Diagram

Description automatically generated

**Python Code**

#Program to create Figure 2.9 from Giordano

from math import cos,sin,exp,sqrt,pi,radians

from matplotlib import pyplot as plt

#lists to store data from calculations

xlist = []

ylist = []

zlist = []

#Constants and initial conditions in mks units

dt = 0.01 #time step

g = 9.8 #gravity

delta = 5

vd = 35

omega = 60\*pi #rate of rotation

Snaughtm = 0.00041 #Snaught over m ratio

x = 0 #inital x position

y = 1 #inital y position

z = 0 #inital z position

vx = 31.3 #initial x velocity

vy = 0 #initial y velocity

vz = 0 #initial z velocity

vtot = sqrt(vx\*\*2 + vy\*\*2 + vz\*\*2) #total velocity

#Function to generate new velocity at each interval

def getNewV (vtot, Vx, Vy, Vz, xyz):

if xyz == 0:

V = Vx

V -= vtot\*V\*(0.0039 + (0.0058/(1 + exp((vtot - vd)/delta))))\*dt

elif xyz == 1:

V = Vy

V -= (g\*dt)

elif xyz == 2:

V = Vz

V += Snaughtm\*omega\*Vx\*dt

return V;

#Euler method to calculate x, y, and z values

for i in range(1000):

xfeet = x\*3.28084

yfeet = y\*3.28084

zfeet = z\*3.28084

xlist.append(xfeet)

ylist.append(yfeet)

zlist.append(zfeet)

x += (vx\*dt)

y += (vy\*dt)

z += (vz\*dt)

vtot = sqrt(vx\*\*2 + vy\*\*2 + vz\*\*2)

vx = getNewV(vtot, vx, vy, vz, 0)

vy = getNewV(vtot, vx, vy, vz, 1)

vz = getNewV(vtot, vx, vy, vz, 2)

if x >= 17:

break

#Creates a plot with both sets of data

%matplotlib

fig = plt.gcf()

fig.set\_size\_inches(5.55, 5)

plt.plot(xlist, ylist, 'k:', lw=1.5)

plt.plot(xlist, zlist, 'k')

plt.plot([0, 0], [-3, 3.5], 'k--', lw=1)

plt.plot([60, 60], [-3, 3.5], 'k--', lw=1)

plt.yticks(ticks = (-4,-2,0,2,4))

plt.xticks(ticks = (0,20,40,60))

plt.title("Sidearm curve ball")

plt.xlabel("x (feet)")

plt.ylabel("y or z (feet)")

plt.annotate('home plate', xy=(59.5, -2.2), xytext=(50,-2.8),

arrowprops=dict(arrowstyle = '->',shrinkA = 3,

connectionstyle = 'arc3,rad=-0.1'),ha='center',va = 'top', fontsize=12)

plt.annotate('horizontal deflection (z)', xy=(28, 0.2), xytext=(23,-0.8),

arrowprops=dict(arrowstyle = '->',shrinkA = 3,

connectionstyle = 'arc3,rad=0.1'),ha='center',va = 'bottom', fontsize=12)

plt.annotate('vertical deflection (y)', xy=(31, 1.8), xytext=(35,2.9),

arrowprops=dict(arrowstyle = '->',shrinkA = 3,

connectionstyle = 'arc3,rad=-0.1'),ha='center',va = 'bottom', fontsize=12)

plt.annotate('pitcher', xy=(0.5,-1.8), xytext=(8,-2.8),

arrowprops=dict(arrowstyle = '->',shrinkA = 3,

connectionstyle = 'arc3,rad=0.1'),ha='center',va = 'top', fontsize=12)

plt.axis([-10,70,-4,4])

plt.tick\_params(direction = 'in', right = True, top = True)

plt.show()