

Analysis of Steady State Flow in Systems

by
Ryan Mitchell
for
MECH2700
Numerical Analysis I

19 Oct 2015.

Contents

1	Introduction	2
2	General Gauss-Jordan Function	3
3	A Chebyshev Filter For Electrical Signals	4
3.1	Conclusion	5
4	A Water-Supply Network	6
4.1	Conclusion	6

List of Figures

1	Chebyshev Filter Circuit	4
2	A Water-Supply Network	6

1 Introduction

Most systems display attributes of both a transient and steady state response. A transient response, also known as the warm up period, is the initial flow that moves through the network and charges / fills storage devices. The network has to be able to contain these surges and influxes without breaking. As the system reaches an equilibrium condition, also known as the steady state, it becomes easier and more predictable to model.

Once the system has reached steady state the system can be analysed analytically in a number of ways. Simple systems can be analysed by hand but as the complexity of the system grows the time requirement for hand calculations increases dramatically. If a steady state system can be described in the form of a set of constraints on all the unknown flow quantities, and the vector of values satisfies all the equations simultaneously, a computer can be used to decrease computational time. If the simultaneous equations can be modelled as a matrix, the processing time is further reduced.

This assignment looks at two network systems and the methods behind solving them:

1. A Chebyshev Filter
2. A Water-Supply Network

A Chebyshev Filter is a high-order filter excellent performance. It is built with passive components and filters electrical signals. The filter contains a system of linear constraint equations but the coefficients as well as the unknown values can be complex numbers. The Water-Supply Network is a network of pipes with the flow of water being described / given. The network contains a system of non-linear constraint equations but the coefficients are real numbers.

This assignment makes use of the Gauss-Jordan matrix elimination method. This method uses row-reduction to modify the values of the lower left-hand corner of the matrix until it is filled with zeros. Once complete the matrix is said to be an upper triangular matrix in row echelon form. Once all the leading non-zero entries on the diagonal are reduced to 1 the matrix is a reduced row echelon matrix. There are three types of elementary row operations:

1. Swapping two rows,
2. Multiplying a row by a non-zero number
3. Adding a multiple of one row to another row

This solver has been designed to handle floats and complex numbers and the code can be seen in Section 2.

2 General Gauss-Jordan Function

```
# file: gjSolver.py
"""
Python program for MECH2700, 2015, Semester 2.
Contains a generic Gauss-Jordan matrix solver that reduces a matrix
until a Gauss-Jordan Matrix is achieved. This solver is equipped to
handle floats and complex numbers.

R. C. Mitchell
2015 --> 19 Oct 2015
"""

#-----
# Import appropriate modules
from math import *
import numpy as np

#-----
# Global definitions of the variables for ease of editing

#-----
# Start function definition

def GaussJordan():
    """
    Comments
    """
    j = np.complex(1j)
    # print 'j = ', j, ' & Type(j)', type(j)

    return
```

3 A Chebyshev Filter For Electrical Signals

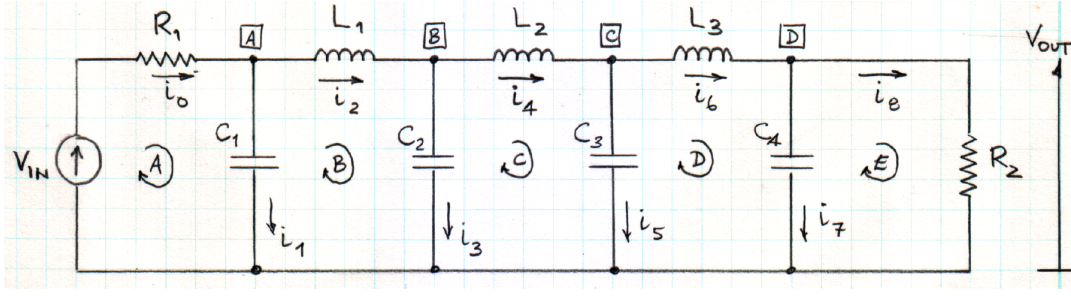


Figure 1: Chebyshev Filter Circuit

To analyse the circuit provided in Figure 1, Mesh and Nodal Analysis techniques are used. Mesh Analysis is derived from Kirchoff's Voltage Law which states that 'the directed sum of the electrical potential differences (voltage) around any closed network is zero'. Nodal Analysis is derived from Kirchoff's Current Law which states that 'at any node (junction) in an electrical circuit, the sum of currents flowing into that node is equal to the sum of currents flowing out of that node'. Both tools are required to derive the nine equations in order to solve for the unknowns, ω and V_{IN} , in the system. The five Mesh equations (equations 1 to 5) and four Node equations (equations 6 to 9) were found to be

