RECURDYN

Expression Handbook

written by Solution Group of FunctionBay, Inc.

FunctionBay

Table of Contents

<OPERATORS>

< FORTRAN FUNCTIONS >

< SIMULATION CONSTANTS >

< DISPLACEMENT >

< VELOCITY >

< ACCELERATION >

< GENERIC FORCE >

< SPECIFIC FORCE >

< SYSTEM ELEMENT >

< ARITHMETIC IF >

< INTERPOLATION >

< GENERAL >

< CONTROL & HYDRAULIC >

< SENSOR >

< STRESS & STRAIN >

OPERATORS

Arithmetic Operators
Relational Operators
Logical Operators

Arithmetic Operators

The following arithmetic operators can be inserted in expressions.

Symbols

+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Involution

Example

Arithmetic operators use standard arithmetic symbols. Note that two asterisks symbolize involution, so 2**n means 2 to the n^{th} power. Also, when dividing (/) a number by zero (0), zero (0) must be expressed as 1e-20.

Relational Operators

Relational operators return the value 0 if false and the value 1 if true.

Format and Examples

<	The left operand is less than the right operand. 1.0<2.0 returns the value 1. 2.0<1.0 returns the value 0.
<=	The left operand is less than or equal to the right operand. 1.0<=2.0 returns the value 1. 2.0<=1.0 returns the value 0.
=	The left operand is equal to the right operand. A=B returns the value 1 (if |A-B|<IF Tolerance). A=B returns the value 0 (if |A-B|>=IF Tolerance).
>	The left operand is greater than the right operand. 2.0>1.0 returns the value 1 1.0>2.0 returns the value 0.
>=	The left operand is greater than or equal to the right operand. 2.0>=1.0 returns the value 1. 1.0>=2.0 returns the value 0.
<>	The left operand is not equal to the right operand. A<>B returns the value 1 (if |A-B|>=IF Tolerance). A<>B returns the value 0 (if |A-B|<IF Tolerance).

Logical Operators

The following logical operators can be inserted in expressions.

Operators

||	Logical OR
&&	Logical AND

Examples

Values of A||B		
	A is true	A is false
B is true	1	1
B is false	1	0

Values of A&&B		
	A is true	A is false
B is true	1	0
B is false	0	0

FORTRAN FUNCTIONS

ABS
ACOS
AINT
ANINT
ASIN
ATAN
ATAN2
COS
COSH
DIM
EXP
LOG
LOG10
MAX
MIN
MOD
SIGN
SIN
SINH
SQRT
TAN
TANH

ABS

Returns the absolute value of the input value.

Format

ABS(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{ABS}(x) = |x|$$

Example

ABS(-21)

ABS(DX(1,2)) <Argument (1) body1.Marker1, (2) body2.Marker2>

ACOS

Returns the inverse cosine, in radians, of the input value. The input must be a real number. The returned value is within the range of π - π.

Format

ACOS(x)

Argument

x	A real number or a function that returns a real number

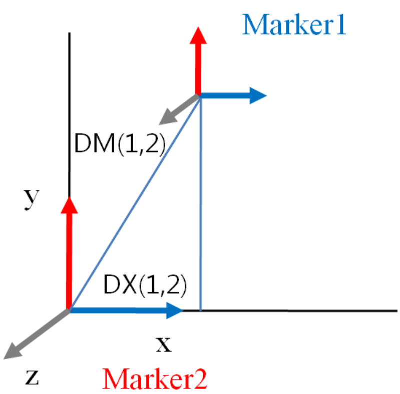
Formula

$$\text{ACOS}=\cos^{-1}(x)$$

Example

As shown in the following figure, the ACOS function calculates the inverse cosine (in radians) of Marker 1 on the InertiaMarker's x-axis.

ACOS(DX(1,2)/DM(1,2)) <Argument: (1)body1.marker1, (2)body2.marker2 >



AINT

Returns the closest integer that is less than the input value.

Format

AINT(x)

Argument

x	A real number or a function that returns a real number

Examples

Ex. 1) Function: AINT(1.2)

Result: 1

Ex. 2) Function: AINT(0.7)

Result: 0

ANINT

Returns the closest integer to the input value.

Format

ANINT(x)

Argument

x	A real number or a function that returns a real number

Examples

Ex. 1) Function: ANINT(1.6)

Result: 2

Ex. 2) Function: ANINT(0.4)

Result: 0

ASIN

Returns the inverse sine, in radians, of the input value. The input must be a real number. The returned value is within the range of -π/2 - π/2.

Format

ASIN(x)

Argument

x	A real number or a function that returns a real number

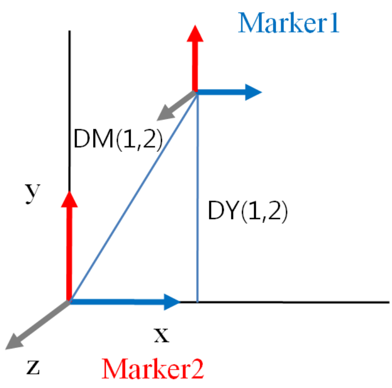
Formula

$$\text{ASIN}=\sin^{-1}\left(x\right)$$

Example

As shown in the following figure, the ASIN function calculates the inverse sine (in radians) of Marker 1 on the InertiaMarker's x-axis.

ASIN(DY(1,2)/DM(1,2)) <Argument: (1)body1.marker1, (2)body2.marker2 >



ATAN

Returns the inverse tangent, in radians, of the input value. The input must be a real number. The returned value is within the range of -π/2 - π/2.

Format

ATAN(x)

Argument

x	A real number or a function that returns a real number

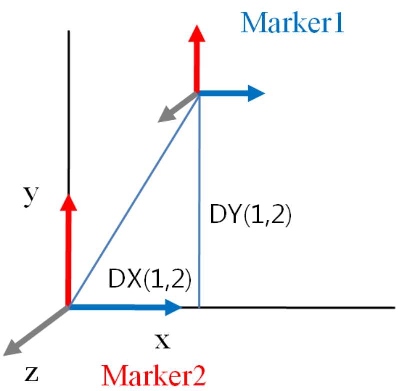
Formula

$$\mathrm{ATAN} = \tan^{-1}(x)$$

Example

As shown in the following figure, the ATAN function calculates the inverse tangent (in radians) of Marker 1 on the InertiaMarker's x-axis.

ATAN(DX(1,2)/DY(1,2)) <Argument: (1)body1.marker1, (2)body2.marker2 >



ATAN2

Returns the inverse tangent, in radians, of the input values. Both input values (numerator and denominator) must be real numbers. The returned value is within the range of $-\pi$ - π.

Format

ATAN2(dx,dy)

Arguments

dx	A real number or a function that returns a real number • This value must belong to the x-axis in a tangent function.
dy	A real number or a function that returns a real number • This value must belong to the y-axis in a tangent function.

Formula

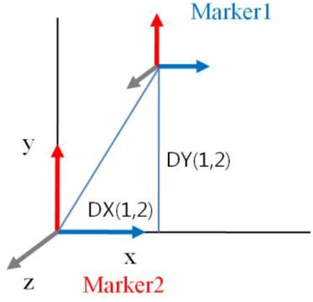
$$\mathrm{ATAN2} = \begin{cases} \tan^{-1}(dx/dy) & dy > 0,\ dx > 0 \\ -\tan^{-1}(dx/dy) & dy < 0,\ dx > 0 \\ \tan^{-1}(dx/dy)+\pi & dy > 0,\ dx < 0 \\ \tan^{-1}(dx/dy)-\pi & dy < 0,\ dx < 0 \\ +\pi/2 & dy > 0,\ dx = 0 \\ -\pi/2 & dy < 0,\ dx = 0 \\ \text{undefined} & dy = 0,\ dx = 0 \end{cases}$$

Example

As shown in the following figure, the ATAN2 function calculates the inverse tangent (in radians) of Marker 1 on the InertiaMarker's x-axis.

ATAN2(DX(1,2),DY(1,2))

<Argument: (1)body1.marker1, (2)body2.marker2 >



COS

Returns the cosine of the input value. The input must be a real number.
The returned value is within the range of -1 - 1.

Format

COS(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{COS} = \cos(x)$$

Examples

Ex. 1) Function: COS(π/2)

Result: 0

Ex. 2) Function: COS(1)

Result: 0.5403

COSH

Returns the hyperbolic cosine of the input value.

Format

COSH(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{COSH} = \frac{e^{x} + e^{-x}}{2}$$

Examples

Ex. 1) Function: COSH(0)

Result: 1

Ex. 2) Function: COSH(1)

Result: 1.54308

DIM

Returns the actual value if the difference between *x* and y is positive and returns 0 if the difference between *x* and y is negative.

Format

DIM(x,y)

Arguments

x	A real number or a function that returns a real number
y	A real number or a function that returns real number

Formula

$$\text{DIM} = \begin{cases} 0, & x \le y \\ x - y, & x > y \end{cases}$$

Examples

Ex. 1) Function: DIM(15,5)
Result: 10

Ex. 2) Function: DIM(5,10)
Result: 0

EXP

Returns the natural logarithm (base e) constant raised to the input value.

Format

EXP(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{EXP} = e^{x}$$

Examples

Ex. 1) Function: EXP(0)

Result: 1

Ex. 2) Function: EXP(3)

Result: 20.0855

LOG

Returns the natural logarithm of the input value.

Format

LOG(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{LOG} = \log_e(x)$$

Examples

Ex. 1) Function: LOG(1)

Result: 0

Ex. 2) Function: LOG(10)

Result: 2.3026

LOG10

Returns the base-10 logarithm of the input value.

Format

LOG10(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\text{LOG10} = \log_{10}(x)$$

Examples

Ex. 1) Function: LOG10 (1)
Result: 0

Ex. 2) Function: LOG10 (10)
Result: 1

MAX

Returns the largest of the input values.

Format

MAX(x1,x2{,x3})

Arguments

x1	A real number or a function that returns a real number
x2	A real number or a function that returns a real number
x3	A real number or a function that returns a real number (omissible)

Formula

$$MAX=max(x1,x2,x3)$$

Example

MAX(1,2,3)

MAX(DX(1), DX(2))   <Argument: (1)body1.marker1, (2)body1.marker2 >

MIN

Returns the smallest of the input values.

Format

MAX(x1,x2{,x3})

Arguments

x1	A real number or a function that returns a real number
x2	A real number or a function that returns a real number
x3	A real number or a function that returns a real number (omissible)

Formula

MIN=min(x1,x2,x3)

Example

MIN (1,2,3)

MIN(DX(1), DX(2))   <Argument: (1)body1.marker1, (2)body1.marker2 >

MOD

Returns the remainder when x is divided by y.

Format

$\mathrm{MOD}(x,y)$

Arguments

x	A real number or a function that returns a real number
y	A real number or a function that returns a real number

Formula

$$\mathrm{MOD} = x - \mathrm{int}(x / \mathrm{y}) \cdot y$$

Examples

Ex. 1) Function: MOD (1,2)

Result: 1

Ex. 2) Function: MOD (5,3)

Result: 2

SIGN

Returns an absolute value that has the magnitude x and the sign of y.

Format

SIGN(x,y)

Arguments

x	A real number or a function that returns a real number
y	A real number or a function that returns a real number

Formula

$$\mathrm{SIGN}=\begin{cases} |x| & (\mathrm{y}>0) \\ 0 & (\mathrm{y}=0) \\ -|x| & (\mathrm{y}<0) \end{cases}$$

Examples

Ex. 1) Function: SIGN (-4.7,1.2)

Result: 4.7

Ex. 2) Function: MOD (5.3,-6.5)

Result: -5.3

Ex. 3) Function: MOD (-2.5,-5.2)

Result: -2.5

SIN

Returns the sine of the input value. The returned value is within the range of -1-1.

Format

SIN(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{SIN} = \sin(x)$$

Examples

Ex. 1) Function: SIN(π/2)
Result: 1

Ex. 2) Function: SIN(1)
Result: 0.8415

SINH

Returns the hyperbolic sine of the input value.

Format

SINH(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{SINH} = \frac{e^{x} - e^{-x}}{2}$$

Examples

Ex. 1) Function: SINH (0)
Result: 0

Ex. 2) Function: SINH (1)
Result: 1.1752

SQRT

Returns the square root of the input value.

Format

SQRT(x)

Argument

x	A real number or a function that returns a real number • If the input value is negative, then no value is returned.

Formula

$$SQRT = \sqrt{x}$$

Examples

Ex. 1) Function: SQRT(2)
Result: 1.41421

Ex. 2) Function: SQRT(-2)
Result: N/A

TAN

Returns the tangent of the input value.

Format

TAN(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\mathrm{TAN} = \tan\left(x\right)$$

Examples

Ex. 1) Function: TAN(π)

Result : 0

Ex. 2) Function: TAN(π/4)

Result: 1

TANH

Returns the hyperbolic tangent of the input value.

Format

TANH(x)

Argument

x	A real number or a function that returns a real number

Formula

$$\text{TANH}=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

Examples

Ex. 1) Function: TANH(1)

Result: 0.76159

Ex. 2) Function: TANH(-3)

Result: -0.99505

SIMULATION CONSTANTS

TIME
PI
DTOR
RTOD

TIME

Returns the simulation time.

Format

TIME

PI

Returns pi (π).

Format

PI

Formula

$$PI = 3.1415926......$$

Example

sin(2*PI*time) = $\sin(2\pi t)$

45*PI/180=$\pi/4$

RTOD

Converts a radian value to a degree value.

Format

RTOD

Formula

$$RTOD=180°/\pi$$

Example

RTOD*pi=180

DTOR

Converts a degree value to a radian value.

Format

DTOR

Formula

$$DTOR=\pi/180°$$

Example

DTOR*180=3.141592

※Caution: If d is inserted at the end of a real number (such as 180d) in an expression, then it is recognized as a degree and automatically converted to a radian value. Therefore, 180d*DTOR will return an incorrect calculation.

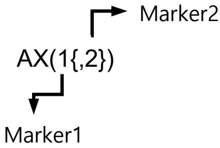
DISPLACEMENT

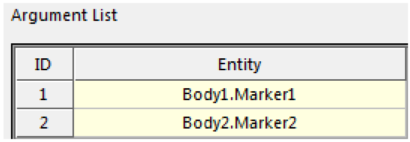
AX
AY
AZ
DM
DX
DY
DZ
PHI
PITCH
PSI
ROLL
THETA
YAW

AX

Returns the rotation angle on the x-axis for one marker's InertiaMarker or the rotation angle on the x-axis for two markers.

