Bibliography

[1] World Bank, “Countries Data”, 2012. [Online]. Available: http://data.worldbank.org/country. [Accessed 5 April 2012].

[2] H. H. Nguyen, V. Uraikul, C. V. Chan and P. Tontiwachwuthikul, “A comparison of automation techniques for optimization of compressor scheduling”, *Advances in Engineering Software,* vol. 39, no. 3, pp. 178–188, 2008.

[3] C. Voudouris, G. Owusu, R. Dorne and D. Lesaint, Service Chain Management: Technology Innovation for the Service Business, Berlin: Springer, 2008.

[4] H. I. H. Saravanamuttoo, G. F. C. Rogers, H. Cohen and P. V. Straniznicky, Gas Turbine Theory, Harlow: Prentice Hall, Pearson Education, 2009.

[5] K. H. Lüdtke, Process Centrifugal Compressors: Basics, Function, Operation, Design, Application, Berlin: Springer, 2004.

[6] D. Japikse and N. C. Baines, Introduction to Turbomachinery, Oxford: Oxford University Press, 1994.

[7] H. P. Bloch, A Practical Guide to Compressor Technology, New Jersey: John Whiley & Sons, 2006.

[8] A. T. Sayers, Hydraulic and Compressible Flow Turbomachines, London, New York: McGraw-Hill, 1990.

[9] K. G. Budinski and M. K. Budinski, Engineering Materials: Properties and selection, New Jersey: Pearson Prentice Hall, 2005.

[10] H. P. Bloch, Improving machinery reliability, Houston: Gulf Publishing Company, 1998.

[11] G. K. Sahu, Handbook Of Piping Design, New Delhi: New Age International, 1998.

[12] J. N. du Plessis and R. Pelzer, “Development of an intelligent control system for mine compressors”, in *Industrial and commercial use of energy (ICUE)*, Cape Town, 2011.

[13] Republic of South Africa. “Guidance notes for medical practitioners – Summary – Emergency Preparedness and Response”. [Online]. Available: www.dmr.gov.za/guidance-notes-for-medical-practitioners/. [Accessed 27 August 2012]. Department of Mineral Resources.

[14] NASA, “National Aeronautics and Space Adminstration”. [Online]. Available: http://www.grc.nasa.gov/. [Accessed 20 October 2012].

[15] U.S. Department of Energy, “Improving Compressed Air System Performance: a sourcebook for industry”. [Online]. Available: http://www1.eere.energy.gov. [Accessed 15 September 2012].

[16] PDHengineer, “Compressed Air Energy Efficiency”. [Online]. Available: http://www.pdhengineer.com/. [Accessed 15 September 2012].

[17] U.S. Department of Energy, “Energy Efficiency and Renewable Energy – Compressed Air Tip Sheet #3”, 2012. [Online]. Available: http://www1.eere.energy.gov/. [Accessed 26 August 2012].

[18] Y. A. Çengel and M. A. Boles, Thermodynamics: An Engineering Approach, New York: McGraw Hill, 2006.

[19] Republic of South Africa “Acts online”. [Online]. Available: http://www.acts.co.za/mhs/index.htm. [Accessed 26 August 2012]. Department of Minerals and Energy.

[20] Prime-Air Blowers Inc., “Prime-Air Blowers, Inc. – Worldwide Ventilation Specialists”, 2012. [Online]. Available: http://www.primeairblowers.com/Hurricane.htm. [Accessed 26 August 2012].

[21] U.S. Department of Energy, “Energy Efficiency and Renewable Energy – Compressed Air Tip Sheet #2”, 2012. [Online]. Available:

http://www1.eere.energy.gov/. [Accessed 27 August 2012].

[22] L. Sun, “Mathematical modeling of the flow in a pipeline with a leak”, *Mathematics and Computers in Simulation,* vol. 82, no. 11, pp. 2 253–2 267, 2012.

[23] S. L. Ke and H. C. Ti, “Transient analysis of isothermal gas flow in pipeline network”, *Chemical Engineering Journal,* vol. 76, no. 2, pp. 169–177, 2000.

[24] GE Sensing and Control, “Druck Industrial Pressure Sensors”. [Online]. Available: http://www.ge-mcs.com/. [Accessed 20 October 2012].

[25] L. Dingle and M. Tooley, Aircraft Engineering Principles, Oxford: Elsevier Butterworth Heinemann, 2005.

[26] D. S. Viswanath, T. K. Ghosh, D. H. L. Prasad, N. V. K. Dutt and K. Y. Rani, Viscosity of Liquids: Theory, Estimation, Experiment, and Data, Dordrecht: Springer, 2007.

[27] B. R. Munson, D. F. Young and T. H. Okiishi, Fundamentals of Fluid Mechanics, Hoboken, New Jersey: J. Wiley & Sons, 2006.

[28] J. L. Mathieu and J. Scott, An Introduction to Turbulent Flow, Cambridge: Cambridge University Press, 2000.

[29] D. Halliday, R. Resnick and J. Walker, Fundamentals of Physics, New York: J. Wiley & Sons, 2001.

[30] R. E. Sonntag, C. Borgnakke and G. J. Van Wylen, Fundamentals of Thermodynamics, New York: J. Wiley & Sons, 2003.

[31] Fuji Electric Co., “Fuji Electric France”. [Online]. Available: http://www.fujielectric.fr/. [Accessed 23 October 2012].

[32] Veris, “Verabar”. [Online]. Available: http://www.veris-inc.com/literature/-v150.pdf. [Accessed 14 October 2012].

[33] D. Brkić, “Iterative Methods for Looped Network Pipeline”, *Water Resources Management,* vol. 25, no. 12, pp. 2 951–2 987, 2011.

[34] E. Kreyszig, Advanced Engineering Mechanics, Singapore: John Wiley & Sons, 1999.

[35] G. Rizzoni, Principles and Applications of Electrical Engineering, Boston: McGraw-Hill , 2004.

[36] M. C. Potter and D. C. Wiggert, Schaum's outlines: Fluid Mechanics, New York: McGraw-Hill, 2008.

[37] R. Saidur, S. Mekhilef, M. B. Ali, A. Safari and H. A. Mohammed, “Applications of variable speed drive (VSD) in electrical motors energy savings”, *Renewable and Sustainable Energy Reviews,* vol. 16, no. 1, pp. 543-550, 2012.

[38] H. M. Deitel, P. J. Deitel and T. R. Nieto, Visual Basic .NET: How to program, New Jersey: Prentice Hall, 2002.

Appendix A: Simplified network solved

Equation 3.21 is used to solve the simple network.

$$P_1 - P_{t1} = B * v_1^2$$
$$P_{t2} - P_2 = B * v_2^2$$
$$P_{t3} - P_3 = B * v_3^2$$
$$m_1 = m_2 + m_3$$

With the known pressures chosen as P_1 = 700 kPa, P_2 = 650 kPa and P_3 = 645kPa. The constant A will be set equal to 75 for this illustration.

