Verkefni_1

February 26, 2019

Verkefni I - Kinematics of the Stewart Platform

Numerical Analysis, 2nd ed., Sauer, Chapter 1

Töluleg Greining STAE405

Háskóli Íslands

Kennari:

- Sigurður Freyr Hafstein

Nemendur:

- Erling Óskar Kristjánsson eok4@hi.is
- Davíð Freyr Björnsson dfb2@hi.is

In [0]:
```python
# -*- coding: utf-8 -*-
import numpy as np
import math
import scipy.optimize as scOpt
import matplotlib.pyplot as plt

"""
Ytri breytur (stikar) sem við skilgreinum sem
global eða viðværar breytur
"""

# global x1, x2, y2, L1, L2, L3, gamma, p1, p2, p3
x1=y2    = 4      # tekid af fig 1.15
x2       = 0      # tekid af fig 1.15
L1       = 2
L2 = L3 = np.sqrt(2)
gamma    = np.pi/2
p1=p2=p3= np.sqrt(5)
```

Suggested Activity 1

Parameters

$$L_1, L_2, L_3, \gamma, x_1, x_2, y_2$$

are fixed constants, strut lengths

$$p_1, p_2, p_3$$

will be known for a given pose.

In [0]:
```python
# Fallið f skilar f(theta, p1, p2, p3, L1, L2, L3, gamma, x1, x2, y2)
# fyrir gefin gildi á theta, p1, p2, p3, L1, L2, L3, gamma, x1, x2 og y2
def f(theta, p1, p2, p3, L1, L2, L3, gamma, x1, x2, y2):
    A2      = L3*np.cos(theta)-x1
    B2      = L3*np.sin(theta)
    A3      = L2*np.cos(theta+gamma)-x2
    B3      = L2*np.sin(theta+gamma)-y2
    aNum    = p2**2-p1**2-A2**2-B2**2
    bNum    = p3**2-p1**2-A3**2-B3**2
    D       = 2*(A2*B3-B2*A3)
    N1      = B3*aNum-B2*bNum
    N2      =-A3*aNum+A2*bNum
    return N1**2+N2**2-p1**2*D**2
```

Látum $\theta_1 = -\pi/4$ og $\theta_2 = \pi/4$.
Nú fæst að $f(\theta_1) =$

In [0]:
```python
print(f(np.pi/4, p1, p2, p3, L1, L2, L3, gamma, x1, x2, y2))
```

```
-4.547473508864641e-13
```

og $f(\theta_2) =$

```
In [0]: print(f(-np.pi/4, p1, p2, p3, L1, L2, L3, gamma, x1, x2, y2))
```

```
-4.547473508864641e-13
```

Eins og sést þá eru þessar tölur mjög nálægt núlli, innan skekkju 10^{-12}, svo allt lítur vel út hingað til.

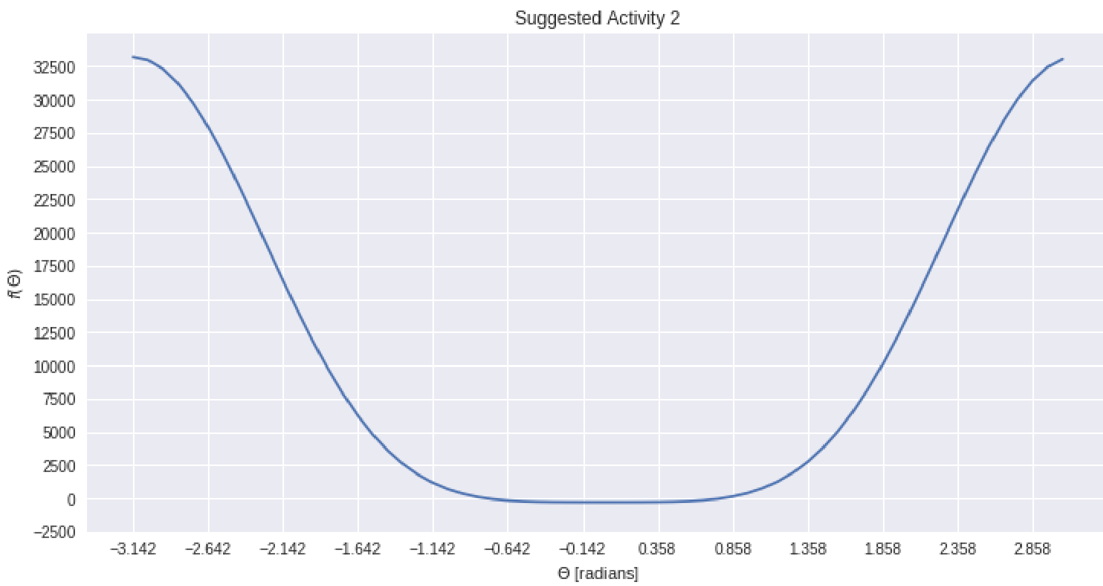
Suggested Activity 2

Plot f(θ) on $[-\pi, \pi]$

```
In [0]: def fPlot(thetaL, thetaH, fileName, graphTitle,
                  minGraphVal, maxGraphVal, interval):
            # Setjum bilið á theta sem við viljum
            # teikna f(theta) fyrir
            theRange = np.arange(thetaL, thetaH, 0.1)
            # Finnum hámarks og lágmarks gildi f á því bili
            maxFunVal = max(f(theRange, p1, p2, p3,
                                  L1, L2, L3, gamma, x1, x2, y2))
            minFunVal = min(f(theRange, p1, p2, p3,
                                  L1, L2, L3, gamma, x1, x2, y2))
            # Upphafsstillum myndina og ásana
            figure, axis = plt.subplots(figsize=(12, 6))
            # Teiknum grafið
            axis.plot(theRange, f(theRange, p1, p2, p3,
                                      L1, L2, L3, gamma, x1, x2, y2))
            # Setjum titil á grafið og ásana
            axis.set(xlabel=r"${\Theta}$ [radians]",
                    ylabel="$f({\Theta})$",
                    title = graphTitle)
            # Setjum hæsta og lægsta gildið á y-ás
            axis.set_ylim([minGraphVal,maxGraphVal])
            axis.grid()
            # Vistum myndina
            fileString = fileName + ".png"
            figure.savefig(fileString)

            # Setjum minnsta sýnilega bilið á x - ásnum
            # og y - ásnum
            plt.xticks(np.arange(thetaL, thetaH, 0.5))
            plt.yticks(np.arange(minGraphVal,maxGraphVal,interval))
            plt.grid(True)
            plt.show()
```

Teiknum $f(\theta)$ fyrir θ frá $-\pi$ upp í π:

```
In [0]: fPlot(-np.pi, np.pi, "sa2", 'Suggested Activity 2', -2500, 35000, 2500)
```



Python fallið fSolver skilar núllstöð f. Þarf upphafságiskun sem byggir á því að skoða graf af $f(\theta)$

```
In [0]: def fSolver(f, thetaGuess):
            func = lambda theta : f(theta, p1, p2, p3,
                                    L1, L2, L3, gamma, x1, x2, y2)
            thetaSol = scOpt.fsolve(func, thetaGuess)
            return thetaSol
```

Skoðum graf af $f(\theta)$ og sjáum þá fyrir hvaða gildi á θ, $f(\theta)$ er nálægt núlli

```
In [0]: thetaCalculated = fSolver(f, np.pi/5)
        thetaGiven = np.pi/4
        print("Núllstöð f(theta) sem fékkst með ítrun er:", thetaCalculated)
        print("Gefin núllstöð f(theta) er:", thetaGiven)

        thetaCalculated = fSolver(f, -np.pi/5)
        thetaGiven = -np.pi/4
        print("Núllstöð f(theta) sem fékkst með ítrun er:", thetaCalculated)
        print("Gefin núllstöð f(theta) er:", thetaGiven)
```

```
Núllstöð f(theta) sem fékkst með ítrun er: [0.78539816]
Gefin núllstöð f(theta) er: 0.7853981633974483
Núllstöð f(theta) sem fékkst með ítrun er: [-0.78539816]
Gefin núllstöð f(theta) er: -0.7853981633974483
```

