

OptiSystem と Matlab のコ・シミュレーション事例

周波数領域と時間領域で同時に解析可能な光通信システムデザインツール OptiSystem は光通信・光ネットワークのあらゆるタイプの物理的なレイヤリンク、ロングホールから LAN、MAN までの設計と解析ができます。OptiSystem は関係製品のある光通信コンポーネント設計ツールとの統合性に加えて ADS、Microwave Office、Matlab との統合もできます。Matlab とのコ・シミュレーション機能により、Matlab を利用して新しい光デバイスを開発したり、デジタル、電気信号を処理（データフレームを作り、FEC 処理など）を行います。要するに、OptiSystem と Matlab のコ・シミュレーションにより、デジタルデータ処理、電気通信と光通信とのトータル通信システムの設計と解析が可能となります。ここで OptiSystem と Matlab のコ・シミュレーションの事例を紹介します。

Matlab を利用した LiNbO₃ Mach-Zehnder Modulator の作成例

OptiSystem はすでに 9 種類の光変調器を用意していますが、シミュレーションの精度をアップさせるため、Matlab とのコ・シミュレーション機能を利用することにより更に現実的な変調器、LiNbO₃ Mach-Zehnder Modulator を作成することができます。

Matlab 素子で作った LiNbO₃ Mach-Zehnder Modulator とそれにより構成された変調器モジュールは以下ようになります。

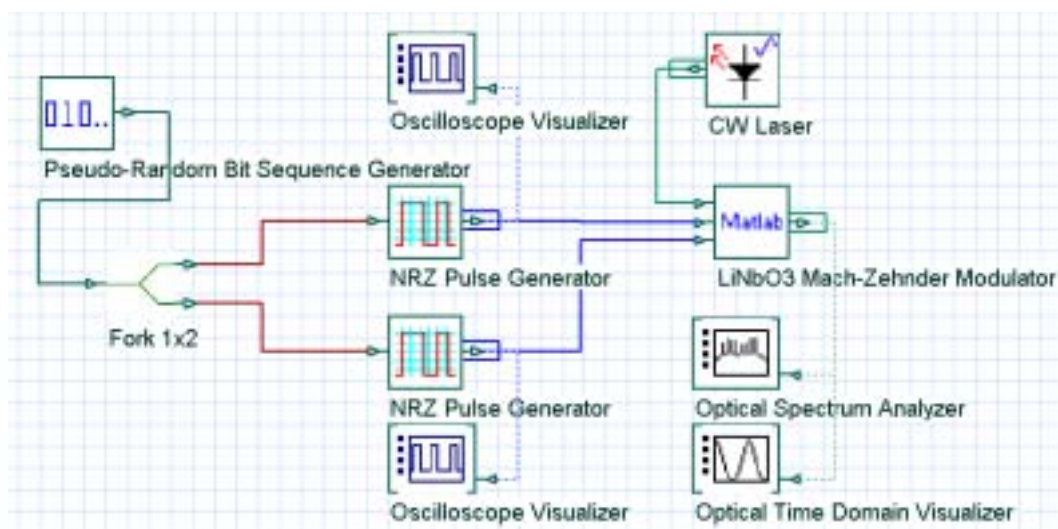


図 1 LiNbO₃ Mach-Zehnder Modulator モジュールレイアウト

LiNbO₃ Mach-Zehnder Modulator には以下のM ファイルを使っています。

```
1 OutputPort1 = InputPort1;
2
3 % Parameter0 = Vdc;
4 % Parameter1 = Vrf;
5 % Parameter2 = Vb1;
6 % Parameter3 = Vb2;
7 % Parameter4 = Loss;
8 % Parameter5 = ExtRatio;
9 % Parameter6 = Number of Samples
10
11 for i = 1:Parameter6;
12
13     Or = real(InputPort1.Sampled.Signal(i))/10^(Parameter4/10/2);
14     Oi = imag(InputPort1.Sampled.Signal(i))/10^(Parameter4/10/2);
15     Out(i) = Or +j*Oi;
16
17
18     Er = 10^(Parameter5/10);
19     Err = (1 - 1/sqrt(Er))/2
20
21
22     E1(i) = pi * InputPort3.Sampled.Signal(i)/Parameter1 + pi * Parameter3 / Parameter0;
23     E2(i) = pi * InputPort2.Sampled.Signal(i)/Parameter1 + pi * Parameter2 / Parameter0;
24
25     E11(i) = Err * exp(j#E1(i));
26     E22(i) = (1 - Err) * exp(j#E2(i));
27
28     OutputPort1.Sampled.Signal(i) = Out(i) * (E11(i) + E22(i));
29
30 end
```

