TPA 1 Grupo 10

A continuación se resuelve el trabajo de aplicación 1.

Inciso 1

El rango admisible de la entrada/salida es de 5V.

Inciso 2

Se hizo un código en c que permite generar una señal de PWM con ciclo de trabajo variable.

Con esta señal de entrada, se ingresó primero al 20% durante

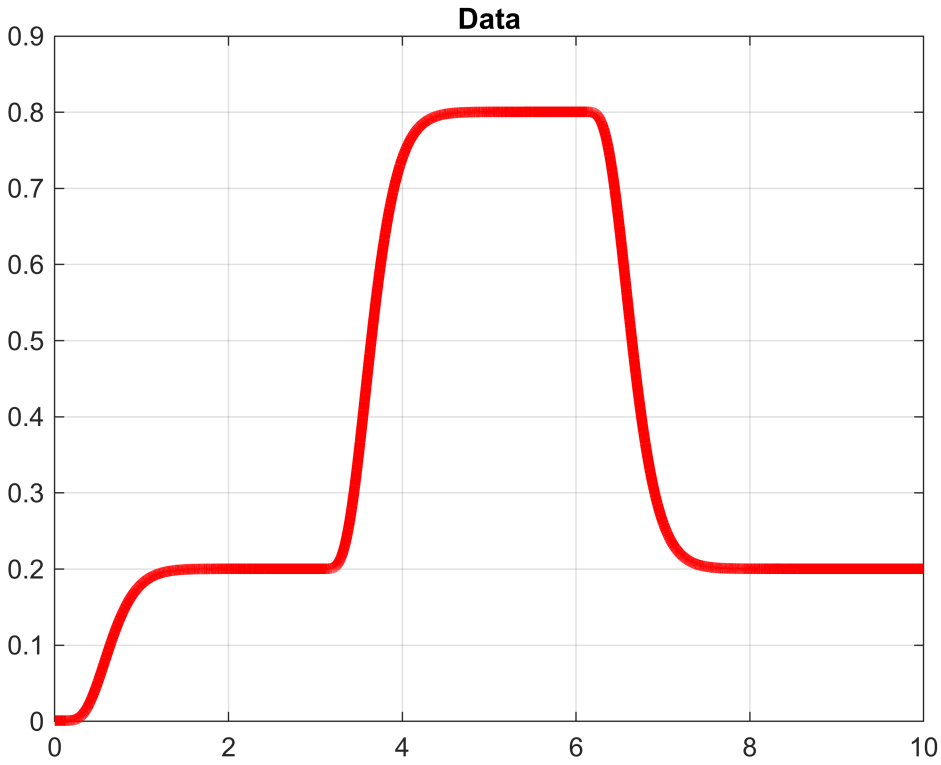
```
clear;
close all;
s = tf('s');

TIME_DIFF = 1e-3;
```

