```
close("all"); clear; clc;
setmadsympath;
```

```
t = sym('t');
px = DynamicVariable('x');
pz = DynamicVariable('z');

q = [
    px;
    pz
    ];
```

```
N = Frame;
```

```
O = Point;
Ob = O.locateNew(px.state*N.x + pz.state*N.z);
```

```
m = sym('m',{'real','positive'});
B = Body(N,Ob,zeros(3,'sym'),m);
```

```
g = sym('g',{'real','positive'});
B.applyForce(N,Ob,-m*g*N.z);
```

```
vx = DynamicVariable('u');
vz = DynamicVariable('w');

u = [
    vx;
    vz
    ];
```

```
Vb = Twist(Pose(N,Ob)).vector;

kdes = [
    vx.state - Vb(4);
    vz.state - Vb(end)
    ];
```

```
eom = kanesMethod(q,u,kdes,B);
```

```
x = [q;u];
M = simplify(expand(eom.MassMatrix));
f = simplify(expand(eom.ForcingVector));
```

```
sys = [q.rate;u.rate] - simplify(expand(M\f));
pprint(sys)
```

$$\begin{bmatrix} \dot{x} - u \\ \dot{z} - w \\ \dot{u} \\ g + \dot{w} \end{bmatrix}$$

```
px0 = sym('x_0');
pz0 = sym('z_0');
vx0 = sym('u_0');
vz0 = sym('w_0');

x0 = [
    px0;
    pz0;
    vx0;
    vz0
    ];
```

```
pxf = sym('l',{'real','positive'});
pzf = 0;
vxf = sym('u_f',{'real','positive'});
vzf = sym('w_f',{'real'});
assume(vzf < 0);

xf = [
    pxf;
    pzf;
    vxf;
    vzf
    ];

tr = sym('t_r',{'real','positive'});
tf = sym('t_f',{'real','positive'});

fx = symfun(x);

cond = [
    fx(0) == x0;
    fx(tf) == xf
    ];

sol = dsolve(sys,fx(tr) == x0);

sol = [
    sol.x;
    sol.z;
    sol.u;
    sol.w
    ];

sol = simplify(expand(x.state - sol));

pprint(sol);
```

$$\begin{bmatrix} x - x_0 - t\,u_0 + t_r\,u_0 \\ \frac{g\,t^2}{2} - g\,t\,t_r - w_0\,t + \frac{g\,{t_r}^2}{2} + w_0\,t_r + z - z_0 \\ u - u_0 \\ w - w_0 + g\,t - g\,t_r \end{bmatrix}$$

```
r = sym('r',{'real','positive'});
pitch = DynamicVariable('theta');
Np = N.orientNew('y',pitch.state);
Op = O.locateNew(r*Np.z);
pprint(posFrom(Op));
```

$$\begin{bmatrix} r\,s_\theta \\ 0 \\ c_\theta\,r \end{bmatrix}$$

```
Vp = Twist(Pose(Np,Op)).vector;
pprint(Vp)
```

$$\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \\ r\,\dot{\theta} \\ 0 \\ 0 \end{bmatrix}$$

```
ovars = [
    px0;
    pz0;
    px.state;
    pz.state;
    vx0;
    vz0;
    vx.state;
    vz.state
    ];
```

```
nvars = [
    r*sin(pitch.State);
    r*cos(pitch.State);
    pxf;
    0;
    Vp(4);
    0;
    Vp(4);
    vzf
    ];

eqns = subs(sol,ovars,nvars);

w = sym('omega',{'real','positive'});
pitch0 = sym('theta_0',{'real'});
assume(pitch0 < 0);

ovars = [
    pitch.rate;
    pitch.state
    ];

nvars = [
    w;
    w*tr + pitch0
    ];

eqns = subs(eqns,ovars,nvars);
eqns = subs(eqns,t,tf);
eqns = simplify(expand(eqns));

eqns = nonzeros(eqns);

pprint(eqns);
```

$$\begin{bmatrix} l - r\sin(\theta_0 + \omega\, t_r) - \omega\, r\, t_f + \omega\, r\, t_r \\ \frac{g\, {t_f}^2}{2} - r\cos(\theta_0 + \omega\, t_r) + \frac{g\, {t_r}^2}{2} - g\, t_f\, t_r \\ w_f + g\, t_f - g\, t_r \end{bmatrix}$$

```
vars = [tf,pxf,vzf];
s = solve(eqns,vars,'ReturnConditions',1);
sol_tf = s.t_f;
sol_l = s.l;
sol_vz = s.w_f;

sol_sys = [
    sol_tf;
    sol_l;
    sol_vz
    ];

sol_sys = simplify(expand(sol_sys));
pprint(sol_sys);
```

$$\begin{bmatrix} t_r + \dfrac{\sqrt{2}\ \sqrt{r}\ \sqrt{\cos(\theta_0 + \omega\, t_r)}}{\sqrt{g}} \\ r\sin(\theta_0 + \omega\, t_r) + \dfrac{\sqrt{2}\ \omega\, r^{3/2}\ \sqrt{\cos(\theta_0 + \omega\, t_r)}}{\sqrt{g}} \\ -\sqrt{2}\ \sqrt{g}\ \sqrt{r}\ \sqrt{\cos(\theta_0 + \omega\, t_r)} \end{bmatrix}$$