Format





Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

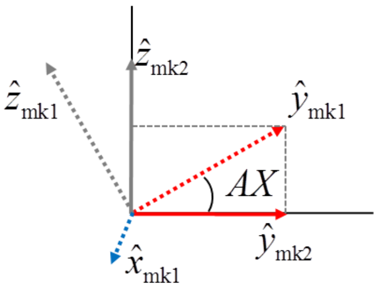
Formula

$$\mathrm{AX} = ATAN2(\hat{y}_{\mathrm{mk1}} \cdot \hat{z}_{\mathrm{mk2}}, \hat{y}_{\mathrm{mk1}} \cdot \hat{y}_{\mathrm{mk2}})$$

$\hat{y}_{\mathrm{mk1}}$: y-direction unit vector of Marker1

$\hat{z}_{\mathrm{mk2}}$: z-direction unit vector of Marker2

$\hat{y}_{\mathrm{mk2}}$: y-direction unit vector of Marker2



Example

AX (body1.Marker1)

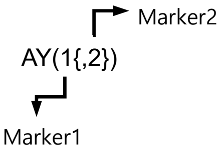
AX (body1.Marker1, body2.Marker2)

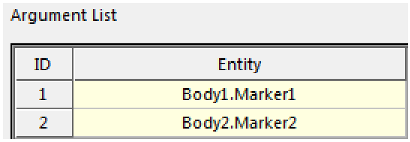
AX (1,2) <Argument: (1)body1.Marker1, (2)body2.Marker2 >

AY

Returns the rotation angle on the y-axis for one marker's InertiaMarker or the rotation angle on the y-axis for two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

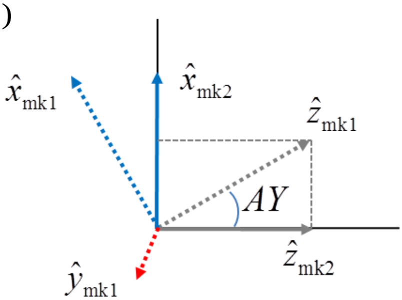
Formula

$$\mathrm{AY} = ATAN2(\hat{z}_{\mathrm{mk1}} \cdot \hat{x}_{\mathrm{mk2}}, \hat{z}_{\mathrm{mk1}} \cdot \hat{z}_{\mathrm{mk2}})$$

$\hat{z}_{\mathrm{mk1}}$: z-direction unit vector of Marker1

$\hat{x}_{\mathrm{mk2}}$: x-direction unit vector of Marker2

$\hat{z}_{\mathrm{mk2}}$: z-direction unit vector of Marker2



Example

AY (body1.Marker1)

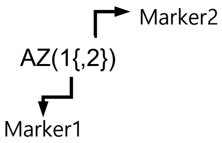
AY (body1.Marker1, body2.Marker2)

AY (1,2)   <Argument: (1)body1.Marker1, (2)body2.Marker2 >

AZ

Returns the rotation angle on the z-axis for one marker's InertiaMarker or the rotation angle on the z-axis for two markers.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

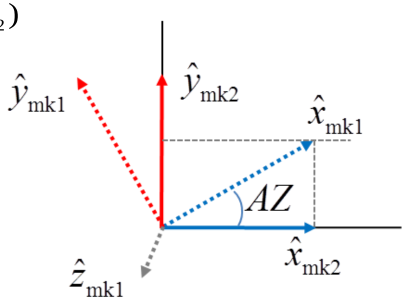
Formula

$$\mathrm{AZ} = ATAN2(\hat{x}_{\mathrm{mk1}} \cdot \hat{y}_{\mathrm{mk2}}, \hat{x}_{\mathrm{mk1}} \cdot \hat{x}_{\mathrm{mk2}})$$

$\hat{x}_{\mathrm{mk1}}$: x-direction unit vector of Marker1

$\hat{y}_{\mathrm{mk2}}$: y-direction unit vector of Marker2

$\hat{x}_{\mathrm{mk2}}$: x-direction unit vector of Marker2



Example

AZ (body1.Marker1)

AZ (body1.Marker1, body2.Marker2)

AZ (1,2)   <Argument: (1)body1.Marker1, (2)body2.Marker2 >

DM

Returns the absolute distance from one marker to the InertiaMarker or the absolute distance between two markers.

Format

Marker2
DM(1{,2})
Marker1

Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

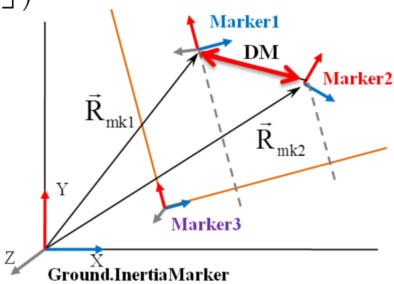
Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{DM} = \left(\left[\vec{R}_{\mathrm{mk1}} - \vec{R}_{\mathrm{mk2}} \right] \cdot \left[\vec{R}_{\mathrm{mk1}} - \vec{R}_{\mathrm{mk2}} \right] \right)^{1/2}$$

$\vec{R}_{\mathrm{mk1}}$: Position vector of Marker1

$\vec{R}_{\mathrm{mk2}}$: Position vector of Marker2



Example

DM (body1.Marker1)

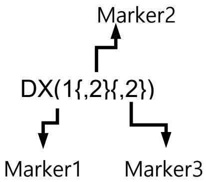
DM (body1.Marker1, body2.Marker2)

DM (1,2)   <Argument: (1)body1.Marker1, (2)body2.Marker2 >

DX

Returns the x-direction location of one marker or the distance between two markers in relation to a third marker’s x-direction.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

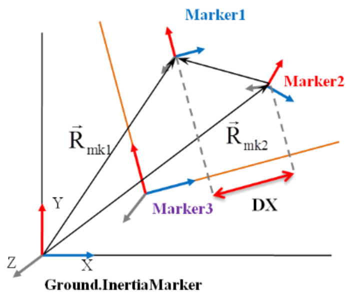
Formula

$$\mathrm{DX} = \left[\vec{R}_{\mathrm{mk1}} - \vec{R}_{\mathrm{mk2}}\right] \cdot \hat{x}_{\mathrm{mk3}}$$

$\vec{R}_{\mathrm{mk1}}$ Position vector of Marker1

$\vec{R}_{\mathrm{mk2}}$ Position vector of Marker2

$\hat{x}_{\mathrm{mk3}}$ x-direction unit vector of Marker3



Example

DX(body1.Marker1)

DX(body1.Marker1, body2.Marker1)

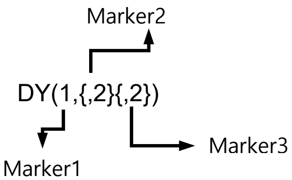
DX(body1.Marker1, body2.Marker1, body3.Marker1)

DX(1,2,3) <Argument: (1)body1.Marker1, (2)body2.Marker1, (3)body3.Marker1>

DY

Returns the y-direction location of one marker or the distance between two markers in relation to a third marker’s y-direction.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

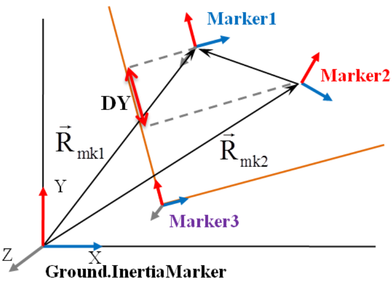
Formula

$$\mathrm{DY} = \left[\vec{R}_{\mathrm{mk1}} - \vec{R}_{\mathrm{mk2}} \right] \cdot \hat{y}_{\mathrm{mk3}}$$

$\vec{R}_{\mathrm{mk1}}$: Position vector of Marker1

$\vec{R}_{\mathrm{mk2}}$: Position vector of Marker2

$\hat{y}_{\mathrm{mk3}}$: y-direction unit vector of Marker3



Example

DY(body1.Marker1)

DY(body1.Marker1, body2.Marker1)

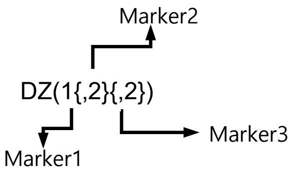
DY(body1.Marker1, body2.Marker1, body3.Marker1)

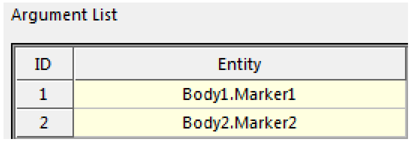
DY(1,2,3) <Argument: (1)body1.Marker1, (2)body2.Marker1, (3)body3.Marker1>

DZ

Returns the z-direction location of one marker or the distance between two markers in relation to a third marker's z-direction.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

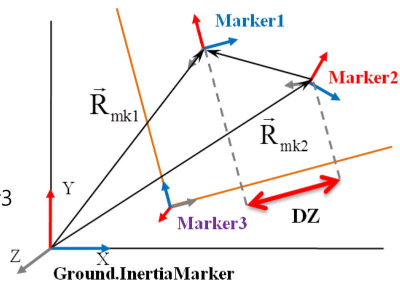
Formula

$$\mathrm{DZ} = \left[\vec{R}_{\mathrm{mk1}} - \vec{R}_{\mathrm{mk2}} \right] \cdot \hat{z}_{\mathrm{mk3}}$$

$\vec{R}_{\mathrm{mk1}}$: Position vector of Marker1

$\vec{R}_{\mathrm{mk2}}$: Position vector of Marker2

$\hat{z}_{\mathrm{mk3}}$: z-direction unit vector of Marker3



Example

DZ (body1.Marker1)

DZ(body1.Marker1, body2.Marker1)

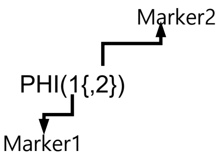
DZ(body1.Marker1, body2.Marker1, body3.Marker1)

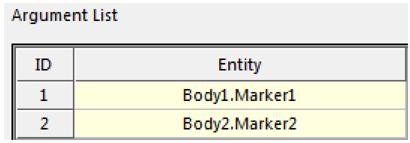
DZ (1,2,3) <Argument: (1)body1.Marker1, (2)body2.Marker1, (3)body3.Marker1>

PHI

Returns the rotation angle of the third rotation in a z-x-z sequence of Euler angles.

Format



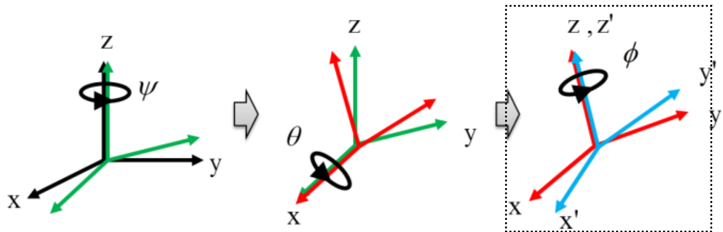

Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

In line with the sequentially defined rotation order, return rotation angle ϕ on the z-axis, which is the third rotation.

Example

PHI(body1.Marker1)

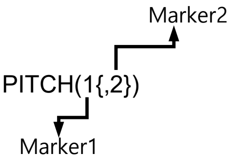
PHI(body1.Marker1, body2.Marker2)

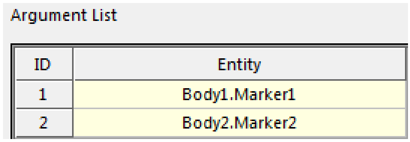
PHI(1,2)　<Argument: (1)body1.Marker1, (2)body2.Marker2 >

PITCH

Returns the rotation angle for the second rotation in a z-y-x sequence of Euler angles.

Format





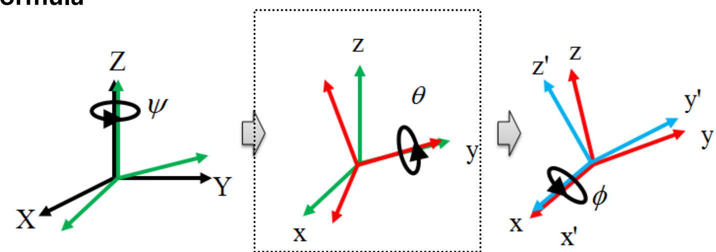
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

In line with the sequentially defined rotation order, return rotation angle θ on the y-axis, which is the second rotation.

Example

PITCH (body1.Marker1)

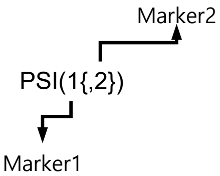
PITCH (body1.Marker1, body2.Marker2)

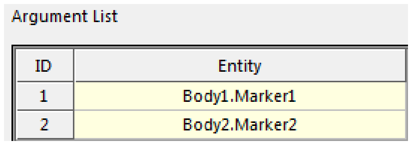
PITCH (1,2)   <Argument: (1)body1.Marker1, (2)body2.Marker2 >

PSI

Returns the rotation angle of the first rotation in a z-x-z sequence of Euler angles.

Format





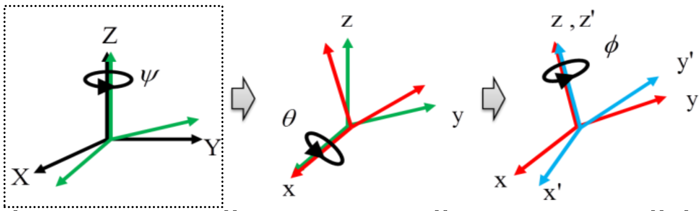
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

In line with the sequentially defined rotation order, return rotation angle ψ on the z-axis, which is the first rotation.

Example

PSI (body1.Marker1)

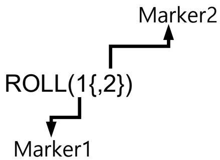
PSI (body1.Marker1, body2.Marker2)

PSI (1,2) <Argument: (1)body1.Marker1, (2)body2.Marker2 >

ROLL

Returns the rotation angle of the third rotation in a z-y-x sequence of Euler angles.

Format



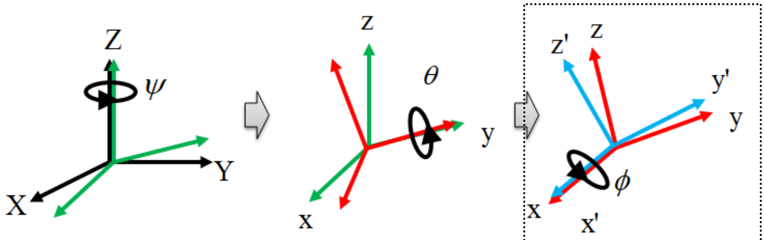
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

In line with the sequentially defined rotation order, return rotation angle ϕ on the x-axis, which is the third rotation.

Example

ROLL (body1.Marker1)

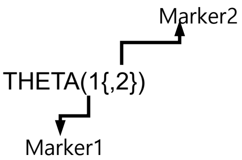
ROLL (body1.Marker1, body2.Marker2)

ROLL (1,2) <Argument: (1)body1.Marker1, (2)body2.Marker2 >

THETA

Returns the rotation angle of the second rotation in a z-x-z sequence of Euler angles.

Format



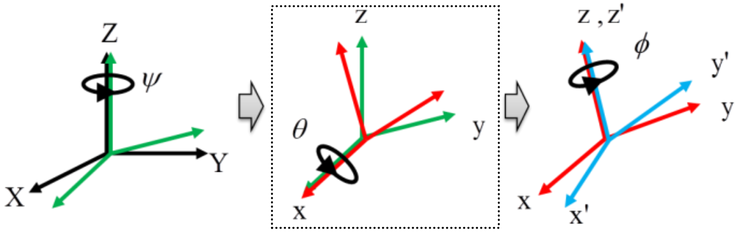
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

In line with the sequentially defined rotation order, return rotation angle θ on the x-axis, which is the second rotation.

Example

THETA (body1.Marker1)

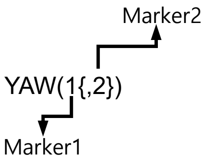
THETA (body1.Marker1, body2.Marker2)

THETA (1,2) <Argument: (1)body1.Marker1, (2)body2.Marker2 >

YAW

Returns the rotation angle of the first rotation in a z-y-x sequence of Euler angles.

Format



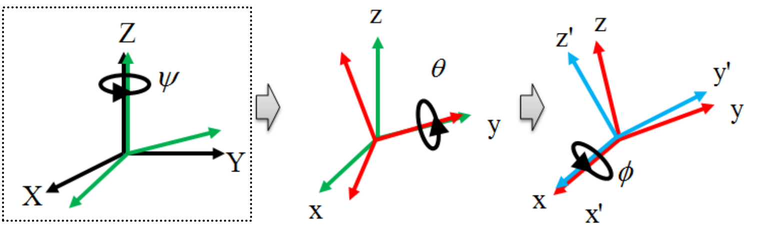
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula



$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

In line with the sequentially defined rotation order, return rotation angle ψ on the z-axis, which is the first rotation.

