Balancing a Chemical Equation

G V V Sharma*

Contents

***Abstract*—This manual shows how to balance chemical equations using matrices.**

Download python codes using

```
svn co https://github.com/gadepall/school/trunk/
    training
```

1 Chemistry

1. Express the problem of balancing the following chemical equation as a matrix equation.

$$Fe + H_2O \rightarrow Fe_3O_4 + H_2 \tag{1.1.1}$$

Solution: Let the balanced version of (1.1.1) be

$$x_1 Fe + x_2 H_2O \rightarrow x_3 Fe_3O_4 + x_4 H_2 \tag{1.1.2}$$

which results in the following equations

$$\begin{aligned} (x_1 - 3x_3)\, Fe &= 0 \\ (2x_2 - 2x_4)\, H &= 0 \\ (x_2 - 4x_3)\, O &= 0 \end{aligned} \tag{1.1.3}$$

which can be expressed as

$$\begin{aligned} x_1 + 0.x_2 - 3x_3 + 0.x_4 &= 0 \\ 0.x_1 + 2x_2 + 0.x_3 - 2x_4 &= 0 \\ 0.x_1 + x_2 - 4x_3 + 0.x_4 &= 0 \end{aligned} \tag{1.1.4}$$

resulting in the matrix equation

$$\begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 2 & 0 & -2 \\ 0 & 1 & -4 & 0 \end{pmatrix} \mathbf{x} = \mathbf{0} \tag{1.1.5}$$

where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \tag{1.1.6}$$

2. Solve (1.1.2) by row reducing the matrix in (1.1.5).

Solution: (1.1.5) can be row reduced as follows

$$\begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 2 & 0 & -2 \\ 0 & 1 & -4 & 0 \end{pmatrix} \xleftrightarrow{R_2 \leftarrow \frac{R_2}{2}} \begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & -4 & 0 \end{pmatrix} \tag{1.2.1}$$

$$\xleftrightarrow{R_3 \leftarrow R_3 - R_2} \begin{pmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -4 & 1 \end{pmatrix} \tag{1.2.2}$$

$$\xleftrightarrow{R_1 \leftarrow 4R_1 - 3R_3} \begin{pmatrix} 4 & 0 & 0 & -3 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -4 & 1 \end{pmatrix} \tag{1.2.3}$$

$$\xleftrightarrow[R_3 \leftarrow -\frac{1}{4}R_3]{R_1 \leftarrow \frac{1}{4}} \begin{pmatrix} 1 & 0 & 0 & -\frac{3}{4} \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -\frac{1}{4} \end{pmatrix} \tag{1.2.4}$$

Thus,

$$x_1 = \frac{3}{4}x_4, x_2 = x_4, x_3 = \frac{1}{4}x_4 \tag{1.2.5}$$

$$\tag{1.2.6}$$

$$\implies \mathbf{x} = x_4 \begin{pmatrix} \frac{3}{4} \\ 1 \\ \frac{1}{4} \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 1 \\ 4 \end{pmatrix} \tag{1.2.7}$$

upon substituting $x_4 = 4$. (1.1.2) then becomes

$$3Fe + 4H_2O \rightarrow Fe_3O_4 + 4H_2 \tag{1.2.8}$$

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

3. Verify your answer through a python code.
Solution: Execute

```
codes/chembal.py
```

2 Mathematics

1. Find the equation of the plane P that contains the point $\begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$ and is perpendicular to each of the planes

$$P_1 : \quad \begin{pmatrix} 2 & 3 & -2 \end{pmatrix} \mathbf{x} = 5 \tag{2.1.1}$$

$$P_2 : \quad \begin{pmatrix} 1 & 2 & -3 \end{pmatrix} \mathbf{x} = 8 \tag{2.1.2}$$

From (**??**), the normals to P_1, P_2 are

$$\begin{aligned} \mathbf{n}_1 &= \begin{pmatrix} 2 & 3 & -2 \end{pmatrix} \\ \mathbf{n}_2 &= \begin{pmatrix} 1 & 2 & -3 \end{pmatrix} \end{aligned} \tag{2.1.3}$$

