Simcenter Motion Solver

Function Expressions
Proprietary and restricted rights notice

This software and related documentation are proprietary to Siemens Product Lifecycle Management Software Inc.

© 2016 Siemens Product Lifecycle Management Software Inc. All Rights Reserved. All trademarks belong to their respective holders.
## Contents

Overview ........................................................................................................................................... 4
Motion Variables .................................................................................................................................. 4
Calculation of Wrap-up ...................................................................................................................... 24
Force and Torque Variables ............................................................................................................. 24
How to use the Arithmetic IF Function .......................................................................................... 28
How to use the Interpolation Functions .......................................................................................... 29
Mathematical Functions .................................................................................................................... 30
Overview

This document contains the functions supported by the Simcenter Motion Solver.

Motion Variables

Motion Variables measure position, velocity, or acceleration for angular or translational motion. The following Motion Variables are available:

<table>
<thead>
<tr>
<th>Type</th>
<th>Displacement</th>
<th>Velocity</th>
<th>Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translational</td>
<td>DM, DX, DY, DZ</td>
<td>VM, VX, VY, VZ, VR</td>
<td>ACCM, ACCX, ACCY, ACCZ</td>
</tr>
<tr>
<td>Rotational</td>
<td>AX, AY, AZ, ROLL, PITCH, YAW, THETA, PHI, PSI</td>
<td>WM, WX, WY, WZ</td>
<td>WDTM, WDTX, WDTY, WDTZ</td>
</tr>
</tbody>
</table>

Description of Motion Variables

A marker name in brackets (“{ }”) is optional. M3 defaults to the global coordinate system if not defined.

ACCM

Measures the magnitude of acceleration between two objects. Valid objects include: Markers (M). If the object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the value returned by this variable are length/time^2.

Syntax

ACCM (M1{, M2})

where

M1 Existing Marker
M2 Existing Marker

Formulation

\[
ACCM (M1, M2) = \left\| \frac{d^2}{dt^2} d_{12} \right\| \frac{d^2}{dt^2} d_{12}
\]

where

\(d_{12}\) is the global vector from the origin of M2 to the origin of M1.
**ACCX, ACCY, ACCZ**

Measures a component of relative acceleration between two objects as measured in the coordinates of a third object. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the values returned by these variables are \( \frac{\text{length}}{\text{time}^2} \).

**Syntax**

\[
\text{ACCX} (M_1, M_2, M_3)\]

Measures projection of relative acceleration between \( M_1 \) and \( M_2 \) along the positive x axis of \( M_3 \).

where

- \( M_1 \) Existing Marker
- \( M_2 \) Existing Marker
- \( M_3 \) Existing Marker

**Formulation**

\[
\text{ACCX} (M_1, M_2, M_3) = f_3^T \frac{d^2}{dt^2} d_{12}
\]

\[
\text{ACY} (M_1, M_2, M_3) = g_3^T \frac{d^2}{dt^2} d_{12}
\]

\[
\text{ACZZ} (M_1, M_2, M_3) = h_3^T \frac{d^2}{dt^2} d_{12}
\]

where

- \( d_{12} \) is the global vector from the origin of \( M_2 \) to the origin of \( M_1 \) and
- \( f_3, g_3, \) and \( h_3 \) are unit vectors along the x, y, and z axes of \( M_3 \), respectively.

**AX, AY, AZ**

Measures relative angles of rotation between two objects about their x, y, or z axes. Valid objects include: Markers (M) and an Axis. If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The computed angle is measured in radians. An attempt is made to compute the total angle of rotation including wrap-up.
Definition of angle with the x axis selected as the argument.
Definition of angle with the y axis selected as the argument.

Definition of angle with the z axis selected as the argument.
Syntax

AX (M1, M2) Measures the angle of rotation (in radians) between objects M1 and M2 about their common axis of rotation x. The angle is measured from the y (or z) axis of M2 to the y (or z) axis of M1 as shown.

| Warning: The simulation will fail if the y axis of M1 becomes orthogonal to both the y and z axes of M2.
angle (M1, M2, "Axis") Measures the angle of rotation (in radians) between objects M1 and M2 about their common axis of rotation y. The angle is measured from the x (or z) axis of M2 to the x (or z) axis of M1 as shown.
| Warning: The simulation will fail if the z axis of M1 becomes orthogonal to both the x and z axes of M2.
angle (M1, M2, "Axis") Measures the angle of rotation (in radians) between objects M1 and M2 about their common axis of rotation z. The angle is measured from the x (or y) axis of M2 to the x (or y) axis of M1 as shown.
| Warning: The simulation will fail if the x axis of M1 becomes orthogonal to both the x and y axes of M2.

where

M1 Existing Marker.
M2 Existing Marker.

| Warning: The angle variables will fail if they reach a singular configuration as described above. The information in the section Calculation of Wrap-up (../mot-theory/CalculationofWrap-up.htm) applies to this variable.

Formulation

\[
AX (M1, M2) = \tan^{-1} \left( \frac{g_1^T h_2}{g_1^T g_2} \right)
\]

\[
AY (M1, M2) = \tan^{-1} \left( \frac{h_1^T f_2}{h_1^T h_2} \right)
\]

\[
AZ (M1, M2) = \tan^{-1} \left( \frac{f_1^T g_2}{f_1^T f_2} \right)
\]

where

\( f_1, g_1, \text{and} \ h_1 \) are unit vectors along the x, y, and z axes of M1, respectively, and
\( f_2, g_2, \text{and} \ h_2 \) are unit vectors along the x, y, and z axes of M2, respectively.
**WM**

Measures the magnitude of relative angular velocity between two objects. Valid objects include: Variable Axis (M) Systems. If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the value returned by this variable are \( \text{radians} / \text{time} \).