Example

YAW (body1.Marker1)

YAW (body1.Marker1, body2.Marker2)

YAW (1,2) <Argument: (1)body1.Marker1, (2)body2.Marker2 >

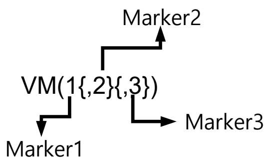
VELOCITY

VM
VR
VX
VY
VZ
WM
WX
WY
WZ

VM

Returns the absolute velocity of one marker or the absolute value of the relative velocity between two markers.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{VM} = \sqrt{\left({}^{(\mathrm{mk3})}\vec{V}_{\mathrm{mk1}} - {}^{(\mathrm{mk3})}\vec{V}_{\mathrm{mk2}} \right)^2}$$

${}^{(mk3)}\vec{V}_{\mathrm{mk1}}$: Marker1's velocity vector from Marker3

${}^{(mk3)}\vec{V}_{\mathrm{mk2}}$: Marker2's velocity vector from Marker3

Example

VM (body1.marker1)

VM (body1.marker1, body2.marker2)

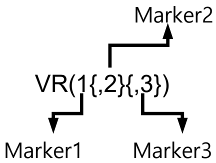
VM(body1.marker1, body2.marker2, body3.marker3)

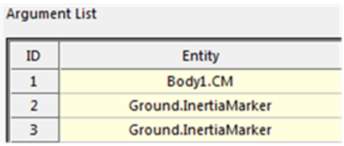
VM (1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

VR

Returns the change in two markers' relative velocity and/or breakaway direction. A positive value (+) indicates that the distance between the two markers is growing. A negative value (-) value indicates that the distance between the two markers is shrinking.

Format





Argument List

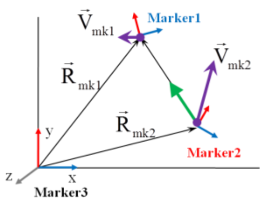
ID	Entity
1	Body1.CM
2	Ground.InertiaMarker
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard marker for the velocity vector • If omitted, then the InertiaMarker is applied.

Formula

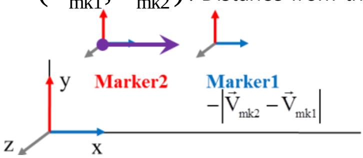
$$\mathrm{VR} = \frac{\left(\left[{}^{(\mathrm{mk3})}\vec{V}_{\mathrm{mk1}} - {}^{(\mathrm{mk3})}\vec{V}_{\mathrm{mk2}}\right]\cdot\left[\vec{R}_{\mathrm{mk1}} - \vec{R}_{\mathrm{mk2}}\right]\right)}{\mathrm{DM}(\vec{R}_{\mathrm{mk1}}, \vec{R}_{\mathrm{mk2}})}$$



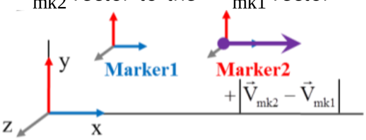
${}^{(\mathrm{mk3})}\vec{V}_{\mathrm{mk1}}$: Marker1's velocity vector from Marker3

${}^{(\mathrm{mk3})}\vec{V}_{\mathrm{mk2}}$: Marker2's velocity vector from Marker3

$\mathrm{DM}(\vec{R}_{\mathrm{mk1}}, \vec{R}_{\mathrm{mk2}})$: Distance from the $\vec{R}_{\mathrm{mk2}}$ vector to the $\vec{R}_{\mathrm{mk1}}$ vector



<Approaching>



<Breakaway>

Example

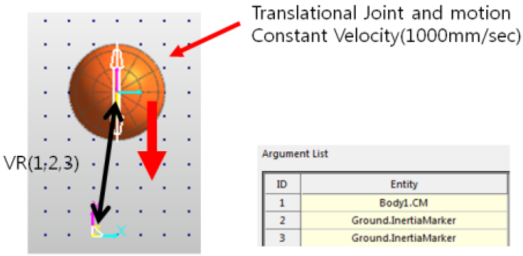
VR (body1.marker1)

VR (body1.marker1, body2.marker2)

VR(body1.marker1, body2.marker2, body3.marker3)

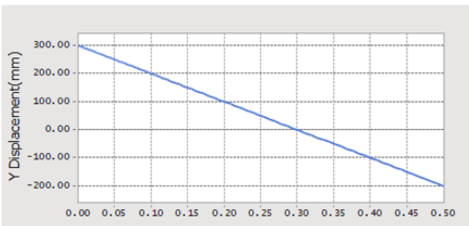
VR (1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

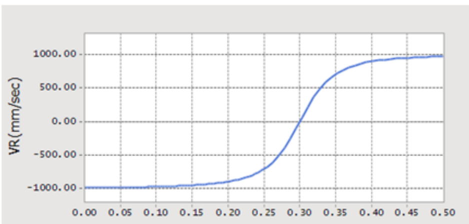
As shown in the following figure, Body1 is placed at (50,300,0) and a translational joint is modeled. This joint is assigned a motion velocity of 1000 mm/sec. By applying the VR function for the InertiaMarker and CM marker to InertiaMarker and Body1, the displacement in the y direction and the VR function results are retrieved, as shown in the following graphs. One of the key examples of a real life application of the VR function is an expression-used to model dampers (shock Absorbers) on automobile suspension systems.



Argument List

ID	Entity
1	Body1.CM
2	Ground.InertiaMarker
3	Ground.InertiaMarker

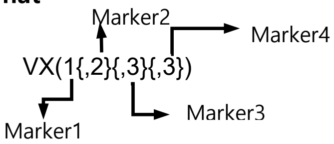


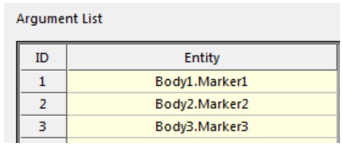


VX

Returns the x-direction velocity for one marker or the relative velocity of two markers in the x-direction.

Format





ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the marker to use in the relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating velocity • The formula measures the velocity relative to Marker4's angular velocity. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$VX = \left(\left[{}^{(mk4)}\vec{V}_{mk1} - {}^{(mk4)}\vec{V}_{mk2} \right] \cdot \hat{x}_{mk3} \right)$$

${}^{(mk4)}\vec{V}_{mk1}$: Velocity vector of Marker1 relative to Marker4's angular velocity

${}^{(mk4)}\vec{V}_{mk2}$: Velocity vector of Marker2 relative to Marker4's angular velocity

$\hat{x}_{mk3}$: x-direction unit vector of Marker3

Example

VX(body1.marker1, body2.marker2)

VX(body1.marker1, body2.marker2, body3.marker3)

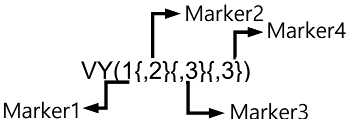
VX(body1.marker1, body2.marker2, body3.marker3, body4.marker4)

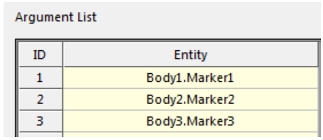
VX (1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

VY

Returns the y direction velocity for one marker or the relative velocity of two markers in the y direction.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the marker to use in the relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating velocity • The formula measures the velocity relative to Marker4's angular velocity. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{VY} = \left(\left[{}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk2}} \right] \cdot \hat{y}_{\mathrm{mk3}} \right)$$

${}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk1}}$: Velocity vector of Marker1 relative to Marker4's angular velocity

${}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk2}}$: Velocity vector of Marker2 relative to Marker4's angular velocity

$\hat{y}_{\mathrm{mk3}}$: y-direction unit vector of Marker3

Example

VY(body1.marker1, body2.marker2)

VY(body1.marker1, body2.marker2, body3.marker3)

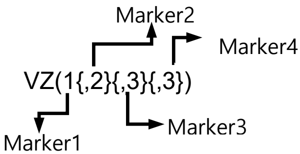
VY(body1.marker1, body2.marker2, body3.marker3, body4.marker4)

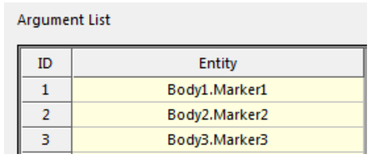
VY(1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

VZ

Returns the z-direction velocity for one marker or the relative velocity of two markers in the z-direction.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the marker to use in the relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating velocity • The formula measures the velocity relative to Marker4's angular velocity. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{VZ} = \left(\left[{}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk2}} \right] \cdot \hat{z}_{\mathrm{mk3}} \right)$$

${}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk1}}$: Velocity vector of the Marker1 relative to Marker4's angular velocity

${}^{(\mathrm{mk4})}\vec{V}_{\mathrm{mk2}}$: Velocity vector of the Marker2 relative to Marker4's angular velocity

$\hat{z}_{\mathrm{mk3}}$: z-direction unit vector of Marker3

Example

VZ(body1.marker1, body2.marker2)

VZ(body1.marker1, body2.marker2, body3.marker3)

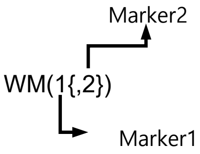
VZ(body1.marker1, body2.marker2, body3.marker3, body4.marker4)

VZ(1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

WM

Returns the absolute value of a marker's angular velocity or the relative angular velocity between two markers.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WM} = \left(\left[\vec{\omega}_{\mathrm{mk1}} - \vec{\omega}_{\mathrm{mk2}}\right]\cdot\left[\vec{\omega}_{\mathrm{mk1}} - \vec{\omega}_{\mathrm{mk2}}\right]\right)^{1/2}$$

$\vec{\omega}_{\mathrm{mk1}}$: Marker1's angular velocity vector from the InertiaMarker

$\vec{\omega}_{\mathrm{mk2}}$: Marker2's angular velocity vector from the InertiaMarker

Example

WM (body1.marker1)

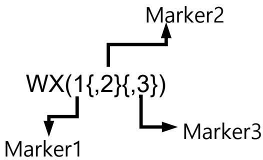
WM (body1.marker1, body2.marker2)

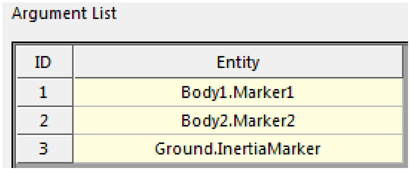
WM (1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

WX

Returns one marker's x-axis angular velocity or the x-axis relative angular velocity between two markers.

Format





Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WX} = \left[\vec{\omega}_{\mathrm{mk1}} - \vec{\omega}_{\mathrm{mk2}}\right] \cdot \hat{x}_{\mathrm{mk3}}$$

$\vec{\omega}_{\mathrm{mk1}}$: Marker1's angular velocity vector from the InertiaMarker

$\vec{\omega}_{\mathrm{mk2}}$: Marker2's angular velocity vector from the InertiaMarker

$\hat{x}_{\mathrm{mk3}}$: x-direction unit vector of Marker3

Example

WX(body1.marker1)

WX(body1.marker1, body2.marker2)

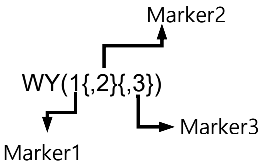
WX(body1.marker1, body2.marker2, body3.marker3)

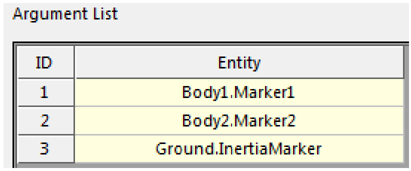
WX(1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

WY

Returns one marker's y-axis angular velocity or the y-axis relative angular velocity between two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WY} = \left[\vec{\omega}_{\mathrm{mk1}} - \vec{\omega}_{\mathrm{mk2}}\right] \cdot \hat{y}_{\mathrm{mk3}}$$

$\vec{\omega}_{\mathrm{mk1}}$: Marker1's angular velocity vector from the InertiaMarker

$\vec{\omega}_{\mathrm{mk2}}$: Marker2's angular velocity vector from the InertiaMarker

$\hat{y}_{\mathrm{mk3}}$: y-direction unit vector of Marker3

Example

WY(body1.marker1)

WYbody1.marker1, body2.marker2)

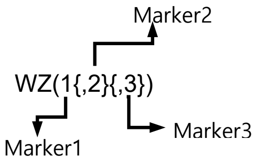
WY(body1.marker1, body2.marker2, body3.marker3)

WY(1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

WZ

Returns one marker's z-axis angular velocity or the z-axis relative angular velocity between two markers.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the reference marker that will serve as the standard for direction • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WZ} = \left[\vec{\omega}_{\mathrm{mk1}} - \vec{\omega}_{\mathrm{mk2}}\right] \cdot \hat{z}_{\mathrm{mk3}}$$

$\vec{\omega}_{\mathrm{mk1}}$: Marker1's angular velocity vector from the InertiaMarker

$\vec{\omega}_{\mathrm{mk2}}$: Marker2's angular velocity vector from the InertiaMarker

$\hat{z}_{\mathrm{mk3}}$: z-direction unit vector of Marker3

Example

WZ(body1.marker1)

WZ (body1.marker1, body2.marker2)

WZ(body1.marker1, body2.marker2, body3.marker3)

WZ(1,2,3) <Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3>

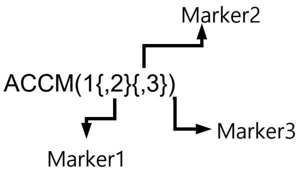
ACCELERATION

ACCM
ACCX
ACCY
ACCZ
WDTM
WDTX
WDTY
WDTZ

ACCM

Returns the absolute value for the acceleration of one marker or the relative acceleration between two markers.

Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	A standard marker for calculating acceleration • The formula calculates the acceleration of Marker1 and Marker2 relative to the angular velocity and the angular acceleration of Marker3. • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{ACCM} = \left(\left[{}^{(\mathrm{mk3})}\vec{a}_{\mathrm{mk1}} - {}^{(\mathrm{mk3})}\vec{a}_{\mathrm{mk2}} \right] \cdot \left[{}^{(\mathrm{mk3})}\vec{a}_{\mathrm{mk1}} - {}^{(\mathrm{mk3})}\vec{a}_{\mathrm{mk2}} \right] \right)^{1/2}$$

${}^{(\mathrm{mk3})}\vec{a}_{\mathrm{mk1}}$: Acceleration vector of Marker1 relative to the angular velocity and the angular acceleration of Marker3

${}^{(\mathrm{mk3})}\vec{a}_{\mathrm{mk2}}$: Acceleration vector of Marker2 relative to the angular velocity and the angular acceleration of Marker3

Example

ACCM(body1.Marker1)

ACCM(body1.Marker1, body2.Marker2)

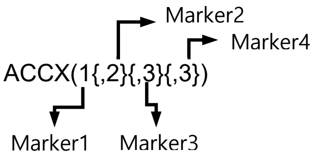
ACCM(body1.Marker1, body2.Marker2, body3.Marker3)

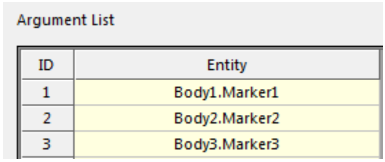
ACCM(1,2,3) <Argument (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

ACCX

Returns the x-axis acceleration of one marker or the x-axis relative acceleration between two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating acceleration • The formula calculates acceleration relative to the angular velocity and the angular acceleration of Marker4. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{ACCX} = \left[{}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk2}} \right] \cdot \hat{x}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk1}}$: Acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker4

${}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk2}}$: Acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker4

$\hat{x}_{\mathrm{mk3}}$: x-direction unit vector of Marker3

Example

ACCX(body1.Marker1, body2.Marker2)

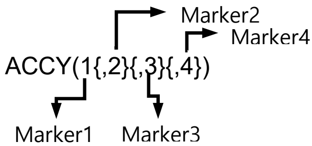
ACCX(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

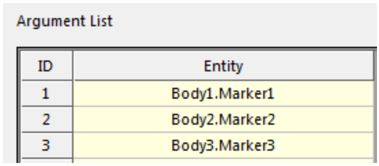
ACCX(1,2,3) <Argument: (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

ACCY

Returns the y-axis acceleration for one marker or the y-axis relative acceleration between two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating acceleration • The formula calculates acceleration relative to the angular velocity and the angular acceleration of Marker4. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{ACCY} = \left[{}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk2}} \right] \cdot \hat{y}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk1}}$: Acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker4

${}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk2}}$: Acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker4

$\hat{y}_{\mathrm{mk3}}$: y-direction unit vector of Marker3

Example

ACCY(body1.Marker1, body2.Marker2)

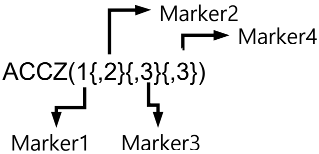
ACCY(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

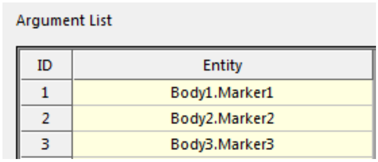
ACCY(1,2,3)<Argument: (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

ACCZ

Returns the z-axis acceleration of one marker or the z-axis relative acceleration between two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of a marker to be calculated • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating acceleration • The formula calculates acceleration relative to the angular velocity and the angular acceleration of Marker4. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{ACCZ} = \left[{}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk2}} \right] \cdot \hat{z}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk1}}$: Acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker4

${}^{(\mathrm{mk4})}\vec{a}_{\mathrm{mk2}}$: Acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker4

$\hat{z}_{\mathrm{mk3}}$: z-direction unit vector of Marker3

Example

ACCZ(body1.Marker1, body2.Marker2)

ACCZ(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

ACCZ(1,2,3)<Argument: (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

WDTM

Returns the absolute value for the angular acceleration of one marker or the relative angular acceleration between two markers.

