

# RECURDYN

Expression Handbook

평선베이(주) 솔루션그룹 역음



FunctionBay



## 목차

### < OPERATORS >

Arithmetic Operator .....	8
Relational Operators.....	9
Logical Operators .....	10

### < FORTRAN FUNCTIONS >

ABS.....	12
ACOS.....	13
AINT.....	14
ANINT .....	15
ASIN .....	16
ATAN .....	17
ATAN2 .....	18
COS .....	19
COSH .....	20
DIM .....	21
EXP.....	22
LOG .....	23
LOG10 .....	24
MAX .....	25
MIN .....	26
MOD .....	27
SIGN .....	28
SIN.....	29
SINH .....	30
SQRT.....	31
TAN.....	32
TANH.....	33

### < SIMULATION CONSTANTS >

TIME .....	35
PI .....	36
RTOD.....	37
DTOR.....	38

## < DISPLACEMENT >

AX .....	40
AY .....	41
AZ .....	42
DM .....	43
DX .....	44
DY .....	46
DZ .....	47
PHI .....	48
PITCH .....	49
PSI .....	50
ROLL .....	51
THETA .....	53
YAW .....	54

## < VELOCITY >

VM .....	56
VR .....	57
VX .....	59
VY .....	60
VZ .....	61
WM .....	62
WX .....	63
WY .....	64
WZ .....	65

## < ACCELERATION >

ACCM .....	67
ACCX .....	68
ACCY .....	69
ACCZ .....	70
WDTM .....	71
WDTX .....	72
WDTY .....	73
WDTZ .....	74

## < GENERIC FORCE >

FM.....	76
FX.....	77
FY.....	78
FZ.....	79
TM.....	80
TX.....	81
TY.....	82
TZ.....	83

**< SPECIFIC FORCE >**

CONTACT.....	85
MOTION.....	86
JFRICTION.....	87
COUPLER.....	88
GEAR.....	89
SHT3D_CF.....	90
JOINT.....	91
PTCV.....	92
CVCV.....	93
AXIAL.....	94
TFORCE.....	95
RFORCE.....	96
SCREWFORCE.....	97
BEAM.....	98
BUSH.....	99
SPRING.....	100
MATRIXFORCE.....	101

**< SYSTEM ELEMENT >**

DIF.....	103
DIF1.....	105
VARVAL.....	106

**< ARITHMETIC IF >**

IF.....	108
---------	-----

**< INTERPOLATION >**

AKISPL.....	110
CUBSPL.....	111

**< GENERAL >**

BISTOP.....	113
CHEBY.....	115
FORCOS .....	116
FORSIN .....	117
HAVSIN.....	118
IMPACT .....	119
POLY .....	121
SHF.....	122
STEP.....	123
STEP5.....	124
SWEEP.....	125

**< CONTROL & HYDRAULIC >**

PIN.....	128
POUT .....	129
HIN.....	130
HOUT.....	131

**< SENSOR >**

SNSR .....	133
EVTIME.....	134
SENSOR.....	135
SENSORONTIME.....	136
SENSOROFTIME .....	137
SENSORDISTANCE .....	138
SENSORDETECTEDPOINT .....	139

**< STRESS & STRAIN >**

STRESS.....	141
STRAIN.....	142

<b>&lt;색인&gt;</b> .....	<b>143</b>
-------------------------	------------

# OPERTATORS

*Arithmetic Operator*

*Relational Operators*

*Logical Operators*

# Arithmetic Operator

Expression 에서 입력할 수 있는 산술 연산자

## Format

+	더하기
-	빼기
*	곱하기
/	나누기
**	거듭제곱

## Example

일반적인 산술기호의 사용법과 동일하며 특히 '\*\*'의 경우  $2^{**n}$  은 2의 n 제곱을 의미한다. 또한 '/'의 경우 정확히 0으로 어떤 수를 나누게 되면 0 대신 1e-20으로 변환시켜 연산한다.

# Relational Operators

참일 경우 1, 거짓일 경우 0을 반환하는 관계연산자

## Format & Example

<	Is less than 1.0<2.0, return the value 1 2.0<1.0, return the value 0
<=	Is less than or equal to 1.0<=2.0, return the value 1 2.0<=1.0, return the value 0
=	Is equal to A=B, return the value 1( If  A-B <IF Tolerance) A=B, return the value 0( If  A-B >=IF Tolerance)
>	Is greater than 1.0>2.0, return the value 0 2.0>1.0, return the value 1
>=	Is greater than or equal to 1.0>=2.0, return the value 0 2.0>=1.0, return the value 1
<>	Is not equal to A<>B, return the value 0( If  A-B <IF Tolerance) A<>B, return the value 1( If  A-B >=IF Tolerance)

# Logical Operators

Expression 에서 입력할 수 있는 논리 연산자

## Definitions

	Logical OR
&&	Logical AND

## Example

The Value of A  B		
	A is true	A is false
B is true	1	1
B is false	1	0

The Value 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

: 입력 값의 절대값을 반환하는 함수

## Format

ABS(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

$ABS(x) = |x|$

## Example

ABS(-21)

ABS(DX(1,2)) <Argument (1) body1.Maker1, (2) body2.Maker2>

# ACOS

: 코사인(cos)의 역함수를 계산하는 함수로써 실수 입력 값에 대한 각도를 계산하여 반환함, 반환되는 값은  $0 \sim \pi$  의 값을 가짐

## Format

ACOS(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

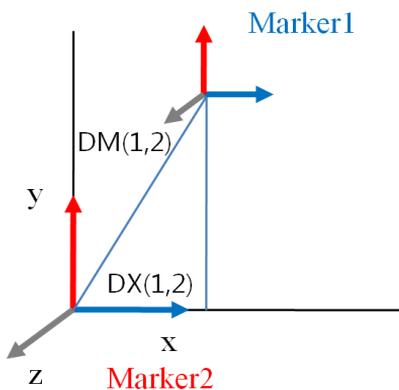
## Formulation

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

## Example

아래 그림과 같이 InertiaMarker의 x축에 대한 Marker1의 각도를 계산하는 함수로 표현 가능

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



# AINT

: 입력된 값보다 작은 가장 가까운 정수를 반환하는 함수

## Format

AINT(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Example

ex1) Function : AINT(1.2)

Result : 1

ex2) Function : AINT(0.7)

Result : 0

# ANINT

: 입력된 값과 가장 가까운 정수를 반환하는 함수

## Format

ANINT(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Example

ex1) Function : ANINT(1.6)

Result : 2

ex2) Function : ANINT(0.4)

Result : 0

# ASIN

: 사인(sin)의 역함수를 계산하는 함수로써 실수 입력 값에 대한 각도를 계산하여 반환함, 반환되는 값은  $-\pi/2 \sim \pi/2$  의 값을 가짐

## Format

ASIN(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

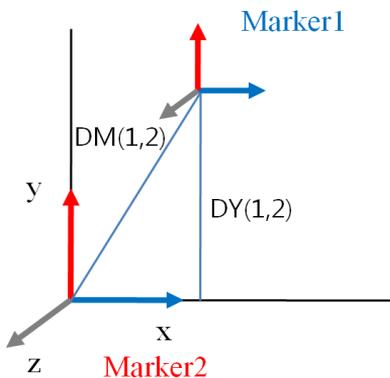
## Formulation

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

## Example

아래 그림과 같이 InertiaMarker의 x축에 대한 Marker1의 각도를 계산하는 함수로 표현 가능

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



# ATAN

: 탄젠트(tangent)의 역함수를 계산하는 함수로써 실수 입력 값에 대한 각도를 계산하여 반환함, 반환되는 값은  $-\pi/2 \sim \pi/2$  의 값을 가짐

## Format

ATAN(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

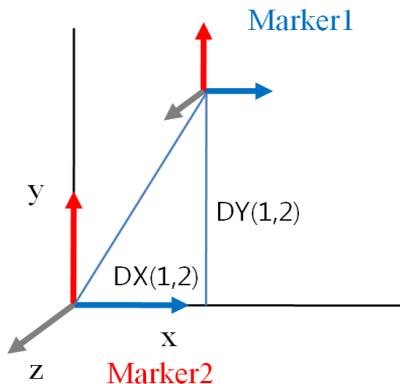
## Formulation

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

## Example

아래 그림과 같이 InertiaMarker의 x축에 대한 Marker1의 각도를 계산하는 함수로 표현 가능

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



# ATAN2

: 탄젠트(tangent)의 역함수를 계산하는 함수로서 분모와 분자의 실수 입력 값에 대한 각도를 계산하여 반환함, 반환되는 값은  $-\pi \sim \pi$ 의 값을 가짐

## Format

ATAN2(dx,dy)

## Arguments definition

dx	실수 또는 실수를 출력하는 함수, 탄젠트 함수에서의 x변위에 해당하는 값
dy	실수 또는 실수를 출력하는 함수, 탄젠트 함수에서의 y변위에 해당하는 값

## Formulation

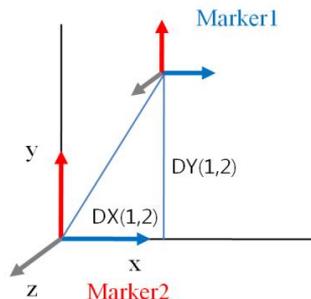
$$\text{ATAN2} = \begin{cases} \tan^{-1}(dx/dy) & dy > 0, dx > 0 \\ \tan^{-1}(dx/dy) + \pi & dy > 0, dx < 0 \\ \tan^{-1}(dx/dy) - \pi & dy < 0, dx < 0 \\ +\pi/2 & dy > 0, dx = 0 \\ -\pi/2 & dy < 0, dx = 0 \\ \text{undefined} & dy = 0, dx = 0 \end{cases}$$

## Example

아래 그림과 같이 InertiaMarker의 x축에 대한 Marker1의 각도를 계산하는 함수로 표현 가능

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

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



# COS

: Cosine 함수 값을 계산하여 -1~ +1 사이의 값을 반환하는 삼각함수

## Format

$\text{COS}(x)$

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

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

## Example

- ex1)      Function :  $\text{COS}(\pi/2)$   
            Result : 0
- ex2)      Function :  $\text{COS}(1)$   
            Result : 0.5403

# COSH

: Hyperbolic 코사인(cos)함수 값을 반환하는 함수

## Format

COSH(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

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

## Example

ex1) Function : COSH(0)

Result : 1

ex2) Function : COSH(1)

Result : 1.54308

# DIM

: 입력한 두 값  $x$  과  $y$  에 대하여 차이가 양일경우 값을 반환하고 음일 경우 0 을 반환하는 함수

## Format

DIM( $x,y$ )

## Arguments definition

$x$	실수 또는 실수를 출력하는 함수
$y$	실수 또는 실수를 출력하는 함수

## Formulation

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

## Example

ex1) Function : DIM(15,5)

Result : 10

ex2) Function : DIM(5,10)

Result : 0

# EXP

: 자연로그의 base 값 e 에 대해 입력한 값을 지수로 계산하여 값을 반환하는 함수

## Format

EXP(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

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

## Example

- ex1)      Function : EXP(0)  
            Result : 1
- ex2)      Function : EXP(3)  
            Result : 20.0855

# LOG

: 입력한 값에 대한 자연로그(base e) 값을 반환하는 함수

## Format

LOG(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

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

## Example

- ex1)      Function : LOG(1)  
            Result : 0
- ex2)      Function : LOG(10)  
            Result : 2.3026

# LOG10

: 입력한 값에 대한 대수로그(base 10) 값을 반환하는 함수

## *Format*

LOG10(x)

## *Arguments definition*

x	실수 또는 실수를 출력하는 함수
---	-------------------

## **Formulation**

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

## **Example**

- ex1)      Function : LOG10(1)  
            Result : 0
- ex2)      Function : LOG10 (10)  
            Result : 1

# MAX

: 입력한 2 개 또는 3 개의 값 중 최대값을 반환하는 함수

## Format

MAX(x1,x2{,x3})

## Arguments definition

x1	실수 또는 실수를 출력하는 함수
x2	실수 또는 실수를 출력하는 함수
x3	실수 또는 실수를 출력하는 함수 (생략가능)

## Formulation

MAX=max(x1,x2,x3)

## Example

MAX(1,2,3)

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

# MIN

: 입력한 2 개 또는 3 개의 값 중 최소값을 반환하는 함수

## Format

MIN(x1,x2{,x3})

## Arguments definition

x1	실수 또는 실수를 출력하는 함수
x2	실수 또는 실수를 출력하는 함수
x3	실수 또는 실수를 출력하는 함수 (생략가능)

## Formulation

MIN=min(x1,x2,x3)

## Example

MIN(1,2,3)

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

# MOD

: 두 입력변수  $x, y$  에 대하여  $x$  을  $y$  로 나눈 나머지 값을 반환하는 함수

## Format

MOD( $x, y$ )

## Arguments definition

$x$	실수 또는 실수를 출력하는 함수.
$y$	실수 또는 실수를 출력하는 함수

## Formulation

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

## Example

ex1) Function : MOD(1,2)

Result : 1

ex2) Function : MOD (5,3)

Result : 2

# SIGN

: 입력된 두 변수 x와 y에 대하여 x의 절대값이 y의 부호를 따르도록 계산하여 반환하는 함수

## Format

SIGN(x,y)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
y	실수 또는 실수를 출력하는 함수

## Formulation

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

## Example

- ex1) Function : SIGN(-4.7,1.2)  
Result : 4.7
- ex2) Function : MOD (5.3,-6.5)  
Result : -5.3
- ex3) Function : MOD (-2.5,-5.2)  
Result : -2.5

# SIN

: Sine 함수값을 계산하여 -1~ +1 사이의 값을 반환하는 삼각함수

## Format

SIN(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

$SIN = \sin(x)$

## Example

- ex1)      Function : SIN( $\pi/2$ )  
            Result : 1
- ex2)      Function : SIN(1)  
            Result : 0.8415

# SINH

: Hyperbolic 사인(sin)함수 값을 반환하는 함수

## Format

SINH(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

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

## Example

ex1) Function : SINH(0)

Result : 0

ex2) Function : SINH (1)

Result : 1.1752

# SQRT

: 입력한 값의 제곱근을 반환하는 함수

## Format

SQRT(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수, 음의 값이 입력되면 값이 반환되지 않음
---	--

## Formulation

$$\text{SQRT} = \sqrt{x}$$

## Example

- ex1)      Function : SQRT(2)  
            Result : 1.41421
- ex2)      Function : SQRT(-2)  
            Result : N/A

# TAN

: 탄젠트(tangent)함수 값 계산하여 실수 값을 반환하는 함수

## Format

TAN(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

$TAN = \tan(x)$

## Example

ex1) Function : TAN( $\pi$ )

Result : 0

ex2) Function : TAN( $\pi/4$ )

Result : 1

# TANH

: Hyperbolic 탄젠트(tan)함수 값을 반환하는 함수

## Format

TANH(x)

## Arguments definition

x	실수 또는 실수를 출력하는 함수
---	-------------------

## Formulation

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

## Example

ex1) Function : TANH(1)

Result : 0.76159

ex2) Function : TANH(-3)

Result : -0.99505

# SIMULATION CONSTANTS

*TIME*

*PI*

*DTOR*

*RTOD*

# TIME

: Simulation Time 을 반환해주는 함수

**Format**

TIME

# PI

: 원주율(3.141592)을 반환하는 함수

## Format

PI

## Formulation

PI = 3.1415926.....