Arbitrarily choosing v_1 = 10, v_2 = 4 and v_3 = 6 we have

$$700\,000 - P_{t1} = 75 * 10^2$$
$$P_{t2} - 650\,000 = 75 * 4^2$$
$$P_{t3} - 645\,000 = 75 * 6^2$$

Solving we have

$$P_{t1} = 692\,500 \text{ Pa}$$
$$P_{t2} = 652\,200 \text{ Pa}$$
$$P_{t3} = 647\,700 \text{ Pa}$$

And the average intermediate node pressure for the first iteration is

$$P_{\text{average iteraiton 1}} = 663\,800 \text{ Pa}$$

Using this pressure to calculate the flows yields

$$v_{1\ \text{iteration}\ 1} = 21.97\ \text{m/s}$$

$$v_{2\ \text{iteration}\ 1} = 13.56\ \text{m/s}$$

$$v_{3\ \text{iteration}\ 1} = 15.83\ \text{m/s}$$

Because the pipes have the same cross sectional area and the density was assumed the same in each pipe, the fluid velocity will be used in the continuity equation. Therefore v_1 has to equal the sum of v_2 and v_3. As this is not yet equal v_1 is set equal to v_2 and v_3 to determine the next iteration's pressures.

Repeating this process gives the following results

$$P_{\text{average iteraiton}\ 2} = 657\ 500\ \text{Pa}$$

$$v_{1\ \text{iteration}\ 2} = 23.8\ \text{m/s}$$

$$v_{2\ \text{iteration}\ 2} = 10\ \text{m/s}$$

$$v_{3\ \text{iteration}\ 2} = 12.9\ \text{m/s}$$

$$P_{\text{average iteraiton}\ 3} = 658\ 455\ \text{Pa}$$

$$v_{1\ \text{iteration}\ 3} = 23.54\ \text{m/s}$$

$$v_{2\ \text{iteration}\ 3} = 10.62\ \text{m/s}$$

$$v_{3\ \text{iteration}\ 3} = 13.39\ \text{m/s}$$

$$P_{\text{average iteraiton}\ 4} = 658\ 448\ \text{Pa}$$

$$v_{1\ \text{iteration}\ 4} = 23.537\ \text{m/s}$$

$$v_{2\ \text{iteration}\ 4} = 10.613\ \text{m/s}$$

$$v_{3\ \text{iteration}\ 4} = 13.391\ \text{m/s}$$

$$P_{\text{average iteraiton 5}} = 658\,447.8\ \text{Pa}$$

$$v_{1\ \text{iteration 5}} = 23.5378\ \text{m/s}$$

$$v_{2\ \text{iteration 5}} = 10.6131\ \text{m/s}$$

$$v_{3\ \text{iteration 5}} = 13.390\ \text{m/s}$$

Figure 49 and Figure 50 show how the intermediate pressure and conservation of mass converge respectively.

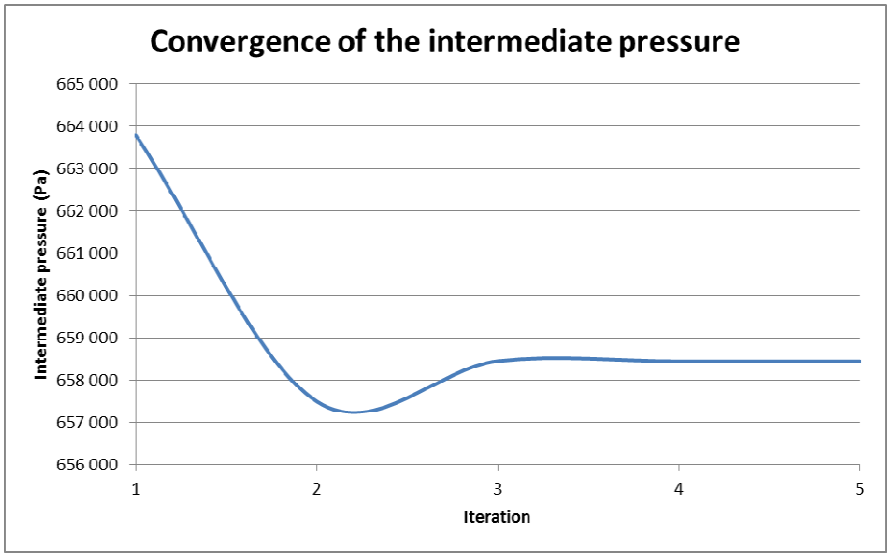

Figure 49: Intermediate pressure's convergence

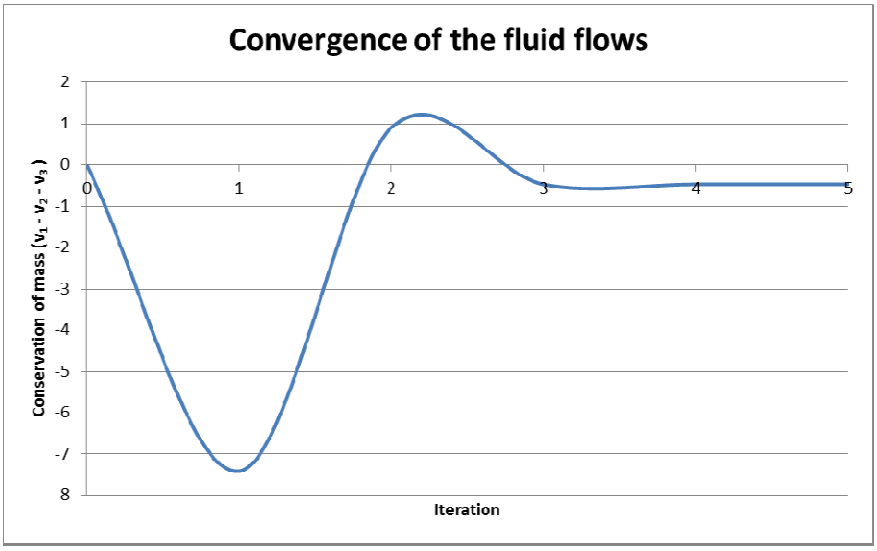

Figure 50: Fluid flow convergence

The reason the conservation of mass equation's result is not closer to zero is because the calculations were done by hand and there was considerable rounding off during each iteration.