Format

Marker2

WDTM(1{,2}{,3})

Marker1 Marker3

Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the comparison marker for a relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	A standard marker for calculating angular acceleration • This formula measures angular acceleration relative to Marker3's angular acceleration. • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WDTM}=\left(\left[{}^{(\mathrm{mk3})}\dot{\vec{\omega}}_{\mathrm{mk1}}-{}^{(\mathrm{mk3})}\dot{\vec{\omega}}_{\mathrm{mk2}}\right]\cdot\left[{}^{(\mathrm{mk3})}\dot{\vec{\omega}}_{\mathrm{mk1}}-{}^{(\mathrm{mk3})}\dot{\vec{\omega}}_{\mathrm{mk2}}\right]\right)^{1/2}$$

${}^{(\mathrm{mk3})}\dot{\vec{\omega}}_{\mathrm{mk1}}$: Angular acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker3

${}^{(\mathrm{mk3})}\dot{\vec{\omega}}_{\mathrm{mk2}}$: Angular acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker3

Example

WDTM(body1.Marker1)

WDTM(body1.Marker1, body2.Marker2)

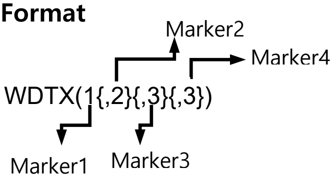
WDTM(body1.Marker1, body2.Marker2, body3.Marker3)

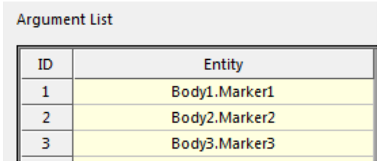
WDTM(1,2,3) <Argument (1)body1.Marker1,(2)body2.Marker2, (3)body3.Marker3>

WDTX

Returns the x-axis angular acceleration of one marker or the x-axis relative angular acceleration between two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the comparison marker for a relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating acceleration • The formula calculates acceleration relative to the angular velocity and the angular acceleration of Marker4. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WDTX} = \left[{}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk2}} \right] \cdot \hat{x}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk1}}$: Angular acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker4

${}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk2}}$: Angular acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker4

$\hat{x}_{\mathrm{mk3}}$: x-direction unit vector of Marker3

Example

WDTX(body1.Marker1, body2.Marker2)

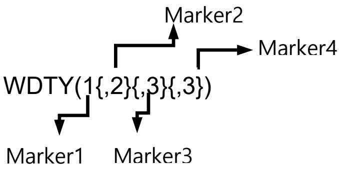
WDTX(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

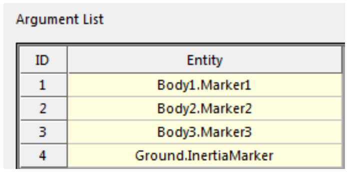
WDTX(1,2,3) <Argument (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

WDTY

Returns the y-axis angular acceleration of one marker or the y-axis relative angular acceleration between two markers.

Format





Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3
4	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the comparison marker for a relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating acceleration • The formula calculates acceleration relative to the angular velocity and the angular acceleration of Marker4. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WDTY} = \left[{}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk2}} \right] \cdot \hat{y}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk1}}$: Angular acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker4

${}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk2}}$: Angular acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker4

$\hat{y}_{\mathrm{mk3}}$: y-direction unit vector of Marker3

Example

WDTY(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

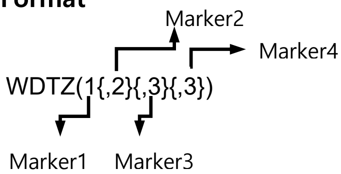
WDTY(body1.Marker1, body2.Marker2)

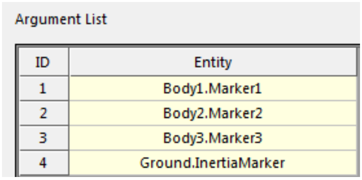
WDTY(1,2,3) <Argument (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

WDTZ

Returns the z-axis angular acceleration of one marker or the z-axis relative angular acceleration between two markers.

Format




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2
3	Body3.Marker3
4	Ground.InertiaMarker

Arguments

Marker1	The name or argument number of a marker to be calculated
Marker2	The name or argument number of the comparison marker for a relative calculation • If omitted, then the InertiaMarker is applied.
Marker3	The name or argument number of the marker for axis definition • If omitted, then the InertiaMarker is applied.
Marker4	A standard marker for calculating acceleration • The formula calculates acceleration relative to the angular velocity and the angular acceleration of Marker4. • Generally, this marker uses the same value as Marker3. If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{WDTZ} = \left[{}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk1}} - {}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk2}} \right] \cdot \hat{z}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk1}}$: Angular acceleration vector of Marker1 relative to the angular velocity and angular acceleration of Marker4

${}^{(\mathrm{mk4})}\dot{\vec{\omega}}_{\mathrm{mk2}}$: Angular acceleration vector of Marker2 relative to the angular velocity and angular acceleration of Marker4

$\hat{z}_{\mathrm{mk3}}$: z-direction unit vector of Marker3

Example

WDTZ(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)
WDTZ(body1.Marker1, body2.Marker2)
WDTZ(1,2,3) <Argument (1)body1.Marker1, (2)body2.Marker2, (3)body3.Marker3>

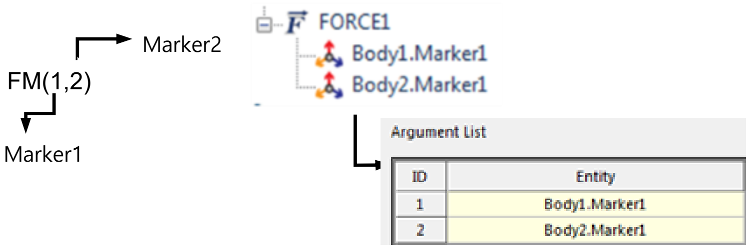
GENERIC FORCE

FM
FX
FY
FZ
TM
TX
TY
TZ

FM

Returns the absolute value of a translational force by using two markers with a force or joint entity.

Format



Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)

Formula

$$\mathrm{FM} = \left({}^{(\mathrm{mk2})}\vec{F}_{\mathrm{mk1}} \cdot {}^{(\mathrm{mk2})}\vec{F}_{\mathrm{mk1}} \right)^{1/2}$$

${}^{(\mathrm{mk2})}\vec{F}_{\mathrm{mk1}}$: Force vector from Marker2 to Marker1

Example

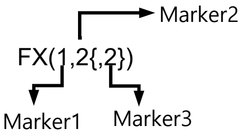
FM(body1.marker1, body2.marker2)

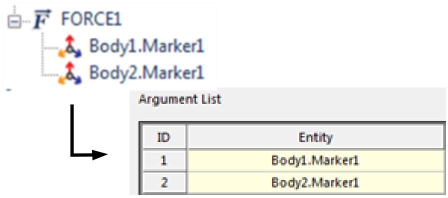
FM(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

FX

Returns the force value, calculated using two markers with a force or joint entity, in the x-direction of a standard marker.

Format





Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)
Marker3	The name or argument number of the standard marker for the direction in which the force is measured • If omitted, then the InertiaMarker is applied.

Formula

$$FX = {}^{(mk2)}\vec{F}_{mk1} \cdot \hat{x}_{mk3}$$

${}^{(mk2)}\vec{F}_{mk1}$: Force vector from Marker2 to Marker1

$\hat{x}_{mk3}$: x-direction unit vector of Marker 3

Example

FX(body1.marker1, body2.marker2)

FX(body1.marker1, body2.marker2, body3.marker3)

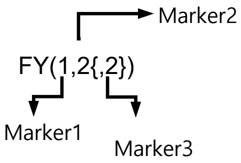
FX(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

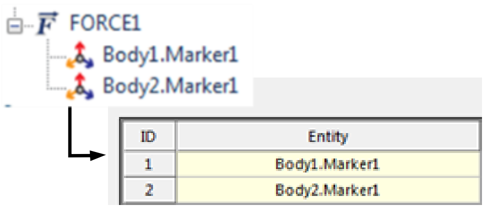
FX(1,2,3) <Argument (1)body1.marker1, (2)body2.marker2 , (3)body3.marker3 >

FY

Returns the force value, calculated using two markers with a force or joint entity, in the y-direction of a standard marker.

Format





Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)
Marker3	The name or argument number of the standard marker for the direction in which the force is measured • If omitted, then the InertiaMarker is applied.

Formula

$$FY={}^{(mk2)}\vec{F}_{\mathrm{mk1}} \cdot \hat{y}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk2})}\vec{F}_{\mathrm{mk1}}$: Force vector from Marker2 to Marker1

$\hat{y}_{\mathrm{mk3}}$: y-direction unit vector of Marker3

Example

FY(body1.marker1, body2.marker2)

FY(body1.marker1, body2.marker2, body3.marker3)

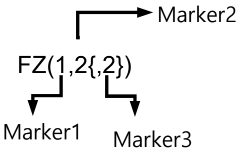
FY(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

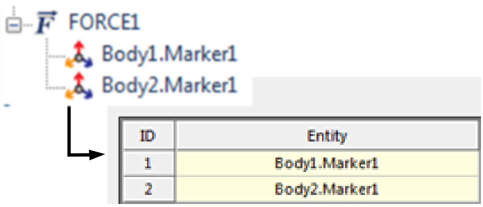
FY(1,2,3) <Argument (1)body1.marker1, (2)body2.marker2 , (3)body3.marker3 >

FZ

Returns the force value, calculated using two markers with a force or joint entity, in the z-direction of a standard marker.

Format





Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)
Marker3	The name or argument number of the standard marker for the direction in which the force is measured • If omitted, then the InertiaMarker is applied.

Formula

$$FZ = {}^{(mk2)}\vec{F}_{mk1} \cdot \hat{z}_{mk3}$$

${}^{(mk2)}\vec{F}_{mk1}$: Force vector from Marker2 to Marker1

$\hat{z}_{mk3}$: z-direction unit vector for Marker3

Example

FZ(body1.marker1, body2.marker2)

FZ(body1.marker1, body2.marker2, body3.marker3)

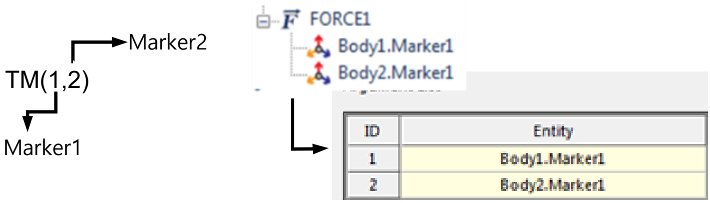
FZ(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

FZ(1,2,3) <Argument (1)body1.marker1, (2)body2.marker2 , (3)body3.marker3 >

TM

Returns the absolute value of the torque force between two markers with a force or joint entity.

Format



Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)

Formula

$$\mathrm{TM} = \left({}^{(\mathrm{mk2})}\vec{T}_{mk2} \cdot {}^{(\mathrm{mk2})}\vec{T}_{\mathrm{mk1}} \right)^{1/2}$$

${}^{(\mathrm{mk2})}\vec{T}_{\mathrm{mk1}}$: Torque vector from Marker2 to Marker1

Example

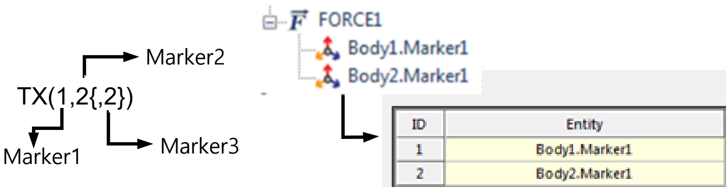
TM(body1.marker1, body2.marker2)

TM(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

TX

Returns the torque force between two markers with a force or joint entity measured along the x-axis of a standard marker.

Format



Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)
Marker3	The name or argument number of the standard marker for the direction in which the force is measured • If omitted, then the InertiaMarker is applied.

Formula

$$TX = {}^{(mk2)}\vec{T}_{mk1} \cdot \hat{x}_{mk3}$$

${}^{(mk2)}\vec{T}_{mk1}$: Torque vector from Marker2 to Marker1

$\hat{x}_{mk3}$: x-direction unit vector of Marker3

Example

TX(body1.marker1, body2.marker2)

TX(body1.marker1, body2.marker2, body3.marker3)

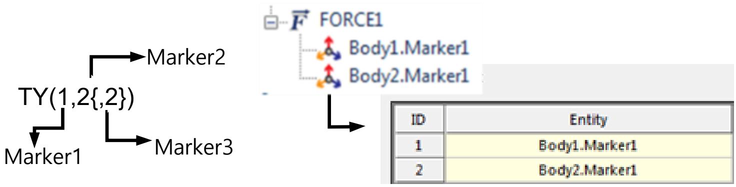
TX(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

TX(1,2,3) <Argument (1)body1.marker1, (2)body2.marker2 , (3)body3.marker3 >

TY

Returns the torque force between two markers with a force or joint entity measured along the y-axis of a standard marker.

Format



Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)
Marker3	The name or argument number of the standard marker for the direction in which the force is measured • If omitted, then the InertiaMarker is applied.

Formula

$$TY = {}^{(mk2)}\vec{T}_{mk1} \cdot \hat{y}_{mk3}$$

${}^{(mk2)}\vec{T}_{mk1}$: Torque vector from Marker2 to Marker1

$\hat{y}_{mk3}$: y-direction unit vector of Marker3

Example

TY(body1.marker1, body2.marker2)

TY(body1.marker1, body2.marker2, body3.marker3)

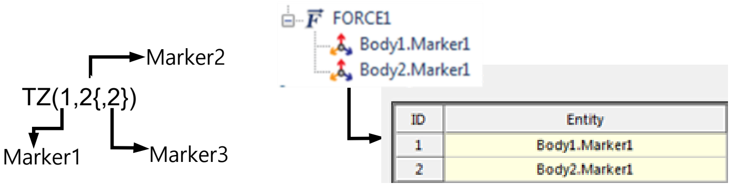
TY(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

TY(1,2,3) <Argument (1)body1.marker1, (2)body2.marker2 , (3)body3.marker3 >

TZ

Returns the torque force between two markers with a force or joint entity measured from the z-axis of a standard marker.

Format



Arguments

Marker1	The name or argument number of a force to be measured, joint entity action, or base marker
Marker2	The name or argument number of a force to be measured, joint entity action, or base marker (not Marker1)
Marker3	The name or argument number of the standard marker for the direction in which the force is measured • If omitted, then the InertiaMarker is applied.