3

Reproduce Figure 1.15
Látum

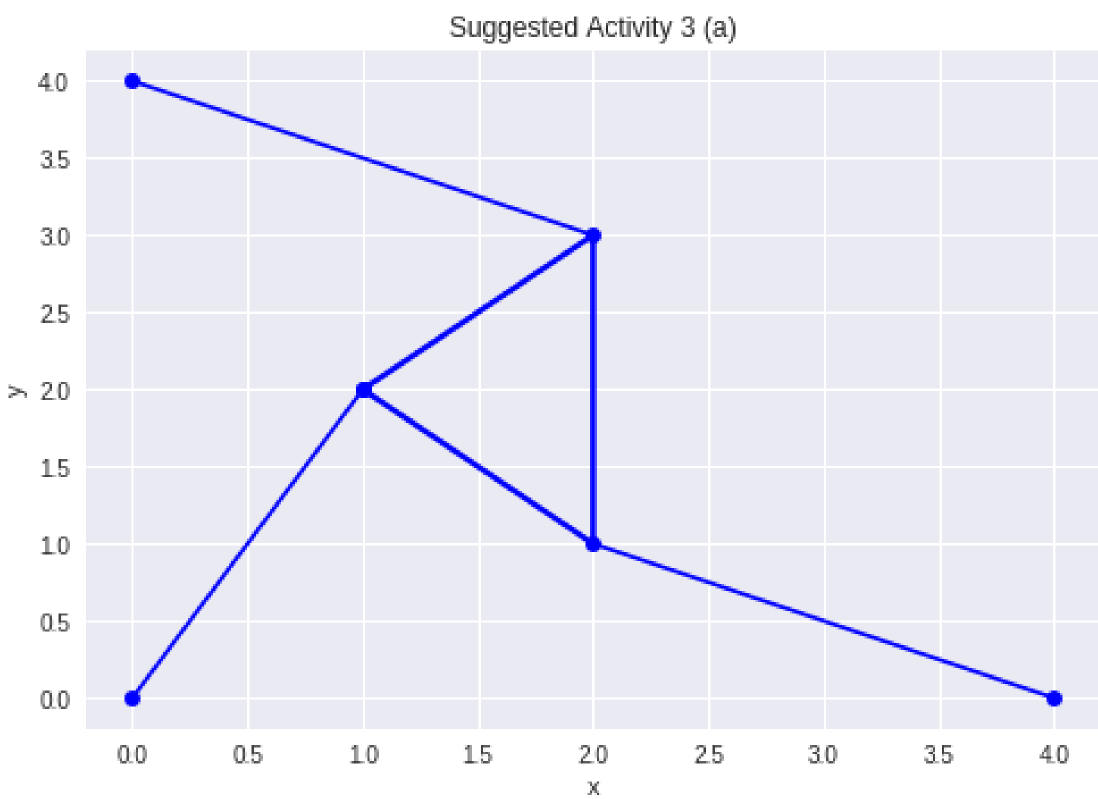
$$
\begin{aligned}
xL_2 &:= x + L_2 \cdot \cos(\theta + \gamma) \\
yL_2 &:= y + L_2 \cdot \sin(\theta + \gamma) \\
xL_3 &:= x + L_3 \cdot \cos(\theta + \gamma) \\
yL_3 &:= y + L_3 \cdot \sin(\theta + \gamma)
\end{aligned}
$$

```
In [0]: # Teiknar Stewart Platform
        def plotStewartPlatform(x, y, x1, x2, y2, xL2, yL2, xL3, yL3,
                                plotTitle="Vantar titil!", plotName="Vantar"):
                # Teiknar þrihyrning. Byrjum lengst til
                # vinstri og forum rettsaelis
                # Her er (x, y) = (1,2),
                #         (xL2, yL2) = (2,3)
                #         (xL3, yL3) = (2,1)
                (triangleX, triangleY)  = [x, xL2, xL3, x],[y, yL2, yL3, y]
                fig, plotObject = plt.subplots()
                plotObject.plot(triangleX, triangleY, 'b', linewidth=2.5)

                # Baetum vid fyrsta akkerinu
                # sem hefur hnit fra (0,0) i (x,y)
                firstAnchorX, firstAnchorY  = [0,x],[0,y]
                plotObject.plot(firstAnchorX, firstAnchorY, 'b')

                # Baetum vid odru akkerinu
                # sem hefur hnit fra (x2, y2)
                # til (xL2, yL2)
                secondAnchorX, secondAnchorY  = [x2,xL2],[y2,yL2]
                plotObject.plot(secondAnchorX, secondAnchorY, 'b')

                # Baetum vid thridja akkerinu
                # sem hefur hnit fra (xL3, yL3)
                # til (x1, 0)
                thirdAnchorX, thirdAnchorY  = [x1,xL3],[0,yL3]
                plotObject.plot(thirdAnchorX, thirdAnchorY, 'b')

                # Baetum vid punktum fyrir
                # akkaeri eitt, tvo og thrju
                # sem hafa hnitin
                # (0,0), (x2, y2) og (x1, 0)
                anchorPointsX, anchorPointsY  = [0, x1, x2], [0, 0, y2]
                plotObject.plot(anchorPointsX, anchorPointsY, 'bo')

                # Baetum vid punktum fyrir
                # hvert horn thrihyrningsins
```

```
        plotObject.plot(triangleX, triangleY, 'bo')

        plotObject.set(xlabel="x",
                ylabel="y",
                title = plotTitle)

        # Vistum myndina
        plotSaveFile = plotName + ".png"
        fig.savefig(plotSaveFile)
        plt.show()
```

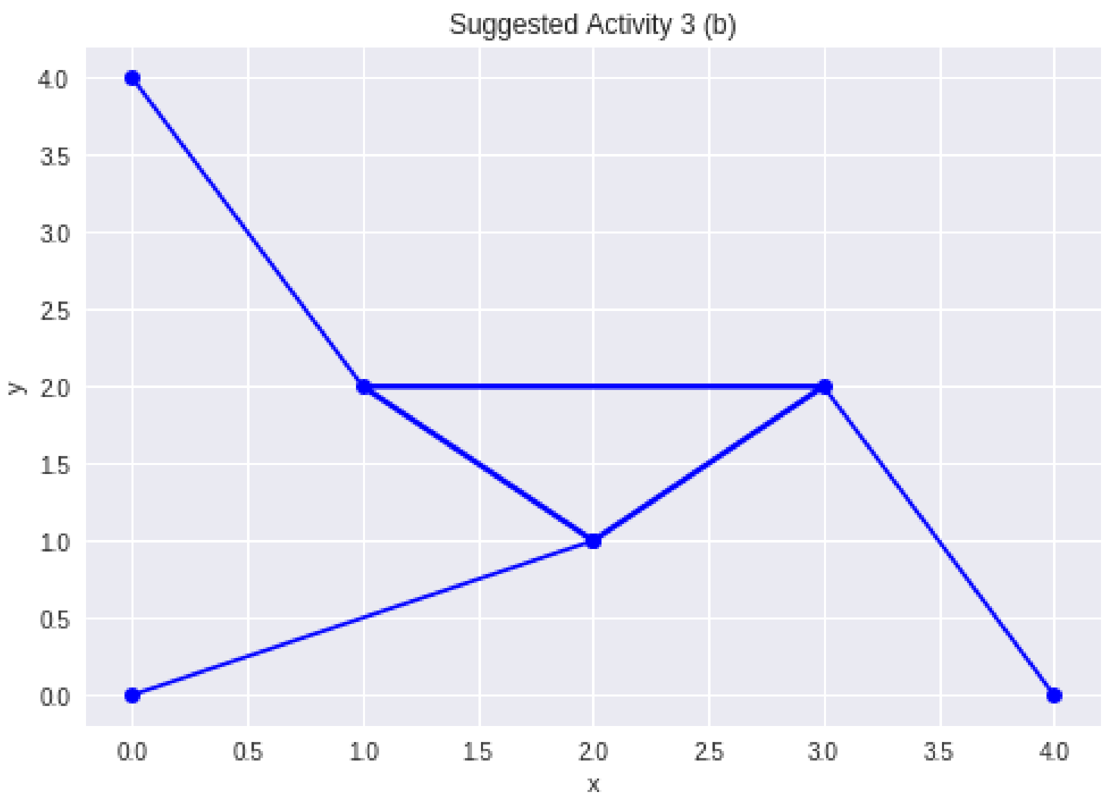
Plot SA3(a):

```
In [0]: x, y = 1, 2
        xL2, yL2 = 2, 3
        xL3, yL3 = 2, 1
        plotName = "sa3a"
        plotTitle = 'Suggested Activity 3 (a)'
        plotStewartPlatform(x, y, x1, x2, y2,
                            xL2, yL2, xL3, yL3, plotTitle, plotName)
```



Plot SA3(b):

