}_k\times(m_k\mathbf{S}_k^{\times}\mathbf{S}_k^{\times}\bar{\omega}_k)\right)\cdot(\bar{\omega}_k)_j \\
&= m_k\bar{a}_{Gk}\cdot(\bar{v}_k)_j + \left(m_k\bar{S}_k\times\bar{a}_k + \mathbf{J}_k\bar{\alpha}_k + \bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot(\bar{\omega}_k)_j
\end{aligned}
\tag{39}
$$

The terms $\bar{\omega}_k$, $\bar{\alpha}_k$, $\bar{v}_k$, $\bar{a}_k$ will be found using the recursive formulations in eqs. 29-32. And the term $\bar{a}_{Gk}$ will be found using eq.35. $\bar{S}_k = [\mathbf{X}_k \quad \mathbf{Y}_k \quad \mathbf{Z}_k]^T$ is the center of mass of the joint represented in the local frame. We will use the symbol $\mathbf{MS}_k$ to represent mass times center of mass $(m_k\mathbf{S}_k)$ which is the vector the components of which are $\mathbf{MS} = [\mathbf{MX} \quad \mathbf{MY} \quad \mathbf{MZ}]^T$. Finally, $\mathbf{J}_k = \begin{bmatrix} \mathbf{XX}_k & \mathbf{XY}_k & \mathbf{XZ}_k \\ \mathbf{XY}_k & \mathbf{YY}_k & \mathbf{YZ}_k \\ \mathbf{XZ}_k & \mathbf{YZ}_k & \mathbf{ZZ}_k \end{bmatrix}$ is the inertia matrix of the joint represented in the local frame.

The terms $(\bar{v}_k)_j = \frac{\partial \bar{v}_k}{\partial \dot{q}_j}$ and $(\bar{\omega}_k)_j = \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j}$ can also be evaluated as follows. If it was a single serial chain:

$$
\begin{aligned}
\bar{\omega}_k &= \dot{q}_0\bar{e}_0 + \dot{q}_1\bar{e}_1 + \dot{q}_2\bar{e}_2 + ... + \dot{q}_k\bar{e}_k \\
\frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} &= \begin{cases} 0 & \text{if } k < j \\ \bar{e}_j & \text{if } k \geq j \end{cases}
\end{aligned}
\tag{40}
$$

Applying this result to our tree-structure we will have $\frac{\partial \bar{\omega}_k}{\partial \dot{x}} = 0$ as none of the angular velocities of the joints $\bar{\omega}_k$ depend on $\dot{x}$. As for $\dot{q}_j \neq \dot{x}$ the result will be $\bar{e}_j$ whenever $\bar{\omega}_k$ refers to speed of the link which is ahead of, or same as, the link whose speed is represented by $\dot{q}_j$ and 0 otherwise. For $(\bar{v}_k)_j = \frac{\partial \bar{v}_k}{\partial \dot{q}_j}$ in case of a single serial chain we get:

$$
\begin{aligned}
\bar{v}_k &= \bar{v}_{k-1} + \bar{\omega}_{k-1}\times\bar{r}_{O_k/O_{k-1}} \\
&\;... \\
&= \bar{v}_0 + \bar{\omega}_0\times\bar{r}_{O_1/O_0} + \bar{\omega}_1\times\bar{r}_{O_2/O_1} + \bar{\omega}_2\times\bar{r}_{O_3/O_2} + ... + \bar{\omega}_{k-1}\times\bar{r}_{O_k/O_{k-1}} \\
&= \bar{v}_0 + \sum_{\kappa=1}^{k}\left(\bar{\omega}_{\kappa-1}\times\bar{r}_{O_\kappa/O_{\kappa-1}}\right)
\end{aligned}
\tag{41}
$$

When partially differentiated wrt $\dot{q}_j$ $(\neq \dot{x})$ we can use eq. 40 to get:

$$
\begin{aligned}
\frac{\partial \bar{v}_k}{\partial \dot{q}_j} &= \frac{\partial \bar{v}_0}{\partial \dot{q}_j} + \sum_{\kappa=1}^{k}\left(\frac{\partial \bar{\omega}_{\kappa-1}}{\partial \dot{q}_j}\times\bar{r}_{O_\kappa/O_{\kappa-1}}\right) \\
&= 0 + \sum_{\kappa=1}^{j}\left(\frac{\partial \bar{\omega}_{\kappa-1}}{\partial \dot{q}_j}\times\bar{r}_{O_\kappa/O_{\kappa-1}}\right) + \sum_{\kappa=j+1}^{k}\left(\frac{\partial \bar{\omega}_{\kappa-1}}{\partial \dot{q}_j}\times\bar{r}_{O_\kappa/O_{\kappa-1}}\right) \\
&= \sum_{\kappa=1}^{j}\left(0\times\bar{r}_{O_\kappa/O_{\kappa-1}}\right) + \sum_{\kappa=j+1}^{k}\left(\bar{e}_j\times\bar{r}_{O_\kappa/O_{\kappa-1}}\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \bar{e}_j \times \sum_{\kappa=j+1}^{k} \bar{r}_{O_\kappa/O_{\kappa-1}} \\
&= \bar{e}_j \times \left(\bar{r}_{O_{j+1}/O_j} + \bar{r}_{O_{j+2}/O_{j+1}} + \bar{r}_{O_{j+3}/O_{j+2}} + ... + \bar{r}_{O_k/O_{k-1}}\right) \\
&= \begin{cases} 0 & \text{for } k \leq j \\ \bar{e}_j \times \bar{r}_{O_k/O_j} & \text{for } k > j \end{cases}
\end{aligned} \tag{42}
$$

In eq. 39, the term $\frac{\partial \bar{v}_k}{\partial \dot{q}_j}$ is operated by dot-product with $m_k \bar{a}_{Gk}$. Using the result above and then applying a triple-product identity, we will get:

$$
m_k \bar{a}_{Gk} \cdot \frac{\partial \bar{v}_k}{\partial \dot{q}_j} = m_k \bar{a}_{Gk} \cdot \left(\bar{e}_j \times \bar{r}_{O_k/O_j}\right) = \left(\bar{r}_{O_k/O_j} \times m_k \bar{a}_{Gk}\right) \cdot \bar{e}_j \qquad \text{for } k > j
$$

For $\dot{q}_j = \dot{x}$ we have:

$$
\begin{aligned}
\frac{\partial \bar{v}_k}{\partial \dot{q}_j} &= \frac{\partial \bar{v}_k}{\partial \dot{q}_j} + \sum_{\kappa=1}^{k} \left( \frac{\partial \bar{\omega}_\kappa}{\partial \dot{q}_j} \times \bar{r}_{O_\kappa/O_{\kappa-1}} \right) \\
&= \frac{\partial \bar{v}_0}{\partial \dot{x}} + 0 \\
&= \bar{i}_0
\end{aligned} \tag{43}
$$

So the sum of the contributions of all joints of the tree-structure for the LHS of

Kane's equations are as follows:

$$
\begin{array}{rrcr}
\dot{x}: & \sum\limits_{k\in\{imu,w,torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} (m_k\bar{a}_{Gk})\cdot\bar{i}_0 & + & 0 \\
\dot{\psi}: & \sum\limits_{k\in\{imu,w,torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(\bar{r}_{O_k/O_0}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_0 & + & \sum\limits_{k\in\{imu,w,torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_0 \\
\dot{q}_{imu}: & \sum\limits_{k\in\{w,torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(\bar{r}_{O_k/O_1}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_1 & + & \sum\limits_{k\in\{imu,w,torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_1 \\
\dot{q}_{w}: & \sum\limits_{k\in\{torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(\bar{r}_{O_k/O_2}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_2 & + & \sum\limits_{k\in\{w,torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_2 \\
\dot{q}_{torso}: & \sum\limits_{k\in\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(\bar{r}_{O_k/O_3}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_3 & + & \sum\limits_{k\in\{torso\}\cup\{4l,\dots,10l\}\cup\{4r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_3 \\
\dot{q}_{1l}: & \sum\limits_{k\in\{5l,\dots,10l\}} \left(\bar{r}_{O_k/O_{4l}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{4l} & + & \sum\limits_{k\in\{4l,\dots,10l\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{4l} \\
\dot{q}_{2l}: & \sum\limits_{k\in\{6l,\dots,10l\}} \left(\bar{r}_{O_k/O_{5l}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{5l} & + & \sum\limits_{k\in\{5l,\dots,10l\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{5l} \\
\dot{q}_{3l}: & \sum\limits_{k\in\{7l,\dots,10l\}} \left(\bar{r}_{O_k/O_{6l}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{6l} & + & \sum\limits_{k\in\{6l,\dots,10l\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{6l} \\
\dot{q}_{4l}: & \sum\limits_{k\in\{8l,9l,10l\}} \left(\bar{r}_{O_k/O_{7l}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{7l} & + & \sum\limits_{k\in\{7l,\dots,10l\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{7l} \\
\dot{q}_{5l}: & \sum\limits_{k\in\{9l,10l\}} \left(\bar{r}_{O_k/O_{8l}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{8l} & + & \sum\limits_{k\in\{8l,9l,10l\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{8l} \\
\dot{q}_{6l}: & \left(\bar{r}_{O_{10l}/O_{9l}}\times m_{10l}\bar{a}_{G\,10l}\right)\cdot\bar{e}_{9l} & + & \sum\limits_{k\in\{9l,10l\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{9l} \\
\dot{q}_{7l}: & 0 & + & \left(m_{10l}\bar{S}_{10l}\times\bar{a}_{10l}+\mathbf{J}_{10l}\bar{\alpha}_{10l}+\bar{\omega}_{10l}\times\mathbf{J}_{10l}\bar{\omega}_{10l}\right)\cdot\bar{e}_{10l} \\
\dot{q}_{1r}: & \sum\limits_{k\in\{5r,\dots,10r\}} \left(\bar{r}_{O_k/O_{4r}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{4r} & + & \sum\limits_{k\in\{4r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{4r} \\
\dot{q}_{2r}: & \sum\limits_{k\in\{6r,\dots,10r\}} \left(\bar{r}_{O_k/O_{5r}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{5r} & + & \sum\limits_{k\in\{5r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{5r} \\
\dot{q}_{3r}: & \sum\limits_{k\in\{7r,\dots,10r\}} \left(\bar{r}_{O_k/O_{6r}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{6r} & + & \sum\limits_{k\in\{6r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{6r} \\
\dot{q}_{4r}: & \sum\limits_{k\in\{8r,9r,10r\}} \left(\bar{r}_{O_k/O_{7r}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{7r} & + & \sum\limits_{k\in\{7r,\dots,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{7r} \\
\dot{q}_{5r}: & \sum\limits_{k\in\{9r,10r\}} \left(\bar{r}_{O_k/O_{8r}}\times m_k\bar{a}_{Gk}\right)\cdot\bar{e}_{8r} & + & \sum\limits_{k\in\{8r,9r,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{8r} \\
\dot{q}_{6r}: & \left(\bar{r}_{O_{10r}/O_{9r}}\times m_{10r}\bar{a}_{G\,10r}\right)\cdot\bar{e}_{9r} & + & \sum\limits_{k\in\{9r,10r\}} \left(m_k\bar{S}_k\times\bar{a}_k+\mathbf{J}_k\bar{\alpha}_k+\bar{\omega}_k\times\mathbf{J}_k\bar{\omega}_k\right)\cdot\bar{e}_{9r} \\
\dot{q}_{7r}: & 0 & + & \left(m_{10r}\bar{S}_{10r}\times\bar{a}_{10r}+\mathbf{J}_{10r}\bar{\alpha}_{10r}+\bar{\omega}_{10r}\times\mathbf{J}_{10r}\bar{\omega}_{10r}\right)\cdot\bar{e}_{10r}
\end{array}
\tag{44}
$$

# 6 Kane's Right-Hand Side

The right hand side of the Kane's forumulation 12 is the sum of some dot product terms. Each term is either the dot product of:

- force applied on the system $\bar{F}_n$
- the linear velocity $\bar{v}_n$ of the point differentiated partially wrt the the unique gerneralized speed $\dot{q}_j$ corresponding to each equation i.e. $\frac{\partial\bar{v}_n}{\partial\dot{q}_j}$

or the dot product of:

- torque applied on the system $\bar{\tau}_n$
- the angular velocity $\bar{\omega}_n$ of the body differentiated partially wrt the the unique gerneralized speed $\dot{q}_j$ corresponding to each equation i.e. $\frac{\partial\bar{\omega}_n}{\partial\dot{q}_j}$

So, in order to analyse the right-hand side of the equation, we need to list down all the forces and torques applied to the system and the points at which they are being applied. They are as follows.