Formula

$$\mathrm{TZ} = {}^{(\mathrm{mk2})}\vec{T}_{mk1} \cdot \hat{z}_{\mathrm{mk3}}$$

${}^{(\mathrm{mk2})}\vec{T}_{\mathrm{mk1}}$: Torque vector from Marker2 to Marker1

$\hat{z}_{\mathrm{mk3}}$: x-direction unit vector of Marker3

Example

TZ(body1.marker1, body2.marker2)

TZ(body1.marker1, body2.marker2, body3.marker3)

TZ(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

TZ(1,2,3) <Argument (1)body1.marker1, (2)body2.marker2 , (3)body3.marker3 >

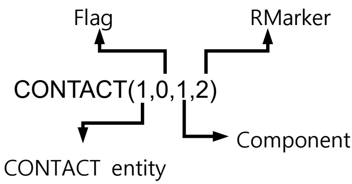
SPECIFIC FORCES

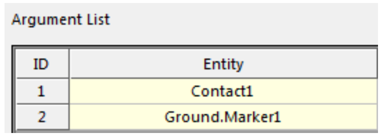
CONTACT
MOTION
JFRICTION
COUPLER
GEAR
SHT3D_CF
JOINT
PTCV
CVCV
AXIAL
TFORCE
RFORCE
SCREWFORCE
BEAM
BUSH
SPRING
MATRIXFORCE

CONTACT

Returns the action force of a contact entity.

Format




Argument List

ID	Entity
1	Contact1
2	Ground.Marker1

Arguments

CONTACT entity	The name or argument number of the contact entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

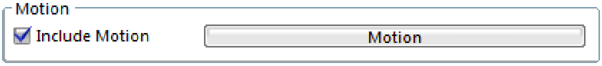
Example

CONTACT (CONTACT1, 0, 1, Ground.InertiaMarker)

CONTACT (1, 0, 1, 2) <Argument: (1) CONTACT1, (2) Ground.InertiaMarker >

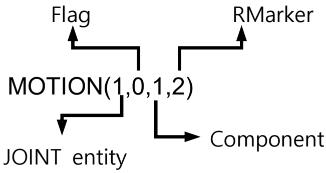
MOTION

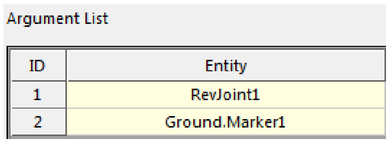
Returns the reaction force of a driving motion acting on a joint. In this case, the driving motion of the joint must be activated.



※ Applicable joints: Revolute, Translational Cylindrical, PTCV

Format





Argument List

ID	Entity
1	RevJoint1
2	Ground.Marker1

Arguments

JOINT entity	The name or argument number of the joint entity that performs the driving motion to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

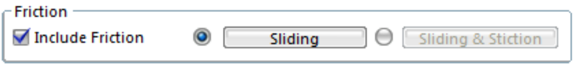
Example

MOTION (RevJoint1, 0, 1, Ground.InertiaMarker)

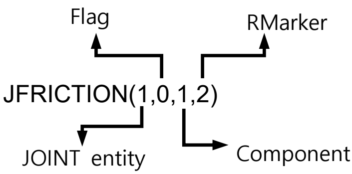
MOTION (1, 0, 1, 2) <Argument: (1) RevJoint1, (2) Ground.InertiaMarker >

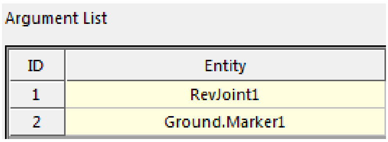
JFRICTION

Returns the reaction force of a friction force acting on a joint. In this case, the friction force of the joint must be activated



Format





Argument List

ID	Entity
1	RevJoint1
2	Ground.Marker1

Arguments

JOINT entity	The name or argument number of the joint entity with the friction to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

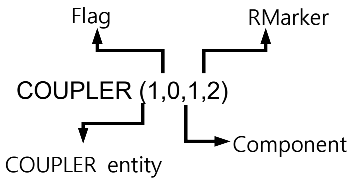
JFRICTION (RevJoint1, 0, 1, Ground.InertiaMarker)

JFRICTION (1, 0, 1, 2) <Argument: (1) RevJoint1, (2) Ground.InertiaMarker >

COUPLER

Returns the reaction force of a coupler entity.

Format



Argument List

ID	Entity
1	Coupler1
2	Ground.Marker1

Arguments

COUPLER entity	The name or argument number of the coupler entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

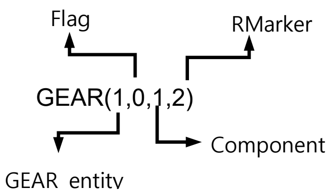
COUPLER (COUPLER1, 0, 1, Ground.InertiaMarker)

COUPLER (1, 0, 1, 2) <Argument: (1) COUPLER1, (2) Ground.InertiaMarker >

GEAR

Returns the reaction force of a gear joint entity.

Format



Argument List

ID	Entity
1	Gear1
2	Ground.Marker1

Arguments

GEAR entity	The name or argument number of the gear entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the 1st joint is returned. • If the flag value is 1, then the force applied to the 2nd joint is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

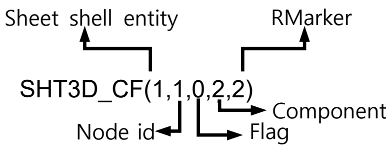
GEAR (GEAR1, 0, 1, Ground.InertiaMarker)

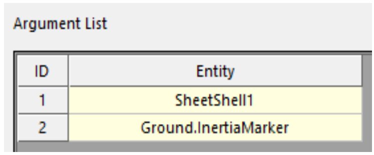
GEAR (1, 0, 1, 2) <Argument: (1) GEAR1, (2) Ground.InertiaMarker >

SHT3D_CF

Returns the contact force of a node. The node must be defined as the output of the sheet shell body defined with the MTT3D toolkit shell sheet entity.

Format




Argument List

ID	Entity
1	SheetShell1
2	Ground.InertiaMarker

Arguments

Sheet shell entity	The name or argument number of the sheet shell entity to be measured
Node id	A nodes selected from the sheet shell body output page
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

SHT3D_CF(SheetShell1,357,0,2, Ground.InertiaMarker)

SHT3D_CF(1,357,0,2,2) <Argument: (1) SheetShell1, (2) Ground.InertiaMarker >

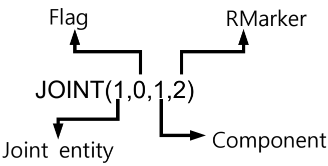
JOINT

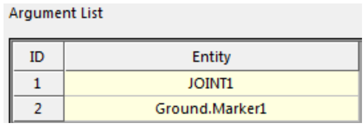
Returns the reaction force of an input joint entity.

※ Applicable joints:

Revolute, Translational, Spherical, Cylindrical, Universal, Planar, Screw, Fixed, Atpoint, Orientation, Distance, Inline, Inplane, Parallel, Perpendicular, Constant Velocity Joint

Format




Argument List

ID	Entity
1	JOINT1
2	Ground.Marker1

Arguments

Joint entity	The name or argument number for the joint entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

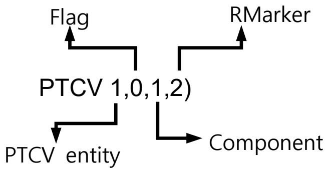
JOINT(JOINT1, 0, 1, Ground.InertiaMarker)

JOINT (1, 0, 1, 2) <Argument: (1)Joint1, (2) Ground.InertiaMarker >

PTCV

Returns the reaction force of a point to curve joint entity.

Format



Argument List

ID	Entity
1	PTCV1
2	Ground.Marker1

Arguments

PTCV entity	The name or argument number for the PTCV entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

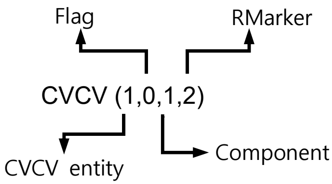
PTCV(PTCV1, 0, 1, Ground.InertiaMarker)

PTCV(1, 0, 1, 2) <Argument: (1) PTCV1, (2) Ground.InertiaMarker >

CVCV

Returns the reaction force of a curve to curve joint entity.

Format



Argument List

ID	Entity
1	CVCV1
2	Ground.Marker1

Arguments

CVCV entity	The name or argument number for the CVCV entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to action marker (body) is returned. • If the flag value is 1, then the force applied to base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

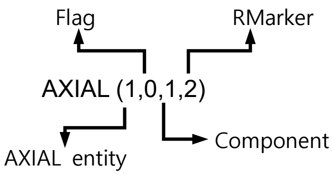
CVCV(CVCV1, 0, 1, Ground.InertiaMarker)

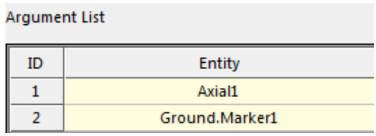
CVCV(1, 0, 1, 2) <Argument: (1)CVCV1, (2) Ground.InertiaMarker >

AXIAL

Returns the action force of an axial force or rotational axial force entity.

Format




Argument List

ID	Entity
1	Axial1
2	Ground.Marker1

Arguments

AXIAL entity	The name or argument number for the axial force or rotational axial force entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

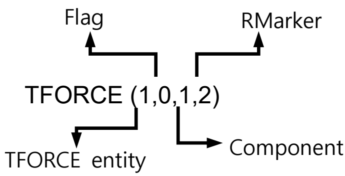
AXIAL(AXIAL1, 0, 1, Ground.InertiaMarker)

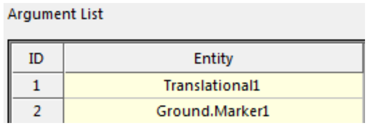
AXIAL(1, 0, 1, 2) <Argument: (1)Joint1, (2) Ground.InertiaMarker >

TFORCE

Returns the action force of a translational force entity.

Format




Argument List

ID	Entity
1	Translational1
2	Ground.Marker1

Arguments

TFORCE entity	The name or argument number for the translational force entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

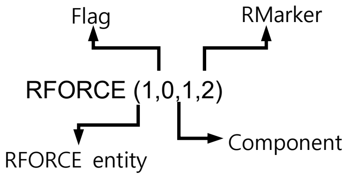
TFORCE (TFORCE1, 0, 1, Ground.InertiaMarker)

TFORCE (1, 0, 1, 2) <Argument: (1) TFORCE1, (2) Ground.InertiaMarker >

RFORCE

Returns the action force of a rotational force entity.

Format



Argument List

ID	Entity
1	Rotational1
2	Ground.Marker1

Arguments

RFORCE entity	The name or argument number of the rotational force entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

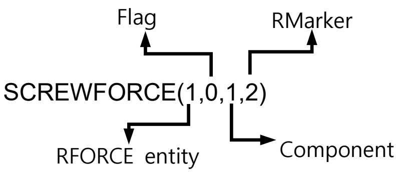
RFORCE (RFORCE1, 0, 1, Ground.InertiaMarker)

RFORCE (1, 0, 1, 2) <Argument: (1) RFORCE1, (2) Ground.InertiaMarker >

SCREWFORCE

Returns the action force of a screw force entity.

Format



Argument List

ID	Entity
1	SCREW1
2	Ground.Marker1

Arguments

SCREWFORCE entity	The name or argument number for the Screw Force entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

SCREWFORCE (SCREWFORCE1, 0, 1, Ground.InertiaMarker)

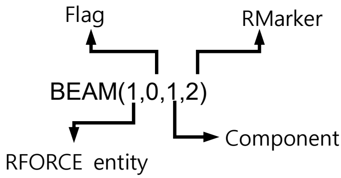
SCREWFORCE (1, 0, 1, 2)

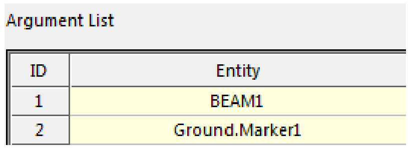
<Argument: (1) SCREWFORCE1, (2) Ground.InertiaMarker >

BEAM

Returns the action force of a beam entity.

Format





Argument List

ID	Entity
1	BEAM1
2	Ground.Marker1

Arguments

BEAM entity	The name or argument number for the Beam entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

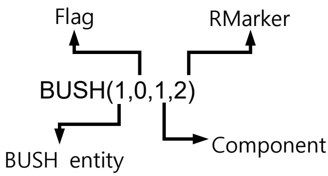
BEAM(Beam1, 0, 1, Ground.InertiaMarker)

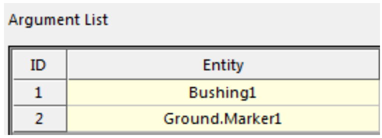
BEAM(1, 0, 1, 2)   <Argument: (1)Beam1, (2) Ground.InertiaMarker >

BUSH

Returns the action force of a Bush entity.

Format




Argument List

ID	Entity
1	Bushing1
2	Ground.Marker1

Arguments

BUSH entity	The name or argument number for the Bush entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

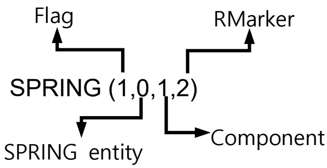
BUSH (BUSH1, 0, 1, Ground.InertiaMarker)

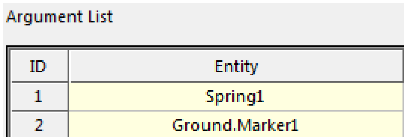
BUSH (1, 0, 1, 2) <Argument: (1) BUSH1, (2) Ground.InertiaMarker >

SPRING

Returns the action force of a translational or rotational spring entity.

Format




Argument List

ID	Entity
1	Spring1
2	Ground.Marker1

Arguments

SPRING entity	The name or argument number for the spring entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

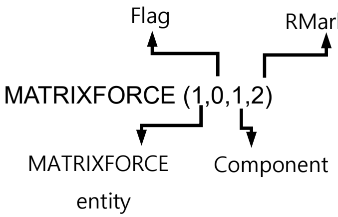
SPRING (SPRING1, 0, 1, Ground.InertiaMarker)

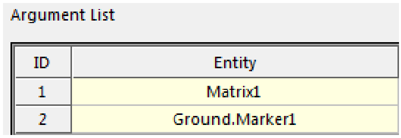
SPRING (1, 0, 1, 2) <Argument: (1) SPRING1, (2) Ground.InertiaMarker >

MATRIXFORCE

Returns the action force of a matrix force entity.

Format





Argument List

ID	Entity
1	Matrix1
2	Ground.Marker1

Arguments

MATRIXFORCE entity	The name or argument number of the matrix force entity to be measured
Flag	A value that specifies the body to which the output load is applied • If the flag value is 0, then the force applied to the action marker (body) is returned. • If the flag value is 1, then the force applied to the base marker (body) is returned.
Component	The components subject to the returned force • One component can be inserted in line with 6 numbered powers and sizes (1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ).
RMarker	The name or argument number of the standard marker for the direction in which the force is measured

Example

MATRIXFORCE(MATRIXFORCE1, 0, 1, Ground.InertiaMarker)

MATRIXFORCE(1,0,1,2) <Argument(1): MATRIXFORCE1,(2)Ground.InertiaMarker>

SYSTEM ELEMENTS

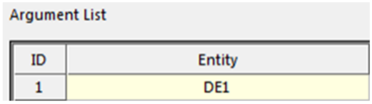
DIF
DIF1
VARVAL

DIF

Returns the integral calculation for the differential equation in a differential equation entity.

Format

DIF(1)
└→ Diff. Eq.



Argument

Diff. Eq.	The name of the differential equation entity or argument number

Formula

$$\dot{\xi} = f(\xi, x, t) \quad \text{(Explicit Type)}$$

$$0 = f(\xi, \dot{\xi}, x, t) \quad \text{(Implicit Type)}$$

(where, ξ : differential variable, x = Other state variable)

Example

DIF(Model1.DE1)

DIF(1) <Argument: (1)DE1>

$$V = RI + L\frac{dI}{dt} \ , \ Torque = K \cdot I$$

$$V = \cos(2t),\ R = 1,\ L = 0.08, K = 6$$

An explicit differential equation can be used to create a torque model for the electrical motor described above.

$$\frac{dI}{dt} = \frac{V - RI}{L} \text{(Explicit Type)}$$

To define the explicit differential equation above:

Create a parametric value for each coefficient (R,L,K), and build models for the body, revolute joint, and rotational axial force. The result is derived by running the organized differential equation through the expression and DIF and applying each entity through the Rotational Axial Force model.