Appendix B: Visual Basic .NET program code

```
Imports System.Math
Imports System.IO
Module Module1

    Sub Main()
        'Variables are declared
        Dim z, g, h, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11,
            T, R, epsilon, L, D1, D2, D3, D4, D5, D6, D7,
            D8, D9, D10, D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, L1, L2,
            L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13,
            L14, L15, L16, L17, L18, L19, L20, L21, k1, k2, k3, k4, k5, k6, k7,
            k8, k9, k10, k11, k12, k13, k14, k15, k16, k17, k18, k19, k20, k21, systemavepressure As Double

        Dim node1(4), node2(3),
            node3(3), node4(4),
            node5(3), node6(3),
            node7(3), node8(3), node9(3),
            W(15), P(15) As Double


        'Constant values are set
        R = 287
        T = 316

        D1 = 0.6
        L1 = 1000
        D2 = 0.6
        L2 = 1000
        D3 = 0.6
        L3 = 1000
        D4 = 0.6
        L4 = 1000
        D5 = 0.6
        L5 = 1000
```

```
D6 = 0.6
L6 = 1000
D7 = 0.6
L7 = 1000
D8 = 0.6
L8 = 1000
D9 = 0.6
L9 = 1000
D10 = 0.6
L10 = 1000
D11 = 0.6
L11 = 1000
D12 = 0.6
L12 = 1000
D13 = 0.6
L13 = 1000
D14 = 0.6
L14 = 1000
D15 = 0.6
L15 = 1000
D16 = 0.6
L16 = 1000
D17 = 0.6
L17 = 1000
D18 = 0.6
L18 = 1000
D19 = 0.6
L19 = 1000
D20 = 0.6
L20 = 1000
D21 = 0.6
L21 = 1000

k1 = 0
k2 = 0
k3 = 5
k4 = 0
```

```
k5 = 0
k6 = 0
k7 = 200
k8 = 0
k9 = 0
k10 = 0
k11 = 0
k12 = 0
k13 = 30
k14 = 0
k15 = 0
k16 = 0
k17 = 0
k18 = 40
k19 = 0
k21 = 0
k21 = 0


'Pipe roughness
epsilon = 45 / 1000000

'Supply and consumer node pressures
p1 = 735000
p2 = 500000
p3 = 500000
p4 = 500000
p5 = 735000
p6 = 500000
p7 = 500000
p8 = 500000
p9 = 735000
p10 = 735000
p11 = 735000

systemavepressure = (p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 + p10 + p11) / 11
```

```
node1(0) = systemavepressure
node2(0) = systemavepressure
node3(0) = systemavepressure
node4(0) = systemavepressure
node5(0) = systemavepressure
node6(0) = systemavepressure
node7(0) = systemavepressure
node8(0) = systemavepressure
node9(0) = systemavepressure


h = 0.1
z = 100000

'Delete file if exist
If (File.Exists("data.csv")) Then
    File.Delete("data.csv")
End If
Dim filewriter As StreamWriter = File.AppendText("data.csv")

Do
    'The nodes setup
    node1 = NodeFour(p1, p2, node2(0), node8(0), D1, D2, D5, D16, R, T, epsilon, h, L1, L2, L5, L16, k1, k2, k5, k16)
    node2 = NodeThree(node1(0), p3, node3(0), D5, D3, D4, R, T, epsilon, h, L5, L3, L4, k5, k3, k4)
    node3 = NodeThree(node2(0), p4, node4(0), D4, D5, D6, R, T, epsilon, h, L4, L6, L7, k4, k5, k6)
    node4 = NodeFour(node3(0), p5, node5(0), node8(0), D7, D8, D9, D17, R, T, epsilon, h, L7, L8, L9, L17, k7, k8, k9, k17)
    node5 = NodeThree(node4(0), p6, node6(0), D9, D10, D11, R, T, epsilon, h, L9, L10, L11, k9, k10, k11)
    node6 = NodeThree(node5(0), p7, node7(0), D11, D12, D13, R, T, epsilon, h, L11, L12, L13, k11, k12, k13)
    node7 = NodeFour(node6(0), p8, p9, node9(0), D13, D14, D15, D20, R, T, epsilon, h, L13, L14, L15, L20, k13, k14, k15,
    k20)
    node8 = NodeFour(node1(0), node4(0), node9(0), p10, D16, D17, D19, D18, R, T, epsilon, h, L16, L17, L19, L18, k16, k17,
    k19, k18)
    node9 = NodeThree(node8(0), p11, node7(0), D19, D20, D21, R, T, epsilon, h, L19, L20, L21, k19, k20, k21)

    'The global system mass balance
    g = Math.Abs(node2(3) - node3(1) + node3(3) - node4(1) + node4(3) - node5(1) + node5(3) - node6(1) + node6(3) - node7(1)
    + node1(4) - node8(1) + node8(3) - node9(1))
```

```
        filewriter.WriteLine(g)
        If Math.Abs(z - g) <= 0.001 Then
            Exit Do
        End If
        If g < 5 Then
            h = 0.1
        End If


        z = g
        Console.WriteLine(g)

    Loop While z > 0

    'Close file
    filewriter.Close()
    filewriter.Dispose()

    'Print to screen command for results
    Console.WriteLine("pt1" & vbTab & node1(0))
    Console.WriteLine("pt2" & vbTab & node2(0))
    Console.WriteLine("pt3" & vbTab & node3(0))
    Console.WriteLine("pt4" & vbTab & node4(0))
    Console.WriteLine("pt5" & vbTab & node5(0))
    Console.WriteLine("pt6" & vbTab & node6(0))
    Console.WriteLine("pt7" & vbTab & node7(0))
    Console.WriteLine("pt8" & vbTab & node8(0))
    Console.WriteLine("pt9" & vbTab & node9(0))

    Console.WriteLine("m1" & vbTab & node1(1))
    Console.WriteLine("m2" & vbTab & node1(2))
    Console.WriteLine("m3" & vbTab & node2(2))
    Console.WriteLine("m4" & vbTab & node2(3))
    Console.WriteLine("m5" & vbTab & node1(3))
    Console.WriteLine("m6" & vbTab & node3(2))
    Console.WriteLine("m7" & vbTab & node3(3))
```

```
        Console.WriteLine("m8" & vbTab & node4(2))
        Console.WriteLine("m9" & vbTab & node4(3))
        Console.WriteLine("m10" & vbTab & node5(2))
        Console.WriteLine("m11" & vbTab & node5(3))
        Console.WriteLine("m12" & vbTab & node6(2))
        Console.WriteLine("m13" & vbTab & node6(3))
        Console.WriteLine("m14" & vbTab & node7(2))
        Console.WriteLine("m15" & vbTab & node7(3))
        Console.WriteLine("m16" & vbTab & node1(4))
        Console.WriteLine("m17" & vbTab & node4(4))
        Console.WriteLine("m18" & vbTab & node8(4))
        Console.WriteLine("m19" & vbTab & node8(3))
        Console.WriteLine("m20" & vbTab & node7(4))
        Console.WriteLine("m21" & vbTab & node9(2))



        Console.ReadLine()

    End Sub

    Function NodeThree(ByVal Pr1 As Double, ByVal Pr2 As Double, ByVal Pr3 As Double, ByVal D1 As Double, ByVal D2 As Double,
                  ByVal D3 As Double, ByVal R As Double, ByVal T As Double, ByVal epsilon As Double, ByVal h As Double,
                  ByVal L1 As Double, ByVal L2 As Double, ByVal L3 As Double,
                   ByVal k1 As Double, ByVal k2 As Double, ByVal k3 As Double) 'Values are called by the function

        'Positive flow is from left to right and from top to bottom

        'Variables are declared
        Dim maxpressure1, minpressure1, v1, v2, v3 As Double
        Dim pt1reset As Double
        Dim Re1, Re2, Re3, rho1, rho2, rho3, f1, f2, f3, c1 As Double
        Dim A1, A2, A3, sig, sig1 As Double
        Dim results(6), W(3), P(3) As Double

        'non zero start values are given for the fluid flows
        v1 = 10
```

```
v2 = 10
v3 = 10

'Average local node pressure is used as an iterative starting point
results(0) = (Pr1 + Pr2 + Pr3) / 3

'If the program reaches the local maximum- or local minimum-pressure the pressure resets
'to pt1reset
pt1reset = results(0)

'Cross-sectional areas of each pipe is determined
'This has to be done every time a next node is calculated
A1 = PI * (D1 / 2) ^ 2
A2 = PI * (D2 / 2) ^ 2
A3 = PI * (D3 / 2) ^ 2

'sign dictating whether the varying pressure is increasing or decreasing
sig1 = -1

Do
    'A local maximum pressure is determined
    If Pr1 >= Pr2 And Pr1 >= Pr3 Then
        maxpressure1 = Pr1
    ElseIf Pr2 >= Pr1 And Pr2 >= Pr3 Then
        maxpressure1 = Pr2
    ElseIf Pr3 >= Pr1 And Pr3 >= Pr2 Then
        maxpressure1 = Pr3
    End If

    'A local minimum pressure is determined
    If Pr1 <= Pr2 And Pr1 <= Pr3 Then
        minpressure1 = Pr1
    ElseIf Pr2 <= Pr1 And Pr2 <= Pr3 Then
        minpressure1 = Pr2
    ElseIf Pr3 <= Pr1 And Pr3 <= Pr2 Then
        minpressure1 = Pr3
    End If
```

```
'Varying pressure reaches local maximum, resets and incrementally decreases