$\bar{\tau}_L, \bar{\tau}_R$ Torques applied by wheel motors (in the body) on the right wheel and left wheel at points $R$ and $L$ on the wheels

$\bar{\tau}_j = \tau_j \bar{e}_j$ The torque applied by each joint motor fixed on the antecedent joint moving the current joint. There are 17 such torques as

$$\bar{\tau}_j \in \{\bar{\tau}_{imu}, \bar{\tau}_w, \bar{\tau}_{torso}, \bar{\tau}_{1l}, ..., \bar{\tau}_{7l}, \bar{\tau}_{1r}, ..., \bar{\tau}_{7r}\}$$

Note that $\bar{\tau}_{imu} = -\bar{\tau}_R - \bar{\tau}_L$ is the sum of reactions torques experienced by the base in response to the wheel torques $\bar{\tau}_L$ and $\bar{\tau}_R$.

$-\bar{\tau}_j = -\tau_j \bar{e}_j$ The reaction torque experienced by each antecedent joint. Note that we do not consider $\tau_{imu}$ in this set of reaction torques as those are already covered in the first item in this list i.e. $\bar{\tau}_L, \bar{\tau}_R$. So in this set of reaction torques we consider only 16 torques $\in \{-\bar{\tau}_w,\ -\bar{\tau}_{torso},\ -\bar{\tau}_{1l},\ ...,\ -\bar{\tau}_{7l},\ -\bar{\tau}_{1r},\ ...,\ -\bar{\tau}_{7r}\}$

$\bar{F}_{gk} = m_k \bar{g}$ is the weight of each joint $k$

$\bar{F}_{el}, \bar{\tau}_{el}$ Force and torque applied by the efor k ¿ jnvironment on the left hand end-effector at point $E_l$

$\bar{F}_{er}, \bar{\tau}_{er}$ Force and torque applied by the environment on the right hand end-effector at point $E_r$

## 6.1 Wheel Torques

The contribution of wheel motor torques on the right-hand side of Kane's equation corresponding to generalized speed $\dot{q}_j$ is:

$$\bar{\tau}_L \cdot \frac{\partial \bar{\omega}_L}{\partial \dot{q}_j} + \bar{\tau}_R \cdot \frac{\partial \bar{\omega}_R}{\partial \dot{q}_j} \tag{45}$$

where,

$$\bar{\tau}_L = \tau_L \bar{j}_0$$

$$\bar{\tau}_R = \tau_R \bar{j}_0$$

$$\bar{\omega}_L = \dot{\psi}\bar{k}_0 + \left(\tfrac{1}{R}\dot{x} - \tfrac{L}{2R}\dot{\psi}\right)\bar{j}_0$$

$$\bar{\omega}_R = \dot{\psi}\bar{k}_0 + \left(\tfrac{1}{R}\dot{x} + \tfrac{L}{2R}\dot{\psi}\right)\bar{j}_0$$

Also,

$$\frac{\partial \bar{\omega}_L}{\partial \dot{q}_j} = \begin{cases} \frac{1}{R}\bar{j}_0 \\ -\frac{L}{2R}\bar{j}_0 + \bar{k}_0 \\ 0 \end{cases} \qquad \frac{\partial \bar{\omega}_R}{\partial \dot{q}_j} = \begin{cases} \frac{1}{R}\bar{j}_0 & \text{for } \dot{q}_j = \dot{x} \\ \frac{L}{2R}\bar{j}_0 + \bar{k}_0 & \text{for } \dot{q}_j = \dot{\psi} \\ 0 & \text{Otherwise} \end{cases} \tag{46}$$

So

$$\bar{\tau}_L \cdot \frac{\partial \bar{\omega}_L}{\partial \dot{q}_j} + \bar{\tau}_R \cdot \frac{\partial \bar{\omega}_R}{\partial \dot{q}_j}$$

$$= \begin{cases} \frac{1}{R}\left(\tau_L + \tau_R\right) & \text{for } \dot{q}_j = \dot{x} \\ \frac{L}{2R}\left(\tau_R - \tau_L\right) & \text{for } \dot{q}_j = \dot{\psi} \\ 0 & \text{Otherwise} \end{cases} \tag{47}$$

The contribution of the reaction torque $\bar{\tau}_{imu}$ experienced by the base, in reaction to the wheel torques $(\bar{\tau}_L, \bar{\tau}_R$, to Kane's RHS for generalized speed $\dot{q}_j$ will be:

$$\bar{\tau}_{imu} \cdot \frac{\partial \bar{\omega}_1}{\partial \dot{q}_j} \tag{48}$$

$$= -\left(\tau_L + \tau_R\right) \bar{j}_0 \cdot \frac{\partial \left(\dot{\psi}\bar{k}_0 + \dot{q}_{imu}\bar{j}_0\right)}{\partial \dot{q}_j}$$

$$= \begin{cases} -\left(\tau_L + \tau_R\right) & \text{for } \dot{q}_j = \dot{q}_{imu} \\ 0 & \text{Otherwise} \end{cases} \tag{49}$$

## 6.2 Joint Torques

We will be discussing the effect of 16 joint torques $\in \{\bar{\tau}_w, \bar{\tau}_{torso}, \bar{\tau}_{1l}, ..., \bar{\tau}_{7l}, \bar{\tau}_{1r}, ..., \bar{\tau}_{7r}\}$ and their reactions. The contribution of motor torque on joint $k$ and its reaction, on the right-hand side of Kane's equation corresponding to generalized speed $\dot{q}_j$ is:

$$\tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} - \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_{a(k)}}{\partial \dot{q}_j} \tag{50}$$

where $a(k)$ = antecedent frame of $k$. The first term is the contribution of action torques and the second term is the contribution of the reaction torques. If we take the summation of all joint torque contibutions each described by expression 50 i.e. for $k = 1...K$, to get get the right-hand side contribution by all joint torques in the kane's equaiton corresponding to a generalized speed $j$, we will get a very simplified solution

$$\sum_{k=1}^{K} \left( \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} - \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_{k-1}}{\partial \dot{q}_j} \right)$$

$$= \sum_{k=1}^{j-1} \left( \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} - \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_{k-1}}{\partial \dot{q}_j} \right)$$

$$+ \tau_j \bar{e}_j \cdot \frac{\partial \bar{\omega}_j}{\partial \dot{q}_j} - \tau_j \bar{e}_j \cdot \frac{\partial \bar{\omega}_{j-1}}{\partial \dot{q}_j}$$

$$+ \sum_{k=j+1}^{K} \left( \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} - \tau_k \bar{e}_k \cdot \frac{\partial \bar{\omega}_{k-1}}{\partial \dot{q}_j} \right)$$

$$= \sum_{k=1}^{j-1} (0 - 0) + \tau_j \bar{e}_j \cdot \bar{e}_j - 0 + \sum_{k=j+1}^{K} \left( \tau_k \bar{e}_k \cdot \bar{e}_j - \tau_k \bar{e}_k \cdot \bar{e}_j \right)$$

$$= \tau_j \tag{51}$$

So the total joint torques contributions on the RHS of each Kane's equation is just the joint torque actuating the generalized speed wrt which the equaiton is being evaluated. This analysis applies directly to the equations corresponding to generalized speeds $\in \{\ \dot{q}_w,\ \dot{q}_{torso},\ \dot{q}_{1l},\ ...,\ \dot{q}_{7l},\ \dot{q}_{1r},\ ...,\ \dot{q}_{7r}\}$. So we know the RHS contributions of the 16 torques to the last 16 of the total of nineteen equations. For the first three equations corresponding to the speeds $\{\dot{x},\ \dot{\psi},\ \dot{q}_{imu}\}$ we can prove that the contribution of all torques amounts to zero. For $\dot{x}$ it is easy to see that none of the angular velocities $\bar{\omega}_k$ depend on $\dot{x}$ hence the partial derivative vanishes. As for the case of $\dot{\psi}$ the contribution (expn. 50) of each torques is $\tau_k \bar{e}_k \cdot \bar{k}_0 - \tau_k \bar{e}_k \cdot \bar{k}_0 = 0$ and for $\dot{q}_{imu}$ it is equal to $\tau_k \bar{e}_k \cdot \bar{j}_0 - \tau_k \bar{e}_k \cdot \bar{j}_0 = 0$.

### 6.3 End-effector Forces and Torques

Let $\bar{F}_e$ and $\bar{\tau}_e$ be the force and torque being applied by the environment on the end-effector. The contribution on the RHS of Kane's equaiton corresponding to generalized speed $\dot{q}_j$ will be as follows assuming a single serial chain of links:

$$\bar{F}_e \cdot \frac{\partial \bar{v}_e}{\partial \dot{q}_j} + \bar{\tau}_e \cdot \frac{\partial \bar{\omega}_K}{\partial \dot{q}_j} \tag{52}$$

where $\bar{v}_e$ is the linear velocity of the point $E$ on the end-effector on which the force is being applied. And $\bar{\omega}_K$ is the angular velocity of the last joint on which the end-effector is mounted.

$$\begin{aligned}
\bar{v}_e &= \bar{v}_K + \bar{\omega}_K \times \bar{r}_{e/O_K} \\
&= \bar{v}_{K-1} + \bar{\omega}_{K-1} \times \bar{r}_{O_K/O_{K-1}} + \bar{\omega}_K \times \bar{r}_{E/O_K} \\
&... \\
&= \bar{v}_0 + \bar{\omega}_0 \times \bar{r}_{O_1/O_0} + \bar{\omega}_1 \times \bar{r}_{O_2/O_1} + \bar{\omega}_2 \times \bar{r}_{O_3/O_2} + ... + \bar{\omega}_{K-1} \times \bar{r}_{O_K/O_{K-1}} + \bar{\omega}_K \times \bar{r}_{O_{K+1}/O_K} \\
&= \bar{v}_0 + \sum_{k=0}^{K} \left(\bar{\omega}_k \times \bar{r}_{O_{k+1}/O_k}\right)
\end{aligned} \tag{53}$$

where we have replace $E$ with $O_{K+1}$ for the convenience of writing the closed-form expression. When partially differentiated wrt $\dot{q}_j$ ($\neq \dot{x}$) we can use eq. 40 to get:

$$\begin{aligned}
\frac{\partial \bar{v}_e}{\partial \dot{q}_j} &= \frac{\partial \bar{v}_0}{\partial \dot{q}_j} + \sum_{k=0}^{K} \left( \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} \times \bar{r}_{O_{k+1}/O_k} \right) \\
&= 0 + \sum_{k=0}^{j-1} \left( \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} \times \bar{r}_{O_{k+1}/O_k} \right) + \sum_{k=j}^{K} \left( \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} \times \bar{r}_{O_{k+1}/O_k} \right) \\
&= \sum_{k=0}^{j-1} \left(0 \times \bar{r}_{O_{k+1}/O_k}\right) + \sum_{k=j}^{K} \left(\bar{e}_j \times \bar{r}_{O_{k+1}/O_k}\right) \\
&= \bar{e}_j \times \sum_{k=j}^{K} \bar{r}_{O_{k+1}/O_k} \\
&= \bar{e}_j \times \left(\bar{r}_{O_{j+1}/O_j} + \bar{r}_{O_{j+2}/O_{j+1}} + \bar{r}_{O_{j+3}/O_{j+2}} + ... + \bar{r}_{O_{K+1}/O_K}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \bar{e}_j \times \bar{r}_{O_{K+1}/O_j} \\
&= \bar{e}_j \times \bar{r}_{E/O_j}
\end{aligned}
$$