<Modeling>

Parametric Value List

Parametric Values

No	DP	Name	Value
1	✔	R	1.
2	✔	L	8.e-002
3	✔	K	6.

Expression List

Expressions

No	Name	Expression
1	Torque	K*DIF(1)
2	EX1	cos(2*time)/L-R*DIF(1)/L

Differential Equation

Name: IDot

Function Type: Explicit

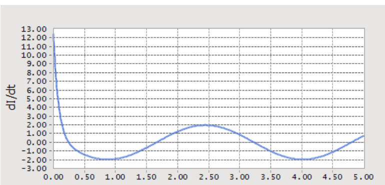
Standard Differential Equation

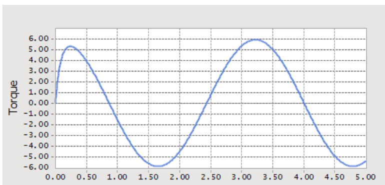
Expression

Name: EX1

cos(2*time)/L-R*DIF(1)/L

<Results>





DIF1

Returns the calculation for a differential equation using a differential equation entity as it is.

Format

DIF1(1)
└→ Diff. Eq.

Argument List

ID	Entity
1	DE1

Argument

Diff. Eq.	The name or argument number of the differential equation entity

Formula

$$\dot{\xi} = f(\xi, x, t) \quad \text{(Explicit Type)}$$

$$0 = f(\xi, \dot{\xi}, x, t) \quad \text{(Implicit Type)}$$

where, ξ : differential variable, x = Other state variable

Example

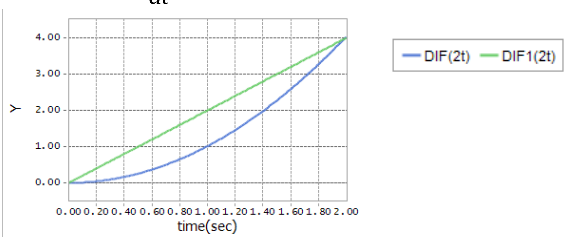
DIF1(Model1.DE1)

DIF1(1) <Argument: (1)DE1>

The following is a comparison of the DIF and DIF1 functions on F(t)= 2t through the expression scope:

$$\text{DIF } (f(t)) = \int 2t \; dt = t^2$$

$$\text{DIF1}(f(t)) = \frac{d}{dt}\int 2t \; dt = 2t$$



VARVAL

Returns the result of a variable equation and enables an expression to use the calculation derived from a variable equation entity.

Format

VARVAL(1)

└→ Variable Eq.

Argument List

ID	Entity
1	VE1
2	VE2

Argument

Variable Eq.	The name or argument number for the variable equation entity

Example

VARVAL (Model1.VE1)

VARVAL(1) <Argument: (1) VE1 >

Execution Sequence

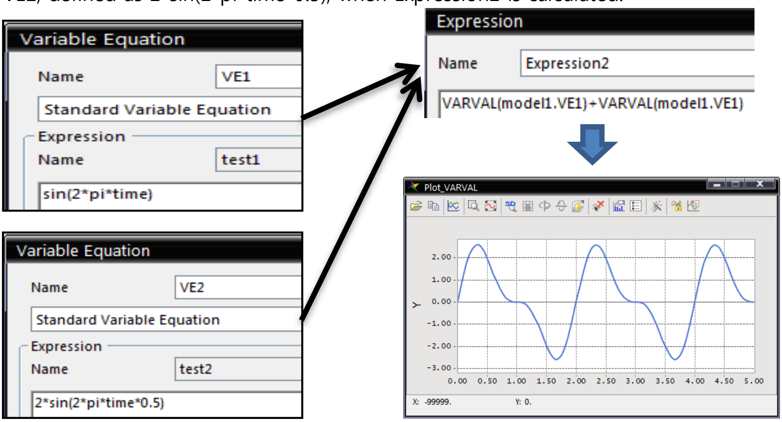
(1) Create Expression 1 to use in the variable equation: sin(2*pi*time).

(2) Create a variable equation entity (VE1).

(3) Connect Expression 1 to the variable equation entity (VE1).

(4) Create Expression2 VARVAL (Model1. VE1).

The following example shows the results of test1 VE1, defined as sin(2*pi*time), and test2 VE2, defined as 2*sin(2*pi*time*0.5), when Expression2 is calculated.



ARITHMETIC IF

IF

IF

Assigns the conditions for calculating functions and returns the value for those functions when the given conditions are satisfied.

Format

$$\mathrm{IF}(\mathrm{f1}\colon \mathrm{f2},\ \mathrm{f3},\ \mathrm{f4})$$

Arguments

f1	Defines the equation that provides the conditions under which to derive the value of the function.
f2	Returns the calculated values when f1 < -IFT. This must be a real number or a function that returns a real number.
f3	Returns the calculated values when -IFT ≤ f1 ≤ IFT. This must be a real number or a function that returns a real number.
f4	Returns the calculated values when f1 > IFT. This must be a real number or a function that returns a real number.

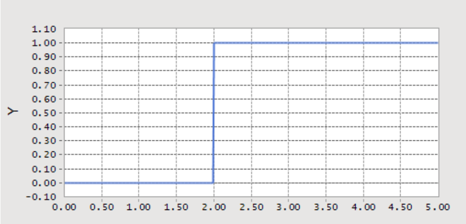
※ IFT (If Tolerance): When an If clause is calculated, the IFT value functions as a parameter that modifies the conditions so that they respond to 0 by allowing a slight range because the result of Expression1 may not be exactly 0. The input values for these parameters can be found on the Home tab of the Simulation menu in the RecurDyn GUI.

Formula

$$\mathrm{IF}(\mathrm{f1}\colon \mathrm{f2},\ \mathrm{f3},\ \mathrm{f4}) = \begin{cases} \mathrm{f2}, & \text{when } \mathrm{f1} < -\mathrm{IFT} \\ \mathrm{f3}, & \text{when } -\mathrm{IFT} \le \mathrm{f1} \le \mathrm{IFT} \\ \mathrm{f4}, & \text{when } \mathrm{IFT} < \mathrm{f1} \end{cases}$$

Example

The figure below shows the results of the following expression using IF functions:
IF(TIME-2.0:0,0,1)



INTERPOLATION

AKISPL
CUBSPL

AKISPL

Uses Akima spline interpolation to return the y values for the x variable input through the spline entity. The Akima spline is only continuous in the 1st derivative. Therefore, it is suitable for force elements, which do not require second order continuity.

Format

AKISPL(X, Z, 1, 0)

Curve name

Order

Argument List

ID	Entity
1	Spline1

Arguments

X	An input variable for the AKISPL function • Generally, this variable is time or a function that returns a real number.
Z	An input variable for the AKISPL function • The second variable is necessary for three dimensional spline functions. • This variable must be a function that returns a real number. Otherwise, 0 is applied.
Curve name	The name or argument number of the spline data defined by the subentity
Order	The interpolation method for the functions (return the value if 0, return calculation for 1st order differential equation if 1, and return calculation for 2nd order differential equation if 2)

Formula

$$\text{AKISPL}=\begin{cases} f(x, spline\_data), & \text{Order=0} \\ df(x, spline\_data)/dx, & \text{Order=1} \\ d^2 f(x, spline\_data)/dx^2, & \text{Order=2} \end{cases}$$

Spline Data

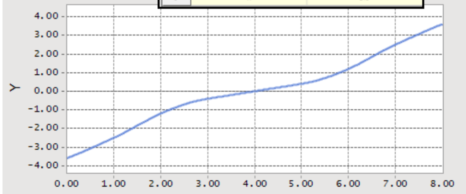
No	X	Y
1	-4.	-3.6
2	-3.	-2.5
3	-2.	-1.2
4	-1.	-0.4
5	0.	0.
6	1.	0.4
7	2.	1.2
8	3.	2.5
9	4.	3.6

Example

The example below was created using the following formula:

AKISPL(time-4,0,1,0)

<Argument: (1) Spline>



CUBSPL

Uses cubic spline interpolation to return the y values for the x variable input through the spline entity. The cubic spline has a continuous 2nd derivative. Therefore, it can be used in specified Motion for joints and bodies, which require continuous second derivatives, as well as for force elements.

Format

CUBSPL(X, Z, 1, 0)

1 → Curve name

0 → Order

Argument List

ID	Entity
1	Spline1

Arguments

X	An input variable for CUBSPL • Generally, this variable is time or a function that returns a real number.
Z	An input variable for CUBSPL • This second variable is necessary for three dimensional spline functions. • This variable must be a function that returns a real number. Otherwise, 0 is applied.
Curve name	The name or argument number for the spline data defined by the subentity
Order	The interpolation methods for the functions (return the value if 0, return calculation for 1st order differential equation if 1, and return calculation for 2nd order differential equation if 2)

Formula

$$\text{CUBSPL}=\begin{cases} f(x, spline\_data), & \text{Order=0} \\ df(x, spline\_data)/dx, & \text{Order=1} \\ d^2 f(x, spline\_data)/dx^2, & \text{Order=2} \end{cases}$$

Spline Data

No	X	Y
1	-4.	-3.6
2	-3.	-2.5
3	-2.	-1.2
4	-1.	-0.4
5	0.	0.
6	1.	0.4
7	2.	1.2
8	3.	2.5
9	4.	3.6

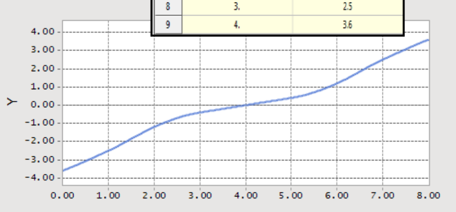
Example

The figure below was created using the following formula:

CUBSPL(time,0,Spline1,0)

CUBSPL(time,0,1,0)

<Argument (1) Spline1>



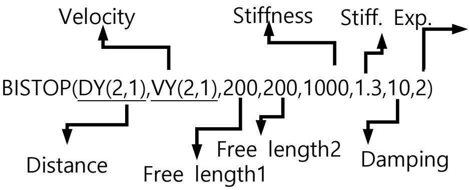
GENERAL

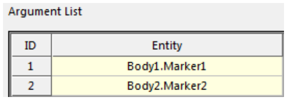
BISTOP
CHEBY
FORCOS
FORSIN
HAVSIN
IMPACT
POLY
SHF
STEP
STEP5
SWEEP

BISTOP

Returns the contact force of a model in a gap defined by the relative location and velocity of two markers.

Format

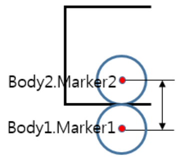
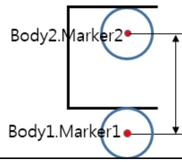




Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Distance(x)	The relative distance between the two markers on the contacting entities	
Velocity($\dot{x}$)	The relative velocity between the two markers on the contacting entities	
Free length1 (x_1)	The contact distance between the two markers on the contacting entities • This value must be a real number or a function that returns a real number. • The free length(x_1) is used to determine whether or not contact is made.	
Free length2 (x_2)	The contact distance between the two markers on the contacting entities • This value must be a real number or a function that returns a real number. • Free length(x_2) is used to determine whether or not contact is made.	
Stiffness(k)	The modulus rigidity on the spring force	
Stiffness exponent(exp)	The nonlinear coefficient value on the surface of the spring force	
Damping(c_{max})	The maximum damping coefficient • This must be a real number or a function that returns a real number	
Penetration(d)	The depth of infiltration that induces the maximum damping coefficient	

Formula

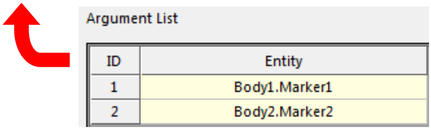
BISTOP =

$$\begin{cases} k(x_1 - x)^{\exp} - \mathrm{STEP}(x, x_1 - d, c_{\max}, x_1, 0)\cdot\dot{x} & \text{when, } x < x_1 \\ 0 & \text{when, } x < x_1 \ \&\ \mathrm{BISTOP} < 0) \\ 0 & \text{when, } x_1 \le x < x_2, \\ -k(x - x_2)^{\exp} - \mathrm{STEP}(x, x_2, 0, x_2 + d, c_{\max})\cdot\dot{x} & \text{when, } x \ge x_2 \\ 0 & \text{when, } x \ge x_2 \ \&\ \mathrm{BISTOP} > 0) \end{cases}$$

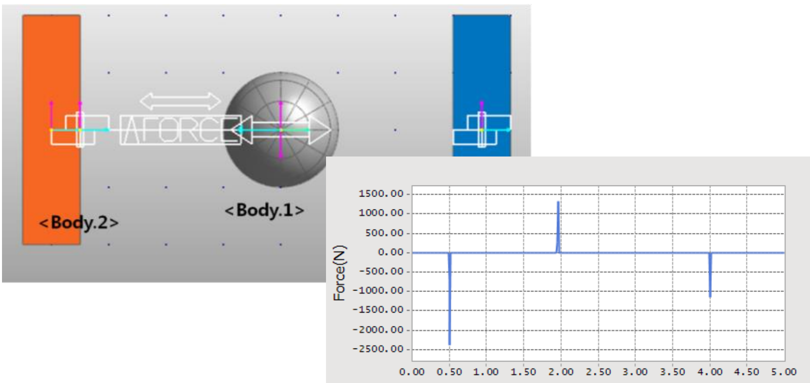
Example

BISTOP(DX(body1.marker1, body2.marker2, body2.marker2), VX(body1.marker1, body2.marker2, body2.marker2),150,550,10000,1.3,100,2)

The following provides an example that defines a model that contacts both sides of a gap through the Axial Force and BISTOP functions. In this example, Body1's initial velocity defines the contact modeling on both sides.
(It is advantageous to select Free length if the contacting sides of the two entities are defined as reference markers when defining the DM function for both markers.)
BISTOP(DX(1,2,2),VX(1,2,2),150,550,10000,1.3,100,2)


Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2



CHEBY

Returns the result of the coefficients and variables evaluated using the Chebyshev polynomial.

Format

CHEBY(x, x_0, c_0, c_1, ... , c_{30})

Arguments

x	The input variable for the Chebyshev equation • Generally, this variable is a real number, function, or time.
x_0	The x-direction offset for the variable (X) in the Chebyshev polynomial
$c_0, c_1, \ldots, c_{30}$	The coefficient values in the Chebyshev polynomial that calculate the linear superposition in line with the Chebyshev polynomial • These values are defined by the number of input coefficients and can include up to 31 coefficients.

Formula

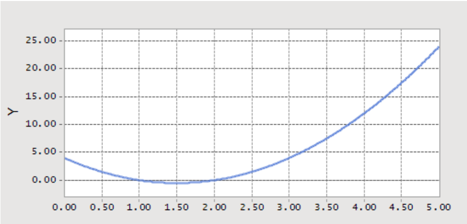
$$\mathrm{C}(\mathrm{x}) = \sum_{j=0}^{n} c_j \cdot T_j(x - x_0), \quad 0 \le j \le n$$

$$\left(\begin{aligned} \text{where, } & T_j(x-x_0) = 2 \cdot (x-x_0) \cdot T_{j-1}(x-x_0) - T_{j-2}(x-x_0) \\ & T_0(x-x_0) = 1, \quad T_1(x-x_0) = x - x_0 \end{aligned} \right)$$

Example

The graph below illustrates the output values of the following expression through the scope:

CHEBY(time,2,1,2,1)



FORCOS

Returns the results of the coefficients and variables evaluated in a Fourier Cosine series.