$\because P \perp P_1, P \perp P_2$, if $\mathbf{n}$ be the normal to P, $\mathbf{n} \perp \mathbf{n}_1, \mathbf{n} \perp \mathbf{n}_2$, which can be expressed using (**??**) as

$$\begin{pmatrix} 2 & 3 & -2 \\ 1 & 2 & -3 \end{pmatrix} \mathbf{n} = 0 \tag{2.1.4}$$

Obtain $\mathbf{n}$ using row reduction.
2. Verify your answer through a python code.
3. Verify that $\mathbf{n} = \mathbf{n}_1 \times \mathbf{n}_2$.

3 Physics

1. A force of $\mathbf{F} = 7\hat{i} + 3\hat{j} - 5\hat{k}$ acts on a particle whose position vector is $\mathbf{r} = \hat{i} - \hat{j} + \hat{k}$. Find the torque about the origin given by $\mathbf{F} \times \mathbf{r}$ using a matrix equation.
2. Verify your answer using python.

4 Maximum Likelihood Sequence Estimation (MLSE)

Let $\mathbf{a} = (a_0, \ldots, a_{m-1})$ be the sequence to be estimated. Where m is the length of the sequence. The received symbol with the channel filter cofficeients $\mathbf{h} = (h_1, \ldots, h_{L-1})$ is

$$\mathbf{r} = \mathbf{s} + \mathbf{n} \tag{2.1}$$

where,

$$\mathbf{s} = \mathbf{a} * \mathbf{h} \tag{2.2}$$

The MLSE or Minimum distance receiver estimation is given by

$$\hat{\mathbf{a}} = \underset{(a_0,\ldots,a_{m-1})}{\operatorname{argmin}}\, d \tag{2.3}$$

where,

$$d = \sum_k |r_k - s_k|^2 \quad s_k = \sum_{l=0}^{L-1} a_l h_{k-l} \tag{2.4}$$

For M-ary modulation formats, $\mathbf{a}$ takes on M^m values. So, Viterbi Algorithm does this in an efficient way.

4.1 Viterbi Algorithm

Let

$$r_k = a_k + 0.5a + k - 1 + n_k \tag{2.5}$$

Where,

$$a_k \in \{0, 1\} \quad n_k \sim N(0, \sigma^2) \tag{2.6}$$

TABLE 2: Truth Table for Trellis

a_k	$\Phi_k = a_{k-1}$	s_k	$\Phi_{k+1} = a_k$
0	$\Phi_0 = 0$	0	$\Phi_0 = 0$
1	$\Phi_0 = 0$	1	$\Phi_1 = 1$
0	$\Phi_1 = 1$	0.5	$\Phi_0 = 0$
1	$\Phi_1 = 1$	1.5	$\Phi_1 = 1$

Where,

$$a_k = current\_symbol \quad \Phi_0 = Current\_state \tag{2.7}$$

$$s_k = a_k + a_{k-1} \quad \Phi_{k+1} = Next\_state \tag{2.8}$$

The fineite state machine for the above truth table is shown in Fig. **??** The Trellis diagram for the FSM shown in Fig. **??** is given in Fig. **??**

The multi-stage trellis diagram is shown in Fig. **??**.

Suppose, we receive $r_k = [0.2, 0.6, 0.9, 0.1]$ symbols at time instants $k = 0, 1, 2, 3$

- With Symbol-by-Symbol detection:
  - Detection threshold = 0.5.
  - Detected Symbols are $[0, 1, 1, 0]$
- ML detection/Minimum distance metric to be minimized