For $\dot{q}_j = \dot{x}$ we have:

$$
\begin{aligned}
\frac{\partial \bar{v}_e}{\partial \dot{q}_j} &= \frac{\partial \bar{v}_0}{\partial \dot{q}_j} + \sum_{k=0}^{K} \left( \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} \times \bar{r}_{O_{k+1}/O_k} \right) \\
&= \frac{\partial \bar{v}_0}{\partial \dot{x}} + 0 \\
&= \bar{i}_0
\end{aligned}
$$

So

$$
\frac{\partial \bar{v}_e}{\partial \dot{q}_j} = \begin{cases} \bar{i}_0 & \text{for } \dot{q}_j = \dot{x} \\ \bar{e}_j \times \bar{r}_{E/O_j} & \text{Otherwise} \end{cases} \tag{54}
$$

Substituting eqs. 54 and 40 in expression 52, we get

$$
\begin{aligned}
\bar{F}_e \cdot \frac{\partial \bar{v}_e}{\partial \dot{q}_j} + \bar{\tau}_e \cdot \frac{\partial \bar{\omega}_K}{\partial \dot{q}_j} &= \begin{cases} \bar{F}_e \cdot \bar{i}_0 + \bar{\tau}_e \cdot 0 & \text{for } \dot{q}_j = \dot{x} \\ \bar{F}_e \cdot \left(\bar{e}_j \times \bar{r}_{E/O_j}\right) + \bar{\tau}_e \cdot \bar{e}_j & \text{Otherwise} \end{cases} \\
&= \begin{cases} \begin{bmatrix} \bar{i}_0^T & O_{1\times 3} \end{bmatrix} \begin{bmatrix} \bar{F}_e \\ \bar{\tau}_e \end{bmatrix} & \text{for } \dot{q}_j = \dot{x} \\ \begin{bmatrix} \left[\bar{e}_j \times \bar{r}_{E/O_j}\right]^T & \bar{e}_j^T \end{bmatrix} \begin{bmatrix} \bar{F}_e \\ \bar{\tau}_e \end{bmatrix} & \text{Otherwise} \end{cases}
\end{aligned}
$$

If we write all kane's equations in the form of a vector then the right hand side contribution of end-effector force and torque will become,

$$
\begin{aligned}
&\begin{bmatrix} \bar{i}_0^T & O_{1\times 3} \\ \left[\bar{e}_0 \times \bar{r}_{E/O_0}\right]^T & \bar{e}_1^T \\ \left[\bar{e}_1 \times \bar{r}_{E/O_1}\right]^T & \bar{e}_2^T \\ \left[\bar{e}_2 \times \bar{r}_{E/O_2}\right]^T & \bar{e}_3^T \\ \dots & \\ \left[\bar{e}_K \times \bar{r}_{E/O_K}\right]^T & \bar{e}_K^T \end{bmatrix} \begin{bmatrix} \bar{F}_e \\ \bar{\tau}_e \end{bmatrix} \\
&= \mathbb{J}^T \mathbb{f}_e
\end{aligned} \tag{55}
$$

where

$$
\begin{aligned}
\mathbb{J} &= \begin{bmatrix} \bar{i}_0 & \bar{e}_0 \times \bar{r}_{E/O_0} & \dots & \bar{e}_K \times \bar{r}_{E/O_K} \\ O_{3\times 1} & \bar{e}_1 & \dots & \bar{e}_K \end{bmatrix} \\
\mathbb{f} &= \begin{bmatrix} \bar{F}_e \\ \bar{\tau}_e \end{bmatrix}
\end{aligned}
$$

The matrix $\mathbb{J}$ is refered to as the Jacobian. And $\mathbb{f}$ is the wrench (formal term to refer to a force/torque couple). This whole theory was assuming a single serial chain of the robot with a single end-effector. For the case of krang, we will have two wrenches $\mathbb{f}_{el}$ and $\mathbb{f}_{er}$ applied at two end-effectors on the right and the left arms respectively. The points on which this wrench is being sensed on the two end-effectors is $E_L$ and $E_R$. So the contribution becomes:

$$\mathbb{J}_L^T \mathbb{f}_{el} + \mathbb{J}_R^T \mathbb{f}_{er} \tag{56}$$

where

- $\mathbb{J}_L = \begin{bmatrix} \bar{i}_0 & \bar{e}_0 \times \bar{r}_{E_L/O_0} & \bar{e}_1 \times \bar{r}_{E_L/O_1} & \bar{e}_2 \times \bar{r}_{E_L/O_2} & \bar{e}_3 \times \bar{r}_{E_L/O_3} & \bar{e}_{4l} \times \bar{r}_{E_L/O_{4l}} & \dots & \bar{e}_{10l} \times \bar{r}_{E_L/O_{10l}} & O_{3\times 7} \\ O_{3\times 1} & \bar{e}_0 & \bar{e}_1 & \bar{e}_2 & \bar{e}_3 & \bar{e}_{4l} & \dots & \bar{e}_{10l} & O_{3\times 7} \end{bmatrix}$
- $\mathbb{J}_R = \begin{bmatrix} \bar{i}_0 & \bar{e}_0 \times \bar{r}_{E_R/O_0} & \bar{e}_1 \times \bar{r}_{E_R/O_1} & \bar{e}_2 \times \bar{r}_{E_R/O_2} & \bar{e}_3 \times \bar{r}_{E_R/O_3} & O_{3\times 7} & \bar{e}_{4r} \times \bar{r}_{E_R/O_{4r}} & \dots & \bar{e}_{10r} \times \bar{r}_{E_R/O_{10r}} \\ O_{3\times 1} & \bar{e}_0 & \bar{e}_1 & \bar{e}_2 & \bar{e}_3 & O_{3\times 7} & \bar{e}_{4r} & \dots & \bar{e}_{10r} \end{bmatrix}$

## 6.4 Gravitational Forces

The contribution of the weight of a joint $k$ to the right-hand side of the equation corresponding to generalized speed $\dot{q}_j$ is

$$m_k \bar{g} \cdot \frac{\partial \bar{v}_{Gk}}{\partial \dot{q}_j} \tag{57}$$

Now

$$\begin{aligned} \bar{v}_{Gk} &= \bar{v}_k + \bar{\omega}_k \times \bar{r}_{Gk/O_k} \\ &= \bar{v}_{k-1} + \bar{\omega}_{k-1} \times r_{O_k/O_{k-1}} + \bar{\omega}_k \times \bar{r}_{Gk/O_k} \\ &= \bar{v}_0 + \sum_{i=1}^{k} \left( \bar{\omega}_{i-1} \times r_{O_i/O_{i-1}} \right) + \bar{\omega}_k \times \bar{r}_{Gk/O_k} \end{aligned}$$

For $\dot{q}_j = \dot{x}$ we have:

$$\begin{aligned} \frac{\partial \bar{v}_{Gk}}{\partial \dot{x}} &= \frac{\partial \bar{v}_0}{\partial \dot{x}} + \sum_{i=1}^{k} \left( \frac{\partial \bar{\omega}_{i-1}}{\partial \dot{x}} \times r_{O_i/O_{i-1}} \right) \\ &= \frac{\partial \dot{x} \bar{i}_0}{\partial \dot{x}} + \sum_{i=1}^{k} \left( 0 \times r_{O_i/O_{i-1}} \right) \\ &= \bar{i}_0 \end{aligned} \tag{58}$$

Now since $\bar{g} = -g\bar{k}_0$, we will have $m_k \bar{g} \cdot \frac{\partial \bar{v}_{Gk}}{\partial \dot{x}} = m_k g \bar{k}_0 \cdot \bar{i}_0 = 0$. So for $\dot{q}_j = \dot{x}$ the total contribution of all joint weights will be zero.

For $\dot{q}_j \neq \dot{x}$ we will have:

$$\begin{aligned} \frac{\partial \bar{v}_{Gk}}{\partial \dot{q}_j} &= \frac{\partial \bar{v}_0}{\partial \dot{q}_j} + \sum_{i=1}^{k} \left( \frac{\partial \bar{\omega}_{i-1}}{\partial \dot{q}_j} \times r_{O_i/O_{i-1}} \right) + \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} \times \bar{r}_{Gk/O_k} \\ &= 0 + \sum_{i=1}^{k} \left( \frac{\partial \bar{\omega}_{i-1}}{\partial \dot{q}_j} \times r_{O_i/O_{i-1}} \right) + \frac{\partial \bar{\omega}_k}{\partial \dot{q}_j} \times \bar{r}_{Gk/O_k} \\ &= \begin{cases} 0 & \text{if } k < j \\ \sum_{i=j+1}^{k} \left( \bar{e}_j \times r_{O_i/O_{i-1}} \right) + \bar{e}_j \times \bar{r}_{Gk/O_k} & \text{if } k \geq j \end{cases} \end{aligned}$$

$$= \begin{cases} 0 & \text{if } k < j \\ \bar{e}_j \times \left( \sum_{i=j+1}^{k} \bar{r}_{O_i/O_{i-1}} + \bar{r}_{Gk/O_k} \right) & \text{if } k \geq j \end{cases}$$

$$= \begin{cases} 0 & \text{if } k < j \\ \bar{e}_j \times \bar{r}_{Gk/O_j} & \text{if } k \geq j \end{cases} \tag{59}$$

The total contributions of all joints (for $\dot{q}_j \neq \dot{x}$) will therefore be:

$$\begin{aligned} &\sum_{k=1}^{K} \left( m_k \bar{g} \cdot \frac{\partial \bar{v}_{Gk}}{\partial \dot{q}_j} \right) \\ &= \bar{g} \cdot \left( \sum_{k=1}^{j-1} m_k \frac{\partial \bar{v}_{Gk}}{\partial \dot{q}_j} + \sum_{k=j}^{K} m_k \frac{\partial \bar{v}_{Gk}}{\partial \dot{q}_j} \right) \\ &= \bar{g} \cdot \left( \sum_{k=1}^{j-1} 0 + \sum_{k=j}^{K} m_k \left( e_j \times \bar{r}_{Gk/O_j} \right) \right) \\ &= \bar{g} \cdot \left( e_j \times \sum_{k=j}^{K} m_k \bar{r}_{Gk/O_j} \right) \\ &= \left( \sum_{k=j}^{K} m_k \bar{r}_{Gk/O_j} \times \bar{g} \right) \cdot e_j \\ &= \sum_{k=j}^{K} \left( \bar{r}_{Gk/O_j} \times m_k \bar{g} \right) \cdot e_j \end{aligned} \tag{60}$$

Note that this derivation is assuming a single serial chain. In the case of Krang however, if $j$ corresponds to a speed of joint before the branching takes place i.e. if $j \in \{imu, w, torso\}$ the range of summation will include all joints in the tree above the current joint. If $j$ corresponds to the speed of joint in one of the branches i.e. if $j \in \{1l, ..., 7l\}$ or $j \in \{1r, ..., 7r\}$ then the range of summation will only be consisting of joints following the current joint in the specific branch.