If results(0) >= maxpressure1 Then
    sig1 = 1
    results(0) = pt1reset
End If

'Varying pressure reaches local minimum, resets and incrementally increases
If results(0) <= minpressure1 Then
    sig1 = -1
    results(0) = pt1reset
End If

'Average density of fluid in pipe is calculated
rho1 = ((results(0) + Pr1) / 2) / (R * T)
rho2 = ((results(0) + Pr2) / 2) / (R * T)
rho3 = ((results(0) + Pr3) / 2) / (R * T)

'Reynolds number is calculated
Re1 = (rho1 * v1 * D1 / 0.0000186)
Re2 = (rho2 * v2 * D2 / 0.0000186)
Re3 = (rho3 * v3 * D3 / 0.0000186)

'Varying pressure mechanism
results(0) = results(0) - sig1 * h 'results(0) denotes the intermediate node pressure

'Ensures that the Reynolds number is positive because negative
'will cause the friction factor equation, containing a logarithmic function,
'to five an error
If Re1 < 0 Then
    Re1 = Re1 * -1
End If
If Re2 < 0 Then
    Re2 = Re2 * -1
End If
If Re3 < 0 Then
    Re3 = Re3 * -1
```

```
End If

f1 = 0.25 / (Log10((epsilon / (3.7 * D1)) + (5.74 / (Re1 ^ 0.9)))) ^ 2
f2 = 0.25 / (Log10((epsilon / (3.7 * D2)) + (5.74 / (Re2 ^ 0.9)))) ^ 2
f3 = 0.25 / (Log10((epsilon / (3.7 * D3)) + (5.74 / (Re3 ^ 0.9)))) ^ 2

'After the friction factor is calculated it is used to calculate the fluid velocity.
'This part also ensures that if the Reynolds number was made positive for calculations, that
'the fluid flow is changed back to negative flow.
If results(0) > Pr1 Then
    sig = -1
    v1 = Sqrt(2 * (sig * Pr1 - sig * results(0)) / (rho1 * (f1 * (L1 / D1) + k1)))
    v1 = v1 * sig
Else
    sig = 1
    v1 = Sqrt(2 * (sig * Pr1 - sig * results(0)) / (rho1 * (f1 * (L1 / D1) + k1)))
End If

If results(0) < Pr2 Then
    sig = -1
    v2 = Sqrt(2 * (-sig * Pr2 + sig * results(0)) / (rho2 * (f2 * (L2 / D2) + k2)))
    v2 = v2 * sig
Else
    sig = 1
    v2 = Sqrt(2 * (-sig * Pr2 + sig * results(0)) / (rho2 * (f2 * (L2 / D2) + k2)))
End If

If results(0) < Pr3 Then
    sig = -1
    v3 = Sqrt(2 * (-sig * Pr3 + sig * results(0)) / (rho3 * (f3 * (L3 / D3) + k3)))
    v3 = v3 * sig
Else
    sig = 1
    v3 = Sqrt(2 * (-sig * Pr3 + sig * results(0)) / (rho3 * (f3 * (L3 / D3) + k3)))
End If
```

```
            results(1) = rho1 * v1 * A1 'results(1) denotes the mass flow in the pipe to the right
            results(2) = rho2 * v2 * A2 'results(2) denotes the mass flow in the pipe to the bottom
            results(3) = rho3 * v3 * A3 'results(3) denotes the mass flow in the pipe to the left

            results(4) = rho1
            results(5) = rho2
            results(6) = rho3

            'The mass flow balance
            c1 = (results(1) - results(3) - results(2)) 'The mass flow balance

            'Checks if the mass flow balance is within predetermined bounds and
            'repeats function if it is not
        Loop While c1 > h Or c1 < -h
        'If the mass flow balance is within the bounds the mass flows of each pipe and the
        'intermediate node pressure is returned to the main program
        Return results

    End Function
    Function NodeFour(ByVal Pr1 As Double, ByVal Pr2 As Double, ByVal Pr3 As Double, ByVal Pr4 As Double, ByVal D1 As Double, ByVal
D2 As Double,
              ByVal D3 As Double, ByVal D4 As Double, ByVal R As Double, ByVal T As Double, ByVal epsilon As Double, ByVal h As
Double, ByVal L1 As Double,
              ByVal L2 As Double, ByVal L3 As Double, ByVal L4 As Double, ByVal k1 As Double, ByVal k2 As Double, ByVal k3 As Double,
ByVal k4 As Double)


        'This function is the same as the 3 pipe function, except that there is an additional pipe.

        Dim maxpressure1, minpressure1, v1, v2, v3, v4 As Double
        Dim pt1reset As Double
        Dim Re1, Re2, Re3, Re4, rho1, rho2, rho3, rho4, f1, f2, f3, f4, c1 As Double
        Dim A1, A2, A3, A4, sig, sig1 As Double
        Dim results(8), W(3), P(3) As Double


        v1 = 10
```

```
v2 = 10
v3 = 10
v4 = 10

results(0) = (Pr1 + Pr2 + Pr3 + Pr4) / 4

pt1reset = results(0)

A1 = PI * (D1 / 2) ^ 2
A2 = PI * (D2 / 2) ^ 2
A3 = PI * (D3 / 2) ^ 2
A4 = PI * (D4 / 2) ^ 2

sig1 = -1

Do

    If Pr1 >= Pr2 And Pr1 >= Pr3 And Pr1 >= Pr4 Then
        maxpressure1 = Pr1
    ElseIf Pr2 >= Pr1 And Pr2 >= Pr3 And Pr2 >= Pr4 Then
        maxpressure1 = Pr2
    ElseIf Pr3 >= Pr1 And Pr3 >= Pr2 And Pr3 >= Pr4 Then
        maxpressure1 = Pr3
    ElseIf Pr4 >= Pr1 And Pr4 >= Pr2 And Pr4 >= Pr3 Then
        maxpressure1 = Pr4
    End If

    If Pr1 <= Pr2 And Pr1 <= Pr3 And Pr1 <= Pr4 Then
        minpressure1 = Pr1
    ElseIf Pr2 <= Pr1 And Pr2 <= Pr3 And Pr2 <= Pr4 Then
        minpressure1 = Pr2
    ElseIf Pr3 <= Pr1 And Pr3 <= Pr2 And Pr3 <= Pr4 Then
        minpressure1 = Pr3
    ElseIf Pr4 <= Pr1 And Pr4 <= Pr2 And Pr4 <= Pr3 Then
        minpressure1 = Pr4
    End If
```

```
If results(0) >= maxpressure1 Then
    sig1 = 1
    results(0) = pt1reset
End If

If results(0) <= minpressure1 Then
    sig1 = -1
    results(0) = pt1reset
End If

rho1 = ((results(0) + Pr1) / 2) / (R * T)
rho2 = ((results(0) + Pr2) / 2) / (R * T)
rho3 = ((results(0) + Pr3) / 2) / (R * T)
rho4 = ((results(0) + Pr4) / 2) / (R * T)

Re1 = (rho1 * v1 * D1 / 0.0000186)
Re2 = (rho2 * v2 * D2 / 0.0000186)
Re3 = (rho3 * v3 * D3 / 0.0000186)
Re4 = (rho4 * v4 * D4 / 0.0000186)

results(0) = results(0) - sig1 * h

If Re1 < 0 Then
    Re1 = Re1 * -1
End If
If Re2 < 0 Then
    Re2 = Re2 * -1
End If
If Re3 < 0 Then
    Re3 = Re3 * -1
End If
If Re4 < 0 Then
    Re4 = Re4 * -1
End If

f1 = 0.25 / (Log10((epsilon / (3.7 * D1)) + (5.74 / (Re1 ^ 0.9)))) ^ 2
f2 = 0.25 / (Log10((epsilon / (3.7 * D2)) + (5.74 / (Re2 ^ 0.9)))) ^ 2
```

```
f3 = 0.25 / (Log10((epsilon / (3.7 * D3)) + (5.74 / (Re3 ^ 0.9)))) ^ 2
f4 = 0.25 / (Log10((epsilon / (3.7 * D4)) + (5.74 / (Re4 ^ 0.9)))) ^ 2

If results(0) > Pr1 Then
    sig = -1
    v1 = Sqrt(2 * (sig * Pr1 - sig * results(0)) / (rho1 * (f1 * (L1 / D1) + k1)))
    v1 = v1 * sig
Else
    sig = 1
    v1 = Sqrt(2 * (sig * Pr1 - sig * results(0)) / (rho1 * (f1 * (L1 / D1) + k1)))
End If

If results(0) < Pr2 Then
    sig = -1
    v2 = Sqrt(2 * (-sig * Pr2 + sig * results(0)) / (rho2 * (f2 * (L2 / D2) + k2)))
    v2 = v2 * sig
Else
    sig = 1
    v2 = Sqrt(2 * (-sig * Pr2 + sig * results(0)) / (rho2 * (f2 * (L2 / D2) + k2)))
End If

If results(0) < Pr3 Then
    sig = -1
    v3 = Sqrt(2 * (-sig * Pr3 + sig * results(0)) / (rho3 * (f3 * (L3 / D3) + k3)))
    v3 = v3 * sig
Else
    sig = 1
    v3 = Sqrt(2 * (-sig * Pr3 + sig * results(0)) / (rho3 * (f3 * (L3 / D3) + k3)))
End If

If results(0) < Pr4 Then
    sig = -1
    v4 = Sqrt(2 * (-sig * Pr4 + sig * results(0)) / (rho4 * (f4 * (L4 / D4) + k4)))
    v4 = v4 * sig
Else
    sig = 1
    v4 = Sqrt(2 * (-sig * Pr4 + sig * results(0)) / (rho4 * (f4 * (L4 / D4) + k4)))
```

```
            End If

            results(1) = rho1 * v1 * A1 'results(1) denotes the mass flow in the pipe that is to the right
            results(2) = rho2 * v2 * A2 'results(2) denotes the mass flow in the pipe that is to the bottom
            results(3) = rho3 * v3 * A3 'results(3) denotes the mass flow in the pipe that is to the left
            results(4) = rho4 * v4 * A4 'results(4) denotes the mass flow in the pipe that is to the top

            results(5) = rho1
            results(6) = rho2
            results(7) = rho3
            results(8) = rho4

            c1 = (results(1) - results(3) - results(2) - results(4))
        Loop While c1 > h Or c1 < -h
        Return results

    End Function

End Module
```