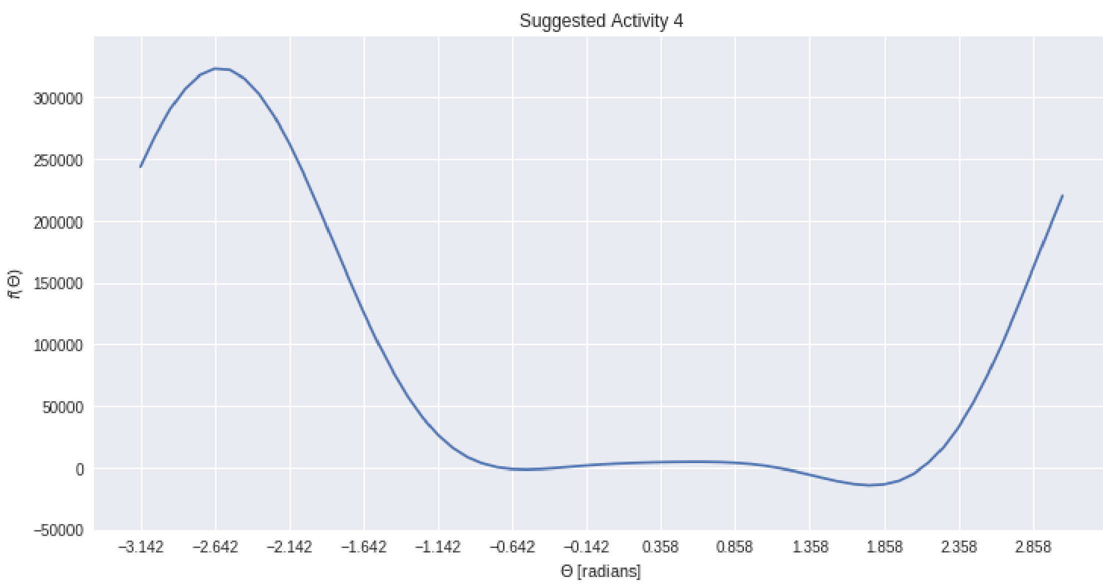
```
In [0]: ''' Plot SA3(b) '''
        x, y = 2, 1
        xL2, yL2 = 1, 2
        xL3, yL3 = 3, 2
        plotName = "sa3b"
        plotTitle = 'Suggested Activity 3 (b)'
        plotStewartPlatform(x, y, x1, x2, y2,
                            xL2, yL2, xL3, yL3, plotTitle, plotName)
```



Suggested Activity 4

Solve the forward kinematics problem for the planar Stewart platform. Plot $f(\theta)$ and then solve $f(\theta) = 0$.

```
In [0]: # Stikar
        x1 = 5
        x2, y2 = 0, 6
        L1 = L3 = 3
        L2 = 3*np.sqrt(2)
        gamma = np.pi/4
        p1 = p2 = 5
        p3 = 3

        # Teiknum f(theta) fyrir theta frá -pi upp í pi
        fPlot(-np.pi, np.pi, "sa4-0", 'Suggested Activity 4', -50000, 350000, 50000)
```



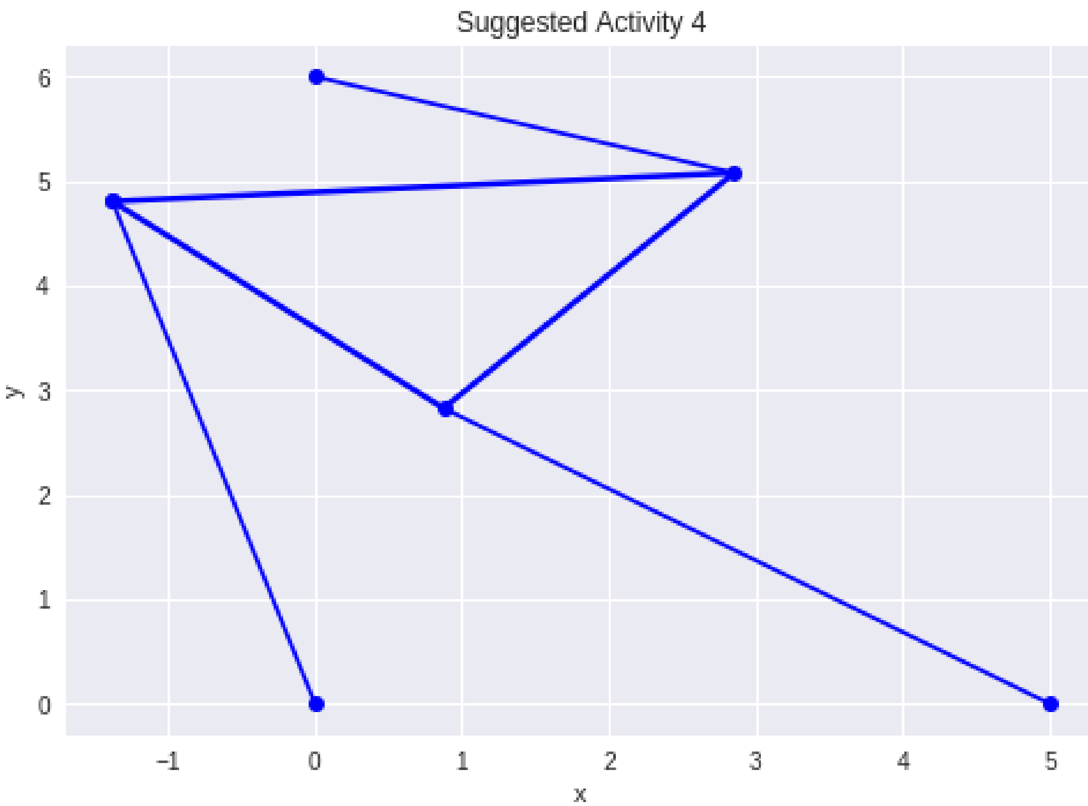
Útfrá mynd sést að ágætis upphafsgisk fyrir þessar fjórar núllstöðvar f(theta) eru -0.7, -0.3, 1.1, 2.1. Finnum núna fjórar núllstöðvar $f(\theta)$

```
In [0]: # Geymum núllstöðvarnar og ágiskanirnar í fylki
        thetas = np.zeros(4, dtype=float)
        thetasGuess = [-0.7, -0.3, 1.1, 2.1]

        # Hér fáum við út réttar núllstöðvar
        print("Ef theta er raunveruleg núllstöð þá ætti")
        print("f(theta) að vera mjög nálægt núlli \n")
        for i in range(len(thetas)):
                thetas[i] = fSolver(f, thetasGuess[i])
```