## Example

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

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

# RTOD

: Radian 값을 Degree 값으로 반환하기 위한 값

## Format

RTOD

## Formulation

$$\text{RTOD} = 180^\circ / \pi$$

## Example

$$\text{RTOD} * \pi = 180$$

# DTOR

: Degree 값을 Radian 값으로 반환하기 위한 값,

## Format

DTOR

## Formulation

$$DTOR = \pi / 180^\circ$$

## Example

$$DTOR * 180 = 3.141592$$

※주의: Expression 에서 실수의 마지막에 D 를 입력하면(ex:180d)  
Degree 의 의미로써 자동으로 Radian 의 값으로 환산하여 사용된다.  
따라서 180d\*DTOR 과 같이 사용하면 잘못된 계산이 이루어짐

# DISPLACEMENT

*AX*

*AY*

*AZ*

*DM*

*DX*

*DY*

*DZ*

*PHI*

*PITCH*

*PSI*

*ROLL*

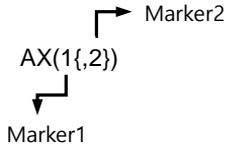
*THETA*

*YAW*

# AX

: 하나의 Marker의 InertiaMarker에 대한 x축 회전각 또는 두 Marker의 x축에 대한 회전각을 계산하는 함수

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

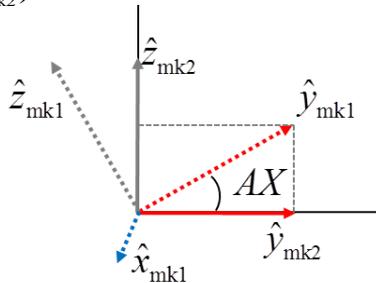
## Formulation

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

$\hat{y}_{mk1}$  : Marker1의 y 방향 단위벡터

$\hat{z}_{mk2}$  : Marker2의 z 방향 단위벡터

$\hat{y}_{mk2}$  : Marker2의 y 방향 단위벡터



## Example

AX (body1.Marker1)

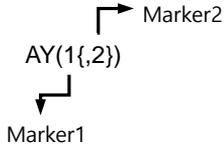
AX (body1.Marker1, body2.Marker2)

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

# AY

: 하나의 Marker 의 InertiaMarker 에 대한 y 축 회전각 또는 두 Marker 의 y 축에 대한 회전각을 계산하는 함수

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

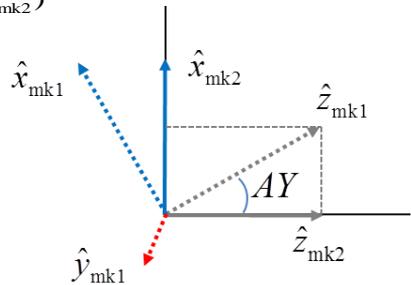
## Formulation

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

$\hat{z}_{mk1}$  : Marker1의 z 방향 단위벡터

$\hat{x}_{mk2}$  : Marker2의 x 방향 단위벡터

$\hat{z}_{mk2}$  : Marker2의 z 방향 단위벡터



## Example

AY (body1.Marker1)

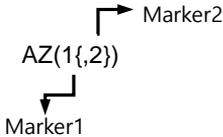
AY (body1.Marker1, body2.Marker2)

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

# AZ

: 하나의 Marker의 InertiaMarker에 대한 z축 회전각 또는 두 Marker의 z축에 대한 회전각을 계산하는 함수

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

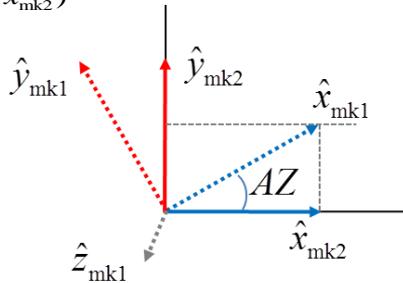
## Formulation

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

$\hat{x}_{mk1}$  : Marker1의 x 방향 단위벡터

$\hat{y}_{mk2}$  : Marker2의 y 방향 단위벡터

$\hat{x}_{mk2}$  : Marker2의 x 방향 단위벡터



## Example

AZ (body1.Marker1)

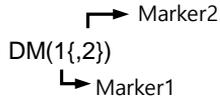
AZ (body1.Marker1, body2.Marker2)

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

# DM

: 하나의 Marker 에 대한 InertiaMarker 로부터의 절대거리 혹은, 두 Marker 사이의 절대 거리를 계산하는 함수

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

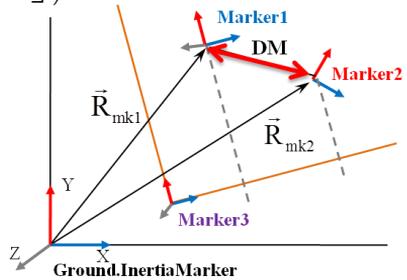
Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

## Formulation

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

$\vec{R}_{mk1}$  : Marker1의 위치 벡터

$\vec{R}_{mk2}$  : Marker2의 위치 벡터



## Example

DM (body1.Marker1)

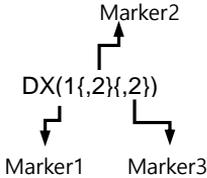
DM (body1.Marker1, body2.Marker2)

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

# DX

하나의 Marker 에 대한 x 방향의 위치 혹은, 두 Marker 사이의 거리를 방향기준 Marker 의 x 방향에 대해 계산하는 함수.

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

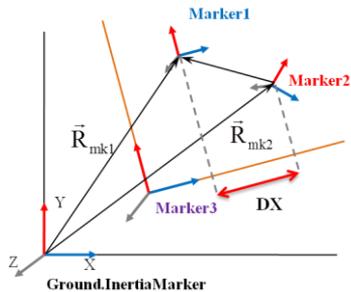
## Formulation

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

$\vec{R}_{mk1}$  Marker1의 위치 벡터.

$\vec{R}_{mk2}$  Marker2의 위치 벡터.

$\hat{x}_{mk3}$  Marker3의 x 방향 단위벡터



## Example

DX(body1.Marker1)

DX(body1.Marker1, body2.Marker1)

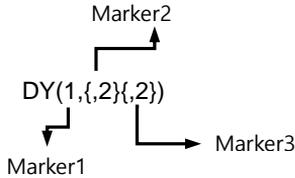
DX(body1.Marker1, body2.Marker1, body3.Marker1)

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

# DY

: 하나의 Marker 에 대한 y 방향의 위치 혹은, 두 Marker 사이의 거리를 방향기준 Marker 의 y 방향에 대해 계산하는 함수

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

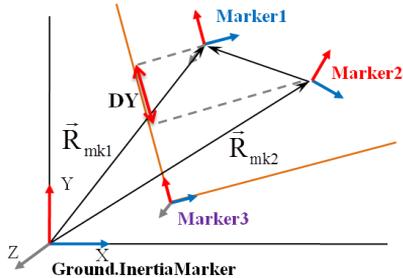
## Formulation

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

$\vec{R}_{mk1}$  : Marker1의 위치 벡터

$\vec{R}_{mk2}$  : Marker2의 위치 벡터

$\hat{y}_{mk3}$  : Marker3의 y 방향 단위벡터



## Example

DY(body1.Marker1)

DY(body1.Marker1, body2.Marker1)

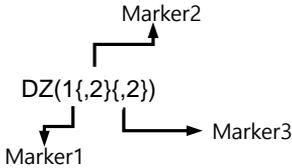
DY(body1.Marker1, body2.Marker1, body3.Marker1)

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

# DZ

: 하나의 Marker 에 대한 z 방향의 위치 혹은, 두 Marker 사이의 거리를 방향기준 Marker 의 z 방향에 대해 계산하는 함수

## Format



Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

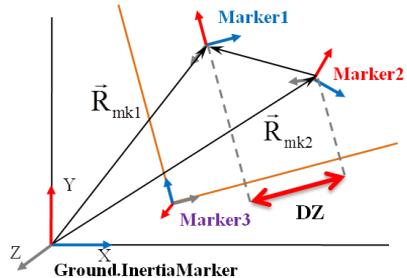
## Formulation

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

$\vec{R}_{mk1}$  : Marker1의 위치벡터

$\vec{R}_{mk2}$  : Marker2의 위치벡터

$\hat{z}_{mk3}$  : Marker3의 z 방향 단위벡터



## Example

DZ (body1.Marker1)

DZ(body1.Marker1, body2.Marker1)

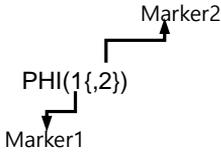
DZ(body1.Marker1, body2.Marker1, body3.Marker1)

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

# PHI

: z-x-z 회전 순서로 정의된 Euler Angle 의 세 번째 회전순서에 해당하는 회전각(PHI)을 반환하는 함수

## Format

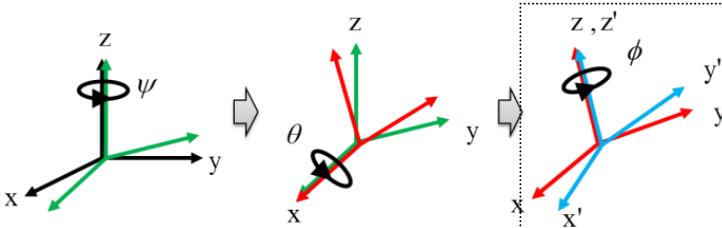


Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

## Formulation



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

순차적으로 정의된 회전 순서에 따라 세 번째 회전인 z축 회전각  $\phi$ 를 반환

## Example

$\text{PHI}(\text{body1.Marker1})$

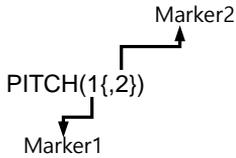
$\text{PHI}(\text{body1.Marker1}, \text{body2.Marker2})$

$\text{PHI}(1,2)$  <Argument: (1)body1.Marker1, (2)body2.Marker2 >

# PITCH

: z-y-x 회전 순서로 정의된 Euler Angle 의 두 번째 회전순서에 해당하는 회전각(PITCH)을 반환하는 함수

## Format



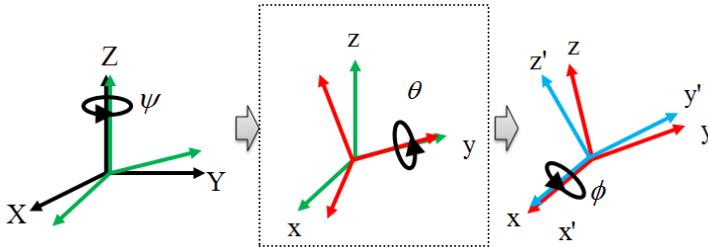
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

## Formulation



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

순차적으로 정의된 회전 순서에 따라 두 번째 회전인 y축 회전각  $\theta$  를 반환

## Example

PITCH (body1.Marker1)

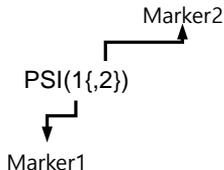
PITCH (body1.Marker1, body2.Marker2)

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

# PSI

: z-x-z 회전 순서로 정의된 Euler Angle 의 첫 번째 회전순서에 해당하는 회전각(PSI)을 반환하는 함수

## Format

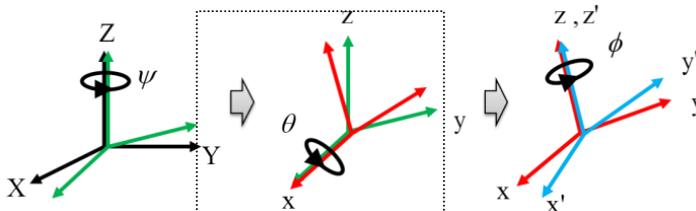


Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

## Formulation



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

순차적으로 정의된 회전 순서에 따라 첫 번째 회전인 Z축 회전각  $\psi$ 를 반환

## Example

PSI (body1.Marker1)

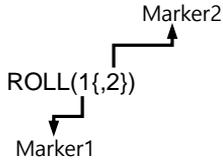
PSI (body1.Marker1, body2.Marker2)

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

## ROLL

: z-y-x 회전 순서로 정의된 Euler Angle 의 세 번째 회전순서에 해당하는 회전각(ROLL)을 반환하는 함수

### Format



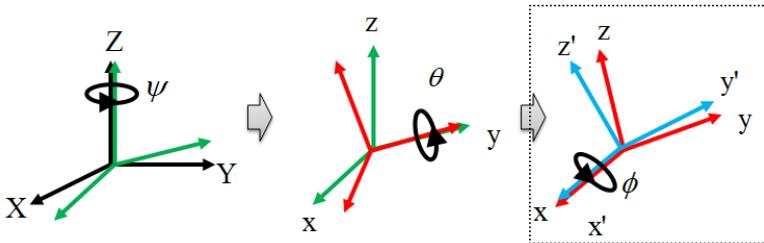
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

### Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

### Formulation



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

순차적으로 정의된 회전 순서에 따라 세 번째 회전인 x축 회전각  $\phi$  를 반환

### Example

ROLL (body1.Marker1)

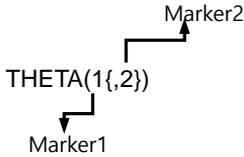
ROLL (body1.Marker1, body2.Marker2)

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

# THETA

: z-x-z 회전 순서로 정의된 Euler Angle 의 두 번째 회전순서에 해당하는 회전각(THETA)을 반환하는 함수

## Format



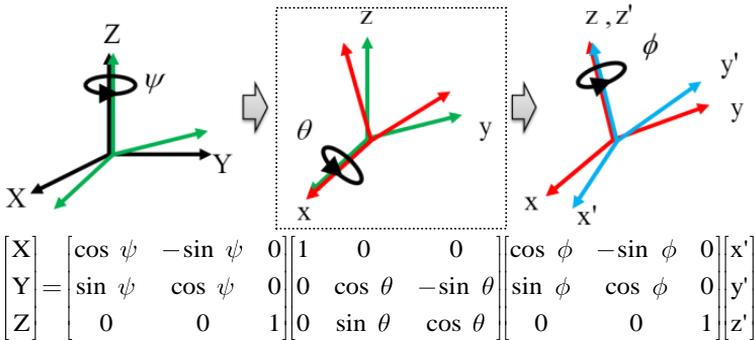
Argument List

ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

## Formulation



순차적으로 정의된 회전 순서에 따라 두 번째 회전인 x축 회전각  $\theta$  를 반환

## Example

THETA (body1.Marker1)