図 2 LiNbO₃ Mach-Zehnder Modulator に使われた Matlab の M ファイル

この光変調器は消光比、挿入損失、両分岐導波路のバイアス電圧、スイッチングバイアス電圧、RF スwitching電圧などあらゆるパラメータを考慮できます。一旦このデバイスを Custom Library に保存すれば、いつでも使えます。

PLL 回路の作成例

高速の信号はジッタ (jitter) に対して非常に敏感です。ジッタの影響で受信信号の正しい時間位置からずれますので、符号誤り率 (BER) が高くなります。ジッタの影響をなくするために PLL 回路を利用することはクロック信号を同期させる最も一般的な手段です。以下は Matlab の Simulink を利用した PLL 回路の作成解析例です。

光通信システムの発信信号にジッタを混入させ、受信回路に Matlab 素子を利用し、クロック信号を同期させます。

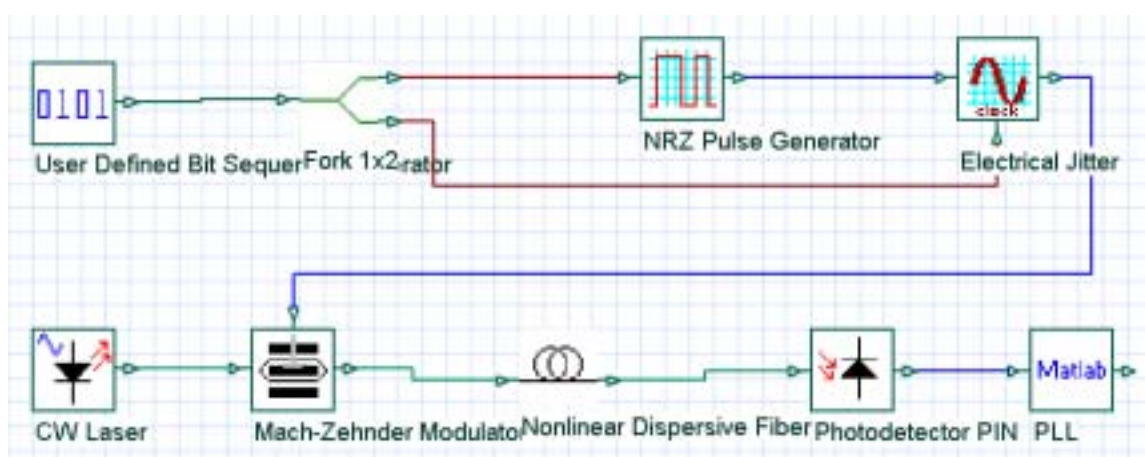


図 3 光通信システムの構成

Matlab 素子 PLL は以下の M ファイルを使っています。

```

1 | cd C:\#Work
2 |
3 | % Parameter0:Bit rate
4 | % Parameter1:Sample rate
5 | % Parameter2:Sequence length
6 | % Parameter3:Number of samples
7 | OutputPort1 = InputPort1;
8 |
9 | %time = InputPort1.Sampled.Time;
10 | %value = real(InputPort1.Sampled.Signal);
11 |
12 | K = 1/mean(real(InputPort1.Sampled.Signal));
13 | % SIMULINK execute routine
14 | open_system('slmodel3');
15 | %sim('slmodel3',[0 (20*Parameter3-1)/Parameter1])
16 | sim('slmodel3',[0 5e-5])
17 | close_system('slmodel3')
18 |
19 | out = squeeze(out)';
20 | OutputPort1.Sampled.Signal = out(1:Parameter3);
21 | clear out

```

図 4 Matlab 素子 PLL に使われた Matlab の M ファイル

この M ファイルには slmodel3 という Matlab の Simulink ファイルを使っています。Slmodel3 は Simulink で PLL 実現された PLL 回路です。Slmodel3 の構成は以下の図面の通りです。