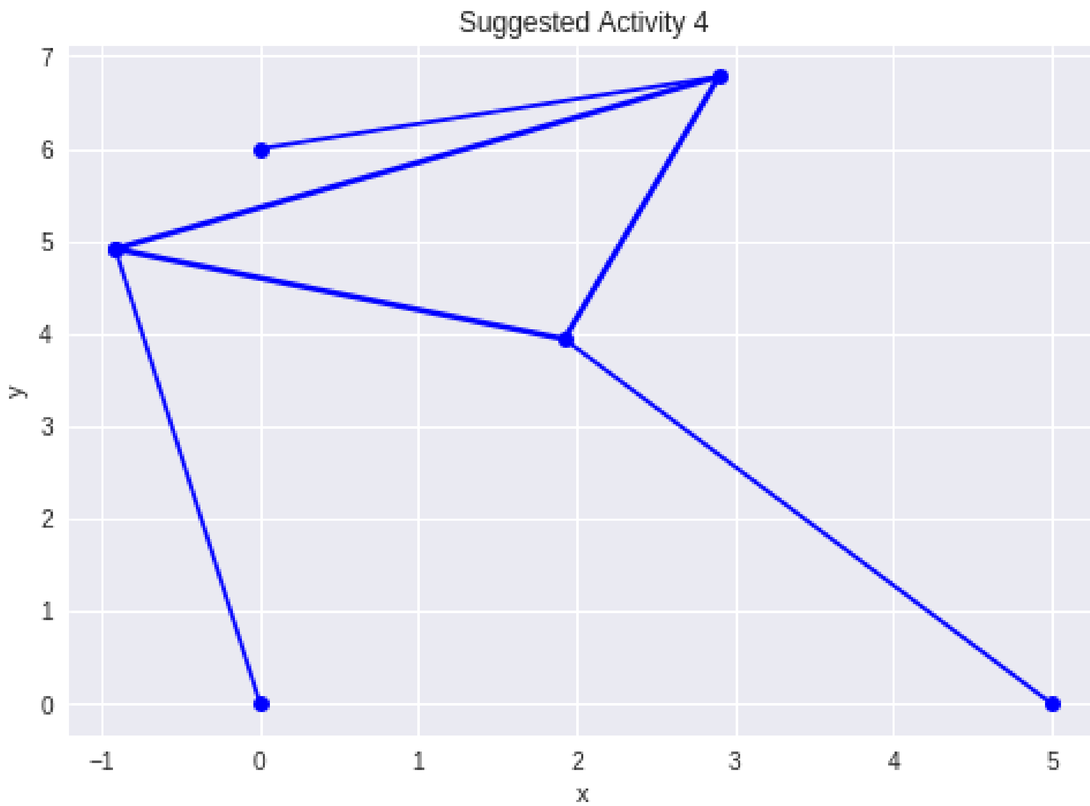
```python
        print("Nú fæst að theta er:", thetas[i])
        print("Svo er f(theta):", f(thetas[i], p1, p2, p3, L1, L2, L3,
                gamma, x1, x2, y2), "\n")

# Geymir x og y gildin fyrir útreiknaðar núllstöðvar f(theta)
xy = np.zeros([4,2], dtype=float)

# Fallið xyCalc reiknar út gildin á x og y miðað við gefin
# gildi á theta, p1, p2, p3, L1, L2, L3, gamma, x1, x2 og y2.
def xyCalc(theta, p1, p2, p3, L1, L2, L3, gamma, x1, x2, y2):
    A2      = L3*np.cos(theta)-x1
    B2      = L3*np.sin(theta)
    A3      = L2*np.cos(theta+gamma)-x2
    B3      = L2*np.sin(theta+gamma)-y2
    aNum    = p2**2-p1**2-A2**2-B2**2
    bNum    = p3**2-p1**2-A3**2-B3**2
    D       = 2*(A2*B3-B2*A3)
    N1      = B3*aNum-B2*bNum
    N2      =-A3*aNum+A2*bNum
    x = N1/D
    y = N2/D
    return [x, y]

for i in range(0, 4):
        xy[i] = xyCalc(thetas[i], p1, p2, p3, L1, L2, L3,
        gamma, x1, x2, y2)

# Útfrá útreiknuðum theta, x og y gildum getum við síðan reiknað xL2, yL2, xL3, yL3
xyL23 = np.zeros([4,4], dtype=float)

def xyLCalc(theta, x, y, p1, p2, p3, L1, L2, L3, gamma, x1, x2, y2):
        xL2 = x + L2*np.cos(theta + gamma)
        yL2 = y + L2*np.sin(theta + gamma)
        xL3 = x + L3*np.cos(theta)
        yL3 = y + L3*np.sin(theta)
        return [xL2, yL2, xL3, yL3]

for i in range(0, 4):
        xyL23[i] = xyLCalc(thetas[i], xy[i,0], xy[i,1], p1, p2, p3, L1, L2, L3,
                gamma, x1, x2, y2)

# Teiknum Stewart platform fyrir þessar fjórar núllstöðvar f(theta)
# og könnum í leiðinni hvort lengdirnar á struts séu p1, p2 og p3
for i in range(0, 4):
        p1Calc = math.hypot(xy[i,0] - 0, xy[i,1] - 0)
        p2Calc = math.hypot(xyL23[i,2] - x1, xyL23[i,3] - 0)
        p3Calc = math.hypot(xyL23[i,0] - x2, xyL23[i,1] - y2)
        print("Fyrir núllstöðina", thetas[i], "þá fást eftirfarandi niðurstöður")
```

```
                print("Nú er p1 = 5 en útreiknað gildi er:", p1Calc)
                print("Nú er p2 = 5 en útreiknað gildi er:", p2Calc)
                print("Nú er p3 = 3 en útreiknað gildi er:", p3Calc)
                print("\n")
                pName = "sa4-" + str(i+1)
                plotStewartPlatform(xy[i,0], xy[i,1], x1, x2, y2, xyL23[i, 0],
                                    xyL23[i, 1], xyL23[i, 2], xyL23[i, 3], "SA 4", pName)
```

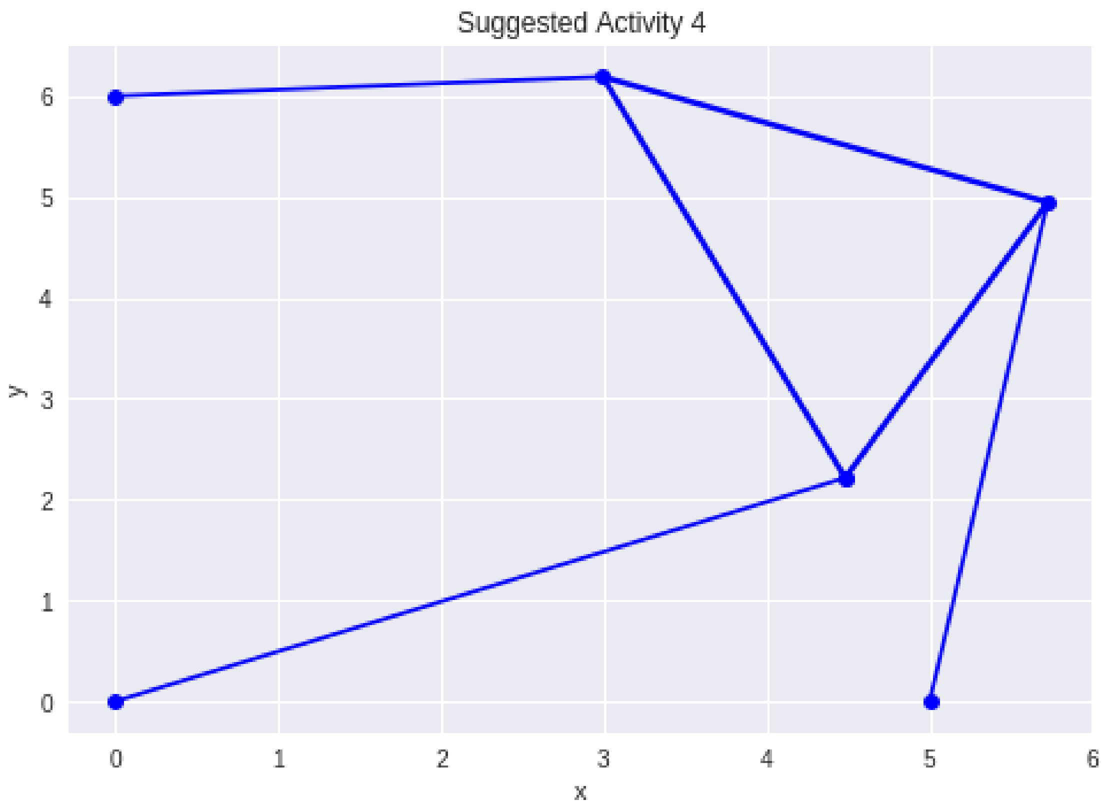
```
Ef theta er raunveruleg núllstöð þá ætti
f(theta) að vera mjög nálægt núlli

Nú fæst að theta er: -0.7208492044603826
Svo er f(theta): -1.3096723705530167e-10

Nú fæst að theta er: -0.33100518428386466
Svo er f(theta): 4.9112713895738125e-11

Nú fæst að theta er: 1.1436855178213738
Svo er f(theta): 5.4569682106375694e-12

Nú fæst að theta er: 2.115909014086863
Svo er f(theta): 3.710738383233547e-08

Fyrir núllstöðina -0.7208492044603826 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 4.999999999999995
Nú er p2 = 5 en útreiknað gildi er: 4.999999999999995
Nú er p3 = 3 en útreiknað gildi er: 2.999999999999991
```



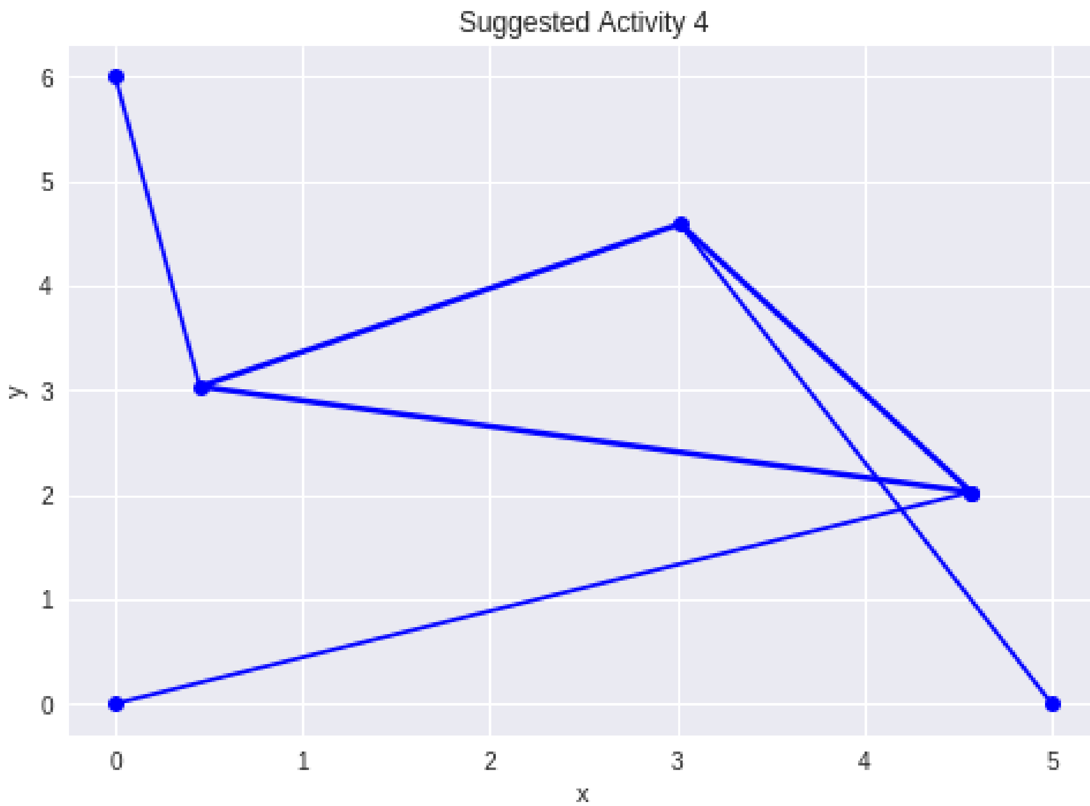

```
Fyrir núllstöðina -0.33100518428386466 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 5.000000000000008
Nú er p2 = 5 en útreiknað gildi er: 5.000000000000008
Nú er p3 = 3 en útreiknað gildi er: 3.0000000000000124
```




```
Fyrir núllstöðina 1.1436855178213738 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 5.000000000000001
Nú er p2 = 5 en útreiknað gildi er: 5.0
Nú er p3 = 3 en útreiknað gildi er: 3.0000000000000004
```




```
Fyrir núllstöðina 2.115909014086863 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 5.000000000000496
Nú er p2 = 5 en útreiknað gildi er: 5.000000000000496
Nú er p3 = 3 en útreiknað gildi er: 3.000000000000825
```