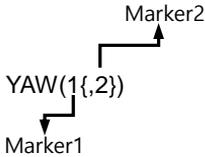
THETA (body1.Marker1, body2.Marker2)

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

# YAW

: z-y-x 회전 순서로 정의된 Euler Angle 의 첫 번째 회전순서에 해당하는 회전각(YAW)을 반환하는 함수

## Format

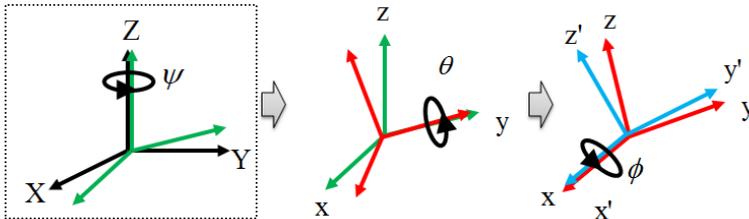


Argument List	
ID	Entity
1	Body1.Marker1
2	Body2.Marker2

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용

## Formulation



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

순차적으로 정의된 회전 순서에 따라 첫 번째 회전인 z축 회전각  $\psi$ 를 반환

## 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*

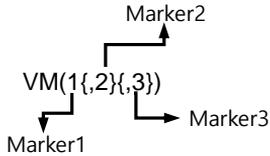
*WZ*

*AY*

# VM

: 하나의 Marker의 절대속도 또는 두 Marker의 상대속도의 절대값을 계산하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk3)}\vec{V}_{mk1}$  : Marker3으로부터의 Marker1의 속도벡터

${}^{(mk3)}\vec{V}_{mk2}$  : Marker3으로부터의 Marker2의 속도벡터

## Example

VM (body1.marker1)

VM (body1.marker1, body2.marker2)

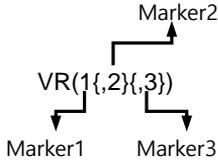
VM(body1.marker1, body2.marker2, body3.marker3)

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

# VR

: 두 Marker 의 상대속도의 절대값에 접근 및 이탈방향에 대해 부호를 부여하여 반환하는 함수로 두 Marker 가 멀어질 경우 양(+)의 값을 가지며 가까워질 경우 음(-)의 값을 반환하는 함수

## Format



Argument List

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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker 적용
Marker3	Reference Marker로서 속도벡터의 기준 Maker가 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

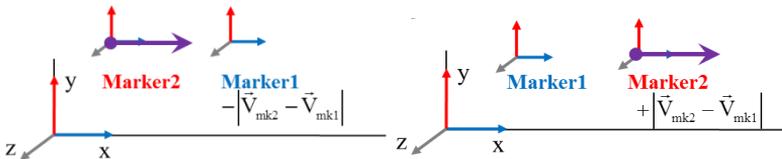
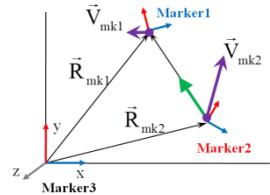
## Formulation

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

$(mk3) \vec{V}_{mk1}$  : Marker3으로부터의 Marker1의 속도벡터

$(mk3) \vec{V}_{mk2}$  : Marker3으로부터의 Marker2의 속도벡터

$DM(\vec{R}_{mk1}, \vec{R}_{mk2})$  :  $\vec{R}_{mk2}$  벡터로부터의  $\vec{R}_{mk1}$  벡터 거리의 크기



### Example

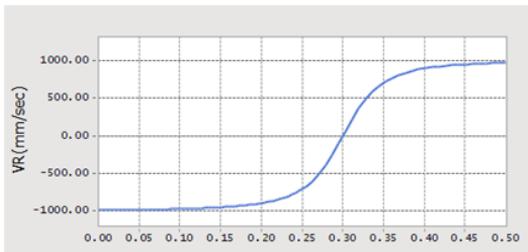
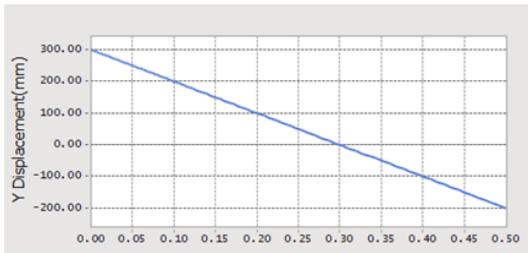
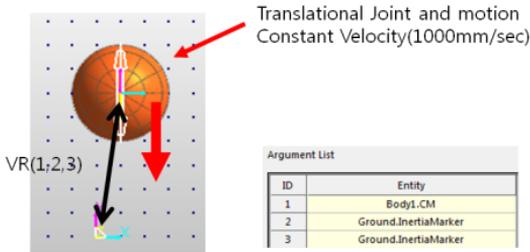
VR (body1.marker1)

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>

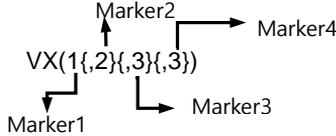
아래 그림과 같이 Body1을 (50,300,0) 위치에 생성하고 Translational Joint를 모델링 한다. 이 Joint에 Motion을 속도 1000mm/sec 부여한다. Inertia Maker와 Body1. 의 CM Marker 간의 VR 함수를 적용하면 Y방향으로의 변위 및 VR 함수의 결과를 다음과 같이 얻을 수 있다. 이러한 VR함수사용의 대표적인 예로 차량 현가 시스템의 Damper(Shock absorber)의 Expression을 이용한 모델링에 주로 활용 된다.



# VX

: 하나의 Marker 에 대한 x 방향 속도 또는 두 Marker 의 상대 속도를 방향기준 Marker 의 x 방향 성분을 반환하는 함수

## Format



Argument List

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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	측 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	속도를 계산하는 기준 Marker로서, Marker4의 각속도에 의한 영향을 고려한 속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\vec{V}_{mk1}$  : Marker4의 각속도를 고려한 Marker1의 속도벡터

${}^{(mk4)}\vec{V}_{mk2}$  : Marker4의 각속도를 고려한 Marker2의 속도벡터

$\hat{x}_{mk3}$  : Marker3의 x방향 단위벡터

## Example

VX(body1.marker1, body2.marker2)

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

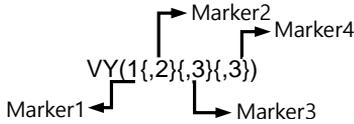
VX(body1.marker1, body2.marker2, body3.marker3, body4.marker4)

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

# VY

: 하나의 Marker 에 대한 y 방향 속도 또는 두 Marker 의 상대 속도를 방향기준 Marker 의 y 방향 성분을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	속도를 계산하는 기준 Marker로서, Marker4의 각속도에 의한 영향을 고려한 속도를 계산할 경우 사용됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\vec{V}_{mk1}$  : Marker4의 각속도를 고려한 Marker1의 속도벡터

${}^{(mk4)}\vec{V}_{mk2}$  : Marker4의 각속도를 고려한 Marker2의 속도벡터

$\hat{y}_{mk3}$  : Marker3의 y방향 단위벡터

## Example

VY(body1.marker1, body2.marker2)

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

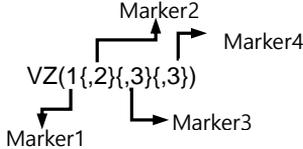
VY(body1.marker1, body2.marker2, body3.marker3, body4.marker4)

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

# VZ

: 하나의 Marker 에 대한 z 방향 속도 또는 두 Marker 의 상대 속도를 방향기준 Marker 의 z 방향 성분을 반환하는 함수

## Format



Argument List

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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	측 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	속도를 계산하는 기준 Marker로서, Marker4의 각속도에 의한 영향을 고려한 속도를 계산할 경우 사용됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\vec{V}_{mk1}$  : Marker4의 각속도를 고려한 Marker1의 속도벡터

${}^{(mk4)}\vec{V}_{mk2}$  : Marker4의 각속도를 고려한 Marker2의 속도벡터

$\hat{z}_{mk3}$  : Marker3의 z방향 단위벡터

## Example

$VZ(\text{body1.marker1}, \text{body2.marker2})$

$VZ(\text{body1.marker1}, \text{body2.marker2}, \text{body3.marker3})$

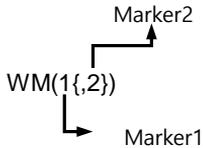
$VZ(\text{body1.marker1}, \text{body2.marker2}, \text{body3.marker3}, \text{body4.marker4})$

$VZ(1,2,3) <\text{Argument: (1)body1.marker1, (2)body2.marker2, (3)body3.marker3}>$

# WM

: 하나의 Marker 의 각속도 또는 두 Marker 사이의 상대 각속도의 절대값을 반환하는 함수

## Format



### Argument List

ID	Entity
1	Body1.Marker1
2	Ground.InertiaMarker

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용

## Formulation

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

$\vec{\omega}_{mk1}$  : InertiaMarker에 대한 Marker1의 각속도벡터

$\vec{\omega}_{mk2}$  : InertiaMarker에 대한 Marker2의 각속도벡터

## Example

WM (body1.marker1)

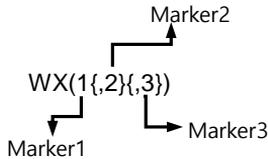
WM (body1.marker1, body2.marker2)

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

# WX

: 하나의 Marker 의 x 축 각속도 또는 두 Marker 에 대한 x 축 방향의 상대 각속도를 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

$\vec{\omega}_{mk1}$  : InertiaMarker에 대한 Marker1의 각속도벡터

$\vec{\omega}_{mk2}$  : InertiaMarker에 대한 Marker2의 각속도벡터

$\hat{x}_{mk3}$  : Marker3의 x 방향 단위벡터

## Example

WX(body1.marker1)

WX(body1.marker1, body2.marker2)

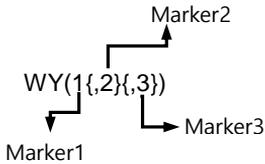
WX(body1.marker1, body2.marker2, body3.marker3)

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

# WY

: 하나의 Marker 의 y 축 각속도 또는 두 Marker 에 대한 y 축 방향의 상대 각속도를 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

$\vec{\omega}_{mk1}$  : InertiaMarker에 대한 Marker1의 각속도벡터

$\vec{\omega}_{mk2}$  : InertiaMarker에 대한 Marker2의 각속도벡터

$\hat{y}_{mk3}$  : Marker3의 y 방향 단위벡터

## Example

WY(body1.marker1)

WYbody1.marker1, body2.marker2)

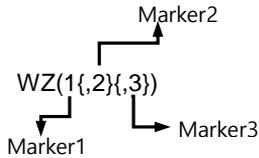
WY(body1.marker1, body2.marker2, body3.marker3)

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

# WZ

: 하나의 Marker 의 z 축 각속도 또는 두 Marker 에 대한 z 축 방향의 상대 각속도를 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	Reference Marker로서 측정하는 방향의 기준이 됨, 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

$\vec{\omega}_{mk1}$  : InertiaMarker에 대한 Marker1의 각속도벡터

$\vec{\omega}_{mk2}$  : InertiaMarker에 대한 Marker2의 각속도벡터

$\hat{z}_{mk3}$  : Marker3의 z 방향 단위벡터

## 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*

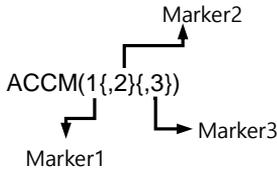
*WDTY*

*WDTZ*

# ACCM

: 하나의 Marker 의 가속도 또는 두 Marker 사이의 상대 가속도에 대한 절대값을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	가속도를 계산하는 기준 Marker로서, Marker3의 각속도 및 각가속도에 의한 영향을 고려한 Marker1과 Marker2의 가속도가 계산됨, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk3)}\vec{a}_{mk1}$  : Marker3의 각속도와 각가속도를 고려한 Marker1의 가속도벡터

${}^{(mk3)}\vec{a}_{mk2}$  : Marker3의 각속도와 각가속도를 고려한 Marker2의 가속도벡터

## Example

ACCM(body1.Marker1)

ACCM(body1.Marker1, body2.Marker2)

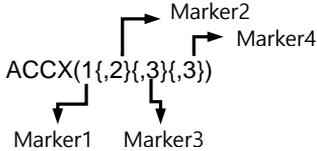
ACCM(body1.Marker1, body2.Marker2, body3.Marker3)

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

# ACCX

: 하나의 Marker 의 x 축 방향 가속도 또는 두 Marker 사이의 상대 가속도를 방향기준 Marker 의 x 축 성분을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	가속도를 계산하는 기준 Marker로서, Marker4의 각속도 및 각가속도에 의한 영향을 고려한 가속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

$\overset{(mk4)}{\vec{a}}_{mk1}$  : Marker4의 각속도와 각가속도를 고려한 Marker1의 가속도벡터

$\overset{(mk4)}{\vec{a}}_{mk2}$  : Marker4의 각속도와 각가속도를 고려한 Marker2의 가속도벡터

$\hat{x}_{mk3}$  : Marker3의 x방향 단위벡터

## Example

ACCX(body1.Marker1, body2.Marker2)

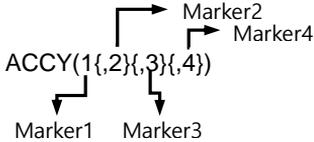
ACCX(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

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

# ACCY

: 하나의 Marker 의 y 축 방향 가속도 또는 두 Marker 사이의 상대 가속도를 방향기준 Marker 의 y 축 성분을 반환하는 함수

## Format



Argument List

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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	가속도를 계산하는 기준 Marker로서, Marker4의 각속도 및 각가속도에 의한 영향을 고려한 가속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\vec{a}_{mk1}$  : Marker4의 각속도와 각가속도를 고려한 Marker1의 가속도벡터

${}^{(mk4)}\vec{a}_{mk2}$  : Marker4의 각속도와 각가속도를 고려한 Marker2의 가속도벡터

$\hat{y}_{mk3}$  : Marker3의 y방향 단위벡터

## Example

ACCY(body1.Marker1, body2.Marker2)

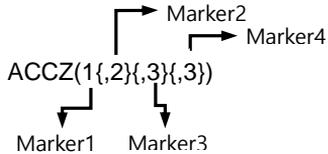
ACCY(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