## 6.5 RHS Final Expressions

If we combine the contributions of all forces to the right hand side of the Kane's equation, this is what we will get:

$$
\begin{array}{rrcrcrcrcrcr}
 & \text{Wheel/Joint Torques} & & \text{Left EE Force} & & \text{Left EE Torque} & & \text{Right EE Force} & & \text{Right EE Torque} & & \text{Gravitational Forces} \\
\dot{x}: & \frac{1}{R}(\tau_L+\tau_R) & + & \bar{F}_{el}\cdot\bar{i}_0 & + & 0 & + & \bar{F}_{er}\cdot\bar{i}_0 & + & 0 & + & 0 \\
\dot{\psi}: & \frac{L}{2R}(\tau_L-\tau_R) & + & \left(\bar{r}_{E_L/O_0}\times\bar{F}_{el}\right)\cdot\bar{e}_0 & + & \bar{\tau}_{el}\cdot\bar{e}_0 & + & \left(\bar{r}_{E_R/O_0}\times\bar{F}_{er}\right)\cdot\bar{e}_0 & + & \bar{\tau}_{er}\cdot\bar{e}_0 & + & \sum\limits_{k\in\{imu,w,torso\}\cup\{4l,\ldots,10l\}\cup\{4r,\ldots,10r\}}\left(\bar{r}_{Gk/O_0}\times m_k\bar{g}\right)\cdot\bar{e}_0 \\
\dot{q}_{imu}: & -(\tau_L+\tau_R) & + & \left(\bar{r}_{E_L/O_1}\times\bar{F}_{er}\right)\cdot\bar{e}_1 & + & \bar{\tau}_{el}\cdot\bar{e}_1 & + & \left(\bar{r}_{E_R/O_1}\times\bar{F}_{er}\right)\cdot\bar{e}_1 & + & \bar{\tau}_{er}\cdot\bar{e}_1 & + & \sum\limits_{k\in\{imu,w,torso\}\cup\{4l,\ldots,10l\}\cup\{4r,\ldots,10r\}}\left(\bar{r}_{Gk/O_1}\times m_k\bar{g}\right)\cdot\bar{e}_1 \\
\dot{q}_w: & \tau_w & + & \left(\bar{r}_{E_R/O_2}\times\bar{F}_{er}\right)\cdot\bar{e}_2 & + & \bar{\tau}_{el}\cdot\bar{e}_2 & + & \left(\bar{r}_{E_R/O_2}\times\bar{F}_{er}\right)\cdot\bar{e}_2 & + & \bar{\tau}_{er}\cdot\bar{e}_2 & + & \sum\limits_{k\in\{w,torso\}\cup\{4l,\ldots,10l\}\cup\{4r,\ldots,10r\}}\left(\bar{r}_{Gk/O_2}\times m_k\bar{g}\right)\cdot\bar{e}_2 \\
\dot{q}_{torso}: & \tau_{torso} & + & \left(\bar{r}_{E_L/O_3}\times\bar{F}_{el}\right)\cdot\bar{e}_3 & + & \bar{\tau}_{er}\cdot\bar{e}_3 & + & \left(\bar{r}_{E_R/O_3}\times\bar{F}_{er}\right)\cdot\bar{e}_3 & + & \bar{\tau}_{el}\cdot\bar{e}_3 & + & \sum\limits_{k\in\{torso\}\cup\{4l,\ldots,10l\}\cup\{4r,\ldots,10r\}}\left(\bar{r}_{Gk/O_3}\times m_k\bar{g}\right)\cdot\bar{e}_3 \\
\dot{q}_{1l}: & \tau_{1l} & + & \left(\bar{r}_{E_L/O_{4l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{4l} & + & \bar{\tau}_{el}\cdot\bar{e}_{4l} & + & 0 & + & 0 & + & \sum\limits_{k\in\{4l,\ldots,10l\}}\left(\bar{r}_{Gk/O_{4l}}\times m_k\bar{g}\right)\cdot\bar{e}_{4l} \\
\dot{q}_{2l}: & \tau_{2l} & + & \left(\bar{r}_{E_L/O_{5l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{5l} & + & \bar{\tau}_{el}\cdot\bar{e}_{5l} & + & 0 & + & 0 & + & \sum\limits_{k\in\{5l,\ldots,10l\}}\left(\bar{r}_{Gk/O_{5l}}\times m_k\bar{g}\right)\cdot\bar{e}_{5l} \\
\dot{q}_{3l}: & \tau_{3l} & + & \left(\bar{r}_{E_L/O_{6l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{6l} & + & \bar{\tau}_{el}\cdot\bar{e}_{6l} & + & 0 & + & 0 & + & \sum\limits_{k\in\{6l,\ldots,10l\}}\left(\bar{r}_{Gk/O_{6l}}\times m_k\bar{g}\right)\cdot\bar{e}_{6l} \\
\dot{q}_{4l}: & \tau_{4l} & + & \left(\bar{r}_{E_L/O_{7l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{7l} & + & \bar{\tau}_{el}\cdot\bar{e}_{7l} & + & 0 & + & 0 & + & \sum\limits_{k\in\{7l,\ldots,10l\}}\left(\bar{r}_{Gk/O_{7l}}\times m_k\bar{g}\right)\cdot\bar{e}_{7l} \\
\dot{q}_{5l}: & \tau_{5l} & + & \left(\bar{r}_{E_L/O_{8l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{8l} & + & \bar{\tau}_{el}\cdot\bar{e}_{8l} & + & 0 & + & 0 & + & \sum\limits_{k\in\{8l,9l,10l\}}\left(\bar{r}_{Gk/O_{8l}}\times m_k\bar{g}\right)\cdot\bar{e}_{8l} \\
\dot{q}_{6l}: & \tau_{6l} & + & \left(\bar{r}_{E_L/O_{9l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{9l} & + & \bar{\tau}_{el}\cdot\bar{e}_{9l} & + & 0 & + & 0 & + & \sum\limits_{k\in\{9l,10l\}}\left(\bar{r}_{Gk/O_{9l}}\times m_k\bar{g}\right)\cdot\bar{e}_{9l} \\
\dot{q}_{7l}: & \tau_{7l} & + & \left(\bar{r}_{E_L/O_{10l}}\times\bar{F}_{el}\right)\cdot\bar{e}_{10l} & + & \bar{\tau}_{el}\cdot\bar{e}_{10l} & + & 0 & + & 0 & + & \left(\bar{r}_{G\,10l/O_{10l}}\times m_{10l}\bar{g}\right)\cdot\bar{e}_{10l} \\
\dot{q}_{1r}: & \tau_{1r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{4r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{4r} & + & \bar{\tau}_{er}\cdot\bar{e}_{4r} & + & \sum\limits_{k\in\{4r,\ldots,10r\}}\left(\bar{r}_{Gk/O_{4l}}\times m_k\bar{g}\right)\cdot\bar{e}_{4r} \\
\dot{q}_{2r}: & \tau_{2r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{5r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{5r} & + & \bar{\tau}_{er}\cdot\bar{e}_{5r} & + & \sum\limits_{k\in\{5r,\ldots,10r\}}\left(\bar{r}_{Gk/O_{5r}}\times m_k\bar{g}\right)\cdot\bar{e}_{5r} \\
\dot{q}_{3r}: & \tau_{3r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{6r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{6r} & + & \bar{\tau}_{er}\cdot\bar{e}_{6r} & + & \sum\limits_{k\in\{6r,\ldots,10r\}}\left(\bar{r}_{Gk/O_{6r}}\times m_k\bar{g}\right)\cdot\bar{e}_{6r} \\
\dot{q}_{4r}: & \tau_{4r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{7r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{7r} & + & \bar{\tau}_{er}\cdot\bar{e}_{7r} & + & \sum\limits_{k\in\{7r,\ldots,10r\}}\left(\bar{r}_{Gk/O_{7r}}\times m_k\bar{g}\right)\cdot\bar{e}_{7r} \\
\dot{q}_{5r}: & \tau_{5r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{8r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{8r} & + & \bar{\tau}_{er}\cdot\bar{e}_{8r} & + & \sum\limits_{k\in\{8r,9r,10r\}}\left(\bar{r}_{Gk/O_{8r}}\times m_k\bar{g}\right)\cdot\bar{e}_{8r} \\
\dot{q}_{6r}: & \tau_{6r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{9r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{9r} & + & \bar{\tau}_{er}\cdot\bar{e}_{9r} & + & \sum\limits_{k\in\{9r,10r\}}\left(\bar{r}_{Gk/O_{9r}}\times m_k\bar{g}\right)\cdot\bar{e}_{9r} \\
\dot{q}_{7r}: & \tau_{7r} & + & 0 & + & 0 & + & \left(\bar{r}_{E_R/O_{10r}}\times\bar{F}_{er}\right)\cdot\bar{e}_{10r} & + & \bar{\tau}_{er}\cdot\bar{e}_{10r} & + & \left(\bar{r}_{G\,10r/O_{10r}}\times m_{10r}\bar{g}\right)\cdot\bar{e}_{10r}
\end{array}
\tag{61}
$$

# 7 MATLAB code For Finding the Dynamic Model

The dynamic model is generated using a script `dynamicModel.m` found in the folder *stableForceInteraction/Implementation/1-ForceControlWhileBalancing/1-ControlProblem1/1-DynamicModeling/2-DynamicModelOfFullRobot/2-matlab/Kanes*. The script populates the frame information in a map container using `getKrangFrames()`. Then it calculates Kane's left-hand side contribution of the wheels using the function `kanesLHSWheels()`. This function uses eq.25 to calculate the expressions for LHS contributions of the wheels. The result of these contributions is stored in the vector `KKw`, a $19 \times 1$ vector of symbolic objects representing the contribution to the LHS of each of the nineteen equations by the two wheels. The LHS contribution by the tree-structure is then evaluated using the funciton `kanesLHSTree()`. This funciton uses eq.39 to evaluate its expression. The result is stored in vector `KKTree`, a $19 \times 1$ vector of symbolic objects representing the contribution to the LHS of each of the nineteen equations by the 17 bodies in the tree structure. The contribution of each body is also stored separately in the second output `KHist`, a $19 \times 17$ matrix with each row corresponding to each equation and each column to each body in tree structure. The sum of vectors `KKw` and `KKTree` represents the final expression for Kane's LHS. This is fed as an input argument to function `getAC()` that generates the matrices **A** and **C** stored in `AA` and `CC` respectively. The variables `f`, `KKw`, `KKTree`, `KHist`, `AA` and `CC` are stored in the MAT file `LHS.mat`. Finally the function `KanesRHS()` generates the right-hand side expressions for Kane's eqautions using eq. 45, 50, 52 and 57. The result is stored in the $19 \times 1$ vector `K`. Other outputs of this function include: I) `KK`: A $19 \times 4$ matrix with each column containing contributions from wheel torques (column 1), joint torques (column 2), end effector forces/torques (column 3) and gravitational forces (column 4) respectively. Sum across all columns is stored in `K`. II) `KK2`: A $19 \times 16$ matrix with each column containing contribution from each of the 16 motors in the tree structure. Sum across all these columns is stored in `KK`'s second column. III) `KK3`: A $19 \times 4$ matrix with each column containing contributions from $\bar{F}_{el}$, $\bar{\tau}_{el}$, $\bar{F}_{er}$ and $\bar{\tau}_{er}$ respectively. The sum across all these columns is stored in `KK`'s third column. IV) `KK4`: A $19 \times 17$ matrix with each column containing the contribution by the weight of each body of the (17-link-long) tree structure. The sum across its columns is stored in the `KK`'s fourth column. The MAT-file `RHS.mat` contains the results of evaluation of the variables `K`, `KK`, `KK2`, `KK3` and `KK4`.

Another script `dynamicModel2.m` found in the same folder uses the explicit forms we evaluated in this report to calculate the same dynamic model. It has the same function and output variables as described above with an underscore added to each (e.g. `kanesLHSWheels_` and `KKw_`) to differentiate from those defined in the last paragraph. The equations used by the functions in this file are eq. 28 and 44 for the evaluation of LHS contributions from wheels and tree-structure respectively, and eq. 47, 51, 55 and 60 for the evaluation of RHS contributions from wheel torques, joint torques, end-effector forces/torques and gravitational forces respectively.

When the results of the outputs of the two scripts was compared, everything matched exactly, giving some form of certainty regarding the correctness of the work.