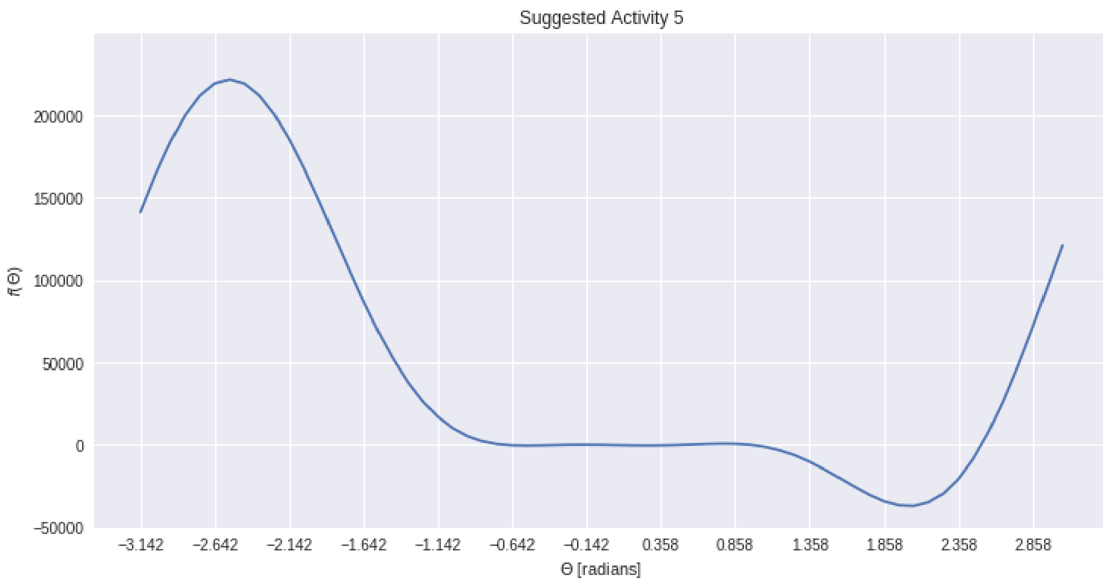

Suggested Activity 4
y
x
0
1
2
3
4
5
6
0
1
2
3
4
5

Suggested Activity 5

Change strut length to $p_2 = 7$ and re-solve the problem. For these parameters, there are six poses.

In [0]:
```
# Change the value of p2
p2 = 7

# Teiknum f(theta) fyrir theta frá -pi upp í pi
fPlot(-np.pi, np.pi, "sa5-0", 'Suggested Activity 5', -50000, 250000, 50000)

# Finnum núna sex núllstöðvar f(theta)
# Hér er ferillinn mjög flatur í kringum fimm
# af sex núllstöðvunum
thetas2 = np.zeros(6, dtype=float)

# Útfrá mynd sést að ágætis upphafsgisk fyrir
# þessar sex núllstöðvar f(theta) eru
# -0.7, -0.4, 0, 0.42, 1, 2.5
thetasGuess2 = [-0.7, -0.4, 0, 0.42, 1, 2.5]

# Hér fáum við út réttar núllstöðvar
print("Ef theta er raunveruleg núllstöð þá ætti")
print("f(theta) að vera mjög nálægt núlli \n")
for i in range(0,len(thetas2)):
        thetas2[i] = fSolver(f, thetasGuess2[i])
        print("Nú fæst að theta er:", thetas2[i])
        print("Svo er f(theta):", f(thetas2[i], p1, p2, p3,
                                   L1, L2, L3,
                                           gamma, x1, x2, y2), "\n")

# Geymir x og y gildin fyrir útreiknaðar núllstöðvar f(theta)
xy2 = np.zeros([6,2], dtype=float)

for i in range(0, 6):
        xy2[i] = xyCalc(thetas2[i], p1, p2, p3,
                        L1, L2, L3, gamma, x1, x2, y2)

# Útfrá útreiknuðum theta, x og y gildum getum við síðan reiknað
# xL2, yL2, xL3, yL3
xy2L23 = np.zeros([6,4], dtype=float)

for i in range(0, 6):
        xy2L23[i] = xyLCalc(thetas2[i], xy2[i,0], xy2[i,1], p1, p2, p3,
                            L1, L2, L3, gamma, x1, x2, y2)

# Teiknum Stewart platform fyrir þessar sex núllstöðvar f(theta)
# og könnum í leiðinni hvort lengdirnar á struts séu p1, p2 og p3
for i in range(0, 6):
        p1Calc2 = math.hypot(xy2[i,0] - 0, xy2[i,1] - 0)
```

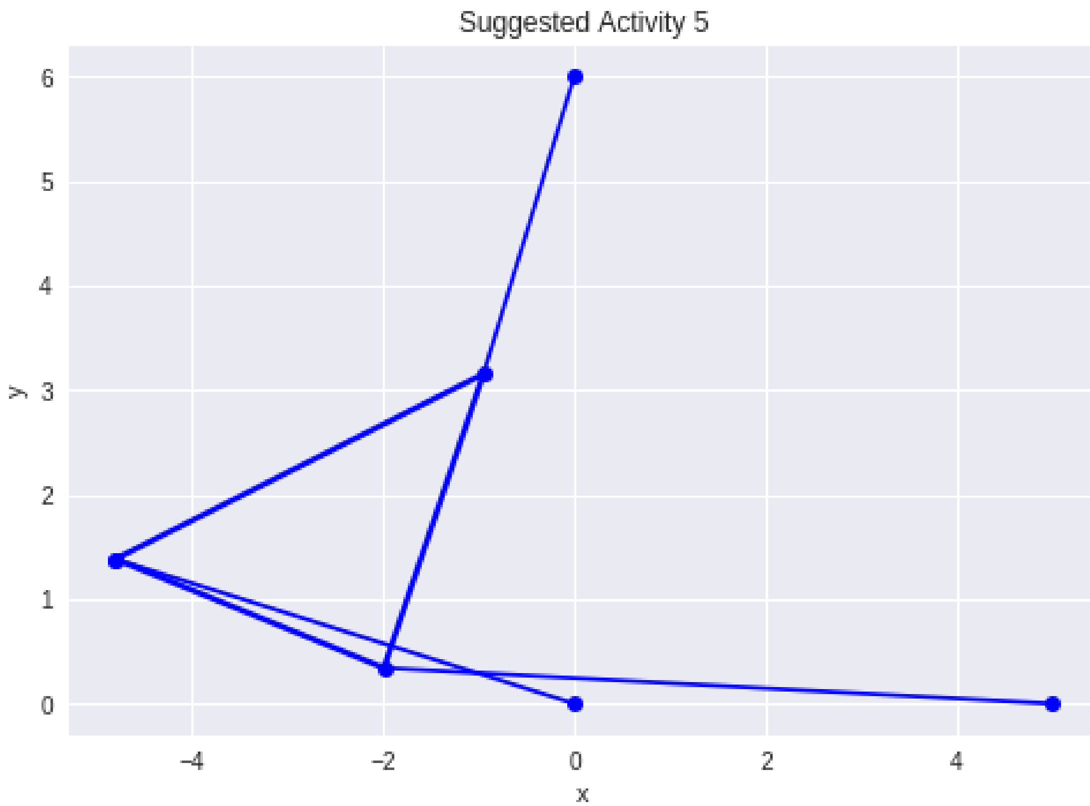
```
p2Calc2 = math.hypot(xy2L23[i,2] - x1, xy2L23[i,3] - 0)
p3Calc2 = math.hypot(xy2L23[i,0] - x2, xy2L23[i,1] - y2)
print("Fyrir núllstöðina", thetas2[i],
      "þá fást eftirfarandi niðurstöður")
print("Nú er p1 = 5 en útreiknað gildi er:", p1Calc2)
print("Nú er p2 = 7 en útreiknað gildi er:", p2Calc2)
print("Nú er p3 = 3 en útreiknað gildi er:", p3Calc2)
print("\n")
pName = "sa5-"+str(i+1)
plotStewartPlatform(xy2[i,0], xy2[i,1], x1, x2, y2, xy2L23[i, 0],
                    xy2L23[i, 1], xy2L23[i, 2], xy2L23[i, 3],
                    "Suggested Activity 5", pName)
```