$$\zeta_i = \sum_k |r_k - s_{k,i}|^2 \tag{2.9}$$

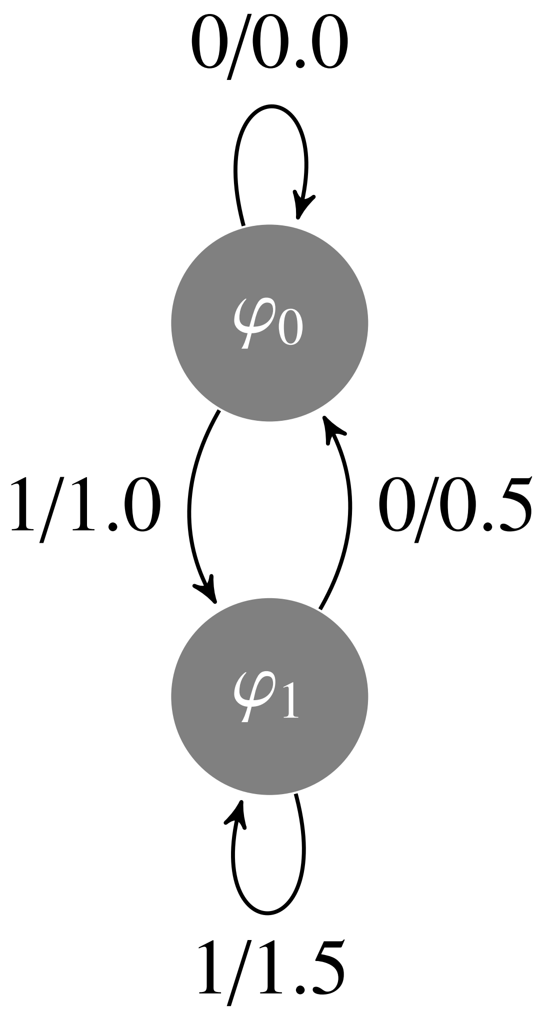

Fig. 2: Finite state machine.

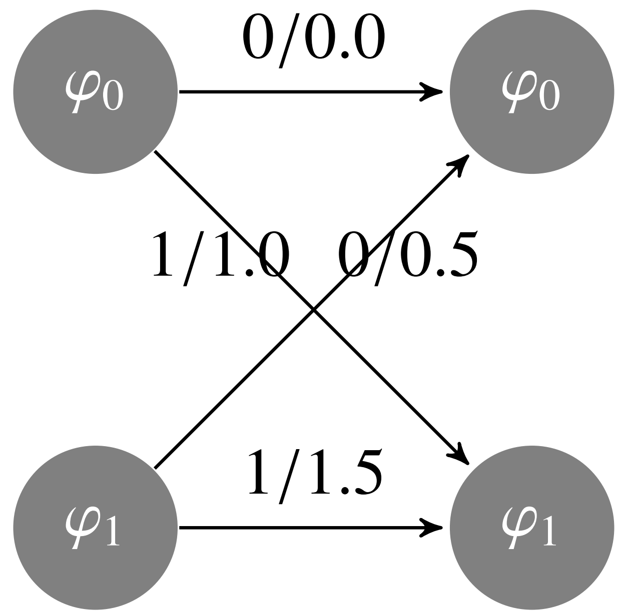

Fig. 2: Trellis Diagram.

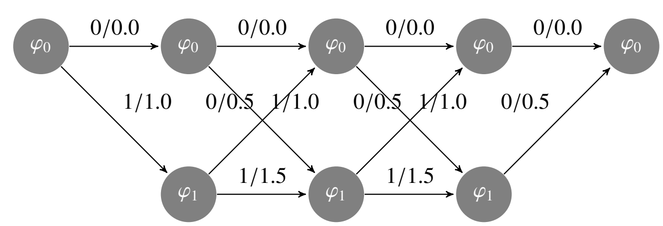

Fig. 2: Trellis Diagram.

Where, i is over different transmit symbol vectors

$$\zeta_i = (r_0 - s_{0,i})^2 + (r_2 - s_{2,i})^2 + (r_2 - s_{2,i})^2 + (r_3 - s_{3,i})^2 \tag{2.10}$$

So, the branc metric at k^{th} symbol period is

$$\zeta_{k,i} = (r_k - s_{k,i})^2 \tag{2.11}$$

Sum of these branch metrics to be minimized in MLSE.

The Trellis Diagram with branch metrics is shown in Fig. **??**. The shortest path using Viterbi Algorithm (VA) is shown in Fig. **??**. The symbol-by-symbol detection gives $[0, 1, 1, 0]$ and Maximum Likelihood sequence estimation (Via VA) gives $[0, 1, 0, 0]$.