## 7.1 Map Container for all the Frame Information

The function `getKrangFrames()` populates the information in a map container `f`. A map container is a data structure in MATLAB that stores a list of data that is retrievable using a key. We store a `frame` structure in each cell of the map and use strings $s \in \mathbf{S} = \{$ `'0'`, `'1'`, `'2'`, `'3'`, `'4l'`, `'5l'`, `'6l'`, `'7l'`, `'8l'`, `'9l'`, `'10l'`, `'4r'`, `'5r'`, `'6r'`, `'7r'`, `'8r'`, `'9r'`, `'10r'` $\}$ as a key to retrieve information. The `frame` structure stores the following elements:

- `x` the unit vector along $x$-axis represented in the antecedent frame
- `y` the unit vector along $y$-axis represented in the antecedent frame
- `z` the unit vector along $z$-axis represented in the antecedent frame
- `P` the position of the origin frame represented in the antecedent frame
- `e` the unit vector along direction of positive rotation of the frame represented in the local frame
- `a` the string $\in \mathbf{S}$ (defined above) that is the key that maps to the antecedent frame
- `q` the symbolic variable used for representing the generalized position $q$ associated with this frame
- `dq` the symbolic variable used for representing the generalized speed $\dot{q}$ associated with this frame
- `ddq` the symbolic variable used for representing the generalized acceleration $\ddot{q}$ associated with this frame

`o` the row number in the inertia matrix $\mathbf{A}$ (i.e. in the final dynamic model $\mathbf{A}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{Q} = \mathbf{F}$) that corresponds to the current joint

`param` array of ten symbolic variables used to represent the inertial parameters of the joint $[m \quad \mathbf{MX} \quad \mathbf{MY} \quad \mathbf{MZ} \quad \mathbf{XX} \quad \mathbf{YY} \quad \mathbf{ZZ} \quad \mathbf{XY} \quad \mathbf{XZ} \quad \mathbf{YZ}]^T$

## 7.2 Functions associated with the Map Container

There are a number of functions that take the map container `f` as the input argument and construct a useful information as an output. Here is a list of those functions:

`isBefore(f, key1, key2)` returns `true` if frame 1 (identified by `key1`) is before frame 2 (identified by `key2`) in the tree structure of the robot

`Rot(f, key1, key2)` returns rotation transform ${}^jA_i$ i.e. represention of frame i (identified by `key1`) in frame j (identified by `key2`)

`Tf(f, key1, key2)` returns rotation transform ${}^jT_i$ i.e. represention of frame i (identified by `key1`) in frame j (identified by `key2`)

`qVec(f)` generates the vector $\mathbf{q}$ containing generalized positions of all frames

`dqVec(f)` generates the vector $\dot{\mathbf{q}}$ containing generalized speeds of all frames

`ddqVec(f)` generates the vector $\ddot{\mathbf{q}}$ containing generalized accelerations of all frames

`mass(f, key)` returns the mass of the frame identified by the `key`

`mCOM(f, key)` returns the mass times COM ($\mathbf{MS} = [\mathbf{MX} \quad \mathbf{MY} \quad \mathbf{MZ}]^T$) of the joint identified by the `key` represented in the local frame

`inertiaMat(f, key)` returns the inertia matrix ($\mathbf{J} = \begin{bmatrix} \mathbf{XX} & \mathbf{XY} & \mathbf{XZ} \\ \mathbf{XY} & \mathbf{YY} & \mathbf{YZ} \\ \mathbf{XZ} & \mathbf{YZ} & \mathbf{ZZ} \end{bmatrix}$) of the joint indentified by `key` represented in the local frame

`angVel(f, key)` returns the symbolic expression for the angular velocity ($\bar{\omega}$) of the joint represented in local frame calculated recursively using eq. 29

`angAcc(f, key)` returns the symbolic expression for the angular acceleration ($\bar{\alpha}$) of the joint represented in local frame calculated recursively using eq. 30

`linVel(f, key)` returns the symbolic expression for the linear velocity ($\bar{v}$) of the joint represented in local frame calculated recursively using eq. 31

`linAcc(f, key)` returns the symbolic expression for the linear acceleration ($\bar{a}$) of the joint represented in local frame calculated recursively using eq. 32

# 8 Conclusion

In this report we evaluated the dynamic model of the robot Golem Krang. We used Kane's formulation for the task because of the use of quasi-velocity in our generalized velocity set that prevents us from using Lagrange formulation. The dynamic model can be used in designing of control systems and simulation of the robot. An accurate simulation will require a few steps that are not performed in this report: The accurate estimation of masses, center of masses, and inertial parameters of each link, accurate modeling of the current-torque relationship of the joint motors (prior experience tells us that the relationship is highly non-linear) and the role of friction in the dynamic model. These are challenging tasks yet necesssary if near-accurate numerical estimation of the dynamic model is to be obtained.

# A Code Listings

We present the code listings of the various functions in this section.

## A.1 Script1

## A.2 `dynamicModel.m`

```
nEqs = 19;
f = getKrangFrames(nEqs-1);
KKw = kanesLHSWheels(f);
[KKTree, KHist] = kanesLHSTree(f);
[AA, CC] = getAC(f, KKw + KKTree);
[K, KK, KK2, KK3, KK4] = kanesRHS(f);
```