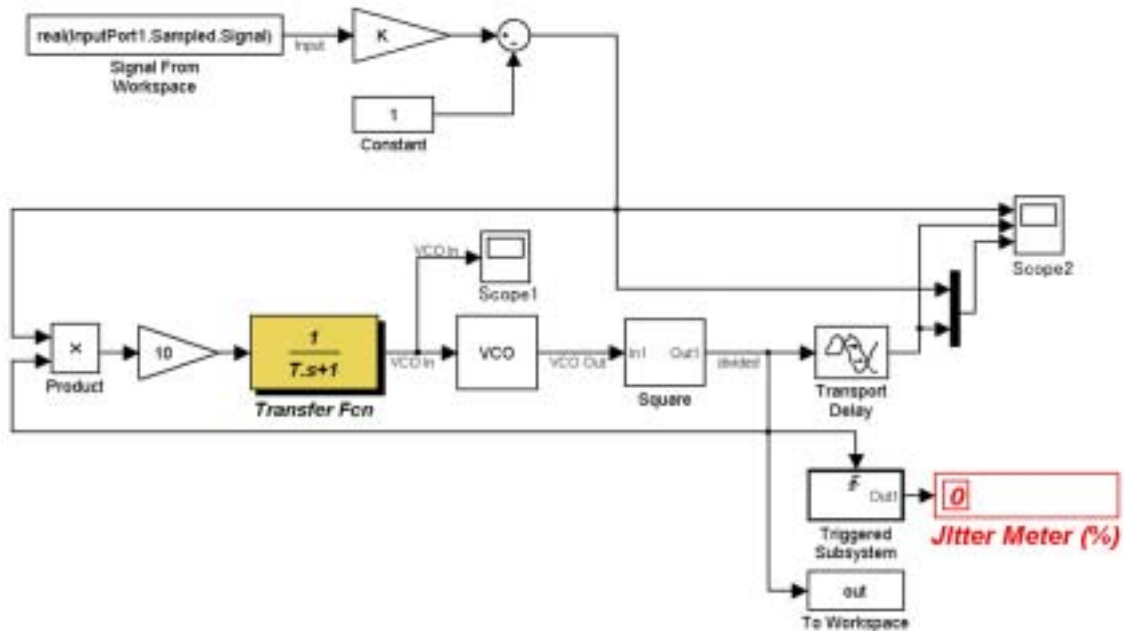
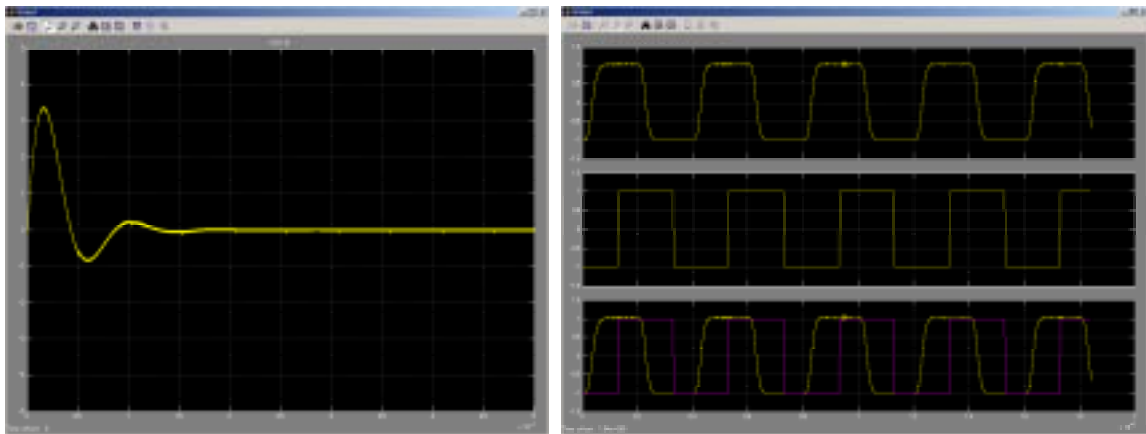


図 5 slmodel3 – Simulink で作った PLL 回路

OptiSystem で解析を行いますとクロックシグナルの動きはスコープで観測できます。



Scope1

Scope2

図 6 シグナルスコープで観測された波形

後書き

OptiSystem と Matlab のコ・シミュレーションの機能により、素子、システムのカスタマーズが可能になります。OptiSystem の応用範囲が更に広がります。どのようなアプリケーションにも対応できるという過言でもないでしょうか。