Format

FORCOS(x, x_0,ω, c_0, c_1, ... , c_{30})

Arguments

x	The input variable for the defined Fourier Cosine series • Generally, this variable is time or a function that results in a real number.
x_0	The x-direction offset for the variable (X) in the Fourier Cosine series
ω	The base frequency for the Fourier Cosine series (in radians)
$c_0, c_1, \ldots, c_{30}$	The coefficient values in the Fourier cosine series that calculates the linear superposition in line with the Fourier cosine series • These values are defined by the number of input coefficients and can include up to 31 coefficients.

Formula

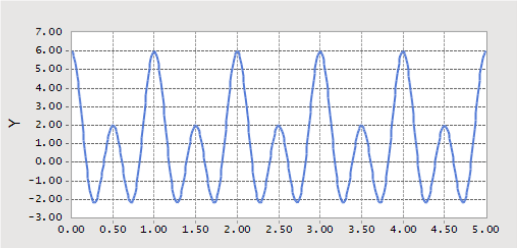
$$\mathrm{F(x)}=c_0+\sum_{j=1}^{n} c_j \cdot T_j(x-x_0) \quad 0 \le j \le n$$

$$\left(where \qquad T_j(x-x_0)=\cos[\,j*\omega*(x-x_0)]\right)$$

Example

The graph below illustrates the output values of the following expression through the scope:

FORCOS(time,0,360d,1,2,3)



FORSIN

Returns the results of the coefficients and variables evaluated in a Fourier sine series.

Format

FORSIN(x, x_0,ω, c_0, c_1, ... , c_{30})

Arguments

x	The input variable for the defined Fourier sine series • Generally, this variable is time or a function that results a real number.
x_0	The x-direction offset for the variable (X) in the Fourier sine series
ω	The base frequency for the Fourier sine series (in radians)
$c_0, c_1, \ldots, c_{30}$	The coefficient values in the Fourier cosine series that calculates the linear superposition in line with the Fourier cosine series • These values are defined by the number of input coefficients and can include up to 31 coefficients.

Formula

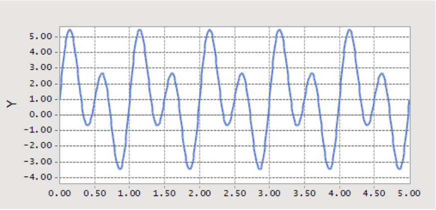
$$F(x)=c_0+\sum_{j=1}^{n} c_j \cdot T_j(x-x_0) \quad 0 \le j \le n$$

$$\left(where \qquad T_j(x-x_0)=\sin[j\cdot\omega\cdot(x-x_0)]\right)$$

Example

The graph below illustrates the output values of the following expression through the scope:

FORSIN(time,0,360d,1,2,3)

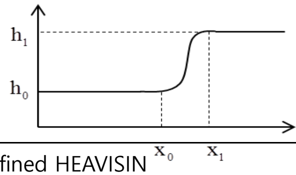


HAVSIN

Interpolates between two markers using a trigonometric function. Generally, this function is used to link two markers in a gradual curve, similar to the Step and Step 5 functions.

Format

$\mathrm{HEAVISIN}(x, x_0, h_0, x_1, h_1)$



Arguments

x	The input variable for the defined HEAVISIN • Generally, this variable is time or a function that returns a real number.
x_0	The starting point for the HEAVISIN function
h_0	The initial value for the input variable (within the range of $x \le x_0$)
x_1	The ending point for the HEAVISIN function
h_1	The initial value for the input variable (within the range of $x \ge x_1$)

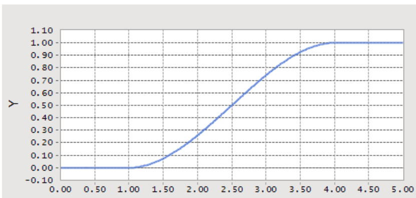
Formula

$$\begin{aligned}\mathrm{HAVSIN} &= h_0, && \text{when } x \le x_0 \\ &= \frac{(h_0 + h_1)}{2} + \frac{(h_1 - h_0)}{2} \cdot \sin\left(\pi \frac{(x - x_0)}{(x_1 - x_0)} - \frac{\pi}{2}\right), && \text{when } x_0 < x \le x_1 \\ &= h_1, && \text{when } x \ge x_1\end{aligned}$$

Example

The graph below illustrates the output values of the following expression through the scope:

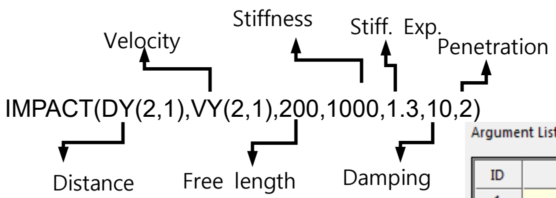
HAVSIN(time, 1.0, 0.0, 4.0, 1.0)



IMPACT

Returns the contact force in a contact model defined by the relative location and velocity of two markers.

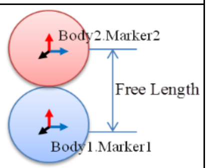
Format



Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

Arguments

Distance(x)	The expression for the relative distance between the two markers on the contacting entities
Velocity($\dot{x}$)	The expression for the relative velocity between the two markers on the contacting entities
Free length(x_1)	The contact distance between the two markers on the contacting entities  • This must be a real number or a function that returns a real number. • The free length(x_1) determines whether or not contact is made.
Stiffness(k)	The modulus rigidity on the spring force
Stiffness exponent(exp)	The nonlinear coefficient value on the surface of the spring force
Damping(c_{max})	The maximum damping coefficient for the calculation of the damping force
Penetration(d)	The depth at which the damping coefficient reaches 0

Formula

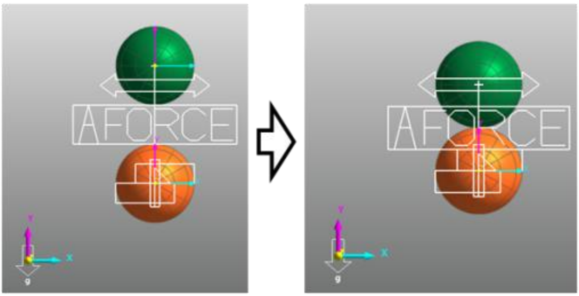
$$\text{IMPACT} = \begin{cases} k(x_1 - x)^{\exp} - \text{STEP}(x, x_1 - d, c_{\max}, x_1, 0)\cdot \dot{x} & \text{when, } x < x_1 \\ 0 & \text{when, } x < x_1 \ \&\ \text{IMPACT} < 0 \\ 0 & \text{when, } x \ge x_1 \end{cases}$$

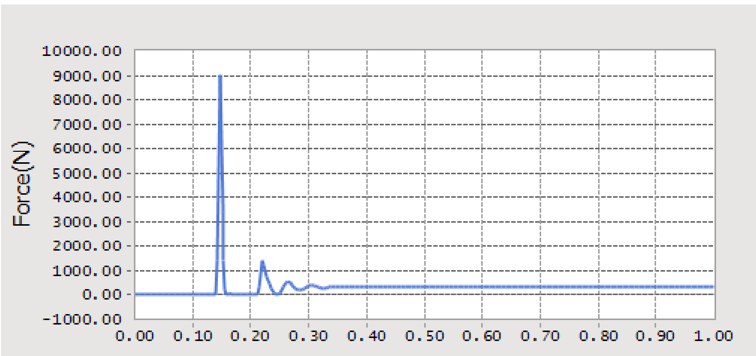
Example

IMPACT(DY(body2.marker2,body1.marker1),VY(body2.marker2,body1.marker1), 200, 1000, 1.3,10,2)

As illustrated in the figure below, after creating the two bodies, define the contact model between the two entities through the Axial Force and IMPACT functions, and produce an Axial Force output graph through the scope.

IMPACT(DY(1,2),VY(1,2),200,1000,1.3,10,2)
<Argument: (1)body1.marker1, (2)body2.marker2 >





POLY

Returns the results of the input variables and coefficients evaluated in a polynomial function.

Format

$POLY(x, x_0, a_0, a_1, \ldots, a_{30})$

Arguments

x	The input variable for the defined polynomial function • Generally, this variable is time or a function that returns a real number.
x_0	The starting point of the polynomial function
$a_0, a_1, \ldots, a_{30}$	The set of polynomial coefficients (up to 31)

Formula

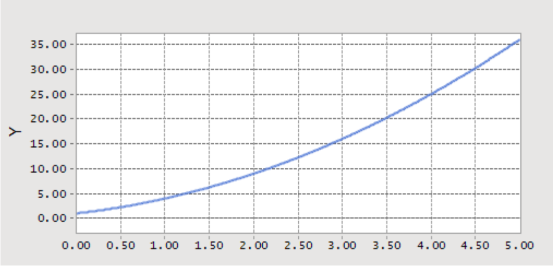
$$POLY = \sum_{j=0}^{n} a_j (x - x_0)^j \qquad \text{where, } 0 \le j \le n$$

$$= a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \cdots + a_n(x - x_0)^n$$

Example

The graph below illustrates the output values of the following expression through the scope:

POLY(time, 0, 1.0, 2.0, 1.0)



SHF

Returns the results of the input variables and coefficients evaluated as a sine wave in a simple harmonic function.

Format

SHF(x,x_0,a,ω,ϕ,b)

Arguments

x	The input variable for the defined SHF function • Generally, this variable is time or a function that returns a real number.
X_0	The starting point of the simple harmonic function
a	The amplitude value for the sine wave of the variable • This value must be a real number or a function that returns a real number
ω	The frequency of the simple harmonic function
ϕ	The phase shift value of the simple harmonic function
b	The average frequency of the simple harmonic function

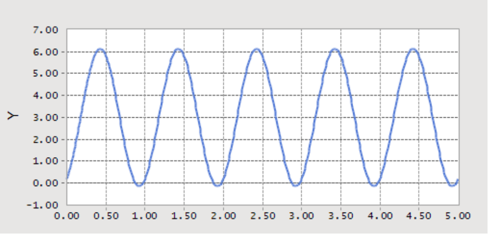
Formula

$$\mathrm{SHF} = a \cdot \sin(\omega \cdot (x - x_0) - \phi) + b$$

Example

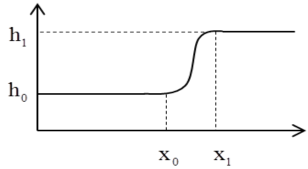
The graph below illustrates the output values of the following expression through the scope:

SHF(time, 10D, PI, 360D, 0,3)



STEP

Interpolates between two markers using a cubic polynomial. Generally, this function is used when linking two markers in a gradual curve.



Format

$$\mathrm{STEP}(x, x_0, h_0, x_1, h_1)$$

Arguments

x	The variable for the defined STEP function • Generally, this variable is time or a function that returns a real number.
x_0	The starting point of the STEP function.
h_0	The initial value for the input variable (within the range of $x \le x_0$)
x_1	The ending point of the STEP function
h_1	The initial value for the input variable (within the range of $x \ge x_1$)

Formula

$$\mathrm{STEP} = h_0 \ , \qquad \text{when } x \le x_0$$

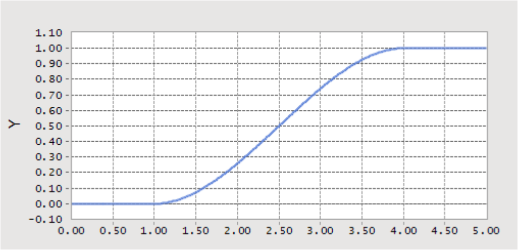
$$= h_0 + (h_1 - h_0)\left[\frac{x - x_0}{x_1 - x_0}\right]^2 \left\{3 - 2\left[\frac{x - x_0}{x_1 - x_0}\right]\right\}, \qquad \text{when } x_0 < x \le x_1$$

$$= h_1 \ , \qquad \text{when } x \ge x_1$$

Example

The graph below illustrates the output values of the following expression through the scope:

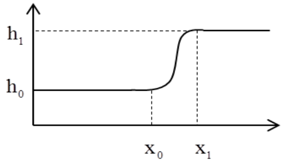
STEP(time, 1.0, 0.0, 4.0, 1.0)



STEP5

Interpolates between two markers using a quantic polynomial. Generally, this function is used when linking two markers in a gradual curve.

Format



STEP5(x,x_0,h_0,x_1,h_1)

Arguments

x	The input variable for the defined STEP5 function • Generally, this variable is time or a function that returns a real number.
x_0	The starting point of the STEP5 function
h_0	The initial value for the input variable (within the range of $x \le x_0$)
x_1	The ending point of the STEP5 function
h_1	The initial value for the input variable (within the range of $x \ge x_1$)

Formula

$$\mathrm{STEP5} = h_0 \ , \qquad \text{when } x \le x_0$$

$$= h_0 + (h_1 - h_0)\cdot\left[\frac{x-x_0}{x_1-x_0}\right]^3\cdot\left\{10 - 15\cdot\frac{x-x_0}{x_1-x_0} + 6\cdot\left[\frac{x-x_0}{x_1-x_0}\right]^2\right\}, \qquad \text{when } x_0 < x \le x_1$$

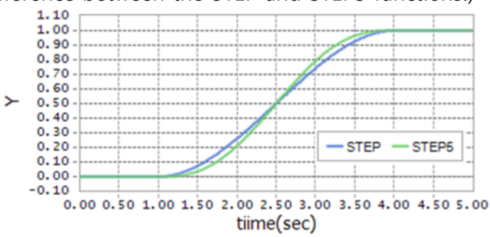
$$= h_1 \ , \qquad \text{when } x \ge x_1$$

Example

The graph below illustrates the output values for the following expression through the scope. (This highlights the difference between the STEP and STEP5 functions.)

STEP (time, 1.0, 0.0, 4.0, 1.0)

STEP5(time, 1.0, 0.0, 4.0, 1.0)

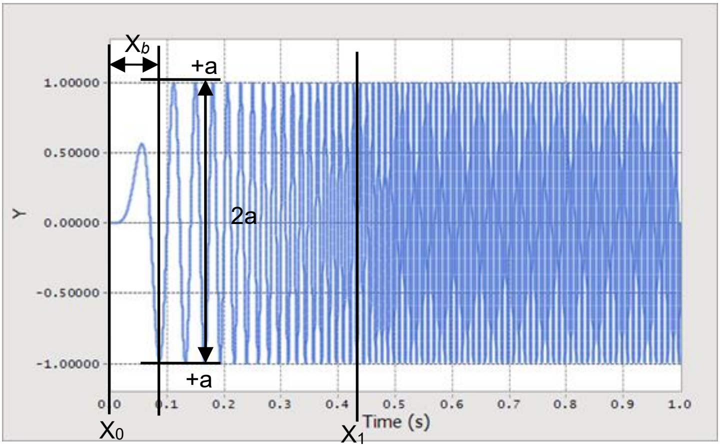


SWEEP

Returns the results of a regular sine wave function with increasing linear frequency.