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

# ACCZ

: 하나의 Marker 의 z 축 방향 가속도 또는 두 Marker 사이의 상대 가속도를 방향기준 Marker 의 z 축 성분을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	가속도를 계산하는 기준 Marker로서, Marker4의 각속도 및 각가속도에 의한 영향을 고려한 가속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\vec{a}_{mk1}$  : Marker4의 각속도와 각가속도를 고려한 Marker1의 가속도벡터

${}^{(mk4)}\vec{a}_{mk2}$  : Marker4의 각속도와 각가속도를 고려한 Marker2의 가속도벡터

$\hat{z}_{mk3}$  : Marker3의 z방향 단위벡터

## Example

ACCZ(body1.Marker1, body2.Marker2)

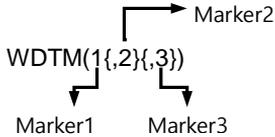
ACCZ(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

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

# WDTM

: 하나의 Marker 의 각가속도 또는 두 Marker 사이의 상대 각가속도에 대한 절대값을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하기 위한 기준 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	각가속도를 계산하는 기준 Marker로서, Marker3의 각가속도에 의한 영향을 고려한 Marker1과 Marker2의 각가속도가 계산됨, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk3)}\dot{\omega}_{mk1}$  : Marker3의 각가속도를 고려한 Marker1의 각가속도벡터

${}^{(mk3)}\dot{\omega}_{mk2}$  : Marker3의 각가속도를 고려한 Marker2의 각가속도벡터

## Example

WDTM(body1.Marker1)

WDTM(body1.Marker1, body2.Marker2)

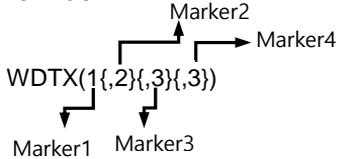
WDTM(body1.Marker1, body2.Marker2, body3.Marker3)

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

# WDTX

: 하나의 Marker 의 x 축에 대한 각가속도 또는 두 Marker 사이의 상대 각가속도를 방향기준 Marker 의 x 축 성분을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	각가속도를 계산하는 기준 Marker로서, Marker4의 각가속도에 의한 영향을 고려한 각가속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

$\overset{(mk4)}{\dot{\omega}}_{mk1}$  : Marker4의 각가속도를 고려한 Marker1의 각가속도벡터

$\overset{(mk4)}{\dot{\omega}}_{mk2}$  : Marker4의 각가속도를 고려한 Marker2의 각가속도벡터

$\hat{x}_{mk3}$  : Marker3의 x방향 단위벡터

## Example

WDTX(body1.Marker1, body2.Marker2)

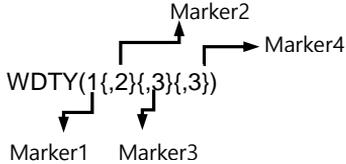
WDTX(body1.Marker1, body2.Marker2, body3.Marker3, body4.Marker4)

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

# WDTY

: 하나의 Marker 의 y 축에 대한 각가속도 또는 두 Marker 사이의 상대 각가속도를 방향기준 Marker 의 y 축 성분을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	각가속도를 계산하는 기준 Marker로서, Marker4의 각가속도에 의한 영향을 고려한 각가속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\dot{\omega}_{mk1}$  : Marker4의 각가속도를 고려한 Marker1의 각가속도벡터

${}^{(mk4)}\dot{\omega}_{mk2}$  : Marker4의 각가속도를 고려한 Marker2의 각가속도벡터

$\hat{y}_{mk3}$  : Marker3의 y방향 단위벡터

## Example

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

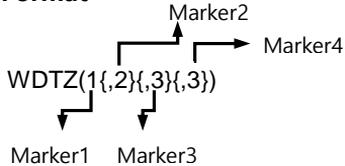
WDTY(body1.Marker1, body2.Marker2)

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

# WDTZ

: 하나의 Marker 의 z 축에 대한 각가속도 또는 두 Marker 사이의 상대 각가속도를 방향기준 Marker 의 z 축 성분을 반환하는 함수

## Format



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

## Arguments definition

Marker1	측정하고자 하는 Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 상대 Marker로서 이름 또는 argument 번호를 입력, 생략될 경우 InertiaMarker를 적용
Marker3	축 정의를 위한 Marker로서 측정하고자 하는 방향의 기준이 되는 Marker 이름 또는 argument번호를 입력하며, 생략될 경우 InertiaMarker 적용
Marker4	각가속도를 계산하는 기준 Marker로서, Marker4의 각가속도에 의한 영향을 고려한 각가속도가 계산됨. 주로 Marker3과 동일한 값을 사용하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk4)}\dot{\omega}_{mk1}$  : Marker4의 각가속도를 고려한 Marker1의 각가속도벡터

${}^{(mk4)}\dot{\omega}_{mk2}$  : Marker4의 각가속도를 고려한 Marker2의 각가속도벡터

$\hat{z}_{mk3}$  : Marker3의 z방향 단위벡터

## 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*

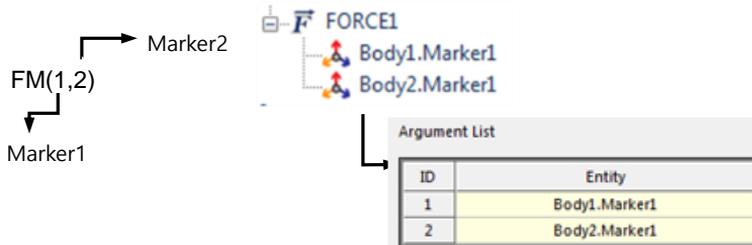
*TY*

*TZ*

# FM

: Force 또는 Joint Entity 가 사용하는 두 Marker 를 이용하여 병진방향 작용력의 절대값을 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로서 이름 또는 argument 번호 입력

## Formulation

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

${}^{(mk2)}\vec{F}_{mk1}$  : Marker2로부터 Marker1으로 작용하는 힘 벡터

## Example

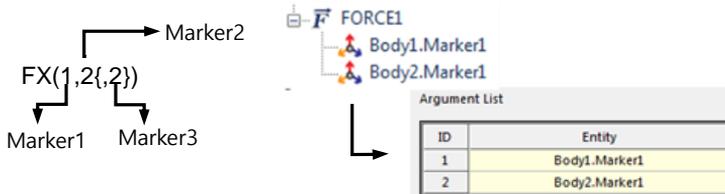
FM(body1.marker1, body2.marker2)

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

# FX

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용력을 계산하여 방향기준 Marker 의 x 방향 성분으로 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로서 이름 또는 argument 번호 입력
Marker3	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력하고, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk2)}\vec{F}_{mk1}$  : Marker2로부터 Marker1으로 작용하는 힘 벡터

$\hat{x}_{mk3}$  : Marker3의 x방향 단위벡터

## Example

FX(body1.marker1, body2.marker2)

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

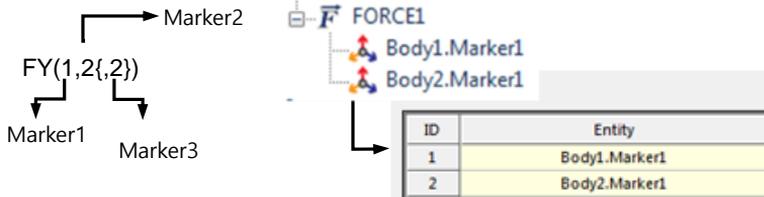
FX(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

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

# FY

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용력을 계산하여 방향기준 Marker 의 y 방향 성분으로 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로서 이름 또는 argument 번호 입력
Marker3	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력하고, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk2)}\vec{F}_{mk1}$  : Marker2로부터 Marker1으로 작용하는 힘 벡터

$\hat{y}_{mk3}$  : Marker3의 y방향 단위벡터

## Example

FY(body1.marker1, body2.marker2)

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

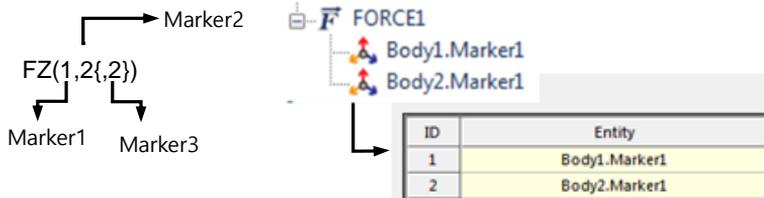
FY(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

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

# FZ

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용력을 계산하여 방향기준 Marker 의 z 방향 성분으로 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로써 이름 또는 argument 번호 입력
Marker3	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력하고, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk2)}\vec{F}_{mk1}$  : Marker2로부터 Marker1으로 작용하는 힘 벡터

$\hat{z}_{mk3}$  : Marker3의 z방향 단위벡터

## Example

FZ(body1.marker1, body2.marker2)

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

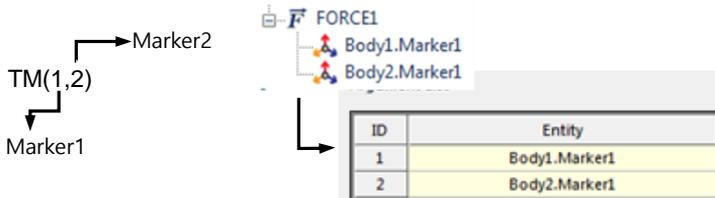
FZ(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

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

# TM

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용토크를 계산하여 절대값을 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로서 이름 또는 argument 번호 입력

## Formulation

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

${}^{(mk2)}\vec{T}_{mk1}$  : Marker2로부터 Marker1으로 작용하는 토크 벡터

## Example

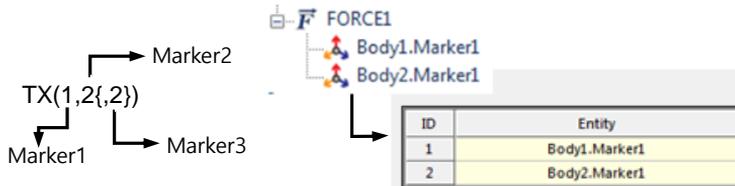
TM(body1.marker1, body2.marker2)

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

# TX

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용토크를 계산하여 방향기준 Marker 의 x 축 성분으로 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로써 이름 또는 argument 번호 입력
Marker3	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk2)}\vec{T}_{mk1}$  : Marker2로부터 Marker1으로 작용하는 토크 벡터

$\hat{x}_{mk3}$  : Marker의 x방향 단위벡터

## Example

TX(body1.marker1, body2.marker2)

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

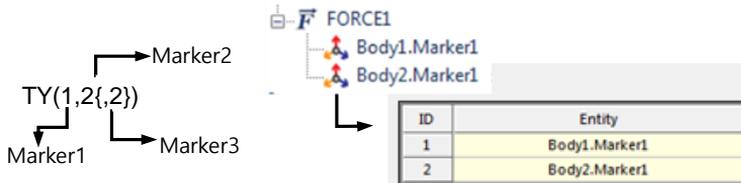
TX(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

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

# TY

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용토크를 계산하여 방향기준 Marker 의 y 축 성분으로 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Force 또는 Joint Entity의 Action 또는 Base Marker(Marker1이 아닌)로서 이름 또는 argument 번호 입력
Marker3	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk2)}\vec{T}_{mk1}$  : Marker2로부터 Marker1의 토크벡터

$\hat{y}_{mk3}$  : Marker3의 y방향 단위벡터

## Example

TY(body1.marker1, body2.marker2)

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

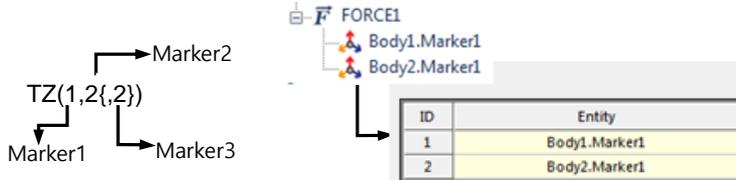
TY(1,2) <Argument: (1)body1.marker1, (2)body2.marker2 >

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

# TZ

: Force 또는 Joint Entity 가 사용하는 두 Marker 사이의 작용토크를 계산하여 방향기준 Marker 의 z 축 성분으로 반환하는 함수

## Format



## Arguments definition

Marker1	측정하고자 하는 Joint 또는 Force Entity의 Action 또는 Base Marker로서 이름 또는 argument 번호 입력
Marker2	측정하고자 하는 Joint 또는 Force Entity의 Action 또는 Base Marker(Marker1이 아닌)로서 이름 또는 argument 번호 입력
Marker3	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력하며, 생략될 경우 InertiaMarker 적용

## Formulation

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

${}^{(mk2)}\vec{T}_{mk1}$  : Marker2로부터 Marker1의 토크벡터

$\hat{z}_{mk3}$  : Marker3의 x방향 단위벡터

## 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 FORCE

*CONTACT*

*MOTION*

*JFRICTION*

*COUPLER*

*GEAR*

*SHT3D\_CF*

*JOINT*

*PTCV*

*CVCV*

*AXIAL*

*TFORCE*

*RFORCE*

*SCREWFORCE*

*BEAM*

*BUSH*

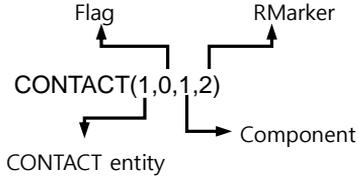
*SPRING*

*MATRIXFORCE*

# CONTACT

: Contact Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	Contact1
2	Ground.Marker1

## Arguments definition

CONTACT entity	결과를 출력하고자 하는 Contact entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

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

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

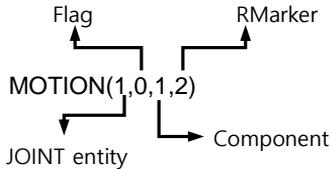
# MOTION

: Joint 에 정의된 Driving Motion 에 의해 발생하는 반력(reaction force)를 계산하여 반환하는 함수, 이 때 Joint 의 Driving Motion 부분은 활성화되어 있어야 함



※ 적용 가능한 joint: Revolute, Translational, Cylindrical, PTCV

## Format



Argument List	
ID	Entity
1	RevJoint1
2	Ground.Marker1

## Arguments definition

JOINT entity	결과를 출력하고자 하는 driving motion이 있는 Joint entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

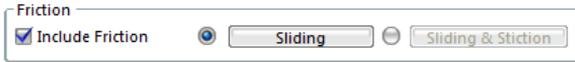
## Example

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

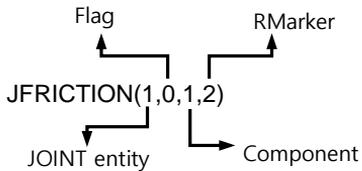
MOTION (1, 0, 1, 2) <Argument: (1) RevJoint1, (2) Ground.InertiaMarker >