**Syntax**

\[ \text{WM} (M_1, M_2) \]

where

- \( M_1 \) Existing Marker.
- \( M_2 \) Existing Marker.

**Formulation**

\[
\text{WM} (M_1, M_2) = \sqrt{\omega_{12}^T \omega_{12}}
\]

where

\[
\omega_{12} = \omega_1 - \omega_2, \quad \omega_1 \quad \text{is the global angular velocity of } M_1, \text{ and } \omega_2 \quad \text{is the global angular velocity of } M_2.
\]

**WX, WY, WZ**

Measures a component of relative angular velocity between two objects along an axis of a third object. Markers and an Axis. If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the values returned by these variables are \( \text{radians} / \text{time} \).

**Syntax**

\[ \text{WX} (M_1, M_2, M_3) \]

Measures the component of relative angular velocity between \( M_1 \) and \( M_2 \) along the positive x axis of \( M_3 \).

where

- \( M_1 \) Existing Marker.
- \( M_2 \) Existing Marker.
- \( M_3 \) Existing Marker.

**Formulation**

\[
\text{WX} (M_1, M_2, M_3) = \omega_{12}^T f_3
\]

\[
\text{WY} (M_1, M_2, M_3) = \omega_{12}^T g_3
\]

\[
\text{WZ} (M_1, M_2, M_3) = \omega_{12}^T h_3
\]
where

\[ \omega_{12} = \omega_1 - \omega_2, \quad \hat{\omega}_1 \] 

is the global angular velocity of \( M_1 \), \( \omega_2 \) is the global angular velocity of \( M_2 \), and \( \hat{f}_3, \hat{g}_3, \text{and} \hat{h}_3 \) are unit vectors along the x, y, and z axes of \( M_3 \), respectively.

### WDTM

Measures the magnitude of relative angular acceleration between two objects. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the value returned by this variable are \( \text{radians} / \text{time}^2 \).

**Syntax**

\[ \text{WDTM} \left( M_1 \{, M_2 \} \right) \]

where

\( M_1 \) Existing Marker.

\( M_2 \) Existing Marker.

**Formulation**

\[ \text{WDTM} \left( M_1, M_2 \right) = \sqrt{\frac{\text{d}}{\text{d}t} \hat{\omega}_1^T} \frac{\text{d}}{\text{d}t} \hat{\omega}_{12} \]

where

\[ \omega_{12} = \omega_1 - \omega_2, \quad \hat{\omega}_1 \] 

is the global angular velocity of \( M_1 \) and \( \omega_2 \) is the global angular velocity of \( M_2 \).

### WDTX, WDTY, WDTZ

Measures a component of relative angular acceleration between two objects along an axis of a third object. Valid objects include: Markers (M) and an Axis. If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the values returned by these variables are \( \text{radians} / \text{time}^2 \).

**Syntax**

\[ \text{WDTX} \left( M_1 \{, M_2 \} \{, M_3 \} \right) \] Measures the component of relative angular acceleration between objects \( M_1 \) and \( M_2 \) along the positive x axis of \( M_3 \).
where

M1 Existing Marker.
M2 Existing Marker.
M3 Existing Marker.
"Axis" x, y, or z axis of M3.

Formulation

\[
\begin{align*}
WDTX (M1, M2, M3) & = \hat{f}_3^T \frac{d}{dt} \omega_{12} \\
WDTY (M1, M2, M3) & = \hat{g}_3^T \frac{d}{dt} \omega_{12} \\
WDTZ (M1, M2, M3) & = \hat{h}_3^T \frac{d}{dt} \omega_{12}
\end{align*}
\]

where

\[\omega_{12} = \omega_1 - \omega_2, \quad \omega_1 \text{ is the global angular velocity of } M1, \quad \omega_2 \text{ is the global angular velocity of } M2, \text{ and } \hat{f}_3, \hat{g}_3, \text{ and } \hat{h}_3 \text{ are unit vectors along the x, y, and z axes of } M3, \text{ respectively.}\]

**DM**

Measures the distance between two object origins. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the value returned by this variable is length.

**Syntax**

\[DM (M1, M2)\]

where

M1 Existing Marker.
M2 Existing Marker.

**Formulation**

\[DM (M1, M2) = \sqrt{d_{12}^T d_{12}}\]

where

\[d_{12} \text{ is the global vector from the origin of } M2 \text{ to the origin of } M1.\]
**DX, DY, DZ**

Measures the component of a distance vector directed between two object origins. The component is measured along the axis of a third object as indicated by the variable name. Valid objects include: Markers and an Axis. If an object is a descendant of a flexible body, then the object must exist at a node on the flexible body. The units of the values returned by these variables is length.

**Syntax**

\[
\text{DX} (\text{M1}, \text{M2}|\text{M3}) \text{ Measures the distance of the vector between M1 and M2 relative to M3 X direction. In other words, the head of the vector is at M1 and the tail is at M2.}
\]

where

- M1 Existing Marker.
- M2 Existing Marker.
- M3 Existing Marker.

