15 Curvilinear Motion: Normal and Tangential Components

Ref: Hibbeler § 12.7, Bedford & Fowler: Dynamics § 2.3

When the path of a particle is known, an n-t coordinate system with an origin at the location of the particle (at an instant in time) can be helpful in describing the motion of the particle. Hibbeler gives a concise procedure for analysis in section 12.7, which we will apply to the following example.

Example: A skateboarder in a quarter pipe.

A skateboarder has dropped into a 3 meter (radius) quarter pipe. When she's dropped a distance of 1 meter her speed is 3.2 m/s and is increasing by 7.8 m/s^2.

Determine the normal and tangential components of the skateboarder's acceleration.

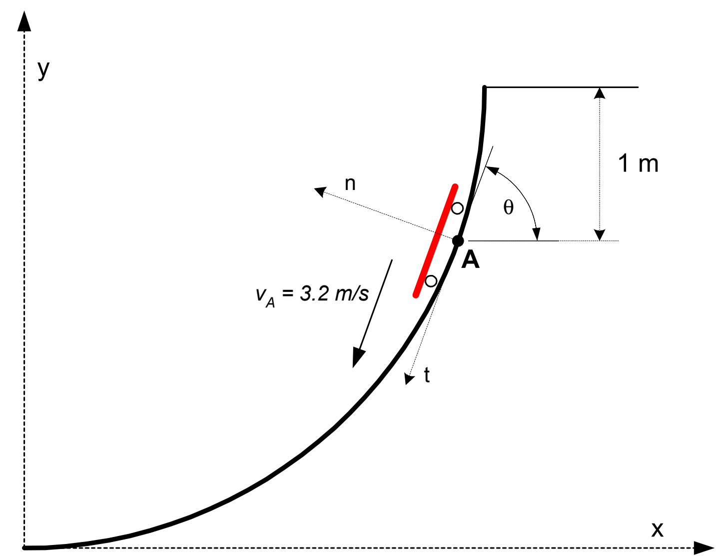
Solution

The equation of the skateboarder's path is that of a quarter circle:

$$y = R - \sqrt{R^2 - x^2}$$

where R is the radius, 3 m.

First, we need to determine the actual location of the skateboarder when she has dropped 1 m.



She is obviously R – 1 = 2 m off the ground, and her x-position can be determined from the path equation using either MATLAB's iterative root finder (fzero), or symbolic math capability. Here, we first create a function (an m-file) whose root we wish to find and then call the fzero iterative solver on the newly created function.

```
%Define the function on which to use fzero.

function F = FindMinimum(x)


R=3;  %3 meters radius
Y=2;  %2 meters off the ground

%The equation describing the scateboarders path is that of
%  and quarter circle => x^2 + y^2  = R^2
%  or similarly  => y = R - sqrt(x^2 + y^2)
%
%  Then setting the equation equal to some arbitrary variable
%  and solving for the root
F = R - sqrt(R^2 - x^2) - Y;
```

```
>> X = fzero('FindMinimum',2)   %Guess: 2

X =

   2.8284
```

So, at the instant of interest, the skateboarder is located at X = 2.828 m, Y = 2 m.

Next, the equation of the slope of the path at this point can be determined using MATLAB's symbolic derivative operator as follows:

```
>> syms x y r                    %Define the x, y , and R as symbolic variables
                                 %Find symbolic result to the first derivative
>> y = r - sqrt( r^2 - x^2 );    %Define symbolic equation
>> dydx = diff(y,x)              %Find first derivative

 dydx =

 1/(r^2-x^2)^(1/2)*x
```

Note: lowercase variables are the symbolic variables while uppercase variables are assigned a numeric value.

By assigning the derivative to an inline function and X and R to their respective values, the numeric value of the derivative can be calculated.

```
>> R = 3;

>> X = 2.8284;

>> DYDX = inline('1/(r^2-x^2)^(1/2)*x','x','r')

DYDX =

    Inline function:

    DYDX(x,r) = 1/(r^2-x^2)^(1/2)*x

>> DYDX(X,R)

ans =

   2.8282
```

The angle in degrees at point **A** can then be determined as:

```
>> atan(2.8282) * 180/pi

ans =

  70.5273
```

The skateboarder's acceleration can be written in terms of normal and tangential velocities, as

$$\mathbf{a} = \dot{v}\,\mathbf{u_t} + \frac{v^2}{\rho}\mathbf{u_n}$$

$$= 7.8\,\tfrac{m}{s^2}\,\mathbf{u_t} + \frac{\left(3.2\,\tfrac{m}{s}\right)^2}{\rho}\mathbf{u_n}$$

The radius of curvature, ρ, of the path at the point of interest (X=2.828 m, Y = 2 m) can be calculated as:

$$\rho = \frac{\left[1+\left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\left|\frac{d^2y}{dx^2}\right|}$$

In MATLAB, this equation is evaluated as follows:

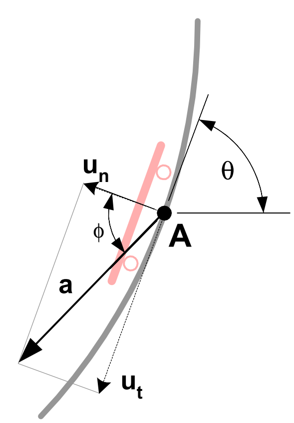
```
>> y = r - sqrt( r^2 - x^2 );                        %Define Symbolic Equation
>> dydx   = diff(y,x)                                %First Derivative
dydx =
1/(r^2-x^2)^(1/2)*x
>> DYDX = inline('1/(R^2-x^2)^(1/2)*x','x','R');     %Create inline function DYDX
>> dy2dx2 = diff(diff(y,x),x)                        %Second Derivative
dy2dx2 =
1/(r^2-x^2)^(3/2)*x^2+1/(r^2-x^2)^(1/2)
                                                     %Create inline function DY2DX2
>> DY2DX2 = inline('1/(r^2-x^2)^(3/2)*x^2+1/(r^2-x^2)^(1/2)','x','r');
>> RHO = (1 + DYDX(X,R)^2)^(3/2) / abs(DY2DX2(X,R))
RHO =
   3.0000
```

The calculated radius of curvature is 3 m – which should come as no surprise since the path is a circle of radius 3 m.

The equation for the acceleration of the skateboarder in terms of tangential and normal components is now:

$$\mathbf{a} = \dot{v}\,\mathbf{u_t} + \frac{v^2}{\rho}\mathbf{u_n}$$

$$= 7.8\,\tfrac{m}{s^2}\,\mathbf{u_t} + \frac{\left(3.2\,\tfrac{m}{s}\right)^2}{3\text{ m}}\mathbf{u_n}$$



The magnitude of the acceleration is found with the following calculation:

>> a = sqrt(7.8^2 + (3.2^2 / 3)^2)   %meters per second squared

a =

8.5142

While the angle (in degrees), ϕ, is found using the atan() function and multiplying by the conversion factor $180^{o}/\pi$.

>> PHI = atan(v_dot / (v^2/rho)) *180/pi

PHI =

66.3655

The angle of the acceleration in degrees from the positive x axis is:

>> ANGLE = THETA + 90 + PHI

ANGLE =

226.8941

Annotated MATLAB Script Solution

```
%Define the function on which to use fzero.
function F = FindMinimum(x)

R=3;  %3 meters radius
Y=2;  %2 meters off the ground

%The equation describing the scateboarders path is that of
%  and quarter circle => x^2 + y^2  = R^2
%  or similarly  => y = R - sqrt(x^2 + y^2)
%
%  Then setting the equation equal to some arbitrary variable
%  and solving for the root
F = R - sqrt(R^2 - x^2) - Y;
```

```
%Call fzero on the FindMinimum function with an initial guess of 2
X = fzero('FindMinimum',2);                        %Guess: 2
fprintf('X = %1.4f m .\n', X)

%Define all other variables using uppercase letters to indicate the
variable has been assigned a value.
V_dot = 7.8;
V = 3.2;
R = 3;
Y = 2;

%Define the x, y , and r as symbolic variables.
%Note:  Lowercase variables are all symbolic variables while
%       uppercase variables are assigned a numeric value.
syms x y r;

%Find symbolic result to first and second derivatives
y = r - sqrt( r^2 - x^2 );                         %Define Symbolic Equation
dydx   = diff(y,x)                                 %First Derivative
```

```
DYDX = inline('1/(R^2-x^2)^(1/2)*x','x','R');
dy2dx2 = diff(diff(y,x),x)                    %Second Derivative
DY2DX2 = inline('1/(r^2-x^2)^(3/2)*x^2+1/(r^2-x^2)^(1/2)','x','r');

%Calculate the angle at point A.
THETA = atan(X) *180/pi;
fprintf('The angle at point a is %1.4f degrees.\n', THETA)

%Calculate the radius of curvature at X = 2.828 m and y = 2 m.
RHO = (1 + DYDX(X,R)^2)^(3/2) / abs(DY2DX2(X,R));
fprintf('The radius of curvature at X = 2.828 m and y = 2 m is %1.4f
m.\n', RHO)

%Calculate the magnitude of acceleration.
A = sqrt(V_dot^2 + (V^2/RHO)^2);
fprintf('The magnitude of acceleration is %1.4f m/s^2.\n', A)

%Calculate the algle of acceleration relative to the t,n coordinate
system.
PHI = atan(V_dot / ( V^2/RHO ))  *180/pi;

%Calculate the angle of acceleration relative to the positive x-axis.
ANGLE = THETA + 90 + PHI;
fprintf('The angle of acceleration relative to the positive x-axis is
%1.4f degrees.\n', ANGLE)
```