# JFRICTION

: Joint 에 정의된 Friction Force 에 의해 발생하는 작용력(action force)를 계산하여 변환하는 함수, 이 때 Joint 의 Friction Force 는 활성화 되어 있어야 함



## Format



Argument List	
ID	Entity
1	RevJoint1
2	Ground.Marker1

## Arguments definition

JOINT entity	결과를 출력하고자 하는 friction이 있는 Joint entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

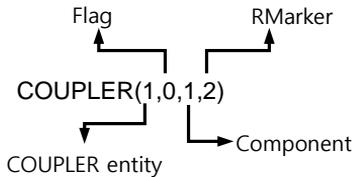
JFRICTION (RevJoint1, 0, 1, Ground.InertiaMarker)

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

# COUPLER

: Coupler Entity 에서 발생하는 반력(reaction force)를 계산하여 반환하는 함수

## Format



Argument List	
ID	Entity
1	Coupler1
2	Ground.Marker1

## Arguments definition

COUPLER entity	결과를 출력하고자 하는 Coupler entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

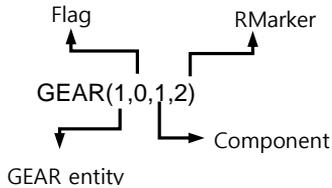
COUPLER (COUPLER1, 0, 1, Ground.InertiaMarker)

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

# GEAR

: Gear Joint Entity 에서 발생하는 반력(reaction force)를 계산하여 반환하는 함수

## Format



Argument List	
ID	Entity
1	Gear1
2	Ground.Marker1

## Arguments definition

GEAR entity	결과를 출력하고자 하는 Gear entity의 이름 또는 Argument 번호
Flag	출력하중의 기준을 설정하는 값으로써 flag 값이 0 이면, 1st Joint에 적용되는 힘이 반환되며 flag 값이 1 이면, 2nd Joint에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

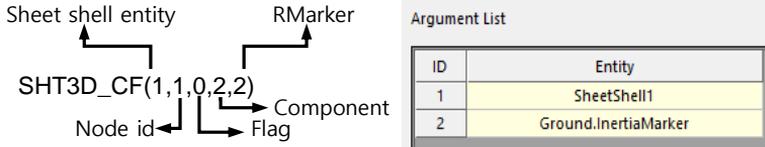
GEAR (GEAR1, 0, 1, Ground.InertiaMarker)

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

# SHT3D\_CF

: MTT3D toolkit 의 Shell Sheet Entity 를 사용하여 정의한 Sheet shell Body 의 Output 으로 정의한 node 에 대해 발생하는 접촉력을 계산하여 반환하는 함수.

## Format



## Arguments definition

Sheet shell entity	결과를 출력하고자 하는 Sheet shell entity의 이름 또는 Argument 번호
Node id	Sheet shell body output page에 설정된 node들 중에서 지정
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

SHT3D\_CF(SheetShell1,357,0,2, Ground.InertiaMarker)

SHT3D\_CF(1,357,0,2,2) <Argument: (1) SheetShell1, (2) Ground.InertiaMarker >

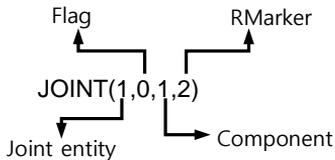
# JOINT

: 입력한 Joint Entity 의 반력(reaction force)을 반환하는 함수

※ 적용 가능한 Joint

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 definition

Joint entity	결과를 출력하고자 하는 Joint entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

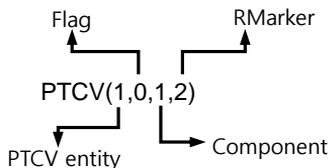
JOINT(JOINT1, 0, 1, Ground.InertiaMarker)

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

# PTCV

: Point to Curve Joint Entity 에서 발생하는 반력(reaction force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	PTCV1
2	Ground.Marker1

## Arguments definition

PTCV entity	결과를 출력하고자 하는 PTCV entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

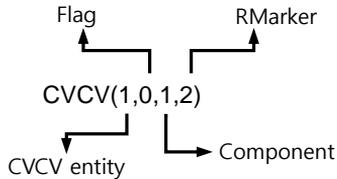
PTCV(PTCV1, 0, 1, Ground.InertiaMarker)

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

# CVCV

: Curve to Curve Joint Entity 에서 발생하는 반력(reaction force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	CVCV1
2	Ground.Marker1

## Arguments definition

CVCV entity	결과를 출력하고자 하는 CVCV entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

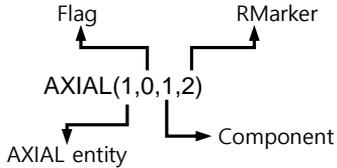
CVCV(CVCV1, 0, 1, Ground.InertiaMarker)

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

# AXIAL

: Axial Force, Rotational Axial Force Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	Axial1
2	Ground.Marker1

## Arguments definition

AXIAL entity	결과를 출력하고자 하는 Axial Force 또는 Rotational Axial Force entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

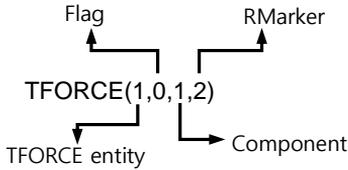
AXIAL(AXIAL1, 0, 1, Ground.InertiaMarker)

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

# TFORCE

: Translational Force Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	Translational1
2	Ground.Marker1

## Arguments definition

TFORCE entity	결과를 출력하고자 하는 Translational Force Entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

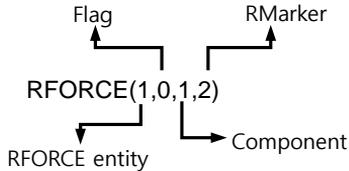
TFORCE (TFORCE1, 0, 1, Ground.InertiaMarker)

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

# RFORCE

: Rotational Force Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	Rotational1
2	Ground.Marker1

## Arguments definition

RFORCE entity	결과를 출력하고자 하는 Rotational Force Entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

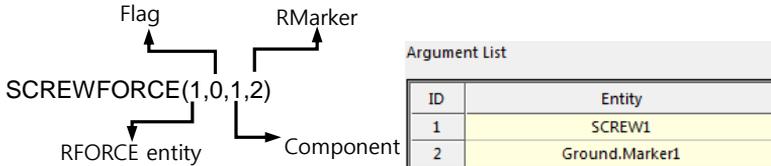
RFORCE (RFORCE1, 0, 1, Ground.InertiaMarker)

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

# SCREWFORCE

: Screw Force Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



## Arguments definition

SCREWFORCE entity	결과를 출력하고자 하는 Screw Force entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

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

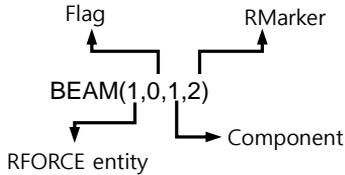
SCREWFORCE (1, 0, 1, 2)

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

# BEAM

: Beam Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	BEAM1
2	Ground.Marker1

## Arguments definition

BEAM entity	결과를 출력하고자 하는 Beam entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

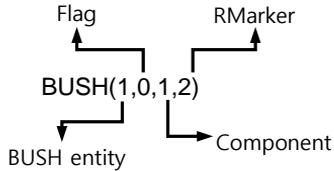
BEAM(Beam1, 0, 1, Ground.InertiaMarker)

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

# BUSH

: Bush Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



Argument List	
ID	Entity
1	Bushing1
2	Ground.Marker1

## Arguments definition

BUSH entity	결과를 출력하고자 하는 Bush entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

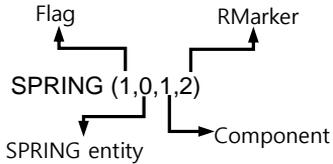
BUSH (BUSH1, 0, 1, Ground.InertiaMarker)

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

# SPRING

: Translational 또는 Rotational Spring Entity 의 작용력(action force)을 반환하는 함수

## Format



Argument List

ID	Entity
1	Spring1
2	Ground.Marker1

## Arguments definition

SPRING entity	결과를 출력하고자 하는 Spring entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력

## Example

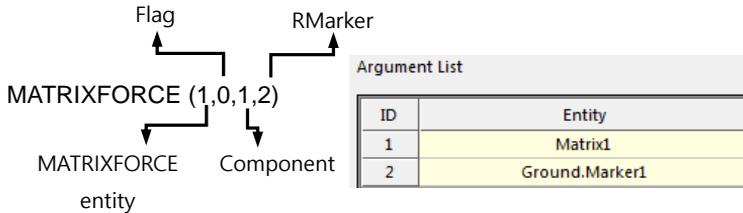
SPRING (SPRING1, 0, 1, Ground.InertiaMarker)

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

# MATRIXFORCE

: Matrix Force Entity 의 계산되는 작용력(action force)을 반환하는 함수

## Format



## Arguments definition

MATRIXFORCE entity	결과를 출력하고자 하는 Matrix force entity의 이름 또는 Argument 번호
Flag	출력하중의 기준Body를 설정하는 값으로써 flag 값이 0 이면, Action Marker (Body)에 적용되는 힘이 반환되며 flag 값이 1 이면, Base Marker (Body)에 적용되는 힘이 반환된다.
Component	반환되는 힘의 구성 요소를 나타내며 숫자에 따라 6분력 및 크기 중 한 요소입력가능 1: FM, 2:FX, 3:FY, 4:FZ, 5:TM, 6:TX, 7:TY, 8:TZ
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument 번호를 입력

## Example

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

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

# SYSTEM ELEMENT

*DIF*

*DIF1*

*VARVAL*

# DIF

: Differential Equation Entity 의 계산식을 이용하여 미분방정식의 적분 계산 값을 반환하는 함수

## Format

DIF(1)  
└─> Diff. Eq

Argument List	
ID	Entity
1	DE1

## Arguments definition

Diff. Eq	DIFFERENTIAL EQUATION entity의 이름 또는 Argument 번호
----------	---

## Formulation

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

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

(where,  $\xi$  : differential variable,  $x$  = Other state variable)

## Example

DIF(Model1.DE1)

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

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

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

Explicit Type의 Differential Equation 이용하여 위와 같은 전기모터의 토크모델의 예를 적용해볼 수 있다.

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

주어진 미분 방정식의 형태를 Explicit Type 으로 정의하면 위 식과 같다.

각 계수(R,L,K)에 대하여 Parametric value를 생성하고, Body, Revolute Joint, Rotational Axial Force를 각각 모델링 한다. 정리된 미분방정식을 Expression, 및 DIF를 각각 형성하여 Rotational Axial Force 모델에 적용한 후 해석한 결과이다.

<모델링>



Parametric Value List

Parametric Values

No	DP	Name	Value	
1	<input checked="" type="checkbox"/>	R	1.	E
2	<input checked="" type="checkbox"/>	L	8.e-002	E
3	<input checked="" type="checkbox"/>	K	6.	E

Expression List

Expressions

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

Differential Equation

Name: IDot

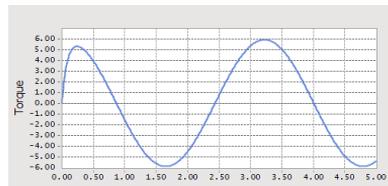
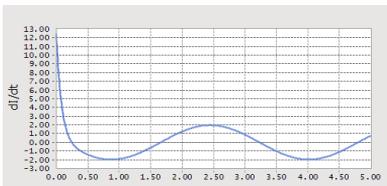
Function Type: Explicit

Standard Differential Equation

Expression Name: EX1

Expression:  $\cos(2 * time) / L - R * DIF(1) / L$

<해석 결과>



# DIF1

: Differential Equation Entity 의 계산식을 이용하여 미분방정식의 시간미분 계산 값을 반환하는 함수

## Format

DIF1(1)  
 ↙ Diff. Eq

Argument List	
ID	Entity
1	DE1

## Arguments definition

Diff. Eq	정의된 미분방정식 entity의 이름 또는 Argument 번호
----------	-------------------------------------

## Formulation

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

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

where,  $\xi$  : differential variable,  $x$  = Other state variable

## Example

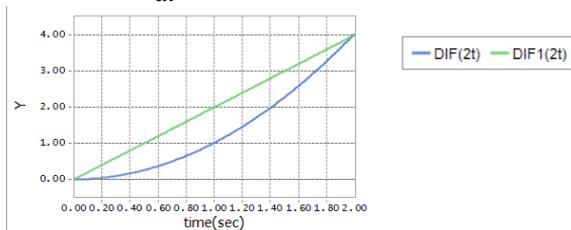
DIF1(Model1.DE1)

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

F(t)= 2 t 에 대하여 DIF 함수 및 DIF1함수를 각각 비교하여 수식과 결과 Expression Scope를 이용하여 비교하면 다음과 같다.

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

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



# VARVAL

: VARIABLE EQUATION Entity 의 계산결과를 Expression 에서 사용하기 위한 함수로써 VARIABLE EQUATION 의 계산 값을 반환

## Format

VARVAL(1)  
 ↘ Variable Eq.

Argument List	
ID	Entity
1	VE1
2	VE2

## Arguments definition

Variable Eq.	VARIABLE EQUATION entity이름 또는 Argument 번호
--------------	---

## Example

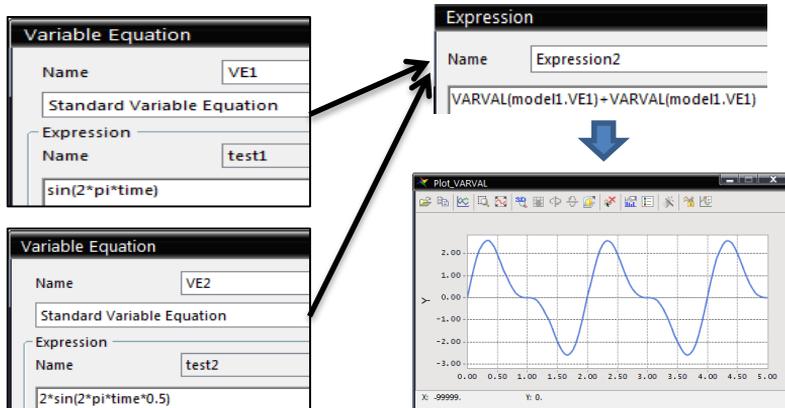
VARVAL (Model1.VE1)

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

<작업 순서>

- (1) Variable Equation에서 사용하기 위한 Expression1 생성:  $\sin(2*\pi*time)$
- (2) Variable Equation Entity 생성(VE1) 후 Expression1을 연결:
- (3) Expression2 생성: VARVAL(Model1. VE1)

아래의 예제는  $\sin(2*\pi*time)$  으로 정의된 test1 VE1과  $2*\sin(2*\pi*time*0.5)$ 로 정의된 test2 함수 VE2를 새로운 Expression에서 아래와 같이 정의하면 볼 수 있는 결과이다.