## A.3 `getKrangFrames()`

```
function f = getKrangFrames(nFrames)

% This function generates a map. The keys to the maps are string literals
% in {'1', '2', '3', '4l', '5l', ... '10l', '4r', '5r', ... '10r'}
% and the values are structs that have members associated with the
% respective joints of the robot.
% 'x', 'y', 'z', 'P' define the unit vectors and position of origin of the
% frame represented in the antecedent frame
% 'e' is the local angular speed of the frame represented in the same frame
% 'a' contains the key to the antecedent frame
% 'param' contains the inertial parameters of the respective link
% 'q','dq' and 'ddq' contain associated joint pos, vel and acc variables
% 'o' defines the row of inertia matrix A (in the dynamic equations) that
% corresponds to the current joint.

syms x psii q_imu q_w q_torso q_1l q_2l q_3l q_4l q_5l q_6l q_7l real
syms q_1r q_2r q_3r q_4r q_5r q_6r q_7r L1 L2 L3 L4 L5 L6 L7 L8 L9 real
syms dx dpsi dq_imu dq_w dq_torso dq_1l dq_2l dq_3l dq_4l dq_5l dq_6l dq_7l real
syms dq_1r dq_2r dq_3r dq_4r dq_5r dq_6r dq_7r L1 L2 L3 L4 L5 L6 L7 L8 L9 real
syms ddx ddpsi ddq_imu ddq_w ddq_torso ddq_1l ddq_2l ddq_3l ddq_4l ddq_5l ddq_6l ddq_7l real
syms ddq_1r ddq_2r ddq_3r ddq_4r ddq_5r ddq_6r ddq_7r real
syms m_1 MX_1 MY_1 MZ_1 XX_1 XY_1 XZ_1 YY_1 YZ_1 ZZ_1 real
syms m_2 MX_2 MY_2 MZ_2 XX_2 XY_2 XZ_2 YY_2 YZ_2 ZZ_2 real
syms m_3 MX_3 MY_3 MZ_3 XX_3 XY_3 XZ_3 YY_3 YZ_3 ZZ_3 real
syms m_4l MX_4l MY_4l MZ_4l XX_4l XY_4l XZ_4l YY_4l YZ_4l ZZ_4l real
syms m_5l MX_5l MY_5l MZ_5l XX_5l XY_5l XZ_5l YY_5l YZ_5l ZZ_5l real
syms m_6l MX_6l MY_6l MZ_6l XX_6l XY_6l XZ_6l YY_6l YZ_6l ZZ_6l real
syms m_7l MX_7l MY_7l MZ_7l XX_7l XY_7l XZ_7l YY_7l YZ_7l ZZ_7l real
syms m_8l MX_8l MY_8l MZ_8l XX_8l XY_8l XZ_8l YY_8l YZ_8l ZZ_8l real
syms m_9l MX_9l MY_9l MZ_9l XX_9l XY_9l XZ_9l YY_9l YZ_9l ZZ_9l real
syms m_10l MX_10l MY_10l MZ_10l XX_10l XY_10l XZ_10l YY_10l YZ_10l ZZ_10l real
syms m_4r MX_4r MY_4r MZ_4r XX_4r XY_4r XZ_4r YY_4r YZ_4r ZZ_4r real
syms m_5r MX_5r MY_5r MZ_5r XX_5r XY_5r XZ_5r YY_5r YZ_5r ZZ_5r real
syms m_6r MX_6r MY_6r MZ_6r XX_6r XY_6r XZ_6r YY_6r YZ_6r ZZ_6r real
syms m_7r MX_7r MY_7r MZ_7r XX_7r XY_7r XZ_7r YY_7r YZ_7r ZZ_7r real
syms m_8r MX_8r MY_8r MZ_8r XX_8r XY_8r XZ_8r YY_8r YZ_8r ZZ_8r real
syms m_9r MX_9r MY_9r MZ_9r XX_9r XY_9r XZ_9r YY_9r YZ_9r ZZ_9r real
syms m_10r MX_10r MY_10r MZ_10r XX_10r XY_10r XZ_10r YY_10r YZ_10r ZZ_10r real

nullSym = sym([0; 0; 0]); nullSymParam = sym(zeros(10,1));
```

```
frame.x = nullSym; frame.y = nullSym; frame.z = nullSym; frame.P = nullSym;
frame.e = nullSym; frame.a = ''; frame.param = nullSymParam;
frame.q = sym(0); frame.dq = sym(0); frame.o = 0;
frame.angVel = sym(0); frame.linVel = sym(0);
frame.angAcc = sym(0); frame.linAcc = sym(0);
frame.gotAngVel = false; frame.gotLinVel = false;
frame.gotAngAcc = false; frame.gotLinAcc = false;

% Frame 0
frame.x = sym([1; 0; 0]); frame.y = sym([0; 1; 0]);
frame.z = sym([0; 0; 1]); frame.P = sym([0; 0; 0]);
frame.e = [0; 0; 1]; frame.a = '-1';
frame.q = psii; frame.dq = dpsi; frame.ddq = ddpsi; frame.o = 2;
frame.param = nullSymParam;
frame.angVel = [0; 0; dpsi];
frame.linVel = [dx; 0; 0];
frame.angAcc = [0; 0; ddpsi];
frame.linAcc = [ddx; 0; 0] + cross(frame.angVel, [1; 0; 0]);
frame.gotAngVel = true; frame.gotLinVel = true;
frame.gotAngAcc = true; frame.gotLinAcc = true;
F{1} = frame;

% Frame 1
frame.x = sym([0; -1; 0]); frame.y = [sin(q_imu); 0; cos(q_imu)];
frame.z = [-cos(q_imu); 0; sin(q_imu)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0]; frame.a = '0';
frame.q = q_imu; frame.dq = dq_imu; frame.ddq = ddq_imu; frame.o = 3;
frame.param = [m_1 MX_1 MY_1 MZ_1 XX_1 XY_1 XZ_1 YY_1 YZ_1 ZZ_1];
frame.angVel = sym(0); frame.linVel = sym(0);
frame.angAcc = sym(0); frame.linAcc = sym(0);
frame.gotAngVel = false; frame.gotLinVel = false;
frame.gotAngAcc = false; frame.gotLinAcc = false;
F{2} = frame;

% Frame 2
frame.x = sym([1; 0; 0]); frame.y = [0; cos(q_w); -sin(q_w)];
frame.z = [0; sin(q_w); cos(q_w)]; frame.P = [0; L1; -L2];
frame.e = [-1; 0; 0];  frame.a = '1';
frame.q = q_w; frame.dq = dq_w; frame.ddq = ddq_w; frame.o = 4;
frame.param = [m_2 MX_2 MY_2 MZ_2 XX_2 XY_2 XZ_2 YY_2 YZ_2 ZZ_2];
F{3} = frame;

% Frame 3
frame.x = [-cos(q_torso); 0; -sin(q_torso)]; frame.y = sym([0; 1; 0]);
frame.z = [sin(q_torso); 0; -cos(q_torso)]; frame.P = [0; L3; L4];
frame.e = [0; -1; 0];  frame.a = '2';
frame.q = q_torso; frame.dq = dq_torso; frame.ddq = ddq_torso; frame.o = 5;
frame.param = [m_3 MX_3 MY_3 MZ_3 XX_3 XY_3 XZ_3 YY_3 YZ_3 ZZ_3];
F{4} = frame;

% Frame 4l
frame.x = [0; cos(q_1l); -sin(q_1l)]; frame.y = sym([1; 0; 0]);
frame.z = [0; -sin(q_1l); -cos(q_1l)]; frame.P = sym([L6; L5; 0]);
frame.e = [0; -1; 0];  frame.a = '3';
frame.q = q_1l; frame.dq = dq_1l; frame.ddq = ddq_1l; frame.o = 6;
frame.param = [m_4l MX_4l MY_4l MZ_4l XX_4l XY_4l XZ_4l YY_4l YZ_4l ZZ_4l];
F{5} = frame;

% Frame 5l
frame.x = sym([-1; 0; 0]); frame.y = [0; -cos(q_2l); -sin(q_2l)];
frame.z = [0; -sin(q_2l); cos(q_2l)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0];  frame.a = '4l';
frame.q = q_2l; frame.dq = dq_2l; frame.ddq = ddq_2l; frame.o = 7;
frame.param = [m_5l MX_5l MY_5l MZ_5l XX_5l XY_5l XZ_5l YY_5l YZ_5l ZZ_5l];
F{6} = frame;

% Frame 6l
frame.x = [-cos(q_3l); 0; sin(q_3l)]; frame.y = sym([0; -1; 0]);
frame.z = [sin(q_3l); 0; cos(q_3l)]; frame.P = sym([0; -L7; 0]);
frame.e = [0; -1; 0];  frame.a = '5l'; frame.o = 8;
frame.q = q_3l; frame.dq = dq_3l; frame.ddq = ddq_3l;
frame.param = [m_6l MX_6l MY_6l MZ_6l XX_6l XY_6l XZ_6l YY_6l YZ_6l ZZ_6l];
F{7} = frame;

% Frame 7l
frame.x = sym([-1; 0; 0]); frame.y = [0; -cos(q_4l); -sin(q_4l)];
frame.z = [0; -sin(q_4l); cos(q_4l)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0];  frame.a = '6l'; frame.o = 9;
frame.q = q_4l; frame.dq = dq_4l; frame.ddq = ddq_4l;
frame.param = [m_7l MX_7l MY_7l MZ_7l XX_7l XY_7l XZ_7l YY_7l YZ_7l ZZ_7l];
F{8} = frame;

% Frame 8l
frame.x = [-cos(q_5l); 0; sin(q_5l)]; frame.y = sym([0; -1; 0]);
frame.z = [sin(q_5l); 0; cos(q_5l)]; frame.P = sym([0; -L8; 0]);
frame.e = [0; -1; 0];  frame.a = '7l'; frame.o = 10;
frame.q = q_5l; frame.dq = dq_5l; frame.ddq = ddq_5l;
frame.param = [m_8l MX_8l MY_8l MZ_8l XX_8l XY_8l XZ_8l YY_8l YZ_8l ZZ_8l];
F{9} = frame;

% Frame 9l
frame.x = sym([-1; 0; 0]); frame.y = [0; -cos(q_6l); -sin(q_6l)];
frame.z = [0; -sin(q_6l); cos(q_6l)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0];  frame.a = '8l'; frame.o = 11;
frame.q = q_6l; frame.dq = dq_6l; frame.ddq = ddq_6l;
frame.param = [m_9l MX_9l MY_9l MZ_9l XX_9l XY_9l XZ_9l YY_9l YZ_9l ZZ_9l];
F{10} = frame;

% Frame 10l
```

```
frame.x = [-cos(q_7l); 0; sin(q_7l)]; frame.y = [-sin(q_7l); 0; -cos(q_7l)];
frame.z = sym([0; -1; 0]); frame.P = [0; -L9; 0];
frame.e = [0; 0; -1];  frame.a = '9l';
frame.q = q_7l; frame.dq = dq_7l; frame.ddq = ddq_7l; frame.o = 12;
frame.param = [m_10l MX_10l MY_10l MZ_10l XX_10l XY_10l XZ_10l YY_10l YZ_10l ZZ_10l];
F{11} = frame;

% Frame 4r
frame.x = [0; cos(q_1r); sin(q_1r)]; frame.y = sym([-1; 0; 0]);
frame.z = [0; -sin(q_1r); cos(q_1r)]; frame.P = [-L6; L5; 0];
frame.e = [0; -1; 0];  frame.a = '3';
frame.q = q_1r; frame.dq = dq_1r; frame.ddq = ddq_1r; frame.o = 13;
frame.param = [m_4r MX_4r MY_4r MZ_4r XX_4r XY_4r XZ_4r YY_4r YZ_4r ZZ_4r];
F{12} = frame;

% Frame 5r
frame.x = sym([-1; 0; 0]); frame.y = [0; -cos(q_2r); -sin(q_2r)];
frame.z = [0; -sin(q_2r); cos(q_2r)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0];  frame.a = '4r';
frame.q = q_2r; frame.dq = dq_2r; frame.ddq = ddq_2r; frame.o = 14;
frame.param = [m_5r MX_5r MY_5r MZ_5r XX_5r XY_5r XZ_5r YY_5r YZ_5r ZZ_5r];
F{13} = frame;

% Frame 6r
frame.x = [-cos(q_3r); 0; sin(q_3r)]; frame.y = sym([0; -1; 0]);
frame.z = [sin(q_3r); 0; cos(q_3r)]; frame.P = sym([0; -L7; 0]);
frame.e = [0; -1; 0];  frame.a = '5r';
frame.q = q_3r; frame.dq = dq_3r; frame.ddq = ddq_3r; frame.o = 15;
frame.param = [m_6r MX_6r MY_6r MZ_6r XX_6r XY_6r XZ_6r YY_6r YZ_6r ZZ_6r];
F{14} = frame;

% Frame 7r
frame.x = sym([-1; 0; 0]); frame.y = [0; -cos(q_4r); -sin(q_4r)];
frame.z = [0; -sin(q_4r); cos(q_4r)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0];  frame.a = '6r'; frame.o = 14;
frame.q = q_4r; frame.dq = dq_4r; frame.ddq = ddq_4r; frame.o = 16;
frame.param = [m_7r MX_7r MY_7r MZ_7r XX_7r XY_7r XZ_7r YY_7r YZ_7r ZZ_7r];
F{15} = frame;

% Frame 8r
frame.x = [-cos(q_5r); 0; sin(q_5r)]; frame.y = sym([0; -1; 0]);
frame.z = [sin(q_5r); 0; cos(q_5r)]; frame.P = sym([0; -L8; 0]);
frame.e = [0; -1; 0];  frame.a = '7r'; frame.o = 15;
frame.q = q_5r; frame.dq = dq_5r; frame.ddq = ddq_5r; frame.o = 17;
frame.param = [m_8r MX_8r MY_8r MZ_8r XX_8r XY_8r XZ_8r YY_8r YZ_8r ZZ_8r];
F{16} = frame;

% Frame 9r
frame.x = sym([-1; 0; 0]); frame.y = [0; -cos(q_6r); -sin(q_6r)];
frame.z = [0; -sin(q_6r); cos(q_6r)]; frame.P = sym([0; 0; 0]);
frame.e = [-1; 0; 0];  frame.a = '8r'; frame.o = 16;
frame.q = q_6r; frame.dq = dq_6r; frame.ddq = ddq_6r; frame.o = 18;
frame.param = [m_9r MX_9r MY_9r MZ_9r XX_9r XY_9r XZ_9r YY_9r YZ_9r ZZ_9r];
F{17} = frame;

% Frame 10r
frame.x = [-cos(q_7r); 0; sin(q_7r)]; frame.y = [-sin(q_7r); 0; -cos(q_7r)];
frame.z = sym([0; -1; 0]); frame.P = [0; -L9; 0];
frame.e = [0; 0; -1];  frame.a = '9r';
frame.q = q_7r; frame.dq = dq_7r; frame.ddq = ddq_7r; frame.o = 19;
frame.param = [m_10r MX_10r MY_10r MZ_10r XX_10r XY_10r XZ_10r YY_10r YZ_10r ZZ_10r];
F{18} = frame;

keys = {'0', '1', '2', '3', '4l', '5l', '6l', '7l', '8l', ...
    '9l', '10l', '4r', '5r', '6r', '7r', '8r', '9r', '10r'};

for i=1:nFrames; keySet{i} = keys{i}; frames{i} = F{i}; end

f = containers.Map( keySet, frames );
```

## A.4 kanesLHSWeels()

```
function KK = kanesLHSWheels(f)

syms x psii phi dpsi dphi dx ddpsi ddphi ddx L g mw mb real
syms XXw XYw XZw YYw YZw ZZw real
syms XXb XYb XZb YYb YZb ZZb real
syms tau_R tau_L R qimu_plus_phi qimu real
syms MXb MYb MZb real

syms t X(t) PSI(t) PHI(t) dX dPSI dPHI ddX ddPSI ddPHI real
dX=diff(X,t);dPSI=diff(PSI,t);dPHI=diff(PHI,t);
ddX=diff(dX,t);ddPSI=diff(dPSI,t);ddPHI=diff(dPHI,t);
q = [x psii phi]'; dq = [dx dpsi dphi]'; ddq = [ddx ddpsi ddphi]';

mydiff = @(H) formula(subs(diff(symfun(subs(H,...
    [x,psii,phi,dx,dpsi,dphi,ddx,ddpsi,ddphi],...
    [X, PSI,PHI,dX,dPSI,dPHI,ddX,ddPSI,ddPHI]),t),t),...
    [X, PSI,PHI,dX,dPSI,dPHI,ddX,ddPSI,ddPHI],...
    [x,psii,phi,dx,dpsi,dphi,ddx,ddpsi,ddphi]));

q_imu = -phi + qimu_plus_phi; % qimu_plus_phi is a constant if pose fixed
```

```
%% Left Wheel

thetaL = x/R - psii*L/(2*R);
dthetaL = dx/R - dpsi*L/(2*R);
iL = [1 0 0]'; jL = [0 1 0]'; kL = [0 0 1]';
i0 = [cos(thetaL) 0 sin(thetaL)]'; j0 = [0 1 0]'; k0 = [-sin(thetaL) 0 cos(thetaL)]';
w0 = dpsi*k0;
v0 = dx*i0;
alpha0 = ddpsi*k0;
a0 = ddx*i0 + dx*(cross(dpsi*k0,i0));
rOL = (L/2)*j0;
Iw=[ZZw 0 0;0 YYw 0;0 0 ZZw];
wL = w0 + dthetaL*j0;
vGL = v0 + cross(w0, rOL);
aGL = a0 + cross(alpha0, rOL) + cross(w0, cross(w0, rOL));
HGL = Iw*wL;
p = mydiff(HGL);
dHGL = p + cross(wL,HGL);

% %% Right Wheel


thetaR = x/R + psii*L/(2*R);
dthetaR = dx/R + dpsi*L/(2*R);
iR = [1 0 0]'; jR = [0 1 0]'; kR = [0 0 1]';
i0 = [cos(thetaR) 0 sin(thetaR)]'; j0 = [0 1 0]'; k0 = [-sin(thetaR) 0 cos(thetaR)]';
w0 = dpsi*k0;
v0 = dx*i0;
alpha0 = ddpsi*k0;
a0 = ddx*i0 + dx*(cross(dpsi*k0,i0));
rOR = -(L/2)*j0;
wR = w0 + dthetaR*j0;
vGR = v0 + cross(w0, rOR);
aGR = a0 + cross(alpha0, rOR) + cross(w0, cross(w0, rOR));
HGR = Iw*wR;
dHGR = mydiff(HGR)+cross(wR,HGR);
dq = dqVec(f); KK = sym(zeros(length(dq),1));
for i=1:2
    KK(i) = mw*aGL'*diff(vGL, dq(i)) + dHGL'*diff(wL, dq(i)) ...
          + mw*aGR'*diff(vGR, dq(i)) + dHGR'*diff(wR, dq(i));
end
```