Format

SWEEP(x,a,x_0,f_0,x_1,f_1,x_b)



Arguments

x	Independent variable for the SWEEP function • This variable must be time, a real number, or a function that returns a real number.
a	Sine function wave
x_0	Starting point of the SWEEP function (x value)
f_0	Initial sine function frequency
x_1	Ending point of the SWEEP function (x value)
f_1	Ending frequency of the sine function
x_b	Range in which the SWEEP function is fully active

Formula

$$\text{SWEEP}=\text{STEP5}(x, x_0, 0, x_0+x_b, 1)\cdot a\cdot \sin(2\pi\cdot\int freq(x)dx)$$

$$freq(x) =$$

$$\begin{cases} f_0\,, & (\;x \le x_0) \\ f_0 + \frac{f_1 - f_0}{x_1 - x_0}\cdot x\;, & (x_0 \le x < x_1) \\ f_1, & (x_1 \le x) \end{cases}$$

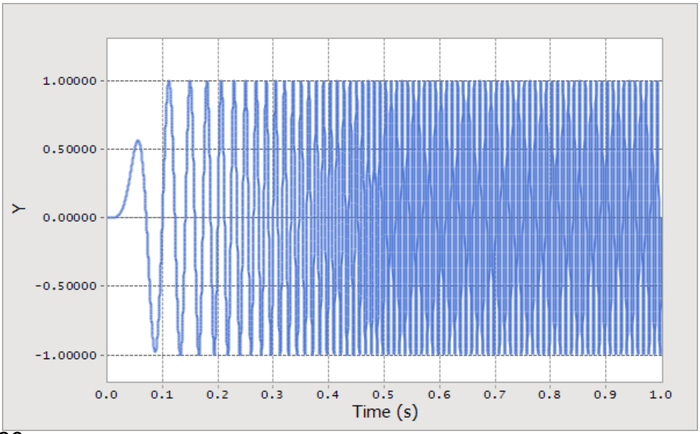
$$\int freq(x)dx =$$

$$\begin{cases} f_0\cdot x\,, & (\;x \le x_0) \\ f_0\cdot x_0 + f_0\cdot(x-x_0)+0.5\frac{f_1-f_0}{x_1-x_0}\cdot(x-x_0)^2\,, & (x_0 \le x < x_1) \\ f_0\cdot x_0 + f_0\cdot(x-x_0)+0.5\frac{f_1-f_0}{x_1-x_0}\cdot(x_1-x_0)^2 + f_1\cdot(x-x_0)\,, & (x_1 \le x) \end{cases}$$

Example

The graph below illustrates the output values of the following expression through the scope:

SWEEP(time, 1.0, 0.0, 0.0, 0.5, 100.0, 0.1)



CONTROL & HYDRAULIC

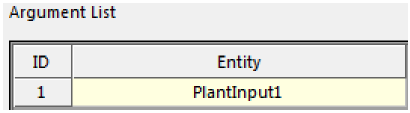
PIN
POUT
HIN
HOUT

PIN

Imports calculated values of control plant from external control programs into the RecurDyn model for the co-simulation using CoLink or Control Toolkit. The values are used as RecurDyn plant inputs and the function returns the imported value.

Format

PIN(1)
└→ Plant Input Entity



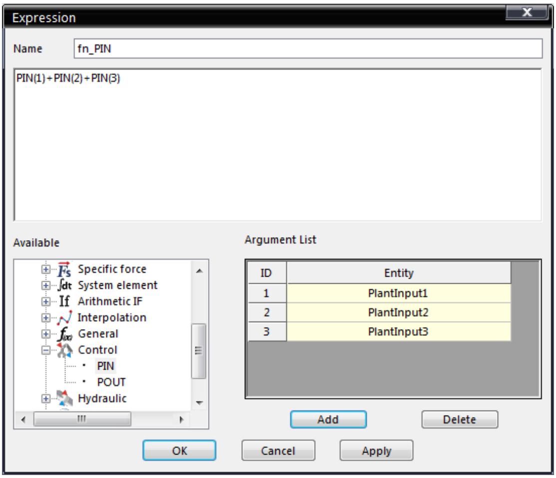
Argument

Plant Input Entity	The name or argument number of the PIN entity

Example

PIN(Mode1.PlantInput1)

PIN(1) <Argument: (1) Mode1.PlantInput1>



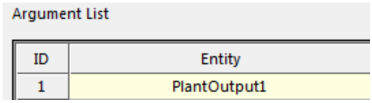
POUT

Imports calculated values of RecurDyn plant from external control programs into the control model for the co-simulation using CoLink or Control Toolkit. The values are used as RecurDyn plant outputs and the function returns the imported value.

Format

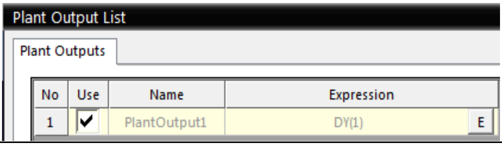
POUT(1)

→ Plant Output Entity



Argument

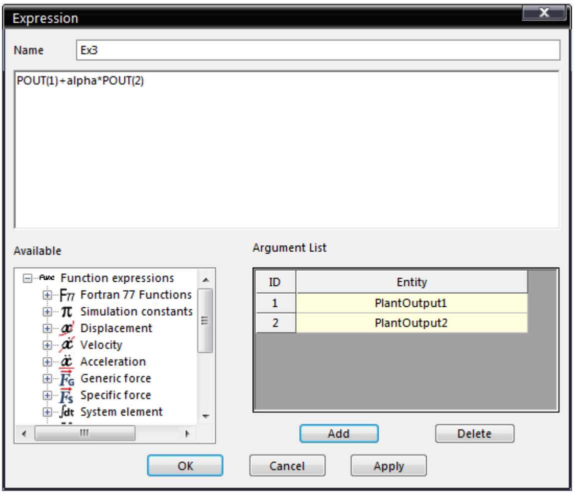
Plant Output Entity	Name or argument number of the POUT entity



Example

POUT(Mode1.PlantOutput1)

POUT(1) <Argument: (1) Mode1. PlantOutput1>



HIN

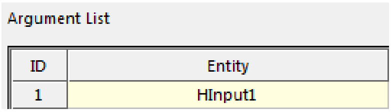
Returns HIN entity value which is calculated in RecurDyn model so that it can be transferred to the external hydraulic software such as AMESim or FMI.

※ The HIN/HOUT functions use a different plant than the PIN/POUT functions. In other words, for PIN/POUT, the RecurDyn model is the plant whereas for HIN/HOUT, hydraulic system model is the plant.

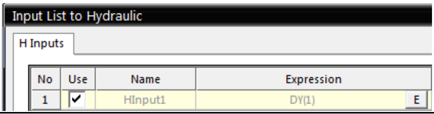
Format

HIN(1)

→ Hyd_Input Entity



Argument

Hyd_Input Entity	The name or argument number of the HIN entity 

Example

HIN(Mode1.HInput1)

HIN(1) <Argument: (1) Mode1. HInput1>

HOUT

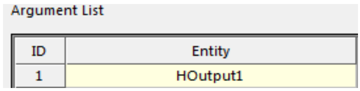
Returns HOUT entity value which is calculated in the hydraulic model by the external hydraulic software such as AMESim or FMI. HOUT means that the hydraulic value is input to RecurDyn, so the returned value is used in RecurDyn model.

※ The HIN/HOUT functions use a different plant than the PIN/POUT functions. In other words, for PIN/POUT, the RecurDyn model is the plant whereas for HIN/HOUT, hydraulic system model is the plant.

Format

HOUT(1)

Hyd_Output Entity


Argument List

ID	Entity
1	HOutput1

Argument

Hyd_Output Entity	The name or argument number of the HOUT entity

Example

HOUT(Mode1.HOutput1)

HOUT(1) <Argument: (1) Mode1. HOutput1>

SENSORS

SNSR
EVTIME
SENSOR
SENSORONTIME
SENSOROFFTIME
SENSORDISTANCE
SENSORDETECTEDPOINT

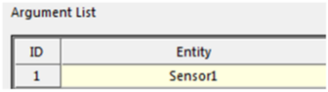
SNSR

Returns the values measured on sensors created using various toolkits.

Format

SNSR(1)

SENSOR entity



Argument List

ID	Entity
1	Sensor1

Argument

SENSOR entity	The name or argument number of the SENSOR entity

※ Supported sensors and returned values

Toolkit	Senor	Returned Value
Belt	Speed	Velocity of the belt segment
	Distance	Distance from the center of the sensor to the belt segment
	Slip	Difference in velocity between the pulley and the belt segment
	Tension	Tension of the belt segment.
	Displacement	X-direction distance from the center of the sensor to the belt segment
Chain	Distance	Distance from the center of the sensor to the chain link
	Speed	Velocity of the chain link
	Tension	Tension of the chain link
MTT2D	Speed	Velocity of the sheet body
	Distance	Distance from the center of the sensor to the sheet body
	Event	On = 1, Off = 0
MTT3D	Speed	Velocity of the sheet body
	Distance	Distance from the center of the sensor to the sheet body
	Event	On = 1, Off = 0
Track_LM	Displacement	X-direction distance from the center of the sensor to the track link
Track_HM	Displacement	X-direction distance from the center of the sensor to the track link

Example

SNSR (mode1.Sensor1)

SNSR (1) <Argument: (1) mode1.Sensor1>

EVTIME

Returns the simulation time at the exact moment when a sensor entity, which is used in MTT2D or 3D, turns on. At all other times, it returns a value of 0.

Format

EVTIME(1)
→ SENSOR entity

Argument List

ID	Entity
1	Sensor1

Argument

SENSOR entity	The name or argument number of the sensor entity

Example

EVTIME (Sensor1)

EVTIME (1) <Argument: (1) Sensor1 >

SENSOR

Returns the status of a Laser or PointInBox sensor entity. The status of the sensor is returned as a value of either 0 or 1 (on=1, off=0).

Format

SENSOR (1)

→ SENSOR entity

Argument List

ID	Entity
1	Sensor1

Argument

SENSOR entity	The name or argument number of the sensor entity

Example

SENSOR (1)

SENSOR (1) <Argument: (1) Sensor1 >

SENSORONTIME

Returns the simulation time at the exact moment when a Laser or PointInBox sensor entity turns on.

The initial value is 0. The time at the moment of changing from OFF to ON is returned. If On/Off is repeated, the most recent ON time is returned.

Format

SENSORONTIME(1)

SENSOR entity

Argument List

ID	Entity
1	Sensor1

Argument

SENSOR entity	The name or argument number of the sensor entity

Example

SENSORONTIME (Sensor1)

SENSORONTIME (1) <Argument: (1) Sensor1 >

SENSOROFFTIME

Returns the simulation time at the exact moment when a Laser or PointInBox sensor entity turns off.

The initial value is 0. The time at the moment of changing from ON to OFF is returned. If On/Off is repeated, the most recent Off time is returned.

Format

SENSOROFFTIME(1)

SENSOR entity

Argument List

ID	Entity
1	Sensor1

Argument

SENSOR entity	The name or argument number of the sensor entity

Example

SENSOROFFTIME (Sensor1)

SENSOROFFTIME (1) <Argument: (1) Sensor1 >

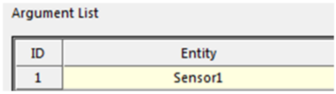
SENSORDISTANCE

Returns the displacement values from the moment when the Laser sensor entity turns on until the sensor's objective geometry. When the sensor is turned off, it returns a value of 0.

Format

SENSORDISTANCE(1)

SENSOR entity


Argument List

ID	Entity
1	Sensor1

Argument

SENSOR entity	The name or argument number of the sensor entity

Example

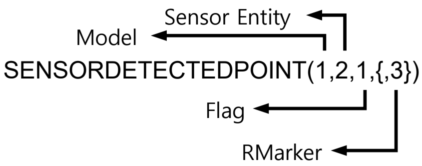
SENSORDISTANCE(Sensor1)

SENSORDISTANCE(1) <Argument: (1) Sensor1 >

SENSORDETECTEDPOINT

Returns the values for location, velocity, and acceleration in the objective area detected by the Laser sensor entity. When the detection status is off, it returns a value of 0.

Format



Argument List

ID	Entity
1	Model1
2	Sensor1
3	Body1.Marker1

Arguments

Model	The name or argument number of the model
SENSOR entity	The name or argument number of the sensor entity
Flag	A flag that indicates the returned displacement, velocity or acceleration type and returns the calculated values for the input characters. Depending on the values, it selects 12 statuses for the detection location. 1:DM, 2:DX, 3:DY, 4:DZ, 5:VM, 6:VX, 7:VY, 8:VZ 9:ACCM, 10:ACCX, 11:ACCY, 12:ACCZ
RMarker	The name or argument number of the standard marker for the direction to be measured

Example

SENSORDETECTEDPOINT (Model1, Sensor1, 1, Body1.Marker1)

SENSORDETECTEDPOINT (1,2,1,3)

<Argument: (1) Model1 (2)Sensor1 (3)Body1.Marker1 >

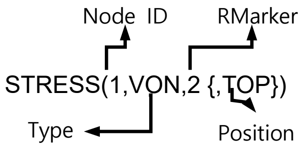
STRESS & STRAIN

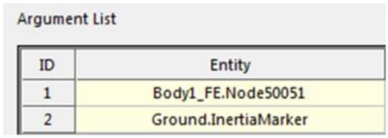
STRESS
STRAIN

STRESS

Returns the stress acting on a designated node on a flexible body.

Format




Argument List

ID	Entity
1	Body1_FE.Node50051
2	Ground.InertiaMarker

Arguments

<table>
<tr><td>Node ID</td><td>Node number, including the names of the FFlex or RFlex Body, or argument number for the nodes to calculated for the stress value
The node must be set as an output node.</td></tr>
<tr><td>Type</td><td>Symbol that indicates the type of stress to measure
<table>
<tr><td>Stress</td><td>Name</td><td>Comment</td></tr>
<tr><td rowspan="6">Basic stress</td><td>X</td><td>x direction from x face</td></tr>
<tr><td>Y</td><td>y direction from y face</td></tr>
<tr><td>Z</td><td>z direction from z face</td></tr>
<tr><td>XY</td><td>y direction from x face</td></tr>
<tr><td>YZ</td><td>z direction from y face</td></tr>
<tr><td>ZX</td><td>z direction from z face</td></tr>
<tr><td rowspan="3">Principal stress</td><td>S1</td><td>First principal</td></tr>
<tr><td>S2</td><td>Second principal</td></tr>
<tr><td>S3</td><td>Third principal</td></tr>
<tr><td>Von-mises stress</td><td>VON</td><td>Von Mises stress</td></tr>
</table></td></tr>
<tr><td>RMarker</td><td>The name or argument number of the standard marker for direction to be measured</td></tr>
<tr><td>Position</td><td>The location of measurement on the shell element (TOP or BOTTOM)</td></tr>
</table>

Example

STRESS(Body1_FE.node1,X,Ground.InertiaMarker)

STRESS(1,VON,2) <Argument: (1) Body1_FE.node1 (2) Ground.InertiaMarker >

STRAIN

Returns the strain acting on a designated node on a flexible body.

Format

Node ID RMarker

STRAIN(1,VON,2{,TOP})

Type Stress position

Argument List

ID	Entity
1	Body1_FE.Node50051
2	Ground.InertiaMarker

Arguments

<table>
<tr><td>Node ID</td><td>Node number, including the names of the FFlex or RFlex Body, or argument number for the nodes to calculated for the stress value
• The node must be set as an output node.</td></tr>
<tr><td>Type</td><td>Symbol that indicates the type of strain to measure
<table>
<tr><td>Strain</td><td>Name</td><td>Comment</td></tr>
<tr><td rowspan="6">Basic strain</td><td>X</td><td>x direction from x face</td></tr>
<tr><td>Y</td><td>y direction from y face</td></tr>
<tr><td>Z</td><td>z direction from z face</td></tr>
<tr><td>XY</td><td>y direction from x face</td></tr>
<tr><td>YZ</td><td>z direction from y face</td></tr>
<tr><td>ZX</td><td>z direction from z face</td></tr>
<tr><td rowspan="3">Principal strain</td><td>E1</td><td>First principal</td></tr>
<tr><td>E2</td><td>Second principal</td></tr>
<tr><td>E3</td><td>Third principal</td></tr>
<tr><td>Von-mises strain</td><td>VON</td><td>Von Mises stress</td></tr>
</table></td></tr>
<tr><td>RMarker</td><td>The name or argument number of the standard marker for direction to be measured</td></tr>
<tr><td>Position</td><td>The location of measurement on the shell element (TOP or BOTTOM)</td></tr>
</table>

Example

STRAIN(Body1_FE.node1,X,Ground.InertiaMarker)

STRAIN(1,VON,2) <Argument: (1) Body1_FE.node1 (2) Ground.InertiaMarker >

Index

RECURDYN

FunctionBay

This book is based on RecurDyn V8R2