# ARITHMETIC IF

*IF*

# IF

: 함수에 대하여 조건을 지정하고 만족 여부에 따라 지정된 함수 값을 반환하는 함수

## Format

IF(f1: f2, f3, f4)

## Arguments definition

f1	함수의 값을 계산하기 위한 조건을 대입하기 위한 계산 식을 정의하는 구문
f2	f1이 -IFT보다 작은 경우 수행되는 함수이며, 계산된 값을 반환한다. 실수 또는 실수를 출력하는 함수
f3	f1이 -IFT보다 크거나 같고 IFT보다 작거나 같은 경우 수행되는 함수이며 계산된 값을 반환한다. 실수 또는 실수를 출력하는 함수
f4	f1이 IFT보다 클 경우 수행되는 함수이며, 계산된 값을 반환한다. 실수 또는 실수를 출력하는 함수

※ IFT(If Tolerance): 이 값은 If문이 분기할 때 Expression1의 값이 수치적으로 정확히 0이 계산되기 어려우므로 대한 약간의 범위를 주어 0의 조건에 대해 응답할 수 있도록 보정해주는 파라미터임. RecurDyn GUI에서 Home Tab의 Simulation메뉴의 설정에 이 파라미터 입력 값이 존재함,

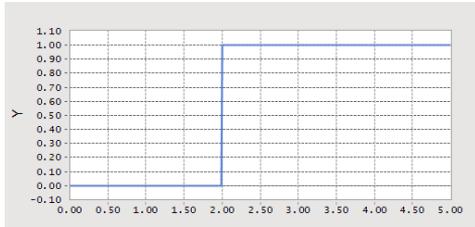
## Formulation

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

## Example

IF함수를 이용하여 Expression을 아래와 같이 구성했을 때 나타낸 결과

IF(TIME-2.0;0,0,1)



# INTERPOLATION

*AKISPL*  
*CUBSPL*

# AKISPL

: Akima spline 함수를 이용하여 곡선보간하는 방법으로써 Spline Entity 를 활용하며, 입력된 변수 x 에 대해 보간한 y 값을 반환하는 함수. Akima spline 함수는 1 차 미분까지 연속이므로 Stiffness 와 같은 Force 요소의 입력값으로 사용하기에 적합함

## Format

AKISPL(X, Z, 1, 0)

Curve name  
Order

Argument List	
ID	Entity
1	Spline1

## Arguments definition

X	AKISPL 함수에 대한 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
Z	AKISPL 함수에 대한 입력변수, 3차원 Spline 함수를 이용할 경우 두 번째 독립변수에 해당한다. 실수를 출력하는 함수를 입력하며, 사용하지 않을 경우 0을 입력
Curve name	SubEntity 에 정의한 Spline Data의 이름 또는 argument 번호
Order	함수의 보간 방법정의 (0 일 경우 그냥 반환, 1 일 경우 함수의 1차 미분 값을 반환, 2일 경우 함수의 2차 미분 값을 반환)

## Formulation

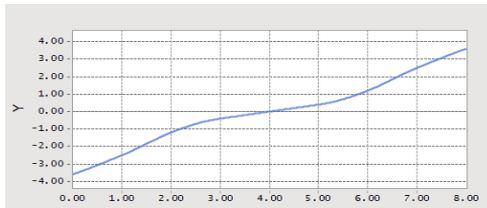
$$AKISPL = \begin{cases} f(x, spline\_data), & \text{Order}=0 \\ df(x, spline\_data) / dx, & \text{Order}=1 \\ d^2 f(x, spline\_data) / dx^2, & \text{Order}=2 \end{cases}$$

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

Spline Data

## Example

아래의 함수를 입력했을 때 나타나는 결과  
AKISPL(time-4,0,1,0)  
<Argument: (1) Spline>



# CUBSPL

: Cubic spline 함수를 이용하여 곡선보간하는 방법으로써 Spline Entity 를 활용하며, 입력된 변수 x 에 대해 보간한 y 값을 반환하는 함수. Cubic spline 함수는 2 차 미분까지 연속이므로 Force 요소는 물론 Motion 의 spline 입력값으로 사용하기에 적합함

## Format

CUBSPL(X, Z, 1, 0)

Curve name  
Order

Argument List

ID	Entity
1	Spline1

## Arguments definition

X	CUBSPL 함수에 대한 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
Z	CUBSPL 함수에 대한 입력변수, 3차원 Spline 함수를 이용할 경우 두 번째 독립변수에 해당한다. 실수를 출력하는 함수를 입력하며, 사용하지 않을 경우 0을 입력
Curve name	SubEntity 에 정의한 Spline Data의 이름 또는 argument 번호
Order	함수의 보간 방법정의 (0 일 경우 그냥 반환, 1 일 경우 함수의 1차 미분 값을 반환, 2일 경우 함수의 2차 미분 값을 반환)

## Formulation

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

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

Spline Data

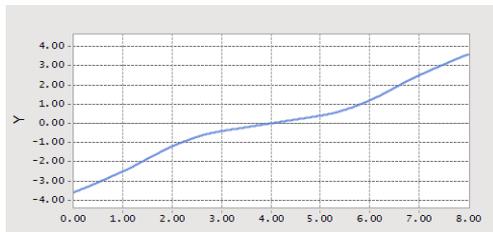
## Example

아래의 함수를 입력했을 때 나타나는 결과

CUBSPL(time,0,Spline1,0)

CUBSPL(time,0,1,0)

<Argument (1) Spline1>



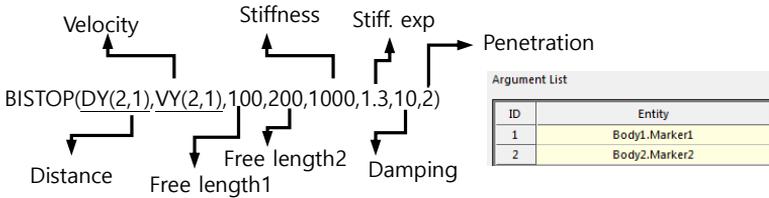
# GENERAL

*BISTOP*  
*CHEBY*  
*FORCOS*  
*FORSIN*  
*HAVSIN*  
*IMPACT*  
*POLY*  
*SHF*  
*STEP*  
*STEP5*  
*SWEEP*

# BISTOP

: 두 Marker의 상대위치와 상대속도를 이용하여 정의된 접촉모델을 이용하여 양 측면에 대하여 접촉력을 계산하여 반환하는 함수

## Format



## Arguments definition

Distance( $\mathcal{X}$ )	접촉 물체의 두 Marker 사이의 상대거리에 해당하는 Expression 함수를 입력
Velocity( $\dot{\mathcal{X}}$ )	접촉 물체의 두 Marker 사이의 상대속도에 해당하는 Expression 함수를 입력
Free length1 ( $\mathcal{X}_1$ )	접촉 물체의 두 Marker 사이에서의 접촉거리에 해당하는 실수 또는 실수를 반환하는 함수, Free length( $\mathcal{X}_1$ )를 이용하여 접촉여부를 판별에 사용
Free length2 ( $\mathcal{X}_2$ )	접촉 물체의 두 Marker 사이에서의 접촉거리에 해당하는 실수 또는 실수를 반환하는 함수, Free length( $\mathcal{X}_2$ )를 이용하여 접촉여부를 판별에 사용
Stiffness( $k$ )	스프링력에 대한 강성계수
Stiffness exponent( $^{exp}$ )	비선형 계수를 의미하며 스프링력에서 변위의 지수 값
Damping( $C_{max}$ )	최대 감쇠계수로, 실수 또는 실수를 반환하는 Expression 함수
Penetration( $d$ )	강쇠력이 최대로 적용되는 침투 깊이

## Formulation

BISTOP =

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

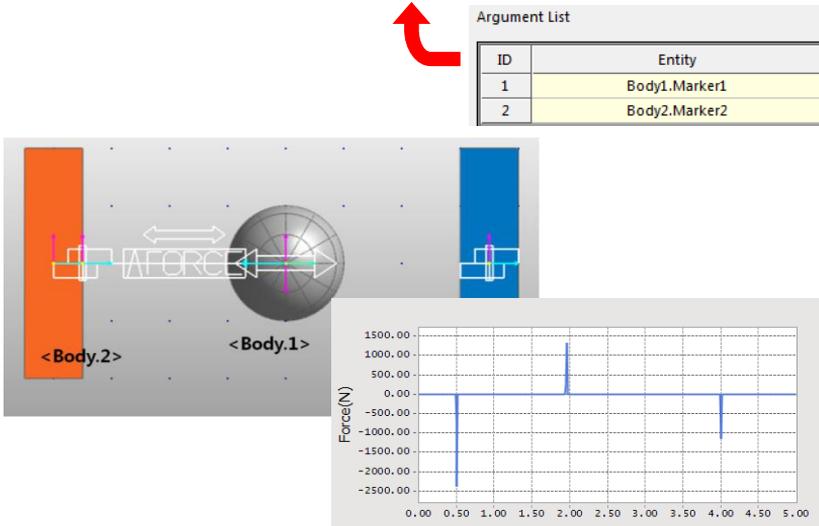
## Example

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

아래 그림과 같이 양 측면에 접촉하는 모델을 Axial Force와 BISTOP 함수를 이용하여 정의하고 Body1의 초기 속도를 부여하여 양 측면에 대한 접촉모델을 정의한 예와 결과

(여기서 두 Marker의 변위함수(DM)에 대한 정의에서 Reference Marker에 두 물체간의 접촉 면을 기준으로 선정 한다면 Free Length를 선정하는데 유리)

BISTOP(DX(1,2,2), VX(1,2,2), 150, 550, 10000, 1.3, 100, 2)



# CHEBY

: 체비쇼프(Chebyshev) 함수의 계산식을 이용하여 각 항의 계수와 변수의 입력을 통해 다항식의 계산 값을 반환하는 함수

## Format

CHEBY(x, x<sub>0</sub>, c<sub>0</sub>, c<sub>1</sub>, ..., c<sub>30</sub>)

## Arguments definition

x	Chebyshev 다항식에 대한 입력변수에 해당하며 일반적으로 실수, Expression 함수 또는 Time 등이 사용.
X <sub>0</sub>	Chebyshev 다항식의 입력변수(x)에 대한 x방향 offset값
c <sub>0</sub> , c <sub>1</sub> , ..., c <sub>30</sub>	Chebyshev 다항식에 사용되는 계수 값으로써 입력된 계수의 수만큼 정의된 Chebyshev 다항식에 따라 식의 선형 중첩계산을 수행, 최대 31개까지 입력가능

## Formulation

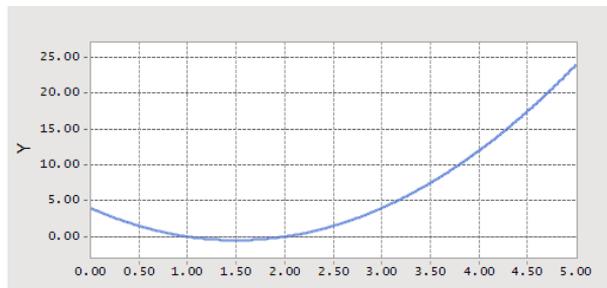
$$C(x) = \sum_{j=0}^n c_j \cdot T_j(x - x_0), \quad 0 \leq j \leq n$$

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

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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



# FORCOS

: Fourier Cosine 급수에 대하여 변수와 계수를 입력하여 함수의 계산 값을 반환하는 함수

## Format

FORCOS(x, x<sub>0</sub>, ω, c<sub>0</sub>, c<sub>1</sub>, ... , c<sub>30</sub>)

## Arguments definition

<b>x</b>	정의된 Fourier Cosine 급수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 사용
<b>X<sub>0</sub></b>	Fourier Cosine 급수의 입력변수( <b>x</b> )에 대한 x방향 offset 값
<b>ω</b>	Fourier Cosine 급수의 기본 주파수로 단위는 Radian
<b>C<sub>0</sub>, C<sub>1</sub>, ... , C<sub>30</sub></b>	Fourier Cosine 급수의 계수로 입력된 계수의 수만큼 정의된 Fourier Cosine 급수에 따라 식의 선형 중첩계산을 수행, 최대 계수 31개 입력가능

## Formulation

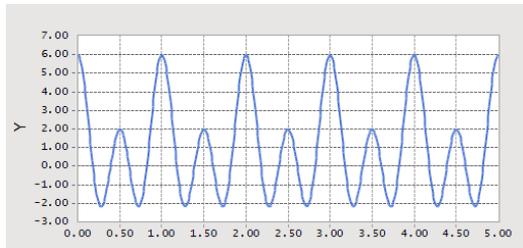
$$F(x) = c_0 + \sum_{j=1}^n c_j \cdot T_j(x - x_0), \quad 0 \leq j \leq n$$

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

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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



# FORSIN

: Fourier Sine 급수에 대하여 변수와 계수를 입력하여 함수의 계산 값을 반환하는 함수

## Format

FORSIN(x, x<sub>0</sub>, ω, c<sub>0</sub>, c<sub>1</sub>, ..., c<sub>30</sub>)

## Arguments definition

x	정의된 Fourier Sine 급수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 사용
X <sub>0</sub>	Fourier Sine 급수의 입력변수(x)에 대한 x방향 offset값
ω	Fourier Sine 급수의 기본 주파수로 단위는 Radian
c <sub>0</sub> , c <sub>1</sub> , ..., c <sub>30</sub>	Fourier Sine 급수의 계수로 입력된 계수의 수만큼 정의된 Fourier Sine 급수에 따라 식의 선형 중첩계산을 수행, 최대 계수 31개 입력가능

## Formulation

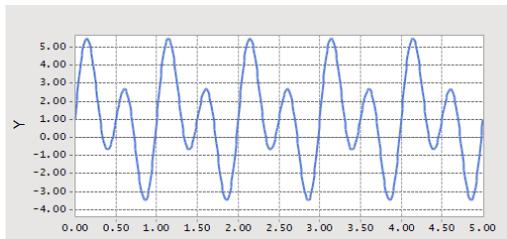
$$F(x) = c_0 + \sum_{j=1}^n c_j \cdot T_j(x - x_0), \quad 0 \leq j \leq n$$

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

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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

