

Guía de Comandos de SymPy Mechanics

Esta guía recopila algunos de los comandos principales del módulo `sympy.physics.mechanics` (SymPy Mechanics) basado en las solicitudes realizadas por algunos del curso, junto con ejemplos básicos para cada uno. Se incluyen funciones como `pos_from` como mecanismo de ayuda de distancia entre puntos en diferentes sistemas coordenados, además de herramientas fundamentales para modelar sistemas dinámicos vistos en el curso:

1. KanesMethod

- `KanesMethod`: Método de Kane para derivar ecuaciones de movimiento.

Ejemplo:

```
from sympy.physics.mechanics import KanesMethod
KM = KanesMethod(N, q_ind=[q1], u_ind=[u1], kd_eqs=[q1.diff()-u1])
KM.kanes_equations([F], [P])
```

2. LagrangesMethod

- `LagrangesMethod`: Método de Lagrange para obtener ecuaciones de movimiento.

Ejemplo:

```
from sympy.physics.mechanics import LagrangesMethod
LM = LagrangesMethod(L, [q1])
LM.lagranges_equations()
```

3. ReferenceFrame

- `ReferenceFrame`: Representa un sistema de referencia.
- `orientnew()`: Crea un nuevo marco orientado respecto a otro.
- `dcm()`: Matriz de cosenos directores.
- `ang_vel_in()`: Velocidad angular relativa.

Ejemplo:

```
N = ReferenceFrame('N')
A = N.orientnew('A', 'Axis', [q1, N.z])
A.dcm(N)
```

4. Point

- `Point`: Define un punto en el espacio.
- `set_vel()`: Define la velocidad de un punto.
- `pos_from()`: Vector de posición desde otro punto.

Ejemplo:

```
O = Point('O')
P = Point('P')
P.set_pos(O, 3*N.x)
P.pos_from(O)
```

5. Particle y RigidBody

- `Particle`: Representa una partícula con masa.
- `RigidBody`: Representa un cuerpo rígido con inercia y referencia.
- `inertia()`: Define el tensor de inercia.

Ejemplo:

```
m, r = symbols('m r')
P = Particle('P', O, m)
I = inertia(N, m*r**2, m*r**2, m*r**2)
B = RigidBody('B', O, N, m, (I, O))
```

6. Dynamicsymbols

- `dynamicsymbols()`: Declara variables dependientes del tiempo.

Ejemplo:

```
from sympy.physics.mechanics import dynamicsymbols
q1, q2 = dynamicsymbols('q1 q2')
dq1 = q1.diff()
```

7. Fuerzas y Torques

- **Fuerzas**: Se representan como tuplas (punto, vector).
- **Torques**: Se representan como tuplas (marco, vector).

Ejemplo:

```
F = (P, 10*N.x)
T = (N, 5*N.z)
```

8. Utilidades adicionales

- `mprint()`: Imprime expresiones legibles.
- `msubs()`: Sustituye valores en expresiones.
- `kinematic_equations()`: Relaciona velocidades angulares.
- `simplify()`: Simplifica expresiones simbólicas.

Ejemplo:

```
expr = (sin(q1)**2 + cos(q1)**2)
simplify(expr) # Retorna 1
```