**Formulation**

\[
\begin{align*}
\text{DX} (\text{M1, M2, M3}) &= d_{12}^T f_3 \\
\text{DY} (\text{M1, M2, M3}) &= d_{12}^T g_3 \\
\text{DZ} (\text{M1, M2, M3}) &= d_{12}^T h_3
\end{align*}
\]

where

- \( d_{12} \) is the global vector from the origin of M2 to the origin of M1 and
- \( f_3, g_3, \text{and} h_3 \) are unit vectors along the x, y, and z axes of M3, respectively.
**PHI**

Measures the third angle of a body-2 313 Euler angle rotation between two objects. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The computed angle is measured in radians.

Euler angles are defined as follows:

1. Begin with an initial right-handed coordinate system \((x_2, y_2, z_2)\).
2. Position an intermediate coordinate system \((x', y', z')\) from the \((x_2, y_2, z_2)\) axes by rotating the \((x', y')\) axes through the psi angle about the common \((z_2, z')\) axes.
3. Position an intermediate coordinate system \((x'', y'', z'')\) from the \((x', y', z')\) axes by rotating the \((y', z')\) axes through the theta angle about the common \((x', x'')\) axes.
4. Position the final coordinate system \((x_1, y_1, z_1)\) from the \((x'', y'', z'')\) axes by rotating the \((x_1, y_1)\) axes through the phi angle about the common \((z'', z_1)\) axes.

**Warnings:** The simulation will fail if the z axis of M2 becomes orthogonal to both the x and y axes of M1. The information in the section *Calculation of Wrap-up* ([..mot-theory/CalculationofWrap-up.htm](../mot-theory/CalculationofWrap-up.htm)) applies to this variable.

**Syntax**

```
PHI (M1[, M2])
```

where

- M 1 Existing Marker.
- M 2 Existing Marker.

**Formulation**

\[
PHI (M1, M2) = \phi_{12} = \tan^{-1}\left(\frac{\mathbf{f}_1 \cdot \mathbf{h}_2}{\mathbf{g}_1 \cdot \mathbf{h}_2}\right)
\]

where

- \(\mathbf{f}_1, \mathbf{g}_1\) are unit vectors along the x and y axes of M1, respectively, and
- \(\mathbf{f}_1, \mathbf{g}_1, \mathbf{h}_2\) is the unit vector along the z axis of M2.
**PITCH**

Measures the second angle of a Body-3 321 yaw-pitch-roll rotation between two objects as shown in the figure below. Valid objects include: Markers (M). If an object is a descendant of a flexible body, then the object must exist at a node on the flexible body. The computed angle is measured in radians.

Definition of yaw-pitch-roll angles.
Yaw-pitch-roll angles are defined as follows:

1. Begin with an initial right-handed coordinate system \((x_2, y_2, z_2)\).

2. Position an intermediate coordinate system \((x', y', z')\) from the \((x_2, y_2, z_2)\) axes by rotating the \((x', y')\) axes through the yaw angle about the common \((z_2, z')\) axes.

3. Position an intermediate coordinate system \((x'', y'', z'')\) from the \((x', y', z')\) axes by rotating the \((x'', z'')\) axes through the pitch angle about the common \((y', y'')\) axes.

4. Position the final coordinate system \((x_1, y_1, z_1)\) from the \((x'', y'', z'')\) axes by rotating the \((y_1, z_1)\) axes through the roll angle about the common \((x'', x_1)\) axes.

**Warning:** The simulation will fail if the x axis of \(M_1\) becomes collinear with the z axis of \(M_2\). The information in the section *Calculation of Wrap-up* (../mot-theory/CalculationofWrap-up.htm) applies to this variable.

**Syntax**

```
PITCH (M1[, M2])
```

where

- \(M_1\) Existing Marker.
- \(M_2\) Existing Marker.

**Formulation**

\[
PITCH (M1, M2) = \phi_{12}^2 = \tan^{-1} \left( \frac{-f_1^T h_2}{q_1^T h_2 \sin \phi_{12}^3 + h_1^T h_2 \cos \phi_{12}^3} \right)
\]

where

- \(f_1\) is the unit vector along the x axis of \(M_1\) and \(f_2, g_2, h_2\) are unit vectors along the x, y, and z axes of \(M_2\), respectively.
PSI

Measures the first angle of a body-2 313 Euler angle rotation between two objects. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The computed angle is measured in radians.

Euler angles are defined as follows:

1. Begin with an initial right-handed coordinate system \((x_2, y_2, z_2)\).
2. Position an intermediate coordinate system \((x', y', z')\) from the \((x_2, y_2, z_2)\) axes by rotating the \((x', y')\) axes through the psi angle about the common \((z_2, z')\) axes.
3. Position an intermediate coordinate system \((x'', y'', z'')\) from the \((x', y', z')\) axes by rotating the \((y', z')\) axes through the theta angle about the common \((x', x')\) axes.
4. Position the final coordinate system \((x_1, y_1, z_1)\) from the \((x'', y'', z'')\) axes by rotating the \((x_1, y_1)\) axes through the phi angle about the common \((z', z_1)\) axes.

**Warning:** The simulation will fail if the z axis of \(i_1\) becomes orthogonal to both the x and y axes of \(i_2\). The information in the section **Calculation of Wrap-up** ([./mot-theory/CalculationofWrap-up.htm](http://simcenter.com/motion_solver/function_expressions/psi)) applies to this variable.

![Definition of Euler angles.](image)

**Syntax**

```
PSI(M1, M2)
```
where

\[ M_1 \text{ Existing Marker.} \]

\[ M_2 \text{ Existing Marker.} \]

**Formulation**

\[
\Psi_{12} = \arctan^{-1}\left( \frac{f_{1z}^T g_{12} \cos \phi_{12} - g_{1z}^T f_{2z} \sin \phi_{12}}{f_{1z}^T f_{2z} \cos \phi_{12} - g_{1z}^T g_{2z} \sin \phi_{12}} \right)
\]

where

- \( h_1 \) is the unit vector along the z axis of \( M_1 \) and \( f_2 \) and \( g_2 \) are unit vectors along the x and y axes of \( M_2 \), respectively.