## A.5 kanesLHSTree()

```
function [K, KHist] = kanesLHSTree(f)

syms g real

dq = dqVec(f);
K = sym(zeros(length(dq), 1));
KHist=sym(zeros(length(dq),length(dq)));

key = keys(f);
for i=1:length(key)

    % Do nothing for frame 0
    if(isequal(key{i}, '0')); continue; end

    % Kinematics and Inertials Params
    w=angVel(f, key{i});
    alpha=angAcc(f, key{i});
    v=linVel(f, key{i});
    a=linAcc(f, key{i});
    J=inertiaMat(f, key{i});
    m=mass(f, key{i});
    mS=mCOM(f, key{i});
    S=mS/m;

    % Inertial Forces and Torques
    vG = v + cross(w, S);
    maG = m*(a + cross(alpha, S) + cross(w, cross(w, S)));
    dHnew = cross(mS, a) + J*alpha + cross(w, J*w);

    % Kanes LHS contributions
    for j=1:length(dq)
        disp(['kanesF: i=', num2str(i), ', j=', num2str(j), ', key=', key{i}]);
        KHist(j,f(key{i}).o) = maG'*diff(v, dq(j)) + dHnew'*diff(w, dq(j));
        K(j) = K(j) + KHist(j,f(key{i}).o);
    end
end
```

## A.6 getAC()

```
function [A, C] = getAC(f, K)

dq = dqVec(f);
ddq = ddqVec(f);
```

```
n=length(dq);
A = sym(zeros(n,n)); C = sym(zeros(n,n));
for i=1:n
    for j=1:n
        A(i,j)=getcoeff(K(i),ddq(j),1);
        C(i,j)=getcoeff(K(i), dq(j),2)*dq(j);
        ccc = getcoeff(K(i),dq(j),1);
        C(i,j) = C(i,j)+ccc;
        for k=1:n
            disp(['getAC: i=', num2str(i), ', j=', num2str(j), ', k=', num2str(k)]);
            C(i,j) = C(i,j) - 0.5*(getcoeff(ccc,dq(k),1))*dq(k);
        end
    end
end

% A=simplify(A);
% C=simplify(C);
```

## A.7 kanesRHS().m

```
function [K, KK, KK2, KK3, KK4] = kanesRHS(f)

dq = dqVec(f);
KK2 = sym(zeros(length(dq), length(dq)-3)); % Stores the contribution of each joint torque (nRows=ndqs, nCols=nTorques)
KK3 = sym(zeros(length(dq), 4)); % Stores the contribution of 1. Fel 2. tae_el 3. Fer 4. tau_er
KK4 = sym(zeros(length(dq), length(dq)-2)); % Stores the contribution of each link weight (nRows=ndqs, nCols=nLinks)
KK = sym(zeros(length(dq),4)); % Store contribution sums of 1. wheel torques 2. joint torques 3. End Eff F/T 4. Weights
K = sym(zeros(length(dq),1));  % Store the sum of all contributions

% Wheel Torques

syms tau_L tau_R L R real
i0 = [1 0 0]'; j0 = [0 1 0]'; k0 = [0 0 1]';
w0 = f('0').angVel;
v0 = f('0').linVel;
theta_L = v0(1)/R - w0(3)*L/(2*R);
theta_R = v0(1)/R + w0(3)*L/(2*R);
wL = w0 + theta_L*j0;
wR = w0 + theta_R*j0;
w1 = simplify(Rot(f, '1', '0') * angVel(f, '1'));
Tau_L = tau_L*j0;
Tau_R = tau_R*j0;
Tau_imu = -(tau_L + tau_R)*j0;

for i=1:length(dq)
    KK(i, 1) = Tau_L'*diff(wL, dq(i)) + Tau_R'*diff(wR, dq(i)) ...
        + Tau_imu'*diff(w1, dq(i));
end

% Joint Torques
syms tau_w tau_torso real
syms tau_1l tau_2l tau_3l tau_4l tau_5l tau_6l tau_7l real
syms tau_1r tau_2r tau_3r tau_4r tau_5r tau_6r tau_7r  real
keysTau = {'2', '3', '4l', '5l', '6l', '7l', '8l', ...
    '9l', '10l', '4r', '5r', '6r', '7r', '8r', '9r', '10r'};
aTau = [tau_w tau_torso tau_1l tau_2l tau_3l tau_4l tau_5l tau_6l tau_7l...
    tau_1r tau_2r tau_3r tau_4r tau_5r tau_6r tau_7r]';
for i=1:length(dq)-3; keySet{i} = keysTau{i}; cTau{i} = aTau(i); end
mTau = containers.Map( keySet, cTau );

keysTau = keys(mTau);
for i=1:length(keysTau)
    key = keysTau{i};
    tau = mTau(key);
    frame = f(key);
    e = frame.e;
    w = angVel(f, key);
    R = Rot(f, key, frame.a);
    wlast = angVel(f, frame.a);

    % Calculate the contribution
    for j=1:length(dq)
        KK2(j, i) = tau*e'*diff(w,dq(j)) + (-R*tau*e)'*diff(wlast,dq(j));
        KK(j, 2) = KK(j, 2) + KK2(j, i);
    end
end

% Contribution of forces/torques on the end—effector

syms Flx Fly Flz Frx Fry Frz Tlx Tly Tlz Trx Try Trz real
Fl = [Flx Fly Flz]'; Tl = [Tlx Tly Tlz]'; % expressed in frame 10l
Fr = [Frx Fry Frz]'; Tr = [Trx Try Trz]'; % expressed in frame 10r
veL = linVel(f, '10l');
weL = angVel(f, '10l');
veR = linVel(f, '10r');
weR = angVel(f, '10r');

for i=1:length(dq)
    KK3(i, 1) = KK3(i, 1) + Fl'*diff(veL, dq(i));
    KK3(i, 2) = KK3(i, 2) + Tl'*diff(weL, dq(i));
    KK3(i, 3) = KK3(i, 3) + Fr'*diff(veR, dq(i));
    KK3(i, 4) = KK3(i, 4) + Tr'*diff(weR, dq(i));
    KK(i, 3) = KK3(i, 1) + KK3(i, 2) + KK3(i, 3) + KK3(i, 4);
```

```
end

% Contribution of link weights
syms g real
key = keys(f);
for i=1:length(key)

    % if key is '0' leave it
    if(isequal(key{i}, '0')); continue; end

    % Get necessary info
    w = angVel(f, key{i});
    v = linVel(f, key{i});
    m = mass(f, key{i});
    S = mCOM(f, key{i})/m;
    T = Tf(f, '0', key{i});

    % Calculate contribution
    vG = v + cross(w, S);
    ROT = T(1:3,1:3);
    mg = m*ROT*[0 0 -g]';
    for j=1:length(dq)
        KK4(j, i) =  mg'*diff(vG, dq(j));
        KK(j, 4) = KK(j, 4) + KK4(j, i);
    end
end

for j=1:length(dq)
    K(j) = KK(j, 1) + KK(j, 2) + KK(j, 3) + KK(j, 4);
end
```

## A.8 Script2

## A.9 dynamicModel2.m

```
nEqs = 19;
f = getKrangFrames(nEqs-1);
KKw = kanesLHSWheels(f);
[KKTree, KHist] = kanesLHSTree(f);
[AA, CC] = getAC(f, KKw + KKTree);
[K, KK, KK2, KK3, KK4] = kanesRHS(f);
```

## A.10 kanesLHSWeels_()

```
function KK = kanesLHSWheels(f)

syms x psii phi dpsi dphi dx ddpsi ddphi ddx L g mw mb real
syms XXw XYw XZw YYw YZw ZZw real
syms XXb XYb XZb YYb YZb ZZb real
syms tau_R tau_L R qimu_plus_phi qimu real
syms MXb MYb MZb real

syms t X(t) PSI(t) PHI(t) dX dPSI dPHI ddX ddPSI ddPHI real
dX=diff(X,t);dPSI=diff(PSI,t);dPHI=diff(PHI,t);
ddX=diff(dX,t);ddPSI=diff(dPSI,t);ddPHI=diff(dPHI,t);
q = [x psii phi]'; dq = [dx dpsi dphi]'; ddq = [ddx ddpsi ddphi]';

mydiff = @(H) formula(subs(diff(symfun(subs(H,...
    [x,psii,phi,dx,dpsi,dphi,ddx,ddpsi,ddphi],...
    [X, PSI,PHI,dX,dPSI,dPHI,ddX,ddPSI,ddPHI]),t),t),...
    [X, PSI,PHI,dX,dPSI,dPHI,ddX,ddPSI,ddPHI],...
    [x,psii,phi,dx,dpsi,dphi,ddx,ddpsi,ddphi]));

q_imu = -phi + qimu_plus_phi; % qimu_plus_phi is a constant if pose fixed

%% Left Wheel


thetaL = x/R - psii*L/(2*R);
dthetaL = dx/R - dpsi*L/(2*R);
iL = [1 0 0]'; jL = [0 1 0]'; kL = [0 0 1]';
i0 = [cos(thetaL) 0 sin(thetaL)]'; j0 = [0 1 0]'; k0 = [-sin(thetaL) 0 cos(thetaL)]';
w0 = dpsi*k0;
v0 = dx*i0;
alpha0 = ddpsi*k0;
a0 = ddx*i0 + dx*(cross(dpsi*k0,i0));
rOL = (L/2)*j0;
Iw=[ZZw 0 0;0 YYw 0;0 0 ZZw];
wL = w0 + dthetaL*j0;
vGL = v0 + cross(w0, rOL);
aGL = a0 + cross(alpha0, rOL) + cross(w0, cross(w0, rOL));
HGL = Iw*wL;
p = mydiff(HGL);
dHGL = p + cross(wL,HGL);

% %% Right Wheel
```

```
thetaR = x/R + psii*L/(2*R);
dthetaR = dx/R + dpsi*L/(2*R);
iR = [1 0 0]'; jR = [0 1 0]'; kR = [0 0 1]';
i0 = [cos(thetaR) 0 sin(thetaR)]'; j0 = [0 1 0]'; k0 = [-sin(thetaR) 0 cos(thetaR)]';
w0 = dpsi*k0;
v0 = dx*i0;
alpha0 = ddpsi*k0;
a0 = ddx*i0 + dx*(cross(dpsi*k0,i0));
r0R = -(L/2)*j0;
wR = w0 + dthetaR*j0;
vGR = v0 + cross(w0, r0R);
aGR = a0 + cross(alpha0, r0R) + cross(w0, cross(w0, r0R));
HGR = Iw*wR;
dHGR = mydiff(HGR)+cross(wR,HGR);
dq = dqVec(f); KK = sym(zeros(length(dq),1));
for i=1:2
    KK(i) = mw*aGL'*diff(vGL, dq(i)) + dHGL'*diff(wL, dq(i)) ...
          + mw*aGR'*diff(vGR, dq(i)) + dHGR'*diff(wR, dq(i));
end
```

## A.11 kanesLHSTree_()

```
function [K, KHist] = kanesLHSTree(f)

syms g real

dq = dqVec(f);
K = sym(zeros(length(dq), 1));
KHist=sym(zeros(length(dq),length(dq)));

key = keys(f);
for i=1:length(key)

    % Do nothing for frame 0
    if(isequal(key{i}, '0')); continue; end

    % Kinematics and Inertials Params
    w=angVel(f, key{i});
    alpha=angAcc(f, key{i});
    v=linVel(f, key{i});
    a=linAcc(f, key{i});
    J=inertiaMat(f, key{i});
    m=mass(f, key{i});
    mS=mCOM(f, key{i});
    S=mS/m;

    % Inertial Forces and Torques
    vG = v + cross(w, S);
    maG = m*(a + cross(alpha, S) + cross(w, cross(w, S)));
    dHnew = cross(mS, a) + J*alpha + cross(w, J*w);

    % Kanes LHS contributions
    for j=1:length(dq)
        disp(['kanesF: i=', num2str(i), ', j=', num2str(j), ', key=', key{i}]);
        KHist(j,f(key{i}).o) = maG'*diff(v, dq(j)) + dHnew'*diff(w, dq(j));
        K(j) = K(j) + KHist(j,f(key{i}).o);
    end
end
```