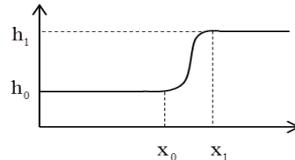


# HAVSIN

: 삼각함수를 이용하여 두 지점을 보간하는 함수로서 일반적으로 두 지점간의 완만한 곡선으로 잇고자 할 경우 사용되며, STEP 및 STEP 5 함수와 유사

## Format

HAVSIN(x,x<sub>0</sub>,h<sub>0</sub>,x<sub>1</sub>,h<sub>1</sub>)



## Arguments definition

x	정의된 HEAVISIN함수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
x <sub>0</sub>	HEAVISIN함수의 시작지점
h <sub>0</sub>	입력변수에 대하여 $x \leq x_0$ 구간의 초기값
x <sub>1</sub>	HEAVISIN함수의 마지막 지점
h <sub>1</sub>	입력변수에 대하여 $x \geq x_1$ 구간의 값

## Formulation

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

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

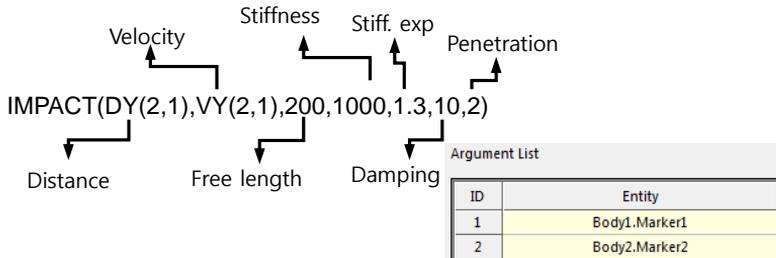
HAVSIN(time, 1.0, 0.0, 4.0, 1.0)



# IMPACT

: 두 Marker 의 상대위치와 상대속도를 이용하여 정의된 접촉모델을 이용하여 접촉력을 반환하는 함수

## Format



## Arguments definition

Distance( $\mathcal{X}$ )	접촉 물체의 두 Marker 사이의 상대거리에 해당하는 Expression 함수를 입력
Velocity( $\dot{\mathcal{x}}$ )	접촉 물체의 두 Marker 사이의 상대속도에 해당하는 Expression 함수를 입력
Free length( $\mathcal{X}_1$ )	<p>접촉 물체의 두 Marker 사이에서의 접촉거리에 해당하는 실수 또는 실수를 반환하는 함수, Free length(<math>\mathcal{X}_1</math>)를 이용하여 접촉여부를 판별에 사용</p>
Stiffness( $k$ )	스프링력에 대한 강성계수
Stiffness exponent( $^{exp}$ )	비선형 계수를 의미하며 스프링력에서 변위의 지수 값
Damping( $C_{max}$ )	감쇠력 계산을 위한 최대 감쇠계수
Penetration( $d$ )	감쇠계수가 0이 되는 깊이

## Formulation

IMPACT =

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

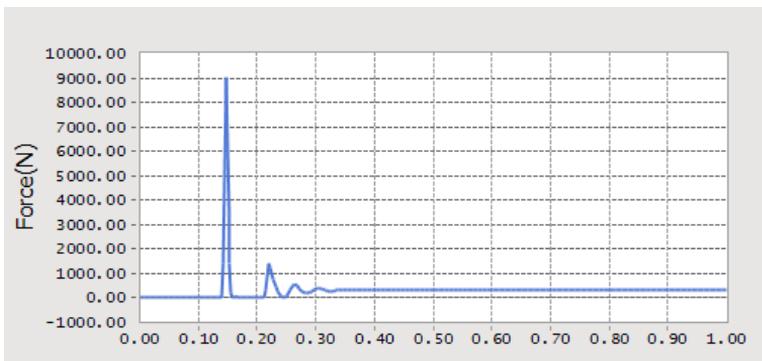
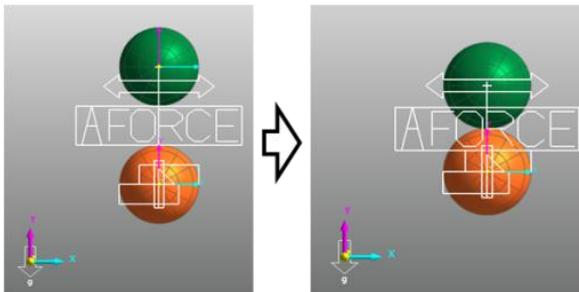
## Example

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

아래에 같이 두 Body를 생성한 후 두 물체간 접촉 모델을 Axial Force와 IMPACT  
함수를 이용하여 정의하고, Axial Force의 출력 그래프를 Scope으로 나타냄

IMPACT(DY(1,2),VY(1,2),200,1000,1.3,10,2)

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



# POLY

: Polynomial 함수에 대하여 입력변수와 다항식의 계수를 입력하여 다항함수 계산 값을 반환하는 함수

## Format

POLY(x,x<sub>0</sub>,a<sub>0</sub>,a<sub>1</sub>,...,a<sub>30</sub>)

## Arguments definition

x	정의된 Polynomial 함수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
X <sub>0</sub>	Poly 함수의 시작지점
a <sub>0</sub> ,a <sub>1</sub> ,...,a <sub>30</sub>	다항식 계수로서 입력된 계수의 수만큼 정의된 다항식 차수를 계산하고, 최대 계수 31개 입력가능

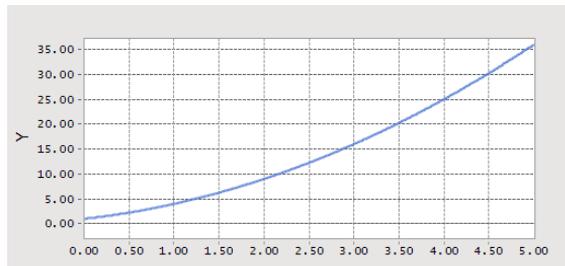
## Formulation

$$\begin{aligned}
 \text{POLY} &= \sum_{j=0}^n a_j (x - x_0)^j && \text{where, } 0 \leq j \leq n \\
 &= a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \dots + a_n(x - x_0)^n
 \end{aligned}$$

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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



# SHF

: 단순 조화(harmonic)함수에 대하여 입력변수와 함수의 계수를 입력하여 Sin 함수의 계산 값을 반환하는 함수

## Format

SHF(x,x<sub>0</sub>,a,ω,φ,b)

## Arguments definition

<b>X</b>	정의된 SHF 함수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
<b>x<sub>0</sub></b>	SHF 함수의 시작지점
<b>a</b>	변수 Sine 파형의 진폭 값, 실수 또는 실수를 반환하는 Expression 함수
<b>ω</b>	Harmonic 함수의 주파수
<b>φ</b>	위상 shift 값
<b>b</b>	Harmonic 함수의 평균값

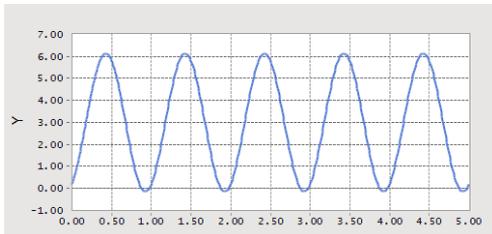
## Formulation

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

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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

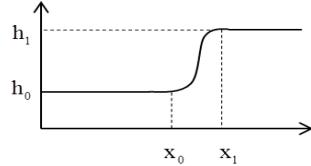


# STEP

: Cubic Polynomial 을 사용하여 두 지점을 보간하는 함수로써  
일반적으로 두 지점간의 완만한 곡선으로 잇고자 할 경우 사용

## Format

STEP(x,x<sub>0</sub>,h<sub>0</sub>,x<sub>1</sub>,h<sub>1</sub>)



## Arguments definition

x	정의된 STEP 함수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
x <sub>0</sub>	STEP 함수의 시작지점
h <sub>0</sub>	입력변수에 대하여 x ≤ x <sub>0</sub> 구간의 초기값
x <sub>1</sub>	STEP함수의 마지막 지점
h <sub>1</sub>	입력변수에 대하여 x ≥ x <sub>1</sub> 구간의 값

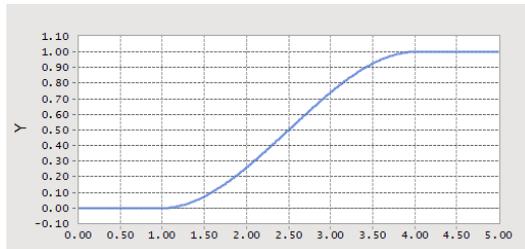
## Formulation

$$\begin{aligned}
 \text{STEP} &= h_0, && \text{when } x \leq x_0 \\
 &= h_0 + (h_1 - h_0) \left[ \frac{x - x_0}{x_1 - x_0} \right]^2 \left\{ 3 - 2 \left[ \frac{x - x_0}{x_1 - x_0} \right] \right\}, && \text{when } x_0 < x \leq x_1 \\
 &= h_1, && \text{when } x \geq x_1
 \end{aligned}$$

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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

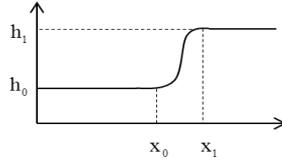


# STEP5

: Quintic Polynomial 을 사용하여 보간하는 함수로서 일반적으로 두 지점간의 완만한 곡선으로 잇고자 할 경우 사용하는 함수

## Format

STEP5(x,x<sub>0</sub>,h<sub>0</sub>,x<sub>1</sub>,h<sub>1</sub>)



## Arguments definition

x	정의된 STEP5 함수에 대하여 입력변수에 해당하며 일반적으로 실수를 출력하는 함수 또는 Time 등이 주로 사용
x <sub>0</sub>	STEP5 함수의 시작지점
h <sub>0</sub>	입력변수에 대하여 x ≤ x <sub>0</sub> 구간의 초기값
x <sub>1</sub>	STEP5함수의 마지막 지점
h <sub>1</sub>	입력변수에 대하여 x ≥ x <sub>1</sub> 구간의 값

## Formulation

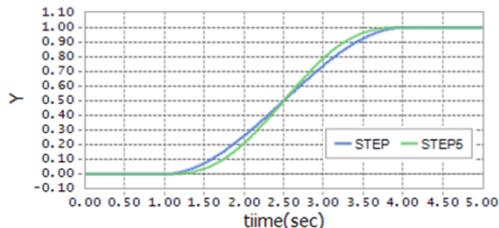
$$\begin{aligned}
 \text{STEP5} &= h_0, && \text{when } x \leq x_0 \\
 &= h_0 + (h_1 - h_0) \cdot \left[ \frac{x - x_0}{x_1 - x_0} \right]^3 \cdot \left\{ 10 - 15 \cdot \frac{x - x_0}{x_1 - x_0} + 6 \cdot \left[ \frac{x - x_0}{x_1 - x_0} \right]^2 \right\}, && \text{when } x_0 < x \leq x_1 \\
 &= h_1, && \text{when } x \geq x_1
 \end{aligned}$$

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄. (STEP 과 STEP5함수를 비교하여 그 차이 비교)

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

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

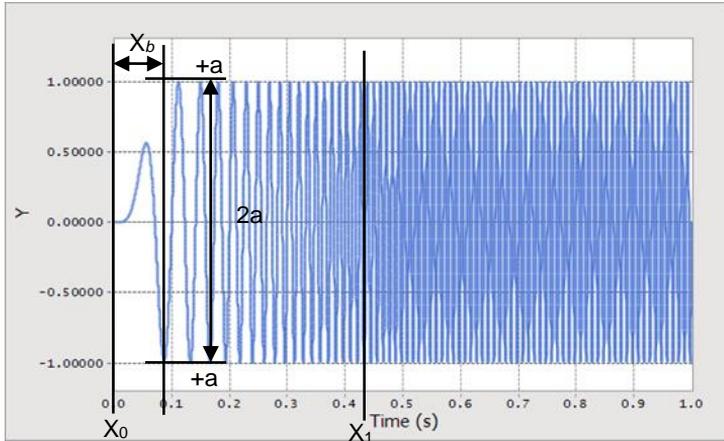


# SWEEP

: 선형적인 주파수 증가와 함께 일정한 진폭의 Sine 함수 값을 반환하는 함수

## Format

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



## Arguments definition

<b>X</b>	SWEEP 함수에 입력되는 독립변수로서 Time, 실수 또는 실수를 반환하는 Expression 함수를 사용
<b>a</b>	Sine 함수의 진폭
$x_0$	SWEEP 함수가 시작되는 $x$ 값
$f_0$	Sine 함수의 최초 주파수
$x_1$	SWEEP 함수가 종료 되는 $x$ 값
$f_1$	Sine 함수의 최종 주파수
$x_b$	SWEEP 함수의 기능이 완전히 활성화 되는 구간

## Formulation

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

$$\text{freq}(x) =$$

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

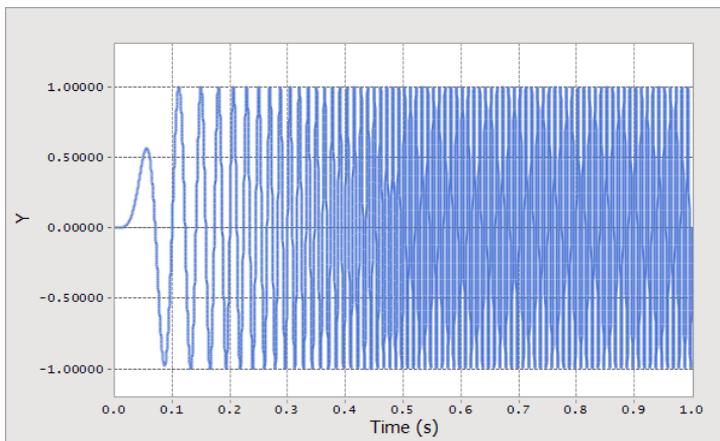
$$\int \text{freq}(x) dx =$$

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

## Example

아래 그래프는 다음 Expression에 대하여 Scope를 이용해 함수에 대한 출력 값을 나타냄

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



# CONTROL & HYDRAULIC

*PIN*

*POUT*

*HIN*

*HOUT*

# PIN

: CoLink 또는 Control toolkit 을 이용한 외부 제어 프로그램 연동해석 수행 시 제어모델에서 계산된 값을 RecurDyn 모델에 입력하기 위한 함수이다. RecurDyn 모델(plant)로 입력(Input)된다는 의미이며, PIN Entity 로 입력되는 값을 반환하는 함수