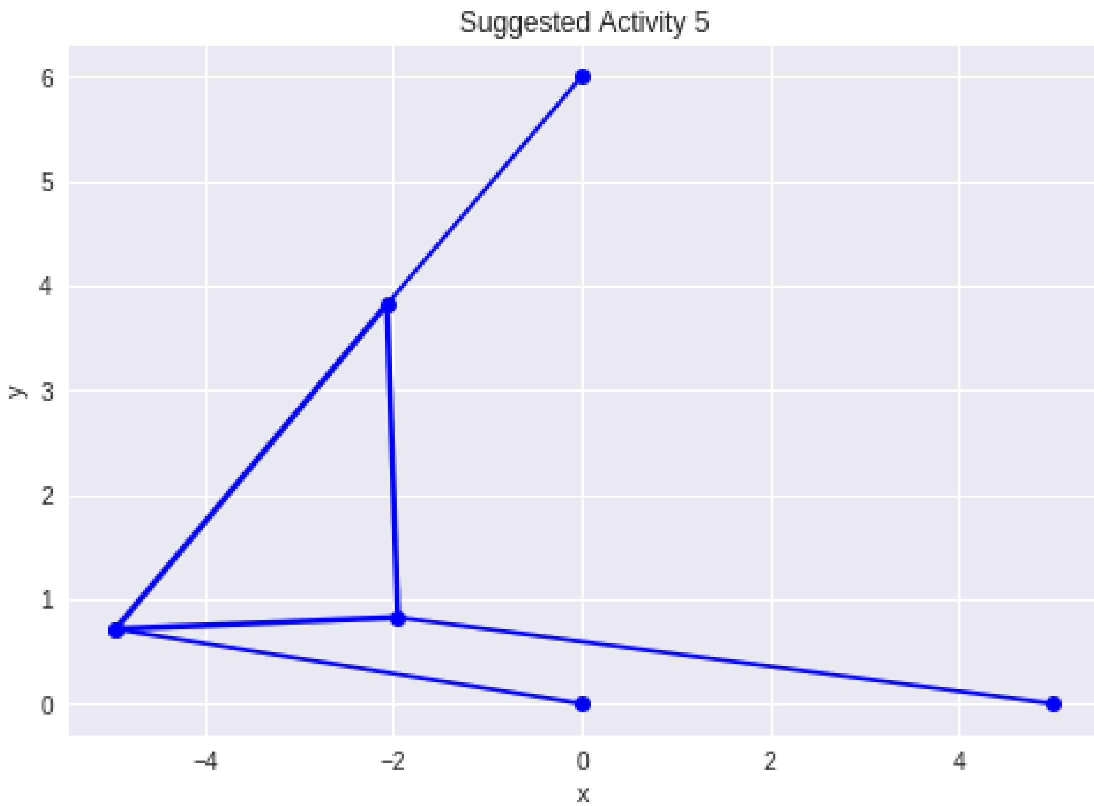

```
Ef theta er raunveruleg núllstöð þá ætti
f(theta) að vera mjög nálægt núlli

Nú fæst að theta er: -0.673157486371671
Svo er f(theta): -1.4551915228366852e-11

Nú fæst að theta er: -0.3547402704156737
Svo er f(theta): 3.637978807091713e-12

Nú fæst að theta er: 0.03776676057591458
Svo er f(theta): -1.8189894035458565e-12

Nú fæst að theta er: 0.45887818104896444
Svo er f(theta): -6.923528417246416e-11
```

```
Nú fæst að theta er: 0.9776728950003634
Svo er f(theta): -1.000444171950221e-11

Nú fæst að theta er: 2.513852799350387
Svo er f(theta): 4.0745362639427185e-10

Fyrir núllstöðina -0.673157486371671 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 4.999999999999998
Nú er p2 = 7 en útreiknað gildi er: 6.999999999999999
Nú er p3 = 3 en útreiknað gildi er: 2.9999999999999987
```

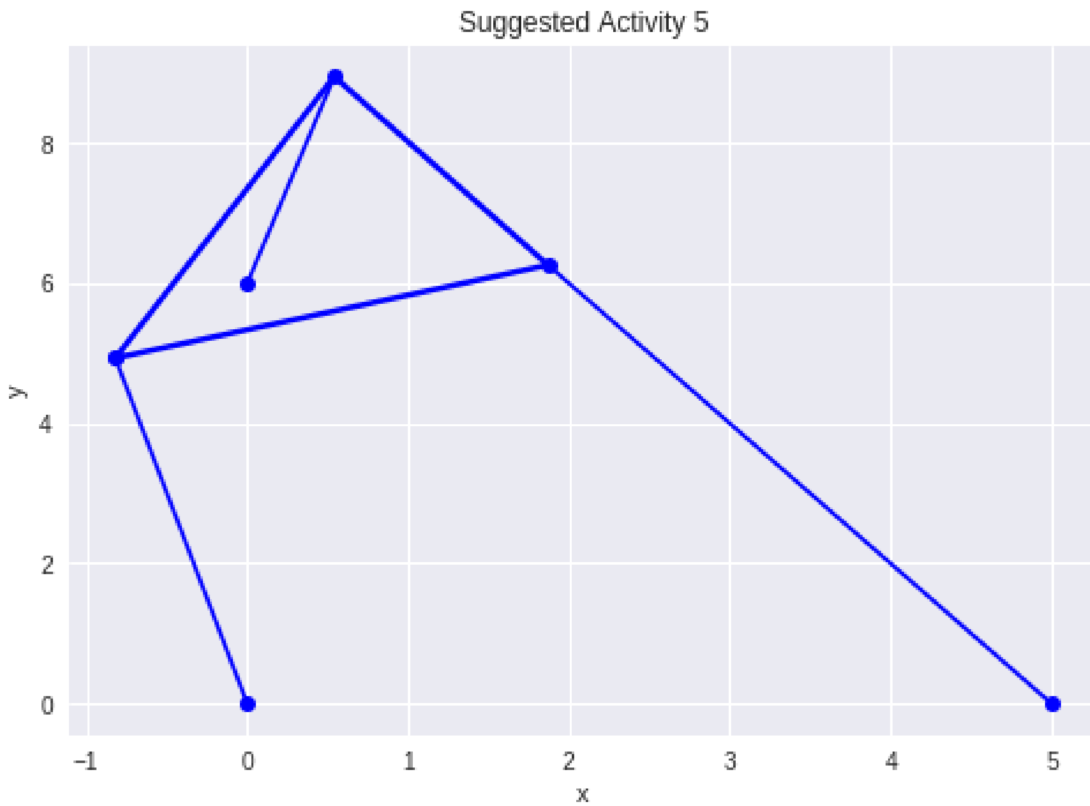

```
Fyrir núllstöðina -0.3547402704156737 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 5.0
Nú er p2 = 7 en útreiknað gildi er: 7.0
Nú er p3 = 3 en útreiknað gildi er: 3.0
```