4.1.1 Trellis structure in Code: There are total M^{L-1} states in the trellis. The complexity of viterbi algorithm increases with the length of the channel filter. Suppose $L = 3$ and number of states in trellis 64. The total number of outpus from the trellis is

$$trellis\_out\_size = 64 * M \tag{2.12}$$

$Trellis\_in\_struct$ and $Trellis\_out\_struct$

The ML sequence estimator structure using viterbi algorithm [**?**] is shown in [**?**] and [**?**]. To estimate a sequence of length m, the receiver complexity is M^m symbols for finding the most likely sequence among M^m sequences. However, this complexity can be reduced from M^m to M^{L-1}, where L is the number of states in trellis used for viterbi Algorithm ($L < m$).

The number of the states in the trellis increaes exponetially (M^{L-1}) with lenghth of the channel impulse response (L). Since the bandwidth requirement is 250 Khz, the length of the impulse response is suficient to assume as 3 and implemneted the viterbi algorithm with 64 states.

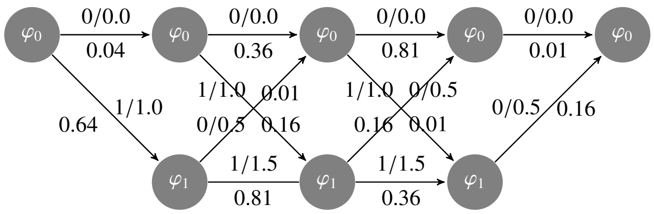

Fig. 2: Trellis Diagram with branch metrics.

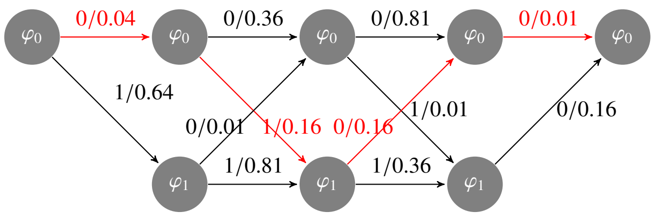

Fig. 2: Trellis Diagram with branch metrics.

5 Channel Estimation

The MLSE via Viterbi Algorithm needs the channel state information for finding the shortest path. The channel can be estimated using Fast Fourier transforms for a sequence of pilot symbols.

We have used 10 pilot symbols in the code for the estimation of channel. The observed symbols at the receiver for the pilot symbols is given by

$$\mathbf{r}_p = \mathbf{h} * \mathbf{x}_p + \mathbf{n}_p \tag{2.13}$$

The steps for channel estimation is given below

- Make $\mathbf{r}_p$ into a circular convolution of $\mathbf{h}$ and $\mathbf{x}_p$ by removing last $L-1$ symbols from $\mathbf{r}_p$ to add them to the first $L-1$ symbols .
- Find

$$Y = fft\left(\mathbf{r}_p\right) \quad X = fft\left(\mathbf{x}_p\right) \tag{2.14}$$

- Find first three taps of h_{est} from

$$h_{est} = ifft\left(\frac{Y}{X}\right) \tag{2.15}$$

The second method for channel estimation based on Toeplitz matrix method with pilot symbols is given below

- Construct a toeplitz matrix X with pilot symbols x_p.
-

$$h_est = (X^* X)^{-1} X^* y_p \tag{2.16}$$

6 Zero Forcing Equalizer and MMSE Equalizer

The Zero-Forcing Equalizer and MMSE are implemente using Toeplitz matrices [**?**].

6.1 Zero Forcing Equalizer

- Construct a toeplitz matrix H with estimated channel h_{est}.
-

$$x_{hat} = (H^* H)^{-1} H^* y \tag{2.17}$$

6.2 MMSE Equalizer

- Construct a toeplitz matrix H with estimated channel h_{est}.
-

$$x_{hat} = \left(H^* H + \frac{I}{SNR}\right)^{-1} H^* y \tag{2.18}$$

Where, I is an identity matrix and SNR is the signal-to-noise ratio.