**ROLL**

Measures the third angle of a Body-3 321 yaw-pitch-roll rotation. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The computed angle is measured in radians.

Yaw-pitch-roll angles are defined as follows:

1. Begin with an initial right-handed coordinate system \((x_2, y_2, z_2)\).
2. Position an intermediate coordinate system \((x', y', z')\) from the \((x, y, z)\) axes by rotating the \((x', y')\) axes through the yaw angle about the common \((z_2, z')\) axes.
3. Position an intermediate coordinate system \((x'', y'', z'')\) from the \((x', y', z')\) axes by rotating the \((x'', z'')\) axes through the pitch angle about the common \((y', y'')\) axes.
4. Position the final coordinate system \((x_1, y_1, z_1)\) from the \((x'', y'', z'')\) axes by rotating the \((y_1, z_1)\) axes through the roll angle about the common \((x'', x_1)\) axes.

**Warning:** The simulation will fail if the z axis of \( M_2 \) becomes orthogonal to both the y and z axes of \( M_1 \). The information in the section *Calculation of Wrap-up* ([/mot-theory/CalculationofWrap-up.htm](../mot-theory/CalculationofWrap-up.htm)) applies to this variable.

**Syntax**

\[
\text{ROLL}(M1[, M2])
\]

where

- \( M_1 \) Existing Marker.
- \( M_2 \) Existing Marker.
Formulation

\[ ROLL (M1, M2) = \phi_2 = \tan^{-1} \left( \frac{\mathbf{g}_1 \cdot \mathbf{h}_2}{\mathbf{h}_1 \cdot \mathbf{h}_2} \right) \]

where

\[ \mathbf{g}_1 \quad \text{and} \quad \mathbf{h}_1 \] are unit vectors along the y and z axes of M2, respectively, and \[ \mathbf{h}_2 \] is the unit vector along the z axis of M1.

---

**THETA**

Measures the second angle of a body-2 313 Euler angle rotation between two objects. Valid objects include: Markers (M). If an object is a descendant of a flexible body, then the object must exist at a node on the flexible body. Euler angles are defined as follows:

1. Begin with an initial right-handed coordinate system \((x_2, y_2, z_2)\).
2. Position an intermediate coordinate system \((x', y', z')\) from the \((x_2, y_2, z_2)\) axes by rotating the \((x', y')\) axes through the psi angle about the common \((z_2, z')\) axes.
3. Position an intermediate coordinate system \((x'', y'', z'')\) from the \((x', y', z')\) axes by rotating the \((y', z')\) axes through the theta angle about the common \((x', x'')\) axes.
4. Position the final coordinate system \((x_1, y_1, z_1)\) from the \((x'', y'', z'')\) axes by rotating the \((x_1, y_1)\) axes through the phi angle about the common \((z'', z_1)\) axes.

**Warning:** The simulation will fail if the z axis of M1 becomes collinear with the z axis of M2. The information in the section *Calculation of Wrap-up* (*Simcenter Motion Solver Theory/CalculationofWrap-up.htm*) applies to this variable.

Syntax

\[ \text{THETA (M1[, M2])} \]

where

M 1 Existing Marker.
M 2 Existing Marker.

Formulation

\[ \text{THETA (M1, M2) = \Theta_2 = \tan^{-1} \left( \frac{f_1 \cdot h_2 \sin \phi_2 + g_1 \cdot h_2 \cos \phi_2}{h_1 \cdot h_2} \right) \]}

where

\[ f_1, g_1, \text{and} \ h_1 \] are unit vectors along the x, y, and z axes of M1, respectively, and \[ h_2 \] is the unit vector along the z axis of M2.
**VM**

Measures the magnitude of the relative velocity between two objects. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the value returned by this variable are \( \frac{\text{length}}{\text{time}} \).

**Syntax**

\[ \text{VM}(\text{M}_1, \text{M}_2) \]

where

- \( \text{M}_1 \) Existing Marker.
- \( \text{M}_2 \) Existing Marker.

**Formulation**

\[ \text{VM}(\text{M}_1, \text{M}_2) = \left( \frac{\partial}{\partial t} \mathbf{d}_{12} \right) \frac{\partial}{\partial t} \mathbf{d}_{12} \]

where

- \( \mathbf{d}_{12} \) is the global vector from the origin of \( \text{M}_2 \) to the origin of \( \text{M}_1 \).

**VR**

Measures the radial velocity between two objects. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the value returned by this variable are \( \frac{\text{length}}{\text{time}} \).

**Warning:** The simulation will fail if the distance between \( t_1 \) and \( t_2 \) becomes zero.

**Syntax**

\[ \text{VR}(\text{M}_1, \text{M}_2) \]

where

- \( \text{M}_1 \) Existing Marker.
- \( \text{M}_2 \) Existing Marker.

**Formulation**

\[ \text{VR}(\text{M}_1, \text{M}_2) = \left( \frac{\partial}{\partial t} \mathbf{d}_{12} \right) \frac{\partial}{\partial t} \mathbf{d}_{12} \]

where

- \( \mathbf{d}_{12} \) is the global vector from the origin of \( \text{M}_2 \) to the origin of \( \text{M}_1 \).
Measures a component of relative velocity between two objects. Valid objects include: Markers (M) and an Axis. If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The units of the values returned by these variables are \( \frac{\text{length}}{\text{time}} \).