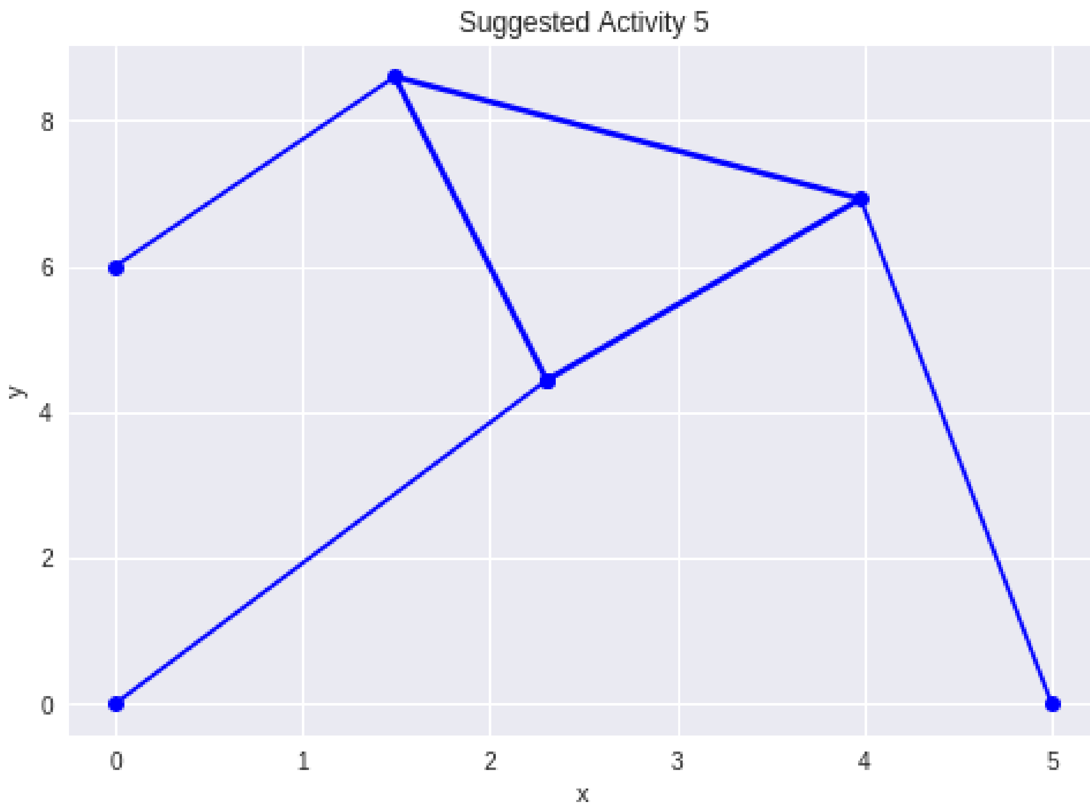

```
Fyrir núllstöðina 0.03776676057591458 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 4.999999999999998
Nú er p2 = 7 en útreiknað gildi er: 6.999999999999998
Nú er p3 = 3 en útreiknað gildi er: 2.999999999999997
```



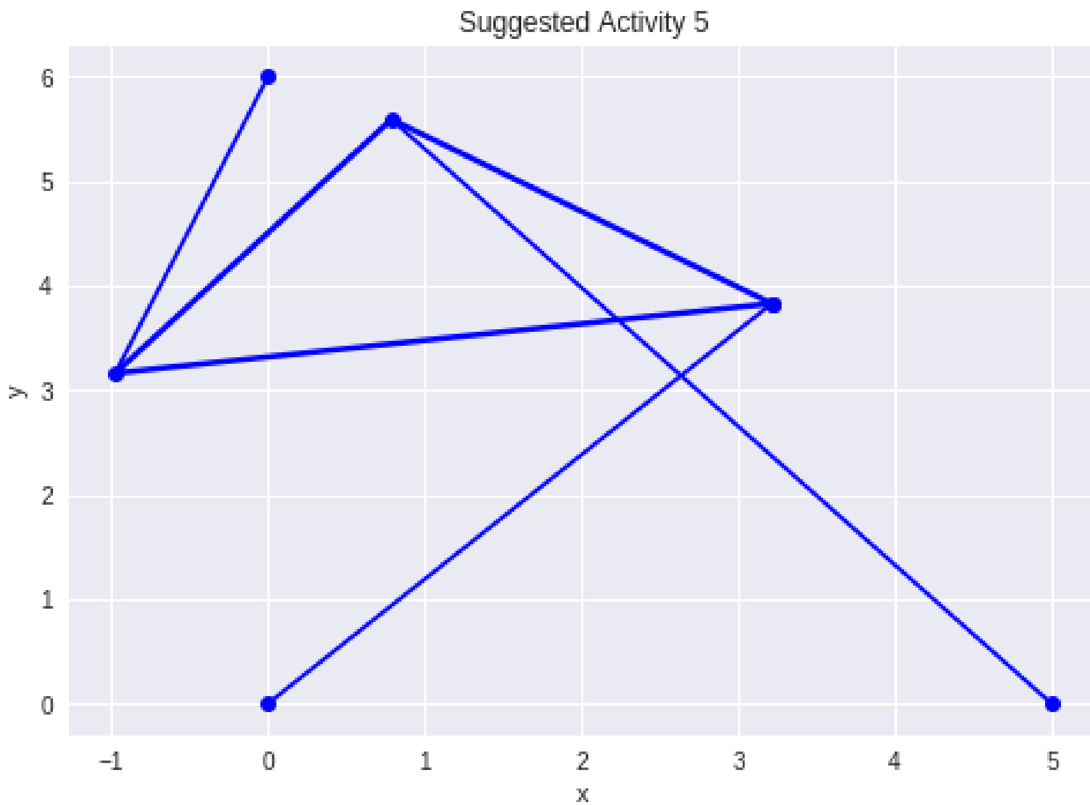

```
Fyrir núllstöðina 0.45887818104896444 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 4.999999999999774
Nú er p2 = 7 en útreiknað gildi er: 6.999999999999839
Nú er p3 = 3 en útreiknað gildi er: 2.9999999999996243
```




```
Fyrir núllstöðina 0.9776728950003634 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 4.9999999999999964
Nú er p2 = 7 en útreiknað gildi er: 6.999999999999997
Nú er p3 = 3 en útreiknað gildi er: 2.9999999999999947
```




```
Fyrir núllstöðina 2.513852799350387 þá fást eftirfarandi niðurstöður
Nú er p1 = 5 en útreiknað gildi er: 5.000000000000003
Nú er p2 = 7 en útreiknað gildi er: 7.000000000000002
Nú er p3 = 3 en útreiknað gildi er: 3.0000000000000053
```



Suggested Activity 5
y
x
6
5
4
3
2
1
0
−1
0
1
2
3
4
5

Suggested Activity 6

Find a strut length p_2, with the rest of the parameters as in Step 4, for which there are only two poses.

```
In [0]: # Stikar úr Suggested Activity 4
        x1 = 5
        x2, y2 = 0, 6
        L1 = L3 = 3
        L2 = 3*np.sqrt(2)
        gamma = np.pi/4
        p1 = 5
        p3 = 3

        # Fall sem telur fjölda róta fallsins f(theta)
        # fyrir gefna stika
        def countRoots(f, p2):
                deltaTheta = 0.001
                thetaOld = -np.pi
                oldSign = f(thetaOld, p1, p2, p3,
                            L1, L2, L3, gamma, x1, x2, y2) > 0
                count = 0
                for theta in np.arange(-np.pi, np.pi+deltaTheta, deltaTheta):
                        thetaNew = theta
                        newSign = f(thetaNew, p1, p2, p3,
                                    L1, L2, L3, gamma, x1, x2, y2) > 0
                        if(newSign != oldSign):
                                count += 1
                        oldSign = newSign
                return count
```

Höfum séð að það eru 4 rætur og þar með 4 stöður þegar $p_2 = 5$ Höfum séð að það eru 6 rætur og þar með 6 stöður þegar $p_2 = 7$ Skoðum gildi þar í kring. Vitum að það eru (líklegast) formerkjaskipti í rótum. Einnig þar sem p_2 er mælikvarði á lengd vitum við að $p > 0$. Því leitum við að formerkjaskiptum á bilinu [$0,12]$

```
In [0]: # Prófum mismunandi gildi á p2
        def p2Roots(p2Min, p2Max, interval, noRoots):
          p2Array = np.arange(p2Min, p2Max+interval, interval)
          p2Roots = list()
          for i in p2Array:
                  if(countRoots(f, i) == noRoots):
                          p2Roots.append(i)

          for i in range(len(p2Roots)):
            print(p2Roots[i])

        p2Roots(0, 12, 0.1, 2)
```

```
3.8000000000000003
3.9000000000000004
4.0
4.1000000000000005
4.2
4.3
4.4
4.5
4.6000000000000005
4.7
4.800000000000001
7.9
8.0
8.1
8.200000000000001
8.3
8.4
8.5
8.6
8.700000000000001
8.8
8.9
9.0
9.1
9.200000000000001
```

Við sjáum að fyrir $p_2 \in [3.8, 4.8] \cup [7.9, 9.2]$ þá fást 2 stöður.