Se cargan los datos

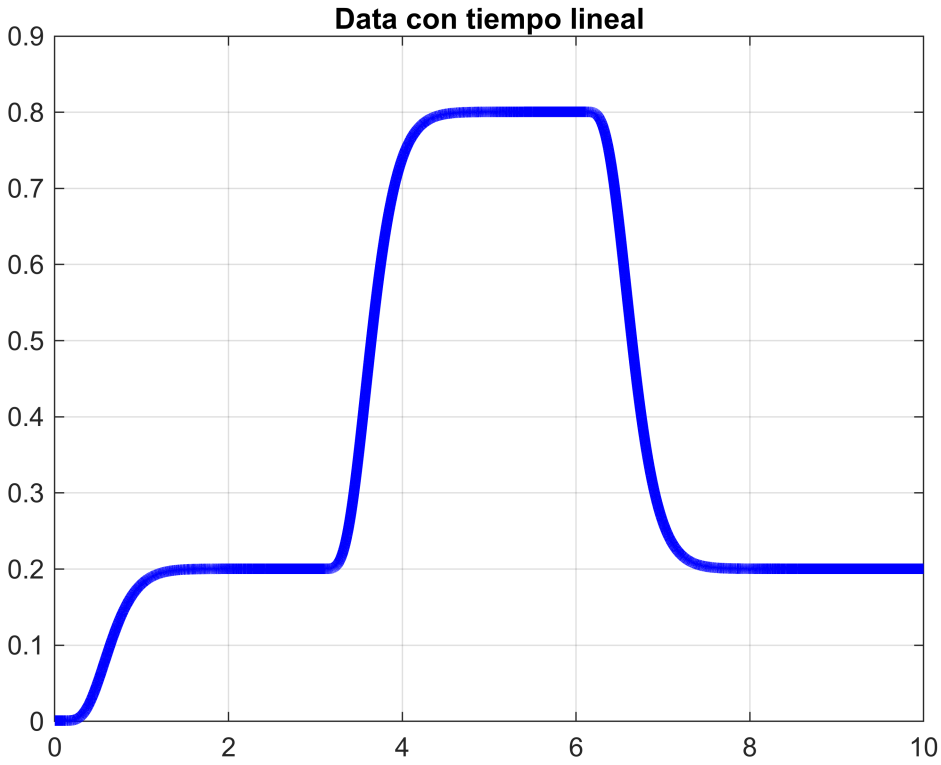
```
% addpath('c:\Users\tomi\Github\facu3\Control 3\TPA\1');
% savepath;
data = open('prot_sim_data.dat');
```

Se grafican los datos de la simulacion

```
time = data.data(:,1);
y6 = data.data(:,7)/5; % se normaliza con respecto a 5V

figure();
plot(time, y6, 'r', 'linewidth', 4);
hold on;
grid on;
title('Data');
```



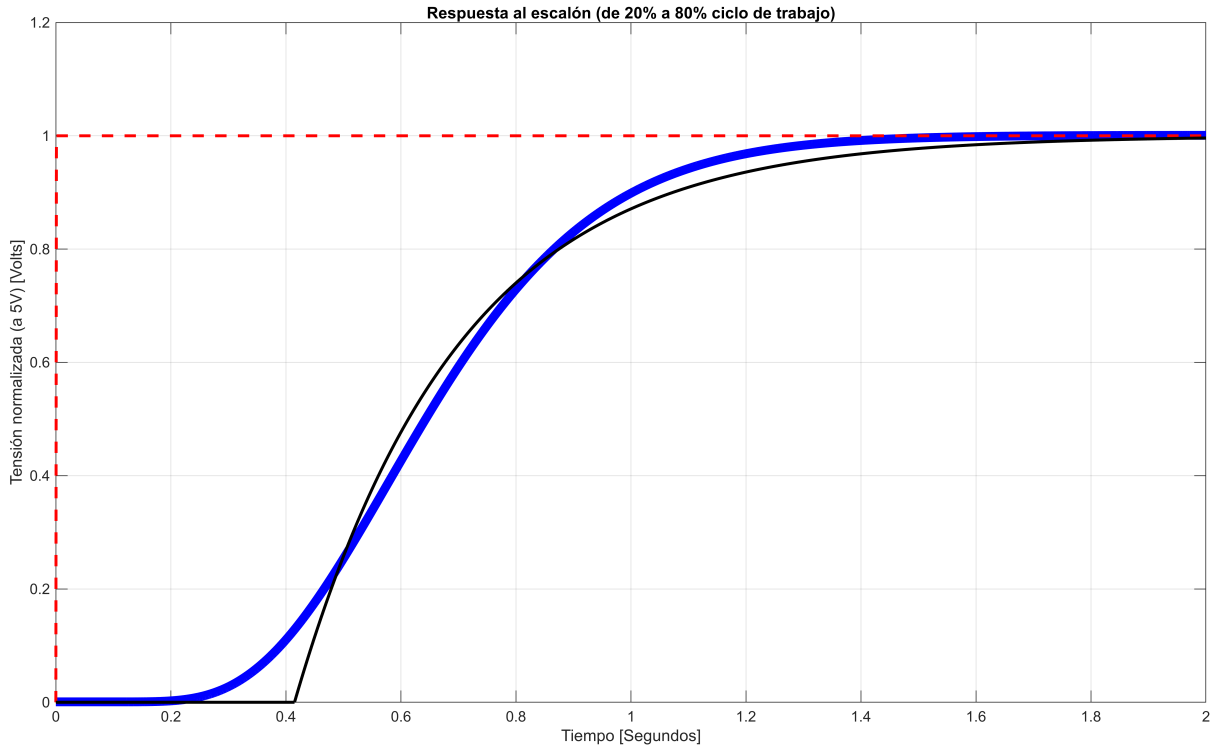

```
lineal_time = 0:TIME_DIFF:10;
lineal_y6 = interp1(time, y6, lineal_time);

figure;
plot(lineal_time, lineal_y6, 'b', 'linewidth', 4);
hold on;
grid on;
title('Data con tiempo lineal');
```