## Format

PIN(1)  
└─ Plant Input Entity

Argument List	
ID	Entity
1	PlantInput1

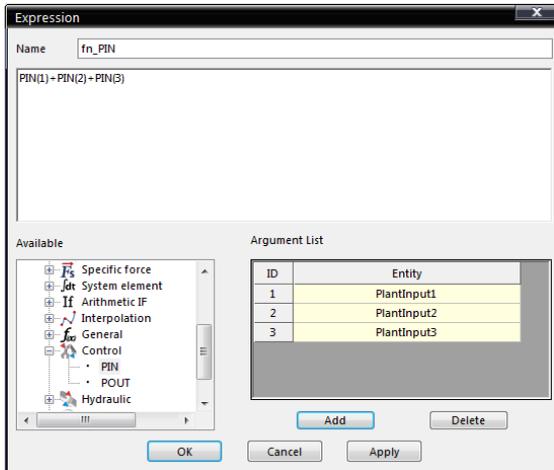
## Arguments definition

Plant Input Entity	PIN entity 의 이름 또는 argument 번호를 사용
--------------------	------------------------------------

## Example

PIN(Mode1.PlantInput1)

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



# POUT

: CoLink 또는 Control toolkit 을 이용한 외부 제어 프로그램 연동해석 수행 시 RecurDyn 모델에서 계산된 값을 제어 모델에 입력하기 위한 함수이다. RecurDyn 모델(plant)에서 출력(Output)된다는 의미이며, POUT Entity 로 입력되는 값을 반환하는 함수

**Format**  
 POUT(1)  
 ↘ Plant Output Entity

Argument List	
ID	Entity
1	PlantOutput1

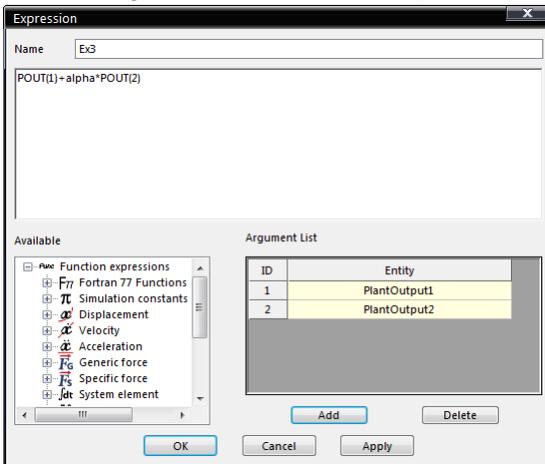
## Arguments definition

Plant Output Entity	POUT entity 의 이름 또는 argument 번호를 사용 
---------------------	--

## Example

POUT(Mode1.PlantOutput1)

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



# HIN

: RecurDyn Model 에서 계산된 값을 Hydraulic System 으로 출력하여 외부 유압계산프로그램(AMESim, FMI)으로 전달하기 위한 함수로서, HIN Entity 의 계산되는 값을 전달함

※ HIN/HOUT 은 PIN/POUT 과 반대의미로 사용됨. 즉, PIN/POUT 은 RecurDyn 모델이 Plant 이며, HIN/HOUT 은 유압 시스템 모델이 Plant 이다.

## Format

HIN(1)  
└─> Hyd\_Input Entity

Argument List	
ID	Entity
1	Hinput1

## Arguments definition

Hyd_Input Entity	HIN entity 의 이름 또는 argument 번호를 사용
	

## Example

HIN(Mode1.HInput1)

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

# HOUT

: 외부 유압시스템 제어 프로그램(AMESim, FMI)의 모델에서 계산된 값을 RecurDyn Model 에서 사용하기 위한 함수이다. 유압시스템 모델(plant)로 입력(Input)된다는 의미이며, HOUT Entity 로 입력되는 값을 RecurDyn 모델이 사용

※ HIN/HOUT 은 PIN/POUT 과 반대의미로 사용됨. 즉, PIN/POUT 은 RecurDyn 모델이 Plant 이며, HIN/HOUT 은 유압 시스템 모델이 Plant 이다.

## Format

HOUT(1)  
└─┬─> Hyd\_Output Entity

ID	Entity
1	HOutput1

## Arguments definition

Hyd_Output Entity	HOUT entity 의 이름 또는 argument 번호를 사용
-------------------	-------------------------------------

## Example

HOUT(Mode1.HOutput1)

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

# SENSOR

*SNSR*

*EVTIME*

*SENSOR*

*SENSORONTIME*

*SENSOROFFTIME*

*SENSORDISTANCE*

*SENSORDETECTEDPOINT*

# SNSR

: 각종 Toolkit 에 만들어진 여러 가지 Sensor 에 대한 측정 결과를 반환하는 함수

## Format

SNSR(1)  
 ↳ SENSOR entity

Argument List	
ID	Entity
1	Sensor1

## Arguments definition

SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
---------------	-----------------------------------

※ 지원하는 Sensor 의 종류 및 반환 값

Toolkit	Sensor종류r	반환 값
Belt	Speed	벨트 segment의 속도.
	Distance	Sensor의 중심위치부터 벨트 segment까지의 거리
	Slip	풀리(pulley)와 벨트의 속도차이
	Tension	Tension of the belt segment.
	Displacement	Sensor의 중심위치부터 벨트 segment까지의 x방향 거리
Chain	Distance	Sensor의 중심위치부터 체인 링크까지의 거리
	Speed	체인 링크의 속도
	Tension	체인 링크의 장력(tension)
MTT2D	Speed	Sheet body의 속도
	Distance	Sensor의 중심위치부터 sheet body까지의 거리
	Event	On = 1, Off = 0
MTT3D	Speed	Sheet body의 속도
	Distance	Sensor의 중심위치부터 sheet body까지의 거리
	Event	On = 1, Off = 0
Track_LM	Displacement	Sensor의 중심위치부터 Track Link까지의 x방향 거리
Track_HM	Displacement	Sensor의 중심위치부터 Track Link까지의 x방향 거리

## Example

SNSR (mode1.Sensor1)

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

# EVTIME

MTT2D 혹은 3D 에서 사용하는 각종 Sensor Entity 가 On 이 되는 순간의 Simulation time 을 반환하는 함수. 그 외의 모든 순간에는 0 이 출력

## Format

EVTIME(1)  
└──┬──> SENSOR entity

ID	Entity
1	Sensor1

## Arguments definition

SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
---------------	-----------------------------------

## Example

EVTIME (Sensor1)

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

# SENSOR

: Laser 및 PointInBox Sensor Entity 에서 계산되는 값을 반환하는 함수로써 센서의 상태 값에 따라 0 과 1 반환(on=1, off=0)

## Format

SENSOR(1)  
└───┬───> SENSOR entity

Argument List	
ID	Entity
1	Sensor1

## Arguments definition

SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
---------------	-----------------------------------

## Example

SENSOR (1)

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

# SENSORONTIME

: Laser 및 PointInBox Sensor Entity 에서 계산되는 값이 On 되는 순간의 Simulation time 을 반환하는 함수. 그 외의 모든 순간에는 0 이 출력

## Format

SENSORONTIME(1)  
SENSOR entity ←

ID	Entity
1	Sensor1

## Arguments definition

SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
---------------	-----------------------------------

## Example

SENSORONTIME (Sensor1)

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

# SENSOROFFTIME

: Laser 및 PointInBox Sensor Entity 에서 계산되는 값이 Off 되는 순간의 Simulation time 을 반환하는 함수. 그 외의 모든 순간에는 0 이 출력

## Format

SENSOROFFTIME(1)  
SENSOR entity ←

ID	Entity
1	Sensor1

## Arguments definition

SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
---------------	-----------------------------------

## Example

SENSOROFFTIME (Sensor1)

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

# SENSORDISTANCE

: Laser Sensor Entity 에서 센서의 상태가 On 된 순간부터 Sensor 의 목표 Geometry 까지의 변위 값을 반환하는 함수, 센서의 상태가 Off 일 경우 0 을 반환

## Format

SENSORDISTANCE(1)

SENSOR entity ←

ID	Entity
1	Sensor1

## Arguments definition

SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
---------------	-----------------------------------

## Example

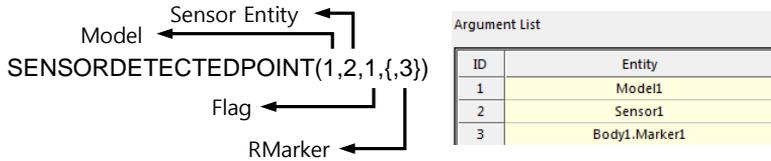
SENSORDISTANCE(Sensor1)

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

# SENSORDETECTEDPOINT

: Laser Sensor Entity 에서 센서가 감지하는 목표 위치에서 발생하는 위치, 속도, 가속도 값을 반환하는 함수, 센서의 감지 상태가 Off 일 경우 0 을 반환

## Format



## Arguments definition

Model	Model의 이름 또는 arguments 번호
SENSOR entity	SENSOR entity의 이름 또는 arguments 번호
Flag	반환되는 변위, 속도 또는 가속도의 종류를 나타내며, 입력 문자에 따라 계산 값을 반환한다. 숫자에 따라 측정위치에 대한 12가지 상태 값을 선택 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	측정하고자 하는 방향의 기준 Marker로서 이름 또는 argument 번호

## Example

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

SENSORDETECTEDPOINT (1,2,1,3)

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

# STRESS & STRAIN

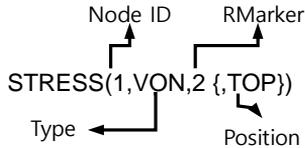
*STRESS*

*STRAIN*

# STRESS

: Flexible Body 의 지정 Node 에 발생하는 응력을 계산하여 반환하는 함수

## Format



Argument List	
ID	Entity
1	Body1_FE.Node50051
2	Ground.InertiaMarker

## Arguments definition

Node ID	응력을 계산하고자 하는 FFlex 또는 RFlex Body의 이름을 포함한 node의 번호 또는 Argument 번호를 입력, 해당 node는 반드시 output node로 설정되어 있어야 사용 가능함		
Type	반환되는 응력의 종류를 지정하는 기호로써 원하는 Type Name을 기입		
	Stress Type	Type Name	Comment
	Basic Stress	X	x direction from x face
		Y	y direction from y face
		Z	z direction from z face
		XY	y direction from x face
		YZ	z direction from y face
		ZX	z direction from z face
	Principal Stress	S1	First Principal
		S2	Second Principal
		S3	Third Principal
	Von-mises Stress	VON	Von Mises stress
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력		
Position	Shell element의 측정위치로써 TOP 또는 BOTTOM 입력		

## Example

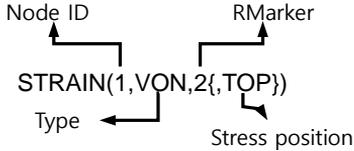
STRESS(Body1\_FE.node1,X,Ground.InertiaMarker)

STRESS(1,VON,2) <Argument: (1) Body1\_FE.node1 (2) Ground.InertiaMarker >

# STRAIN

: Flexible Body 의 지정 Node 에서 발생하는 변형률을 계산하여 반환하는 함수

## Format



Argument List	
ID	Entity
1	Body1_FE.Node50051
2	Ground.InertiaMarker

## Arguments definition

Node ID	변형률 계산하고자 하는 FFlex 또는 RFlex Body의 이름을 포함한 node의 번호 또는 Argument 번호를 입력, 해당 node는 반드시 output node로 설정되어 있어야 사용 가능함		
Type	반환되는 변형률의 형태를 지정하는 기호를 입력한다.		
	Strain Type	Type Name	Comment
	Basic Strain	X	x direction from x face
		Y	y direction from y face
		Z	z direction from z face
		XY	y direction from x face
		YZ	z direction from y face
		ZX	z direction from z face
	Principal Strain	E1	First Principal
		E2	Second Principal
		E3	Third Principal
	Von-mises Strain	VON	Von Mises stress
RMarker	측정하고자 하는 방향의 기준 Marker로서, 이름 또는 argument번호를 입력		
Position	Shell element의 측정위치로써 TOP 또는 BOTTOM 입력		

## Example

STRAIN(Body1\_FE.node1,X,Ground.InertiaMarker)

STRAIN(1,VON,2) <Argument: (1) Body1\_FE.node1 (2) Ground.InertiaMarker >

## 색인

ABS .....	12	DM .....	43
ACCM .....	65	DTOR .....	38
ACCX .....	66	DX .....	44
ACCY .....	67	DY .....	45
AC CZ .....	68	DZ .....	46
ACOS .....	13	EVTIME .....	132
AIN T .....	14	EXP .....	22
AKISPL .....	108	FM .....	74
ANINT .....	15	FORCOS .....	114
ATAN .....	17	FORSIN .....	115
ATAN2 .....	18	FX .....	75
AX .....	40	FY .....	76
AXIAL .....	92	FZ .....	77
AY .....	41	GEAR .....	87
AZ .....	42	HAVSIN .....	116
BEAM .....	96	HIN .....	128
BISTOP .....	111	HOUT .....	129
BUSH .....	97	IF .....	106
CHEBY .....	113	IMPACT .....	117
CONTACT .....	83	JFRICTION .....	85
COS .....	19	JOINT .....	89
COSH .....	20	LOG .....	23
COUPLER .....	86	LOG10 .....	24
CUBSPL .....	109	MATRIXFORCE .....	99
CVCV .....	91	MAX .....	25
DIF .....	101	MIN .....	26
DIF1 .....	103	MOD .....	27
DIM .....	21	MOTION .....	84

PHI .....	47	STRAIN .....	140
PI .....	36	STRESS .....	139
PIN .....	126	SWEEP.....	123
PITCH .....	48	TAN .....	32
POLY.....	119	TANH.....	33
POUT .....	127	TFORCE.....	93
PSI .....	49	THETA.....	51
PTCV.....	90	TIME .....	35
RFORCE .....	94	TM .....	78
ROLL.....	50	TX .....	79
RTOD .....	37	TY .....	80
SCREWFORCE .....	95	TZ .....	81
SENSOR.....	133	VARVAL.....	104
SENSORDETECTEDPOINT ...	137	VM .....	54
SENSORDISTANCE .....	136	VR .....	55
SENSOROFFTIME .....	135	VX .....	57
SENSORONTIME .....	134	VY .....	58
SHF .....	120	VZ .....	59
SHT3D_CF.....	88	WDTM .....	69
SIGN .....	28	WDTX.....	70
SIN .....	29	WDTY.....	71
SINH.....	30	WDTZ .....	72
SNSR .....	131	WM .....	60
SPRING .....	98	WX .....	61
SQRT .....	31	WY .....	62
STEP.....	121	WZ .....	63
STEP5.....	122	YAW,.....	52

# RECURDYN

이 책의 모든 저작권 및 출판권은 평선베이주에 있습니다.

저작권법에 의해 한국 내에서 보호받는 저작물이므로 무단전재, 무단배포와 복사 및 제본을 금합니다.

본 내용은 RecurDyn V8R2를 기반으로 작성되었습니다.