**Syntax**

\[
VX(M1, M2, M3, "Axis") \quad \text{Measures the component of translational velocity between objects } M1 \text{ and } M2 \text{ along the positive x axis of } M3.
\]

where

- \( M1 \) Existing Marker.
- \( M2 \) Existing Marker.
- \( M3 \) Existing Marker.
- "Axis" x, y, or z axis of M3.

**Formulation**

\[
VX(M1, M2, M3) = f_3^T \frac{d}{dt} d_{12} \\
VY(M1, M2, M3) = g_3^T \frac{d}{dt} d_{12} \\
VZ(M1, M2, M3) = h_3^T \frac{d}{dt} d_{12}
\]

where

- \( d_{12} \) is the global vector from the origin of \( M2 \) to the origin of \( M1 \) and
- \( f_3, g_3, \text{and } h_3 \) are unit vectors along the x, y, and z axes of \( M3 \), respectively.
**YAW**

Measures the first angle of a Body-3 321 yaw-pitch-roll rotation between two objects. Valid objects include: Markers (M). If an object is a descendent of a flexible body, then the object must exist at a node on the flexible body. The computed angle is measured in radians.

Yaw-pitch-roll angles are defined as follows:

1. Begin with an initial right-handed coordinate system \((x_2, y_2, z_2)\).
2. Position an intermediate coordinate system \((x', y', z')\) from the \((x_2, y_2, z_2)\) axes by rotating the \((x', y')\) axes through the yaw angle about the common \((z_2, z')\) axes.
3. Position an intermediate coordinate system \((x'', y'', z'')\) from the \((x', y', z')\) axes by rotating the \((x'', z'')\) axes through the pitch angle about the common \((y', y'')\) axes.
4. Position the final coordinate system \((x_1, y_1, z_1)\) from the \((x'', y'', z'')\) axes by rotating the \((y_1, z_1)\) axes through the roll angle about the common \((x'', x_1)\) axes.

**Warning:** The simulation will fail if the x axis of M1 becomes orthogonal to both the x and y axes of M2. The information in the section *Calculation of Wrap-up* ([../mot-theory/CalculationofWrap-up.htm](../mot-theory/CalculationofWrap-up.htm)) applies to this variable.
Definition of yaw-pitch-roll angles.

Syntax

\[ \text{YAW} \left( \text{M}_1, \text{M}_2 \right) \]

where

M\textsubscript{1} Existing Marker.
M\textsubscript{2} Existing Marker.

Formulation

\[
\text{YAW} \left( \text{M}_1, \text{M}_2 \right) = \phi_{12} = \tan^{-1} \left( \frac{f_1^T f_2 \sin \phi_{12} - g_1^T f_2 \cos \phi_{12}}{g_1^T g_2 \cos \phi_{12} - h_1^T g_2 \sin \phi_{12}} \right)
\]

where

\( f_1 \) is the unit vector along the x axis of M\textsubscript{1} and \( f_2 \) and \( g_2 \) are unit vectors along the x and y axes of M\textsubscript{2}, respectively.
Calculation of Wrap-up

Configuration variables that measure angular displacement account for wrap-up. Wrap-up occurs when multiple relative rotations occur between the two referenced triads. Without wrap-up, the measured angle would jump discontinuously by $\pi$ or $2\pi$ at certain values. The discontinuous angle represents the correct orientation. However, in general, it is less useful and would result in discontinuous forces if used to describe a force. Therefore, configuration variables have a wrap-up algorithm built-in to maintain continuous angular measurements.

Since multiple angles can represent the same orientation (for example, 0 and $2\pi$ are the same angle), the initial angle measured by angle variables will be returned in the range $-\pi < \text{angle} < \pi$. If a different initial angle is desired, then add constant multiples of $2\pi$ to achieve the desired initial angle.

If the solver step size is large enough and causes large changes in the relative orientation between the two triads, the wrap-up algorithm may fail. The measured angle then becomes discontinuous without warning. The following actions may help to limit the solver step size:

- In the Dynamic Data element, decrease Max Int Step, Solution Tol, or Integration Tol.
- In the System Data element, decrease Print Interval.
- In the Inverse Dynamic Data element, decrease Step Size or Solution Tol.
- In the Kinematic Data element, decrease Step Size or Solution Tol.
- In the Static Data element, decrease Step Size.

Force and Torque Variables

A marker name in brackets (“{ }”) is optional. The marker names are defined as follows:

(m1, m2,{m3}) : (bodyname.markernname, bodyname.markernname, {bodyname.markernname}).

m1 is the action marker and m2 is base marker. If m3 is defined, m3 becomes the reference marker. You will obtain the force acting on m1 with respect to m2 in the coordinate system of m3 (m3 defaults to the global coordinate system if not defined).
Translational Force (FM, FX, FY and FZ)

**FM**
Magnitude of force
Syntax

FM( m1, m2)

**FX**
X - component force
Syntax

FX( m1, m2 ,m3)

**FY**
Y - component force
Syntax

FY( m1, m2,m3)

**FZ**
Z - component force
Syntax

FZ( m1, m2 ,m3)

NOTE: The markers used must be part of a Force entity

**TM**
Magnitude of torque
Syntax

TM( m1, m2 )
**TX**

X - component torque

**Syntax**

TX( m1, m2 {,m3} )

**TY**

Y - component torque

**Syntax**

TY( m1, m2 {,m3} )