$$\textcircled{A} \quad V_{IN} - Z_{R1}i_0 - Z_{C1}i_1 = 0 \quad (1)$$

$$V_{IN} = Z_{R1}i_0 + Z_{C1}i_1$$

$$\textcircled{B} \quad Z_{C1}i_1 - Z_{L1}i_2 - Z_{C2}i_3 = 0 \quad (2)$$

$$\textcircled{C} \quad Z_{C2}i_3 - Z_{L2}i_4 - Z_{C3}i_5 = 0 \quad (3)$$

$$\textcircled{D} \quad Z_{C3}i_5 - Z_{L3}i_6 - Z_{C4}i_7 = 0 \quad (4)$$

$$\textcircled{E} \quad Z_{C4}i_7 - Z_{R2}i_8 = 0 \quad (5)$$

$$\boxed{A} \quad i_0 - i_1 - i_2 = 0 \quad (6)$$

$$\boxed{B} \quad i_2 - i_3 - i_4 = 0 \quad (7)$$

$$\boxed{C} \quad i_4 - i_5 - i_6 = 0 \quad (8)$$

$$\boxed{D} \quad i_6 - i_7 - i_8 = 0 \quad (9)$$

These equations can then be compiled into matrix form in order to make use of software tools.

$$\begin{bmatrix}
Z_{R1} & Z_{C1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & Z_{C1} & -Z_{L1} & -Z_{C2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & Z_{C2} & -Z_{L2} & -Z_{C3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & Z_{C3} & -Z_{L3} & -Z_{C4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_{C4} & -Z_{R2} \\
1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1
\end{bmatrix} * \begin{bmatrix} i_0 \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \end{bmatrix} = \begin{bmatrix} V_{IN} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Which becomes

$$\begin{bmatrix}
R_1 & \frac{-j}{\omega C_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-j}{\omega C_1} & -j\omega L_1 & \frac{j}{\omega C_2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{-j}{\omega C_2} & -j\omega L_2 & \frac{j}{\omega C_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{-j}{\omega C_3} & -j\omega L_3 & \frac{j}{\omega C_4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-j}{\omega C_4} & -R_2 \\
1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1
\end{bmatrix} * \begin{bmatrix} i_0 \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \end{bmatrix} = \begin{bmatrix} V_{IN} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Need to check for singularity of the matrix as the first step of the function!

3.1 Conclusion

4 A Water-Supply Network

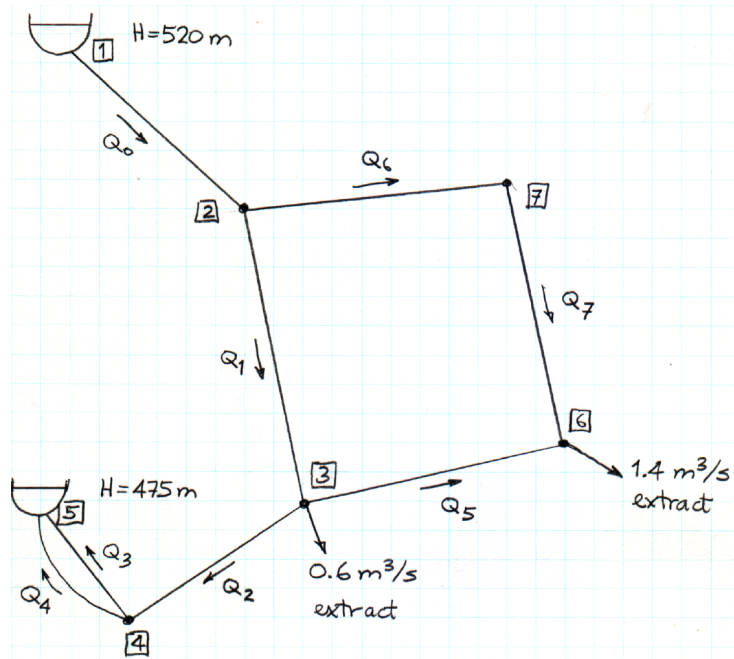


Figure 2: A Water-Supply Network

4.1 Conclusion