Appendix C: Network solving results

Two pipe setup with one intermediate node

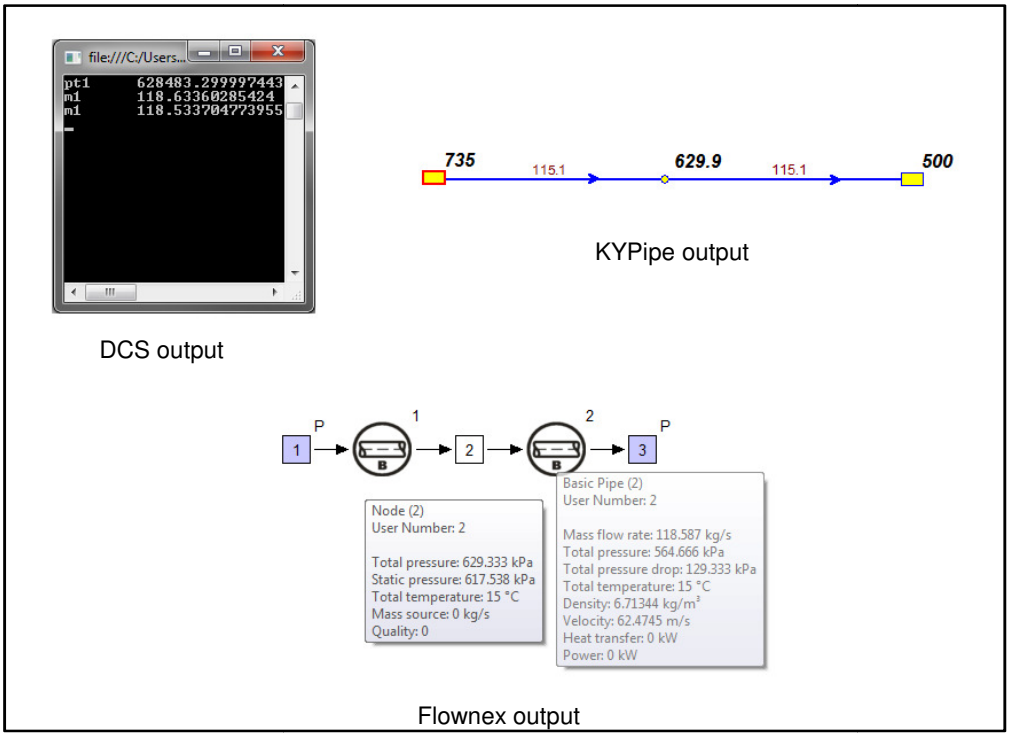

Figure 51: Output for two pipes and an intermediate node

Due to the window output style of Flownex and space constraints only the DCS and KYPipe results are shown from here on.