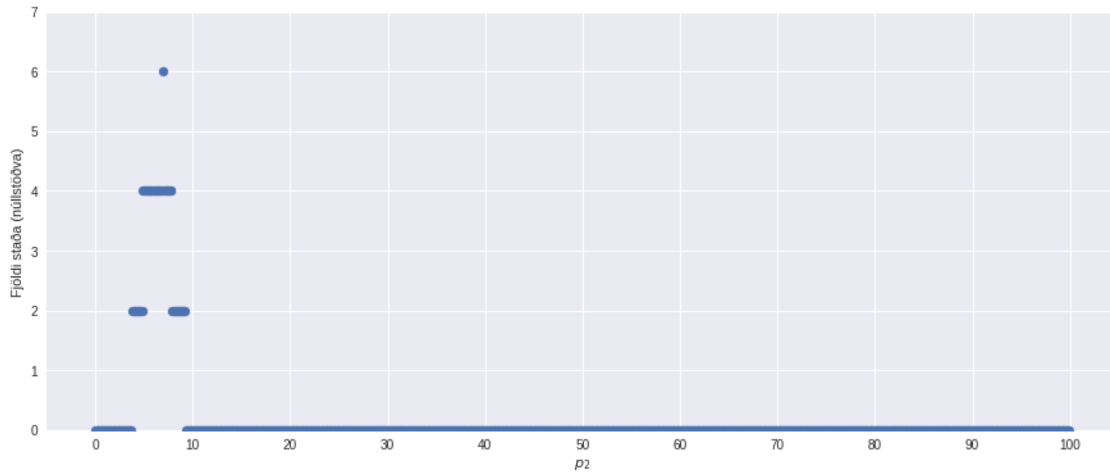
Suggested Activity 7

Calculate the intervals in p_2, with the rest of the parameters as in Step 4, for which there are 0, 2, 4 and 6 poses, respectively

Byrjum á að plotta graf með gildinu á p_2 á x - ás og fjölda staða á y - ás. Til að liðir Stewart platform virki sem skyldi þá er eðlilegt að gera ráð fyrir því að lengdin af einum fætinum sé ekki margfalt lengri en hinar. Því látum við hámarksgildi p_2 vera 100 eða tuttuguföld stærð næstlengsta fótarins, p_1. En það sést á myndinni að nóg hefði verið að hafa 9.2 sem hámarksgildi p_2.

```
In [0]: # Stikar úr Suggested Activity 4
        x1 = 5
        x2, y2 = 0, 6
        L1 = L3 = 3
        L2 = 3*np.sqrt(2)
        gamma = np.pi/4
        p1 = 5
        p3 = 3

        maxVal = 100
        interval = 0.1
        no = int(maxVal/interval)
        noSolutions = np.zeros(no)
        p2 = 0
        for i in range(len(noSolutions)):
          noSolutions[i] = countRoots(f, p2)
          p2 += interval

        theRange = np.arange(0, maxVal, interval)
        matrixRange = np.arange(0, len(noSolutions), 1)

        figure, axis = plt.subplots(figsize=(15, 6))

        # Teiknum grafið
        axis.scatter(theRange, noSolutions[matrixRange])

        # Setjum titil á grafið og ásana
        axis.set(xlabel=r"${p_2}$",
                 ylabel="Fjöldi staða (núllstöðva)",
                 title = "")
        axis.set_ylim([0,6])
        axis.grid()

        # Setjum minnsta sýnilega bilið á x - ásnum
        plt.xticks(np.arange(0, maxVal+1, 10))
        plt.yticks(np.arange(0, 8, 1))
        # Vistum myndina
        fileString = "sa7" + ".png"
        figure.savefig(fileString)
```

```
plt.grid(True)
plt.show()
```



Við sjáum að það fást 0 stöður þegar $10 < p_2 \leq 100$. Til að finna hvar fjöldi staða er ekki núll, takmörkum okkur þá við bilið $0 < p_2 \leq 10$.

Úr Suggested Activity 6 höfm við að það fást 2 stöður þegar $p_2 \in [3.8, 4.8] \cup [7.9, 9.2]$

```
In [0]: p2Roots(0, 10, 0.1, 4)
```

```
4.9
5.0
5.1000000000000005
5.2
5.300000000000001
5.4
5.5
5.6000000000000005
5.7
5.800000000000001
5.9
6.0
6.1000000000000005
6.2
6.300000000000001
6.4
6.5
6.6000000000000005
6.7
6.800000000000001
6.9
7.1000000000000005
7.2
```

```
7.300000000000001
7.4
7.5
7.6000000000000005
7.7
7.800000000000001
```

Við sjáum að það fást 4 stöður þegar $p_2 \in [4.9, 6.9] \cup [7.1, 7.8]$

```
In [0]: p2Roots(0, 10, 0.1, 6)
```

```
7.0
```

Við sjáum að það fást 6 stöður þegar $p_2 = 7$

```
In [0]: p2Roots(0, 10, 0.1, 0)
```

```
0.0
0.1
0.2
0.30000000000000004
0.4
0.5
0.6000000000000001
0.7000000000000001
0.8
0.9
1.0
1.1
1.2000000000000002
1.3
1.4000000000000001
1.5
1.6
1.7000000000000002
1.8
1.9000000000000001
2.0
2.1
2.2
2.3000000000000003
2.4000000000000004
2.5
2.6
2.7
2.8000000000000003
2.9000000000000004
```

```
3.0
3.1
3.2
3.3000000000000003
3.4000000000000004
3.5
3.6
3.7
9.3
9.4
9.5
9.600000000000001
9.700000000000001
9.8
9.9
10.0
```

Við sjáum að það fást 0 stöður þegar $p_2 \in [0, 3.7] \cup [9.3, 100]$

Varðandi leit að núllstöðvum í Suggested Activity 2, 4 og 5

Í þessum þremur æfingum átti að finna núllstöðvar $f(\theta)$ á bili $\theta \in [-\pi, \pi]$. Fyrsta skrefið við að finna núllstöðvarnar var að plotta ferilinn fyrir öll $\theta \in [-\pi, \pi]$. Í öllum æfingunum var ferilinn $f(\theta)$ er mjög flatur í kringum $f(\theta) = 0$, sem bendir til þess að villumögnunin (e. error magnification factor) sé mikil á þessu svæði. Þannig þó $|f(\theta)|$ sé mjög lág tala fyrir eitthvað θ þá getum við samt verið hlutfallslega mjög langt frá hinni réttu rót $f(\theta)$. Því þarf sérstaklega að gæta þess að stop skilyrðið sé ekki of slakt við leit rótarinnar. Fallið fsolve, sem var notað í þessu verkefni, er með strangt stop skilyrði þar sem það gerir kröfu um að munurinn á milli útreiknaðra θ gilda tveggja samliggjandi ítrana sé í mesta lagi 1.49012e-08.

Suggested Activity 8

Derive or look up the equations representing the forward kinematics of the three-dimensional, six-degrees-of-freedom Stewart platform. Write a Matlab program and demonstrate its use to solve the forward kinematics.

Verkefnið snýst um að hanna sex-leggja sviðspall. Til eru mismunandi lausnir á verkefninu. Ein lausn er þannig að hver leggur er í tveimur pörtum, báðir af fastri lengd, annar í snertingu við sviðspallinn, og hinn hluti hvers leggs snýst um sinn eigin snúningsöxul. Önnur lausn er þannig að leggirnir geta lengst, en upphafshæð þeirra er föst.

Skoðum aðeins verkefni sem tekst á við seinni útgáfuna. Sá hluti verkefnisins sem snýr að framvirkri hreyfifræði (e. forward kinematics) snýst um að finna staðsetningu og halla sviðspallsins þegar lengdir leggjanna eru þekktar, svipað því sem er var gert í tvívíða þriggja-leggja verkefninu að ofan. Það er engin góð þekkt jafna til að leysa það verkefni, nema bara að finna núllstöðvar á jöfnu svipaðri og var gert að ofan, þ.e.a.s. fundin er bestunarlausn með reikniritum. Látum vigurinn

$$\bar{t} = [t_x, t_y, t_z]$$

tákna staðsetningu miðju sviðspallsins í þrívídd, og þá eru skilgreind þrjú snúningshorn pallsins α, β, γ um x, y, z ásana. Svo eru leggirnir skilgreindir með sex vigrum af fastri lengd, og sex vigrar af breytilegri lengd (því leggirnir eru lengjanlegir) og staðsetningarhnit þeirra eru $\bar{b_i} = [b_i x, b_i y, 0]$ og $\bar{p_i} = [p_i x, p_i y, 0]$, $i = 1, ..., 6$. z-hnitin eru 0 því þau reikningar miða við að grunnhnit leggjanna séu á jörðinni eða upphafspunkti z-áss. Nú er hægt að tákna lengdir leggjanna $\bar{l_i} = -\bar{b_i} + \bar{t} + \underline{R} \cdot \bar{p_i}$ þar sem \$ $\underline{\text{R}}$\$ er snúningsfylki, reiknað út frá snúningshornunum þremur. Farið er nánar út í jöfnurnar á blaðsíðum 4,5 og 6 í verkefninu í hlekknum að ofan.

Svo er staðsetning og halli sviðspallsins fyrir allar mismunandi lengdir leggjanna reiknaður og þar með hægt að sjá fyrir sér hreyfingu hans. Svo eru mismunandi stöður fyrir fætur af sömu stærð, og það þarf að takast á við það.