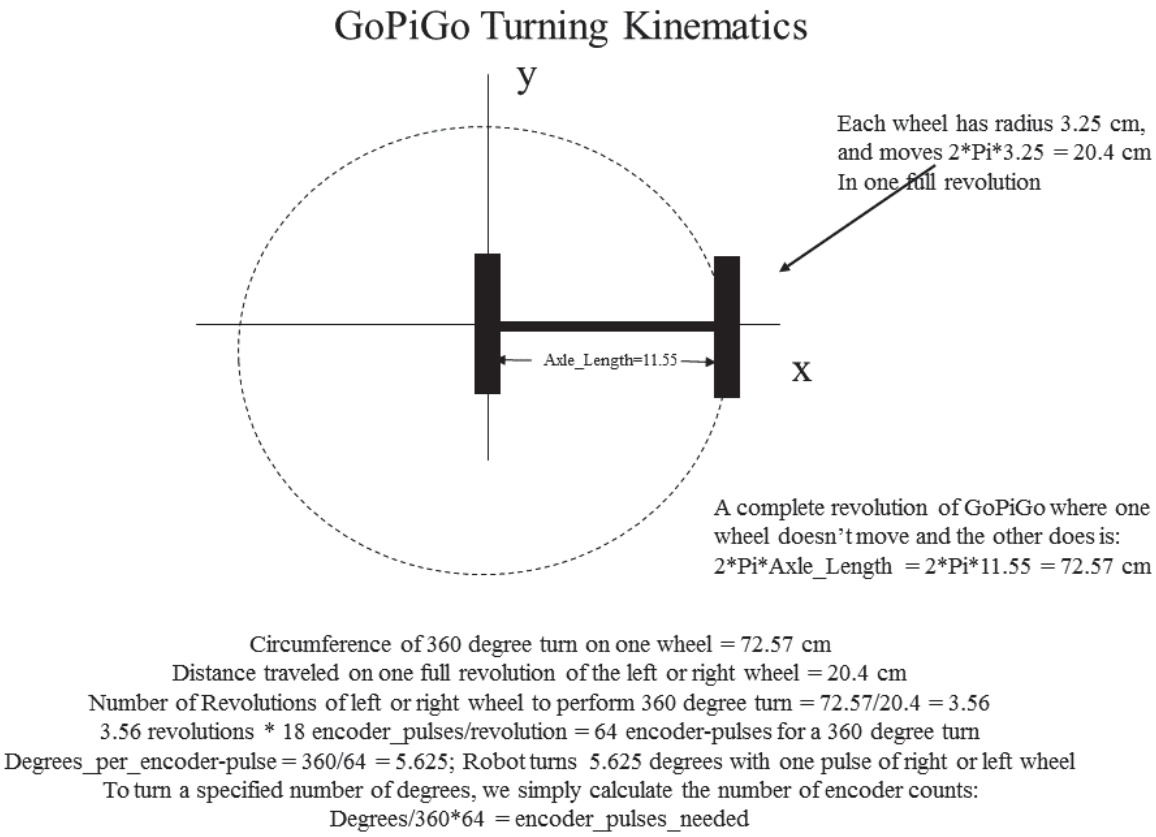

Figure 3: GoPiGo wheel kinematics

4 Understanding GoPiGo Kinematics

The GoPiGo's kinematics use two parameters: the wheel radius R_{wheel} (~ 3.25 cm) and the distance between the two drive wheels, the $AxleLength$ (~ 11.55 cm). Each robot has some difference with these numbers, so you may want to check on them.

The number of encoder counts on each wheel is 18: Measuring 18 pulses of the encoder is one full revolution of the wheel.

To move forward a certain distance $D_{forward}$, we need to find out how many encoder counts map to the distance needed. Given the two parameters above, we note that each revolution of a wheel moves the wheel linearly $2\pi R_{wheel}$. If both wheels are moving at the same rate we get forward motion, and the number of wheel revolutions is is $\frac{D_{forward}}{2\pi R_{wheel}}$, and the number of encoder counts to move this distance is just the number of revolutions of the wheels time 18 pulses per revolution:

$$encoder\_counts = \frac{D_{forward}}{2\pi R_{wheel}} \cdot 18$$

Turning: we can turn left by turning only the right wheel, and turn right by turning only the left wheel. We need to compute how many encoder pulses to use to turn either left or right a certain number of degrees. First, we need to calcuate the ratio **DPR: Degrees Per Pulse** for the wheel encoders. To do this, we note that the GoPiGo wheel radius is 3.25 cm, and the linear distance the wheel travels in one revolution is $2 \cdot 3.25\,\pi \;=\; 20.42$ cm. If we only turn one wheel, then the robot makes a circular motion with the circle's circumference being $2 \cdot AxleLength \cdot \pi \;=\; 72.57$ cm. The number of wheel revolutions for a left or right turn is $72.57/20.4 \;=\; 3.56$, and $3.56 \cdot 18 = 64$ encoder pulses for a full 360 degree turn. Determining the number of encoder pulses for turn less than 360 degrees can be done using just $\frac{Degrees-to-turn}{360} \cdot 64$ encoder pulses on the left wheel (turn clockwise) or right wheel (turn counter-clockwise). The code below is from the Find_Hole programs.

```
from gopigo import *

en_debug=1
## 360 roation is ~64 encoder pulses or 5.625 deg/pulse
## DPR is the Deg:Pulse Ratio or the # of degrees per
##  encoder pulse.
DPR = 360.0/64
WHEEL_RAD = 3.25 # Wheels are ~6.5 cm diameter.

def left_deg(deg=None):
    '''
    Turn chassis left by a specified number of degrees.
    DPR is the #deg/pulse (Deg:Pulse ratio)
    This function sets the encoder to the correct number
     of pulses and then invokes left().
    '''
    if deg is not None:
        pulse= int(deg/DPR)
        enc_tgt(1,0,pulse)
    left()

def right_deg(deg=None):
    '''
    Turn chassis right by a specified number of degrees.
    DPR is the #deg/pulse (Deg:Pulse ratio)
    This function sets the encoder to the correct number
     of pulses and then invokes right().
    '''
    if deg is not None:
        pulse= int(deg/DPR)
        enc_tgt(0,1,pulse)
    right()
```

```
def fwd_cm(dist=None):
    '''
    Move chassis fwd by a specified number of cm.
    This function sets the encoder to the correct number
     of pulses and then invokes fwd().
    '''
    if dist is not None:
        pulse = int(cm2pulse(dist))
        enc_tgt(1,1,pulse)
    fwd()

def bwd_cm(dist=None):
    '''
    Move chassis bwd by a specified number of cm.
    This function sets the encoder to the correct number
     of pulses and then invokes bwd().
    '''
    if dist is not None:
        pulse = int(cm2pulse(dist))
        enc_tgt(1,1,pulse)
    bwd()

def cm2pulse(dist):
    '''
    Calculate the number of pulses to move the chassis dist cm.
    pulses = dist * [pulses/revolution]/[dist/revolution]
    '''
    wheel_circ = 2*math.pi*WHEEL_RAD # [cm/rev] cm traveled per revolution o
    revs = dist/wheel_circ
    PPR = 18 # [p/rev] encoder Pulses Per wheel Revolution
    pulses = PPR*revs # [p] encoder pulses required to move dist cm.
    if en_debug:
        print 'WHEEL_RAD',WHEEL_RAD
        print 'revs',revs
        print 'pulses',pulses
    return pulses
```