Three pipe setup with one intermediate node

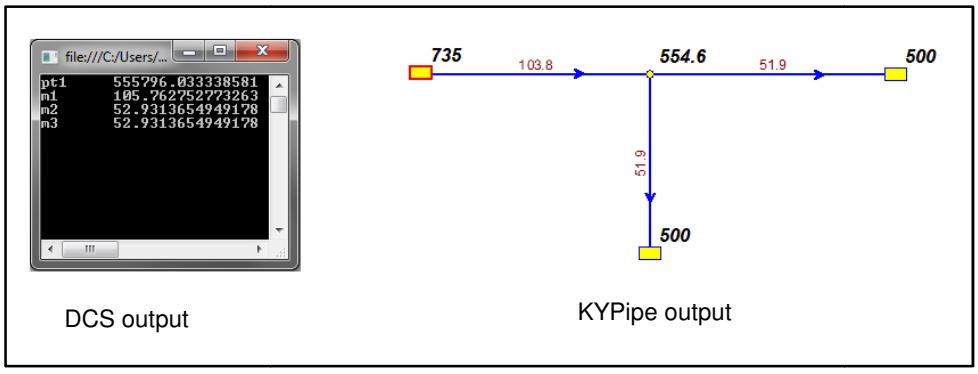

Figure 52: Output for three pipes and an intermediate node

Five pipe setup with two intermediate nodes

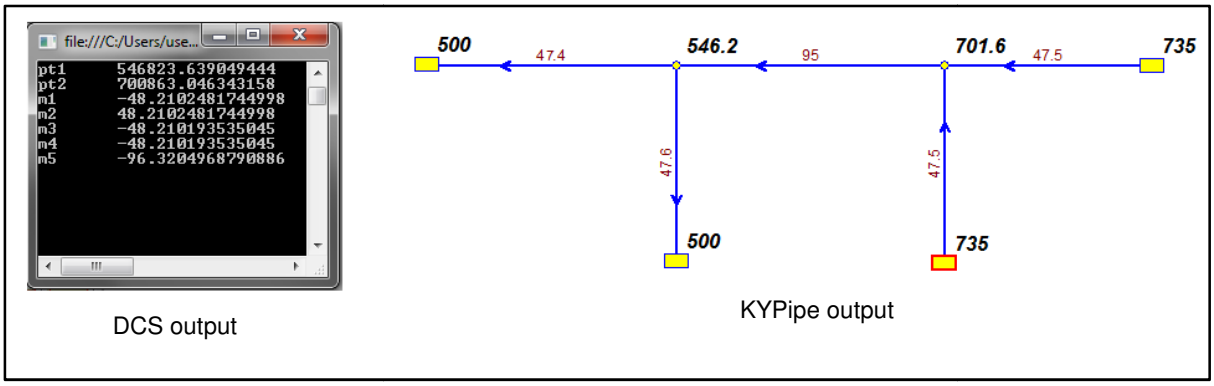

Figure 53: Output for five pipes and two intermediate nodes

Twenty-one pipe setup with nine intermediate nodes

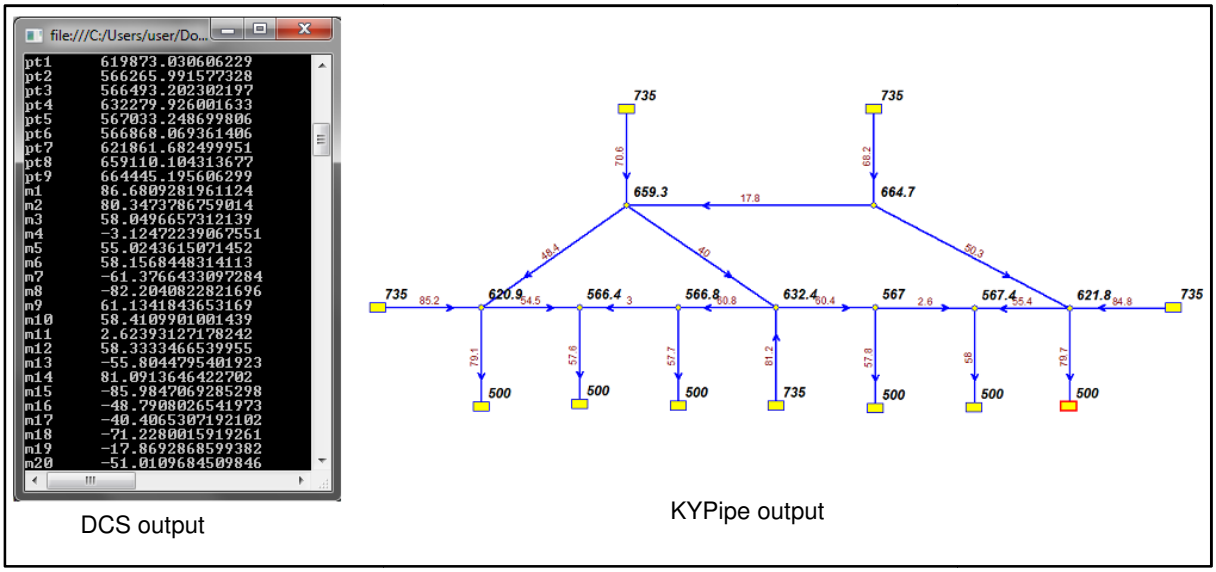

Figure 54: Output for twenty-one pipes and nine intermediate nodes

Appendix D: Compressor data for DCS verification

Actual power consumption for one day:

	VK40 C1	VK50 nr3 C1	VK40 nr1 C2	VK50 nr1 C1	VK40 nr2 C2	VK50 nr2 C1	VK10 nr2 C1	VK10 nr1 C1	Total Power
12:00:00 AM	3302.53	4861.38	0	0	3477.27	4963.45	12.49	841.69	17458.81
12:30:00 AM	3296.15	4853.44	0	0	3476.5	4964.71	1109.73	0	17700.55
01:00:00 AM	3325.78	4902.04	0	0	3488.63	5015.97	1123.96	0	17856.42
01:30:00 AM	3325.57	4896.53	0	0	3499.74	5010.56	1122.49	0	17854.95
02:00:00 AM	3323.45	4889.64	0	0	3486.57	5004.82	1120.73	0	17825.29
02:30:00 AM	3313.77	4868.66	0	0	3483.93	4984.34	1117.19	0	17767.99
03:00:00 AM	3335.83	4897.46	0	0	3503.6	5014.4	1121.97	0	17873.39
03:30:00 AM	3299.74	4895.69	0	0	3501.57	5009.67	1122.54	0	17829.36
04:00:00 AM	3206.97	4888.67	0	0	3498.53	5001.67	1121.07	0	17717.08
04:30:00 AM	3124.48	4916.79	0	0	3506.15	5030.69	1125.13	0	17703.43
05:00:00 AM	2918.97	4931.66	0	0	3546.97	5045.64	1128.54	0	17571.99
05:30:00 AM	2782.03	4934.24	0	0	3586.71	5048.93	1129.25	0	17481.39
06:00:00 AM	2893.06	4943.04	0	0	3606.51	5058.33	1132.45	0	17633.64
06:30:00 AM	3298.51	4900.7	0	0	3590.4	5019.12	1130.81	0	17939.81
07:00:00 AM	3329.39	4876.99	0	0	3565.14	4992.82	1128.64	0	17893.27
07:30:00 AM	3300.49	4853.45	1330.19	0	3557.08	4968.36	1125.08	1065.69	20200.65
08:00:00 AM	3119.88	4801.58	3835.29	0	3518.85	4924.9	1129.67	1125.9	22456.40
08:30:00 AM	3303.77	4851.94	3821.44	0	3529.47	4965.85	1127.16	1120.13	22720.11
09:00:00 AM	3271.75	4835.84	3800.66	151.1	3523.18	4941.17	1114.88	1106.34	22745.30
09:30:00 AM	3228.57	4853.71	3838.07	4323.38	3534.83	4966.1	1128.94	1117.63	26991.63
10:00:00 AM	3195.72	4828.33	3830.7	4357.05	3538.48	4943.81	1126.17	1119.68	26940.36
10:30:00 AM	3270.28	4796.77	3802.21	4582.87	3510.89	4920.33	1125.07	1118.93	27127.79
11:00:00 AM	3212.1	4801.87	3791.86	4277.76	3509.94	4917.46	1119.74	1113.51	26744.70
11:30:00 AM	3173.04	4768.88	3790.68	4226.07	3492.19	4884.97	1118.18	1110.75	26565.24
12:00:00 PM	3157.63	4747.86	3760.99	2473.8	3482.29	4859.71	1108.83	1100.5	24692.11
12:30:00 PM	3255.19	4803.82	3711.62	0	3478.33	4913.45	1108.83	1100.99	22372.75
01:00:00 PM	3258.32	4809.37	2856.53	0	3490.63	4918.89	1109.4	1100.06	21543.74
01:30:00 PM	3229.44	4788.7	1268.38	0	3478.39	4892.05	1098.34	1089.82	19845.68
02:00:00 PM	3223.32	4792.48	0	0	3493.55	4894.96	1099.13	1089.77	18593.79
02:30:00 PM	1037.23	4761.97	0	0	3489.91	4870.93	826.86	1097.81	16085.31
03:00:00 PM	0	4748.31	0	0	3459.7	4852.36	0	981.82	14042.82
03:30:00 PM	0	4772.05	0	0	3481.31	4878.34	1.87	1084.72	14218.94
04:00:00 PM	1818.27	4751.84	0	0	3458.93	4862.06	0	1085.93	15977.70
04:30:00 PM	0	4714.05	2119.46	0	3444.28	4829.84	0	1086.27	16194.59
05:00:00 PM	1147.01	4735.82	648.69	0	3445.17	4842.3	0	1081.99	15901.69
05:30:00 PM	797.25	4721.48	880.06	0	3448.79	4830.83	0	1082.13	15761.27
06:00:00 PM	0	4718.8	2354.21	0	3467.32	4828.64	0	1089.86	16459.58
06:30:00 PM	1651.38	4774.73	0	0	3484.16	4883.08	0	1088.4	15882.52
07:00:00 PM	0	4754.61	1157.94	0	3483.85	4857.45	0	1087.92	15342.56
07:30:00 PM	0	4743.95	1683.26	0	3485.63	4852.17	0	1091.64	15857.46
08:00:00 PM	1818.58	4784.25	0	0	3496.56	4890.86	0	1093.85	16084.93
08:30:00 PM	53.14	4787.59	0	0	3499.9	4891.72	330.67	1090.51	14654.38
09:00:00 PM	0	4820.92	0	0	3515.02	4925.98	911.28	1097.13	15271.21
09:30:00 PM	0	4811.67	0	0	3524.94	4913.86	141.15	1093.07	14485.59
10:00:00 PM	0	4818.89	0	0	3519.89	4924.84	873.92	1097.88	15236.34
10:30:00 PM	0	4833.44	0	0	3533.72	4937.8	0	1097.23	14403.13
11:00:00 PM	0	4864.29	0	0	3544.82	4969.58	0	1102.34	14481.99
11:30:00 PM	0	4827.13	0	0	3537.64	4931.08	438.15	1096.74	14831.72
Average Power	2116.64	4823.69	1089.21	508.17	3505.79	4934.39	727.30	769.76	18475.44

9	0	6	2	0	0	5	2	

Total stops and starts for large compressors	17
Total stops and starts for small compressors	7

Total starts and stops	24
Average power consumption for the day	18475.44

Theoretical power consumption for one day with the DCS selection method:

	VK40 C1	VK50 nr3 C1	VK40 nr1 C2	VK50 nr1 C1	VK40 nr2 C2	VK50 nr2 C1	VK10 nr2 C1	VK10 nr1 C1	Theoretical
12:00:00 AM	0	4943.04	1995.7964	4582.87	0	5058.33	0	0	16580.04
12:30:00 AM	0	4943.04	2494.7455	4582.87	0	5058.33	0	0	17078.99
01:00:00 AM	0	4943.04	2763.4104	4582.87	0	5058.33	0	0	17347.65
01:30:00 AM	0	4943.04	2763.4104	4582.87	0	5058.33	0	0	17347.65
02:00:00 AM	0	4943.04	2763.4104	4582.87	0	5058.33	0	0	17347.65
02:30:00 AM	0	4943.04	2763.4104	4582.87	0	5058.33	0	0	17347.65
03:00:00 AM	0	4943.04	2763.4104	4582.87	0	5058.33	0	0	17347.65
03:30:00 AM	0	4943.04	2494.7455	4582.87	0	5058.33	0	0	17078.99
04:00:00 AM	0	4943.04	2494.7455	4582.87	0	5058.33	0	0	17078.99
04:30:00 AM	0	4943.04	2609.8876	4582.87	0	5058.33	0	0	17194.13
05:00:00 AM	0	4943.04	2417.9841	4582.87	0	5058.33	0	0	17002.22
05:30:00 AM	0	4943.04	2417.9841	4582.87	0	5058.33	0	0	17002.22
06:00:00 AM	0	4943.04	2494.7455	4582.87	0	5058.33	0	0	17078.99
06:30:00 AM	0	4943.04	2494.7455	4582.87	0	5058.33	0	0	17078.99
07:00:00 AM	0	4943.04	2763.4104	4582.87	0	5058.33	0	0	17347.65
07:30:00 AM	1834.7065	4943.04	3838.07	4582.87	0	5058.33	0	0	20257.02
08:00:00 AM	2501.8725	4943.04	3838.07	4582.87	0	5058.33	0	0	20924.18
08:30:00 AM	2668.664	4943.04	3838.07	4582.87	0	5058.33	0	0	21090.97
09:00:00 AM	2735.3806	4943.04	3838.07	4582.87	0	5058.33	0	0	21157.69
09:30:00 AM	2501.8725	4943.04	3838.07	4582.87	3606.51	5058.33	0	0	24530.69
10:00:00 AM	1834.7065	4943.04	3838.07	4582.87	3606.51	5058.33	0	0	23863.53
10:30:00 AM	2001.498	4943.04	3838.07	4582.87	3606.51	5058.33	0	0	24030.32
11:00:00 AM	1934.7814	4943.04	3838.07	4582.87	3606.51	5058.33	0	0	23963.60
11:30:00 AM	1667.915	4943.04	3838.07	4582.87	3606.51	5058.33	0	0	23696.74
12:00:00 PM	0	4943.04	3838.07	4582.87	1983.5805	5058.33	0	0	20405.89
12:30:00 PM	0	4943.04	3838.07	4582.87	1983.5805	5058.33	0	0	20405.89
01:00:00 PM	0	4943.04	3838.07	4582.87	0	5058.33	0	0	18422.31
01:30:00 PM	0	4943.04	3454.263	4582.87	0	5058.33	0	0	18038.50
02:00:00 PM	0	4943.04	2686.649	4582.87	0	5058.33	0	0	17270.89
02:30:00 PM	0	4448.736	0	4582.87	0	5058.33	0	0	14089.94
03:00:00 PM	0	4844.1792	0	4582.87	0	5058.33	0	0	14485.38
03:30:00 PM	0	4943.04	0	4582.87	0	5058.33	0	0	14584.24
04:00:00 PM	0	4448.736	0	4582.87	0	5058.33	0	0	14089.94
04:30:00 PM	0	4844.1792	0	4582.87	0	5058.33	0	0	14485.38
05:00:00 PM	0	4943.04	1535.228	4582.87	0	5058.33	0	0	16119.47
05:30:00 PM	0	4943.04	1919.035	4582.87	0	5058.33	0	0	16503.28
06:00:00 PM	0	4399.3056	0	4582.87	0	5058.33	0	0	14040.51
06:30:00 PM	0	4448.736	0	4582.87	0	5058.33	0	0	14089.94
07:00:00 PM	0	4943.04	0	4582.87	0	5058.33	905.96	1125.9	16616.10
07:30:00 PM	0	4201.584	0	4582.87	0	5058.33	0	0	13842.78
08:00:00 PM	0	4547.5968	0	4582.87	0	5058.33	0	0	14188.80
08:30:00 PM	0	4943.04	0	4582.87	0	5058.33	0	0	14584.24
09:00:00 PM	0	4695.888	0	4582.87	0	5058.33	0	0	14337.09
09:30:00 PM	0	4943.04	0	4582.87	0	5058.33	0	675.54	15259.78
10:00:00 PM	0	4695.888	0	4582.87	0	5058.33	0	0	14337.09
10:30:00 PM	0	4695.888	0	4582.87	0	5058.33	0	0	14337.09
11:00:00 PM	0	4695.888	0	4582.87	0	5058.33	0	0	14337.09
11:30:00 PM	0	4943.04	0	4582.87	0	5058.33	0	1125.9	15710.14
Average Power	410.03	4852.42	1961.41	4582.87	458.33	5058.33	18.87	60.99	17403.25