## A.12 kanesRHS_().m

```
function [K, KK, KK2, KK3, KK4] = kanesRHS(f)

dq = dqVec(f);
KK2 = sym(zeros(length(dq), length(dq)-3)); % Stores the contribution of each joint torque (nRows=ndqs, nCols=nTorques)
KK3 = sym(zeros(length(dq), 4)); % Stores the contribution of 1. Fel 2. tae_el 3. Fer 4. tau_er
KK4 = sym(zeros(length(dq), length(dq)-2)); % Stores the contribution of each link weight (nRows=ndqs, nCols=nLinks)
KK = sym(zeros(length(dq),4)); % Store contribution sums of 1. wheel torques 2. joint torques 3. End Eff F/T 4. Weights
K = sym(zeros(length(dq),1));  % Store the sum of all contributions

% Wheel Torques

syms tau_L tau_R L R real
i0 = [1 0 0]'; j0 = [0 1 0]'; k0 = [0 0 1]';
w0 = f('0').angVel;
v0 = f('0').linVel;
theta_L = v0(1)/R - w0(3)*L/(2*R);
theta_R = v0(1)/R + w0(3)*L/(2*R);
wL = w0 + theta_L*j0;
wR = w0 + theta_R*j0;
w1 = simplify(Rot(f, '1', '0') * angVel(f, '1'));
Tau_L = tau_L*j0;
Tau_R = tau_R*j0;
Tau_imu = -(tau_L + tau_R)*j0;

for i=1:length(dq)
```

```
    KK(i, 1) = Tau_L'*diff(wL, dq(i)) + Tau_R'*diff(wR, dq(i)) ...
        + Tau_imu'*diff(w1, dq(i));
end

% Joint Torques
syms tau_w tau_torso real
syms tau_1l tau_2l tau_3l tau_4l tau_5l tau_6l tau_7l real
syms tau_1r tau_2r tau_3r tau_4r tau_5r tau_6r tau_7r  real
keysTau = {'2', '3', '4l', '5l', '6l', '7l', '8l', ...
    '9l', '10l', '4r', '5r', '6r', '7r', '8r', '9r', '10r'};
aTau = [tau_w tau_torso tau_1l tau_2l tau_3l tau_4l tau_5l tau_6l tau_7l...
    tau_1r tau_2r tau_3r tau_4r tau_5r tau_6r tau_7r]';
for i=1:length(dq)-3; keySet{i} = keysTau{i}; cTau{i} = aTau(i); end
mTau = containers.Map( keySet, cTau );

keysTau = keys(mTau);
for i=1:length(keysTau)
    key = keysTau{i};
    tau = mTau(key);
    frame = f(key);
    e = frame.e;
    w = angVel(f, key);
    R = Rot(f, key, frame.a);
    wlast = angVel(f, frame.a);

    % Calculate the contribution
    for j=1:length(dq)
        KK2(j, i) = tau*e'*diff(w,dq(j)) + (-R*tau*e)'*diff(wlast,dq(j));
        KK(j, 2) = KK(j, 2) + KK2(j, i);
    end
end

% Contribution of forces/torques on the end-effector

syms Flx Fly Flz Frx Fry Frz Tlx Tly Tlz Trx Try Trz real
Fl = [Flx Fly Flz]'; Tl = [Tlx Tly Tlz]'; % expressed in frame 10l
Fr = [Frx Fry Frz]'; Tr = [Trx Try Trz]'; % expressed in frame 10r
veL = linVel(f, '10l');
weL = angVel(f, '10l');
veR = linVel(f, '10r');
weR = angVel(f, '10r');

for i=1:length(dq)
    KK3(i, 1) = KK3(i, 1) + Fl'*diff(veL, dq(i));
    KK3(i, 2) = KK3(i, 2) + Tl'*diff(weL, dq(i));
    KK3(i, 3) = KK3(i, 3) + Fr'*diff(veR, dq(i));
    KK3(i, 4) = KK3(i, 4) + Tr'*diff(weR, dq(i));
    KK(i, 3) = KK3(i, 1) + KK3(i, 2) + KK3(i, 3) + KK3(i, 4);
end

% Contribution of link weights
syms g real
key = keys(f);
for i=1:length(key)

    % if key is '0' leave it
    if(isequal(key{i}, '0')); continue; end

    % Get necessary info
    w = angVel(f, key{i});
    v = linVel(f, key{i});
    m = mass(f, key{i});
    S = mCOM(f, key{i})/m;
    T = Tf(f, '0', key{i});

    % Calculate contribution
    vG = v + cross(w, S);
    ROT = T(1:3,1:3);
    mg = m*ROT*[0 0 -g]';
    for j=1:length(dq)
        KK4(j, i) =  mg'*diff(vG, dq(j));
        KK(j, 4) = KK(j, 4) + KK4(j, i);
    end
end

for j=1:length(dq)
    K(j) = KK(j, 1) + KK(j, 2) + KK(j, 3) + KK(j, 4);
end
```

## A.13 Helper functions

## A.14 `angVel()`

```
function w = angVel(f, key)

% Calculate angular velocity of the current frame represented in the same
% frame.
% 'f' is the map container of the information for all the frames
% 'key' identifies the frame whose angular veclocity is demanded
% One more argument can define non-zero angular velocity of the base frame

disp(['   Computing angVel of ', key]);
```

```
if(f(key).gotAngVel) % This serves two purposes:
                     % 1. Base condition for recursive call as Frame 0 is
                     % setup to be gotAngVel = true
                     % 2. If this link has already been evaluated then
                     % don't go into recursion
    w=f(key).angVel;
else % recursive call
    w = Rot(f, f(key).a, key) * angVel(f, f(key).a) + f(key).e*f(key).dq;
    frame = f(key);
    frame.angVel = w;
    frame.gotAngVel = true;
    f(key) = frame;
end

disp(['   Computed angVel of ', key]);
```

## A.15 angAcc()

```
function alpha = angAcc(f, key)

% Calculate angular acceleration of the current frame represented in the same
% frame.
% 'f' is the map container of the information for all the frames
% 'key' identifies the frame whose angular acceleration is demanded
% One more argument can define non-zero angular velocity of the base frame

disp(['   Computing angAcc of ', key]);

if(f(key).gotAngAcc)
    alpha=f(key).angAcc;
else
    alpha_a = Rot(f, f(key).a, key) * angAcc(f, f(key).a);
    alpha = alpha_a + f(key).e * f(key).ddq ...
        + f(key).dq * cross(angVel(f, key), f(key).e);

    frame = f(key);
    frame.angAcc = alpha;
    frame.gotAngAcc = true;
    f(key) = frame;
end

disp(['   Computed angAcc of ', key]);
```

## A.16 linVel()

```
function V = linVel(f, key)

% Calculate linear velocity of the current frame measured in the world
% frame represented in the current frame
% 'f' is the map conatainer containing the information of the frames of
% the robot
% 'key' identifies the current frame
% Optional arguments:
% 'w0' is the angular velocity of the base frame
% 'V0' is the linear velocity of the base frame

disp(['   Computing linVel of ', key]);

if(f(key).gotLinVel)
    V=f(key).linVel;
else
    V = Rot(f, f(key).a, key) * (linVel(f, f(key).a) + ...
            cross(angVel(f, f(key).a), f(key).P));

    frame = f(key);
    frame.linVel = V;
    frame.gotLinVel = true;
    f(key) = frame;

end

disp(['   Computed linVel of ', key]);
```

## A.17 linAcc()

```
function a = linAcc(f, key)

% Calculate linear velocity of the current frame measured in the world
% frame represented in the current frame
% 'f' is the map conatainer containing the information of the frames of
```

```
% the robot
% 'key' identifies the current frame
% Optional arguments:
% 'w0' is the angular velocity of the base frame
% 'V0' is the linear velocity of the base frame

disp(['   Computing linAcc of ', key]);

if(f(key).gotLinAcc)
    a=f(key).linAcc;
else
    w_a = angVel(f, f(key).a);
    alpha_a = angAcc(f, f(key).a);
    a_a = linAcc(f, f(key).a);

    % Acceleration of the current frame
    a = Rot(f, f(key).a, key) * (a_a + cross(alpha_a, f(key).P) ...
        + cross(w_a, cross(w_a, f(key).P)));

    frame = f(key);
    frame.linAcc = a;
    frame.gotLinAcc = true;
    f(key) = frame;

end

disp(['   Computed linAcc of ', key]);
```

## A.18 Rot()

```
function A = Rot(f, key1, key2)

% Find the rotation transform of a frame
% f denotes the map container that contains information of the frames.
% key1 identifies the frame1 that is being represented
% key2 identifies frame2 in which frame1 is being represented
% This function assumes frame1 immediately follows frame2 or frame2
% immendiately follows frame1

if(isequal(f(key1).a, key2)) % if frame2 is the antecedent of frame1
    A = [f(key1).x f(key1).y f(key1).z];
elseif(isequal(f(key2).a, key1)) % if frame1 is the antecedent of frame2
    A = [f(key2).x'; f(key2).y'; f(key2).z'];
end
```

## A.19 Tf()

```
function T = Tf(f, key1, key2)

% Find the transform of a frame
% f denotes the map container that contains information of the frames.
% key1 identifies the frame1 that is being represented
% key2 identifies frame2 in which frame1 is being represented

if(isBefore(f, key2, key1)) % if frame2 is before frame1 in the chain
    T = [f(key1).x f(key1).y f(key1).z f(key1).P; 0 0 0 1];
    key = f(key1).a;
    while(~isequal(key, key2))
        T = [f(key).x f(key).y f(key).z f(key).P; 0 0 0 1] * T;
        key = f(key).a;
    end
elseif(isBefore(f, key1, key2)) % if frame1 is before frame2 in the chain
    T = [f(key2).x' -f(key2).x'*f(key2).P; ...
         f(key2).y' -f(key2).y'*f(key2).P; ...
         f(key2).z' -f(key2).z'*f(key2).P; ...
         0 0 0 1];
     key = f(key2).a;
     while(~isequal(key, key1))
         T = T * [f(key).x' -f(key).x'*f(key).P; ...
              f(key).y' -f(key).y'*f(key).P; ...
              f(key).z' -f(key).z'*f(key).P; ...
              0 0 0 1];
         key = f(key).a;
     end
elseif(isequal(key1, key2))
    T = sym(eye(4));
end
```

## A.20 mass()

```
function m = mass(f, key)

param = f(key).param;
m = param(1);
```

## A.21 mCOM()

```
function MS = mCOM(f, key)

param = f(key).param;
MS = param(2:4)';
```

## A.22 inertiaMat()

```
function J = inertiaMat(f, key)

param = f(key).param;
J = param([5 6 7; 6 8 9; 7 9 10]);
```

## A.23 qVec()

```
function q = qVec(f)

% Generate q vector in the order defined by memeber 'o' of the frames
% contained in 'f'

key = keys(f);
n = length(key)-1;
q = sym(zeros(n, 1));
for i=1:length(key)
    if(isequal(key{i}, '0')); continue; end
    q(f(key{i}).o) = f(key{i}).q;
end
```

## A.24 dqVec()

```
function dq = dqVec(f)

% Generate dq vector in the order defined by memeber 'o' of the frames
% contained in 'f'

key = keys(f);
n = length(key)+1;
dq = sym(zeros(n, 1));
for i=1:length(key)
    if(isequal(key{i}, '0'));
        w0 = f('0').angVel;
        v0 = f('0').linVel;
        dq(1) = v0(1);
        dq(2) = w0(3);
    else
        dq(f(key{i}).o) = f(key{i}).dq;
    end
end
```

## A.25 ddqVec()

```
function ddq = ddqVec(f)

% Generate ddq vector in the order defined by memeber 'o' of the frames
% contained in 'f'

key = keys(f);
n = length(key)+1;
ddq = sym(zeros(n, 1));
for i=1:length(key)
    if(isequal(key{i}, '0'));
        alpha0 = f('0').angAcc;
        a0 = f('0').linAcc;
        ddq(1) = a0(1);
        ddq(2) = alpha0(3);
    else
        ddq(f(key{i}).o) = f(key{i}).ddq;
    end
end
```

## A.26 getcoeff()

```
function c = getcoeff(P, x, a)

[C, T] = coeffs(P, x);
n=length(C);
exists = 0;
for i=1:n
    if(isequal(T(i),x^a));
        exists = 1;
        break;
    end
end

if(exists)
    c = C(i);
else
    c = 0;
end
```