**TZ**

Z - component torque

**Syntax**

TZ( m1, m2 {,m3} )

**Force Functions**

**CONTACT**

Contact force function

**Syntax**

CONTACT( Element, BodyNum, Axis, M )

**MOTION**

Driving force function of motion

**Syntax**

MOTION( Element, BodyNum, Axis, M)
**JFRICION**

Friction force function of joint

Syntax

\[ \text{JFRICION( Element, BodyNum, Axis, M) } \]

**COUPLER**

Driving or driven force function of coupler

Syntax

\[ \text{COUPLER( Element, BodyNum, Axis, M) } \]

**GEAR**

Driving or driven force function of gear

Syntax

\[ \text{GEAR( Element, BodyNum, Axis, M) } \]

where

- Element is the element producing the reaction force or torque.
- "Axis" is x, y, or z axis of M, or "mg" for magnitude.
- M is the existing marker.
- BodyNum is the reference number, where 1 indicates the first body referenced, and so on.
How to use the Arithmetic IF Function

**IF**

The arithmetic IF statement conditionally defines a function expression.

**Syntax**

```
IF(Expression1: Expression2, Expression3, Expression4)
```

If the value of Expression1 < 0, IF calculates and returns the value of Expression2. If the value of Expression1 = 0, IF calculates and returns the value of Expression3. If the value of Expression1 > 0, IF calculates and returns the value of Expression4.

**Example**

The following example conditionally defines the different values according to the value of the time.

```
IF(TIME-2.0: -1.0, 0.0, 1.0)
```

- **Expression1** = TIME=2.0
- **Expression2** = -1.0
- **Expression3** = 0
- **Expression4** = 1.0

<table>
<thead>
<tr>
<th>Condition</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME &lt; 2.0</td>
<td>IF = -1.0</td>
</tr>
<tr>
<td>TIME = 2.0</td>
<td>IF = 0.0</td>
</tr>
<tr>
<td>TIME &gt; 2.0</td>
<td>IF = 1.0</td>
</tr>
</tbody>
</table>
How to use the Interpolation Functions

**AKISPL**

Interpolation value

**Syntax**

```
AKISPL(x, z, modelname.curvename, order)
```

Higher order derivative of interpolated value

**Syntax**

```
AKISPL(x, z, modelname.curvename, order)
```

where

- \( x \) is the independent variable of the x axis.
- \( z \) is the interpolated value of the z axis. You must put the 0 as a z value.
- Modelname.curvename is the identifier of the curve or spline that is being used.
- Order is the optional differential order.
## Mathematical Functions

In this section, \( u \) and \( v \) are sub-expressions and \( y \) is the value returned by the function.

<table>
<thead>
<tr>
<th>Mathematical Function</th>
<th>Syntax</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>ABS( (u) )</td>
<td>( y =</td>
</tr>
<tr>
<td>acos</td>
<td>ACOS( (u) )</td>
<td>( y = \cos^{-1} u )</td>
</tr>
<tr>
<td>aint</td>
<td>AINT( (u) )</td>
<td>Returns the largest integer not greater than ( u ).</td>
</tr>
<tr>
<td>asin</td>
<td>ASIN( (u) )</td>
<td>( y = \sin^{-1} u )</td>
</tr>
<tr>
<td>atan</td>
<td>ATAN( (u) )</td>
<td>( y = \tan^{-1} u )</td>
</tr>
<tr>
<td>atan2</td>
<td>ATAN2( (u, v) )</td>
<td>( y = \tan^{-1} \frac{u}{v} )</td>
</tr>
</tbody>
</table>
| bistop                | BISTOP\( (x, \dot{x}, x_1, x_2, k, \exp, c_{\text{max}}, d) \) | BISTOP = \( k(\dot{x} - x)\exp - \text{STEP}(x, x_1 - d, c_{\text{max}}, x, 0)^*\dot{x} \) when \( x < x_1 \)  
BISTOP = 0, when \( x_1 \leq x \leq x_2 \)  
BISTOP = \(-k(x - x_2)\exp - \text{STEP}(x, x_2, 0, x_2 + d, c_{\text{max}})^*\dot{x} \) when \( x \geq x_2 \)  
Example:  
BISTOP\( (x, \dot{x}, x_1, x_2, k, \exp, c_{\text{max}}, d) \)  
x: Distance variable \( \dot{x} \)  
\( \dot{x} \): Time derivative of \( x \)  
x_1: Lower bound of \( x \)  
x_2: Upper bound of \( x \)  
k: Stiffness  
\( \exp \): Exponent of force  
c_{\text{max}}: Maximum damping coefficient  
d: Boundary penetration |
| cheby                 | CHEBY\( (x, x_0, c_0, c_1, \ldots, c_{30}) \) | Defines the Chebyshev Polynomial.  
\( C(x) = \sum_{j=0}^{n} c_j * T_j(x - x_0) \)  
\( 0 \leq j \leq n \) |
where:

\[
T_j(x - x_0) = 2 \times (x - x_0) \times T_{j-1}(x - x_0) - T_{j-2}(x - x_0)
\]

\[
T_0(x - x_0) = 1, \quad T_1(x - x_0) = x - x_0
\]

**Example:**

`CHEBY(time, 1, 1, 1)`

\[x = \text{time}: \text{Independent variable}\]

\[x_0 = 1: \text{Shift in the Chebyshev polynomial}\]

\[c_0, c_1, c_2: \text{The coefficients for the Chebyshev polynomial}\]

\[c_0 = 1\]

\[c_1 = 1\]

\[c_2 = 1\]

\[C(x) = 1 + 1 \times (\text{time}-1) + 1 \times (2 \times (\text{time}-1)^2 - 1) + \text{time} + 2 \times \text{time}^2 - 4 \times \text{time} + 2 - 1\]

\[= 2 \times \text{time}^2 - 3 \times \text{time} + 1\]

