TrajectoryDerivations

October 18, 2021

1 Trajectory Derivations

This document shows the derivation of linear velocity v(t) and angular velocity w(t). We will also derive the derivative of xd(t) and yd(t) as well.

The trajectories are defined as:

xd(t) = (W/2)sin([2pi*t]/T)
yd(t) = (H/2)sin([4pi*t]/T)

where W is the width of the figure-eight trajectory, H is the height of the figure-eight tracjectory, and T is the time(in seconds) to finish one figure-eight

The velocities can be expressed as:

(d/dt)xd(t) = v*cos(theta)
(d/dt)yd(t) = v*sin(theta)
(d/dt)theta(t) = w;

1.1 Derive

We can derive by solving for and then taking the time dervative

[35]:

```
import sympy as sym
from sympy.abc import t

print("theta can be expressed as y_dot/x_dot")
theta = sym.symbols(r'theta')
theta_dot = sym.symbols(r'omega')
y_dot_sym = sym.symbols(r'\dot{y}')
x_dot_sym = sym.symbols(r'\dot{x}')
yx_eqn_symbol = sym.Eq(y_dot_sym/x_dot_sym, (sym.sin(theta))/(sym.cos(theta)))
display(yx_eqn_symbol)

print("Solving for theta")
theta_eqn = sym.Eq(theta, sym.atan(y_dot_sym/x_dot_sym))
display(theta_eqn)

print("Find theta_dot, use chain rule for derivative")
theta_exp = sym.atan(y_dot_sym/x_dot_sym)
y_ddot_sym = sym.symbols(r'\ddot{y}')
```

```
x_ddot_sym = sym.symbols(r'\ddot{x}')
theta_exp_dy = theta_exp.diff(y_dot_sym)
theta_exp_dx = theta_exp.diff(x_dot_sym)
theta_exp_full = theta_exp_dy*y_ddot_sym + theta_exp_dx*x_ddot_sym
theta_exp_full = sym.simplify(theta_exp_full)
print("Partial derivative of y")
display(theta_exp_dy*y_ddot_sym)
print("Partial derivative of x")
display(theta_exp_dx*x_ddot_sym)
print("theta_dot a.k.a omega")
display(sym.Eq(theta_dot,theta_exp_full))
```

```
theta can be expressed as y_dot/x_dot
```

$$\frac{\dot{y}}{\dot{x}} = \frac{\sin(\theta)}{\cos(\theta)}$$

```
Solving for theta
```

$$\theta = \operatorname{atan}\left(\frac{\dot{y}}{\dot{x}}\right)$$

```
Find theta_dot, use chain rule for derivative
Partial derivative of y
```

$$\frac{\ddot{y}}{\dot{x}\left(1 + \frac{\dot{y}^2}{\dot{x}^2}\right)}$$

```
Partial derivative of x
```

$$-\frac{\ddot{x}\dot{y}}{\dot{x}^2\left(1 + \frac{\dot{y}^2}{\dot{x}^2}\right)}$$

```
theta_dot a.k.a omega
```

$$\omega = \frac{-\ddot{x}\dot{y} + \ddot{y}\dot{x}}{\dot{x}^2 + \dot{y}^2}$$

1.2 Derive v

We can derive v by taking the x and y component of linear velocity and finding the magnitude.

[33]:
```
import sympy as sym

# Define symbols
x_dot = sym.symbols(r'\dot{x}')
y_dot = sym.symbols(r'\dot{y}')
v = sym.symbols(r'v')
theta = sym.symbols(r'theta')

print(f"Setting up relationship between x_dot, y_dot, and v")
```

```
simple_eqn = sym.Eq(x_dot**2 + y_dot**2, (v**2)*(sym.cos(theta)**2 + sym.
 ↪sin(theta)**2))
display(simple_eqn)
print("Thus...v is")
simple_eqn = sym.Eq(v, sym.sqrt(x_dot**2 + y_dot**2))
display(simple_eqn)
```

Setting up relationship between x_dot, y_dot, and v

$\dot{x}^2 + \dot{y}^2 = v^2 \left(\sin^2(\theta) + \cos^2(\theta)\right)$

Thus…v is

$v = \sqrt{\dot{x}^2 + \dot{y}^2}$

1.3 Derivation of velocities and accelerations

[34]:
```
import sympy as sym
from sympy.abc import t

# Define Constants
W = sym.symbols(r'W')
H = sym.symbols(r'H')
T = sym.symbols(r'T')

# Define Functions
x = (W/2)*sym.sin((2*sym.pi*t)/T)
y = (H/2)*sym.sin((4*sym.pi*t)/T)
print("X Trajectory")
display(x)
print("Y Trajectory")
display(y)

x_dot = x.diff(t)
y_dot = y.diff(t)

print("X_dot Velocity")
display(x_dot)
print("Y_dot Velocity")
display(y_dot)

x_ddot = x_dot.diff(t)
y_ddot = y_dot.diff(t)

print("X_ddot Acceleration")
display(x_ddot)
print("Y_ddot Acceleration")
display(y_ddot)
```

```
X Trajectory
```

$$\frac{W \sin\left(\frac{2\pi t}{T}\right)}{2}$$

```
Y Trajectory
```

$$\frac{H \sin\left(\frac{4\pi t}{T}\right)}{2}$$

```
X_dot Velocity
```

$$\frac{\pi W \cos\left(\frac{2\pi t}{T}\right)}{T}$$

```
Y_dot Velocity
```

$$\frac{2\pi H \cos\left(\frac{4\pi t}{T}\right)}{T}$$

```
X_ddot Acceleration
```

$$-\frac{2\pi^2 W \sin\left(\frac{2\pi t}{T}\right)}{T^2}$$

```
Y_ddot Acceleration
```

$$-\frac{8\pi^2 H \sin\left(\frac{4\pi t}{T}\right)}{T^2}$$