```
% t をパラメータとする角度 theta と独立な距離 r を定義する
% 仮定は宣言できないことは declare_sample.m を参照
syms theta(t)
syms r real

% 位置ベクトルを横ベクトルで定義する
position = r * [cos(theta), sin(theta)]
```

position(t) = $\begin{pmatrix} r\cos(\theta(t)) & r\sin(\theta(t)) \end{pmatrix}$

```

% 速度ベクトルは位置ベクトルの微分で計算できる
% diff( position ) でも diff( position, t )でも構わない
velocity = diff( position )
```

velocity(t) =

$$\begin{pmatrix} -r\sin(\theta(t))\,\frac{\partial}{\partial t}\,\theta(t) & r\cos(\theta(t))\,\frac{\partial}{\partial t}\,\theta(t) \end{pmatrix}$$

```
isequal_diff_vs_difft = isequal( diff( position ), diff( position, t ) )
```

isequal_diff_vs_difft = *logical*
   1

```

% 速度ベクトルの 2 乗を計算する。(r\omega)^2 を期待する
% しかしそうはならない
simplify( velocity * velocity' )
```

ans(t) =

$$r^2\left|\frac{\partial}{\partial t}\,\theta(t)\right|^2\cos(\overline{\theta(t)}-\theta(t))$$

```

% conj(theta) が邪魔
% パラメータを持たない文字として別の文字を定義し、置き換える
% 微分も正しく計算されているのが分かる
syms theta_R omega_R real
subs_R = @(input) subs( input, [theta, diff(theta,t)], [theta_R, omega_R])
```

subs_R = *値をもつ function_handle:*
    @(input)subs(input,[theta,diff(theta,t)],[theta_R,omega_R])

```
velocity_R = subs_R( velocity )
```

velocity_R(t) = $\begin{pmatrix} -\omega_R\, r\sin(\theta_R) & \omega_R\, r\cos(\theta_R) \end{pmatrix}$

```
simplify( subs_R( velocity * velocity' ) )
```

ans(t) = $\omega_R^{\,2}\, r^2$

```

% 加速度ベクトルを定義して、そのあとに置き換える
% diff( X, t, t ) によって 2 階微分が計算できる
```

```
acceleration = diff( position, t, t )
```

acceleration(t) =

$$\left(-r\sin(\theta(t))\frac{\partial^2}{\partial t^2}\theta(t)-r\cos(\theta(t))\left(\frac{\partial}{\partial t}\theta(t)\right)^2\quad r\cos(\theta(t))\frac{\partial^2}{\partial t^2}\theta(t)-r\sin(\theta(t))\left(\frac{\partial}{\partial t}\theta(t)\right)^2\right)$$

```
syms alpha_R real
subs_R = @(input) subs( input, [theta, diff(theta,t), diff(theta,t,t)], [theta_R, omega_R, alph
```

subs_R = *値をもつ function_handle:*

@(input)subs(input,[theta,diff(theta,t),diff(theta,t,t)],[theta_R,omega_R,alpha_R])

```
acceleration_R = simplify( subs_R( acceleration ) )
```

acceleration_R(t) = $\left(-r\,\left(\cos(\theta_R)\,{\omega_R}^2+\alpha_R\sin(\theta_R)\right)\quad -r\,\left({\omega_R}^2\sin(\theta_R)-\alpha_R\cos(\theta_R)\right)\right)$

```
% 動径方向と回転方向に分ける
% ベクトルを分けるのは１組の直行ベクトルとの内積が良い
r_vec = [cos(theta), sin(theta)];
rot_vec = [-sin(theta), cos(theta)];

acceleration_r = acceleration * r_vec';
acceleration_rot = acceleration * rot_vec';

acceleration_r_R = simplify( subs_R( acceleration_r ) )
```

acceleration_r_R(t) = $-{\omega_R}^2\,r$

```
acceleration_rot_R = simplify( subs_R( acceleration_rot ) )
```

acceleration_rot_R(t) = $\alpha_R\,r$

```
% 端点における質量と端点での力を動径方向と回転方向に定義すれば
% 円運動の方程式が導ける
syms m F_r F_rot real
Netwon_eq = [
    m * acceleration_r_R == F_r
    m * acceleration_rot_R == F_rot
    ]
```

Netwon_eq(t) =

$$\begin{pmatrix}-m\,{\omega_R}^2\,r=F_r\\ \alpha_R\,m\,r=F_{\mathrm{rot}}\end{pmatrix}$$

```
% ちなみに、Netwon_eq(t) の各項目にアクセスするためには Netwon_eq(2,1) は正しくない
try
    Netwon_eq(2,1)
catch ME
    ME.message
end
```

```
ans =
'位置 2 の引数が無効です。 シンボリック関数では 1 個の入力引数が必要ですが、受け取ったのは 2 個です。'
```

```
% Netwon_eq(t) の各項目にアクセスするためにはまず formula を取る
Netwon_eq_f = formula( Netwon_eq )
```

```
Netwon_eq_f =
```

$$\begin{pmatrix} -m\,{\omega_R}^2\,r = F_r \\ \alpha_R\,m\,r = F_{\mathrm{rot}} \end{pmatrix}$$

```
Netwon_eq_f(2,1)
```

ans = $\alpha_R\,m\,r = F_{\mathrm{rot}}$