<table>
<thead>
<tr>
<th>Function</th>
<th>Expression</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cos</td>
<td>cos(u)</td>
<td>(y = \cos(u))</td>
<td>The cosine of u.</td>
</tr>
<tr>
<td>cosh</td>
<td>cosh(u)</td>
<td>(y = \cosh(u))</td>
<td>The hyperbolic cosine of u.</td>
</tr>
<tr>
<td>dim</td>
<td>DIM(u,v)</td>
<td>(y = \text{DIM}(u,v))</td>
<td>The positive difference between u and v.</td>
</tr>
<tr>
<td>exp</td>
<td>EXP(u)</td>
<td>(y = e^u)</td>
<td>Exponential function</td>
</tr>
<tr>
<td>forcos</td>
<td>FORCOS(x, x_0, \omega, c_0, c_1, \ldots, c_m)</td>
<td>Defines the Fourier Cosine series.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

`FORCOS(time, 0, 360D, 1, 2, 3)`

\[x = \text{time}: \text{Independent variable}\]

\[x_0 = 0: \text{Shift in the Fourier cosine series}\]

\[\omega = 360D: \text{Frequency of the Fourier cosine series}\]

\[c_0, c_1, c_2: \text{Define coefficient for the Fourier series}\]

\[c_0 = 1\]

\[c_1 = 2\]

\[c_2 = 3\]

\[F(x) = 1 + 2 \times \cos(1 \times 360D \times \text{time}) + 3 \times \cos(2 \times 360D \times \text{time})\]
| forsintab | \( \text{FOR{SIN}(x, x_0, \omega, a_0, a_1, \ldots, a_n)} \) | Defines the Fourier Sine series.  
\[
F(x) = a_0 + \sum_{j=1}^{n} a_j * T_j(x - x_0), \quad 0 \leq j \leq n
\]
where  
\[
T_j(x-x_0) = \sin[j * \omega * (x - x_0)]
\]

**Example:**  
\[
\text{FOR{SIN} (time, 0.25, \pi, 0, 1, 2, 3)}
\]
\[
x = \text{time}: \text{Independent variable}
\]
\[
x_0 = 0.25: \text{Shift in the Fourier sine series}
\]
\[
\omega = \pi: \text{Frequency of the sine series}
\]
\[
a_0, a_1, a_2: \text{Define coefficient for the sine series}
\]
\[
a_0 = 1
\]
\[
a_1 = 2
\]
\[
a_2 = 3
\]
\[
F(x) = \sin (\pi*(\text{time}+0.25)) + 2 * \sin (2 * \pi*(\text{time}+0.25)) + 3 * \sin (3 * \pi*(\text{time}+0.25))
\]

| havstabhavsin | \( \text{hav{SIN}(u, x_{\min}, y_{\min}, x_{\max}, y_{\max})} \) | havsin defines a blending function. For values of \( u \) less than \( x_{\min} \), havsin has the value \( y_{\min} \). For values of \( u \) greater than \( x_{\max} \), havsin has the value \( y_{\max} \). For values of \( u \) between \( x_{\min} \) and \( x_{\max} \), havsin returns the value of a cosine function fitted between the two points \((x_{\min}, y_{\min})\) and \((x_{\max}, y_{\max})\). A half period of the cosine function is used.  