Start/Stop	2	0	3	0	2	0	2	5

Total stops and starts for large compressors	7
Total stops and starts for small compressors	7

Total starts and stops	14
Theoretical average power consumption for the day	17403.25

Theoretical power consumption for one day with the DCS selection method and reduced shaft- and compressor exit pressures:

	VK40 C1	VK50 nr3 C1	VK40 nr1 C2	VK50 nr1 C1	VK40 nr2 C2	VK50 nr2 C1	VK10 nr2 C1	VK10 nr1 C1	Theoretical
12:00:00 AM	0	4943.04	0	4582.87	0	5058.33	1132.45	1125.9	16842.59
12:30:00 AM	0	4943.04	1535.228	4582.87	0	5058.33	0	0	16119.47
01:00:00 AM	0	4943.04	1535.228	4582.87	0	5058.33	0	0	16119.47
01:30:00 AM	0	4943.04	1535.228	4582.87	0	5058.33	0	0	16119.47
02:00:00 AM	0	4349.8752	0	4582.87	0	5058.33	0	0	13991.08
02:30:00 AM	0	4399.3056	0	4582.87	0	5058.33	0	0	14040.51
03:00:00 AM	0	4251.0144	0	4582.87	0	5058.33	0	0	13892.21
03:30:00 AM	0	3954.432	0	4582.87	0	5058.33	0	0	13595.63
04:00:00 AM	0	3954.432	0	4582.87	0	5058.33	0	0	13595.63
04:30:00 AM	0	3954.432	0	4582.87	0	5058.33	0	0	13595.63
05:00:00 AM	0	3954.432	0	4582.87	0	5058.33	0	0	13595.63
05:30:00 AM	0	3954.432	0	4582.87	0	5058.33	0	0	13595.63
06:00:00 AM	2668.664	4943.04	0	4582.87	0	5058.33	0	0	17252.90
06:30:00 AM	2668.664	4943.04	0	4582.87	0	5058.33	0	0	17252.90
07:00:00 AM	3002.247	4943.04	0	4582.87	0	5058.33	0	0	17586.49
07:30:00 AM	3335.83	4943.04	2494.7455	4582.87	0	5058.33	0	0	20414.82
08:00:00 AM	3335.83	4943.04	3070.456	4582.87	0	5058.33	0	0	20990.53
08:30:00 AM	3335.83	4943.04	3070.456	4582.87	0	5058.33	0	0	20990.53
09:00:00 AM	3335.83	4943.04	3377.5016	4582.87	0	5058.33	0	0	21297.57
09:30:00 AM	3335.83	4943.04	2686.649	4582.87	3606.51	5058.33	0	0	24213.23
10:00:00 AM	3335.83	4943.04	2494.7455	4582.87	3606.51	5058.33	0	0	24021.33
10:30:00 AM	3335.83	4943.04	2686.649	4582.87	3606.51	5058.33	0	0	24213.23
11:00:00 AM	3335.83	4943.04	2494.7455	4582.87	3606.51	5058.33	0	0	24021.33
11:30:00 AM	3335.83	4943.04	2149.3192	4582.87	3606.51	5058.33	0	0	23675.90
12:00:00 PM	3335.83	4943.04	0	4582.87	2813.0778	5058.33	0	0	20733.15
12:30:00 PM	3335.83	4943.04	0	4582.87	2596.6872	5058.33	0	0	20516.76
01:00:00 PM	3335.83	4943.04	0	4582.87	1442.604	5058.33	0	0	19362.67
01:30:00 PM	3335.83	4943.04	0	4582.87	0	5058.33	0	0	17920.07
02:00:00 PM	0	4943.04	0	3757.9534	0	5058.33	0	0	13759.32
02:30:00 PM	0	4943.04	0	1833.148	0	5058.33	0	0	11834.52
03:00:00 PM	0	4943.04	0	1833.148	0	5058.33	0	0	11834.52
03:30:00 PM	0	4943.04	0	2062.2915	0	5058.33	0	0	12063.66
04:00:00 PM	0	4943.04	0	1833.148	0	5058.33	0	0	11834.52
04:30:00 PM	0	4943.04	0	1833.148	0	5058.33	0	0	11834.52
05:00:00 PM	0	4943.04	0	2062.2915	0	5058.33	0	0	12063.66
05:30:00 PM	0	4943.04	0	2520.5785	0	5058.33	0	0	12521.95
06:00:00 PM	0	4943.04	0	1833.148	0	5058.33	0	0	11834.52
06:30:00 PM	0	4943.04	0	1833.148	0	5058.33	0	0	11834.52
07:00:00 PM	0	4943.04	0	3574.6386	0	5058.33	0	0	13576.01
07:30:00 PM	0	4943.04	0	2749.722	0	5058.33	0	0	12751.09
08:00:00 PM	0	4943.04	0	2749.722	0	5058.33	0	0	12751.09
08:30:00 PM	0	4943.04	0	3299.6664	0	5058.33	0	0	13301.04
09:00:00 PM	0	4943.04	0	2841.3794	0	5058.33	0	0	12842.75
09:30:00 PM	0	4943.04	0	3666.296	0	5058.33	0	0	13667.67
10:00:00 PM	0	4943.04	0	2841.3794	0	5058.33	0	0	12842.75
10:30:00 PM	0	4943.04	0	2841.3794	0	5058.33	0	0	12842.75
11:00:00 PM	0	4943.04	0	2978.8655	0	5058.33	0	0	12980.24
11:30:00 PM	0	4943.04	0	3666.296	0	5058.33	0	0	13667.67
Average Power	1077.20	4801.96	606.89	3769.41	518.44	5058.33	23.59	23.46	15879.27

Start/Stop	2	0	4	0	2	0	1	1

Total stops and starts for large compressors	8
Total stops and starts for small compressors	2

Total starts and stops	10
Theoretical average power consumption for the day	15879.27