**Assumptions:**  

![Havsin blending function](image)
<table>
<thead>
<tr>
<th>Function</th>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>impact</td>
<td>( \text{IMPACT}(x, \dot{x}, x_1, k, \exp, c_{\text{max}}, d) )</td>
<td>Define collision between two rigid bodies. Ref. BISTOP</td>
</tr>
<tr>
<td></td>
<td>( \text{IMPACT} = k(x_1 - x)^{\exp} \cdot \text{STEP}(x, x_1 - d, c_{\text{max}}, x_1, 0) \cdot \dot{x} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>when ( x &lt; x_1 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \text{IMPACT} = - ) when, ( x &gt; x_1 )</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>( \text{IMPACT}(DZ,VZ,1.0,100,1.5,25,0.1) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x = DZ: \text{Distance variable} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \dot{x} = VZ: \text{The time derivative of } x )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x_1 = 1.0: \text{Free length of } x )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( k = 100: \text{Stiffness} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \exp = 1.5: \text{Exponent of the force} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( c_{\text{max}} = 25: \text{Maximum damping coefficient} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( d = 0.1: \text{Boundary penetration} )</td>
<td></td>
</tr>
<tr>
<td>log</td>
<td>( \text{LOG}(u) )</td>
<td>( y = \log_{e} u )</td>
</tr>
<tr>
<td></td>
<td>Assumptions:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0 &lt; u )</td>
<td></td>
</tr>
<tr>
<td>log10</td>
<td>( \text{LOG10}(u) )</td>
<td>The common logarithm (base10) of ( u ).</td>
</tr>
<tr>
<td>max</td>
<td>( \text{MAX}(A1,A2,{A3}) )</td>
<td>The maximum value in the set of arguments: ( A1, A2, A3 ) (optional).</td>
</tr>
<tr>
<td></td>
<td>( A1, A2, \text{and } A3 ) (optional) must all be of the same type (either integer or real).</td>
<td></td>
</tr>
<tr>
<td>min</td>
<td>( \text{MIN}(A1,A2,{A3}) )</td>
<td>The minimum value in the set of arguments: ( A1, A2, A3 ) (optional).</td>
</tr>
<tr>
<td></td>
<td>( A1, A2, \text{and } A3 ) (optional) must all be of the same type (either integer or real).</td>
<td></td>
</tr>
<tr>
<td>mod</td>
<td>( \text{MOD}(A, P) )</td>
<td>The remainder of the arguments has the sign of the first argument.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>poly</strong></td>
<td>( \text{POLY}(x, x_0, a_0, a_1, \ldots, a_m) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defines the POLYNOMIAL function.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( P(x) = \sum_{j=0}^{n} a_j (x - x_0)^j )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a_0 + a_1*(x - x_0) + a_2*(x - x_0)^2 + \cdots + a_n*(x - x_0)^n )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \text{POLY}(\text{time}, 0, 1, 1, 1) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x = \text{time}: ) Independent variable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x_0 = 0: ) Shift in the polynomial</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a_0, a_1, a_2: ) Define coefficient for the polynomial series</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a_0 = 1 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a_1 = 1 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a_2 = 1 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( P(x) = 1 + 1*\text{time} + 1*\text{time}^2 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( = \text{time}^2 + \text{time} + 1 )</td>
<td></td>
</tr>
<tr>
<td><strong>shf</strong></td>
<td>( \text{SHF}(x, x_0, a, \omega, \phi, b) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Define the Simple Harmonic function.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \text{SHF} = a \times \sin(\omega \times (x - x_0) - \phi) + b )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \text{SHF}(\text{time}, 10D, \pi, 360D, 0, 3) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x = \text{time}: ) Independent variable in the function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x_0 = 10D: ) Offset in the independent variable x</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a = \pi: ) Amplitude of the harmonic function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \omega = 360D: ) The frequency of the harmonic function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \phi = 0: ) Phase shift in the harmonic function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b = 3: ) Average displacement of the harmonic function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \text{SHF} = \pi \times \sin(360D*(\text{time} - 10D)) + 3 )</td>
<td></td>
</tr>
<tr>
<td><strong>sign</strong></td>
<td>( \text{SIGN}(u,v) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the magnitude of ( u ) with the sign of ( v ).</td>
<td></td>
</tr>
<tr>
<td><strong>sin</strong></td>
<td>( \text{SIN}(u) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( y = \sin u )</td>
<td></td>
</tr>
<tr>
<td><strong>sinh</strong></td>
<td>( \text{SINH}(u) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( y = \sinh u )</td>
<td></td>
</tr>
<tr>
<td><strong>sqrt</strong></td>
<td>( \text{SQRT}(u) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( y = \sqrt{u} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Assumptions:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( 0 \leq u )</td>
<td></td>
</tr>
</tbody>
</table>
The step function is defined as:
\[
\text{step}(u, x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}})
\]
where \( u \) is a subexpression and \( x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, \) and \( y_{\text{max}} \) are real numbers.

step defines a blending function. For values of \( u \) less than \( x_{\text{min}} \), step has the value \( y_{\text{min}} \). For values of \( u \) greater than \( x_{\text{max}} \), step has the value \( y_{\text{max}} \). For values of \( u \) between \( x_{\text{min}} \) and \( x_{\text{max}} \), step returns the value of the cubic function fitted between the two points \((x_{\text{min}}, y_{\text{min}})\) and \((x_{\text{max}}, y_{\text{max}})\). The slope of the cubic function is zero at \( x_{\text{min}} \) and \( x_{\text{max}} \).

### Assumptions:

\[
x_{\text{min}} < x_{\text{max}}
\]

### Example:

**Cubic blending function.**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tan</td>
<td>( y = \tan u )</td>
</tr>
<tr>
<td>tanh</td>
<td>( y = \tanh(u) )</td>
</tr>
</tbody>
</table>
First and Second Order Differential Functions for Predefined Functions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>First Order Function Name</th>
<th>Second Order Function name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BISTOP</td>
<td>DBISTOP</td>
<td>DDBISTOP</td>
</tr>
<tr>
<td>CHEBY</td>
<td>DCHEBY</td>
<td>DDCHEBY</td>
</tr>
<tr>
<td>FORCOS</td>
<td>DFORCOS</td>
<td>DDFORCOS</td>
</tr>
<tr>
<td>FORSINE</td>
<td>DFORSINE</td>
<td>DDFORSINE</td>
</tr>
<tr>
<td>HAVSIN</td>
<td>DHAVSIN</td>
<td>DDAVSIN</td>
</tr>
<tr>
<td>IMPACT</td>
<td>DIMPACT</td>
<td>DDIIMPACT</td>
</tr>
<tr>
<td>POLY</td>
<td>DPOLY</td>
<td>DDPOLY</td>
</tr>
<tr>
<td>SHF</td>
<td>DSHF</td>
<td>DDSHF</td>
</tr>
<tr>
<td>STEP</td>
<td>DSTEP</td>
<td>DDSTEP</td>
</tr>
</tbody>
</table>
Siemens Industry Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a leading global provider of product lifecycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

© 2016 Siemens Product Lifecycle Management Software Inc. Siemens and the Siemens logo are registered trademarks of Siemens AG, D-Cubed, Femap, Geolus, GO PLM, I-deas, Insight, JT, NX, Parasolid, Solid Edge, Teamcenter, Tecnomatix and Velocity Series are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other trademarks, registered trademarks or service marks belong to their respective holders.