Modelo FOPDT

```
step_resp = lineal_y6((3/TIME_DIFF):(5/TIME_DIFF));
step_resp=step_resp-0.2; % muevo los datos 0.2 porque se comenzó con 20% de duty
cycle
step_resp=step_resp/0.6; % escalo a que la referencia sea 1, porque se puso 80% de
duty cycle como step
step_time = lineal_time(1:length(step_resp));
step_input = ones(1, length(step_resp)); step_input(1)=0; % esto es para el
programa de estimación que necesita una entrada

fig = figure; set(fig, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
set(fig, 'Toolbar', 'none', 'Menu', 'none');
plot(step_time, step_resp, 'b', 'linewidth', 6);
hold on;
grid on;
title('Respuesta al escalón (de 20% a 80% ciclo de trabajo)');
xlabel('Tiempo [Segundos]');
ylabel('Tensión normalizada (a 5V) [Volts]');
plot(step_time, step_input, '--r', 'linewidth', 2);

% De los graficos generados se estima el delay
fopdt_delay = 0.415;
```

```
% De System Identification se obtuvieron los siguientes parámetros
fopdt_time_constant = 0.28562;
fopdt_gain = 1;

% Modelo FOPDT resultante
fopdt_estimated = fopdt_gain/(s*fopdt_time_constant+1);
fopdt_estimated.inputDelay = fopdt_delay;

[fopdt_est_y, fopdt_est_t] = step(fopdt_estimated, step_time);
fopdt_est_y=reshape(fopdt_est_y, size(step_resp));
plot(fopdt_est_t, fopdt_est_y, 'k', 'linewidth', 2);
```




```
error_fopdt = immse(step_resp, fopdt_est_y);
fprintf('El error cuadrático medio entre la estimada y los datos es: %f\n',
error_fopdt);
```

```
El error cuadrático medio entre la estimada y los datos es: 0.001009
```

Modelo SOPDT

```
% Parámetros obtenidos de System Identification
sopdt_delay = 0.3272;
sopdt_time_constant_1 = 0.1798;
sopdt_time_constant_2 = 5;
sopdt_gain = 1;

% Modelo FOPDT resultante
```

```
sopdt_estimated = sopdt_gain/
((s*sopdt_time_constant_1+1)*(s*sopdt_time_constant_1+1));
sopdt_estimated.inputDelay = sopdt_delay;

[sopdt_est_y, sopdt_est_t] = step(sopdt_estimated, step_time);
sopdt_est_y=reshape(sopdt_est_y, size(step_resp));
plot(sopdt_est_t, sopdt_est_y, 'm', 'linewidth', 2);

error_sopdt = immse(step_resp, sopdt_est_y);
fprintf('El error cuadrático medio (SOPDT) entre la estimada y los datos es: %f\n',
error_sopdt);
```

```
El error cuadrático medio (SOPDT) entre la estimada y los datos es: 0.000259
```

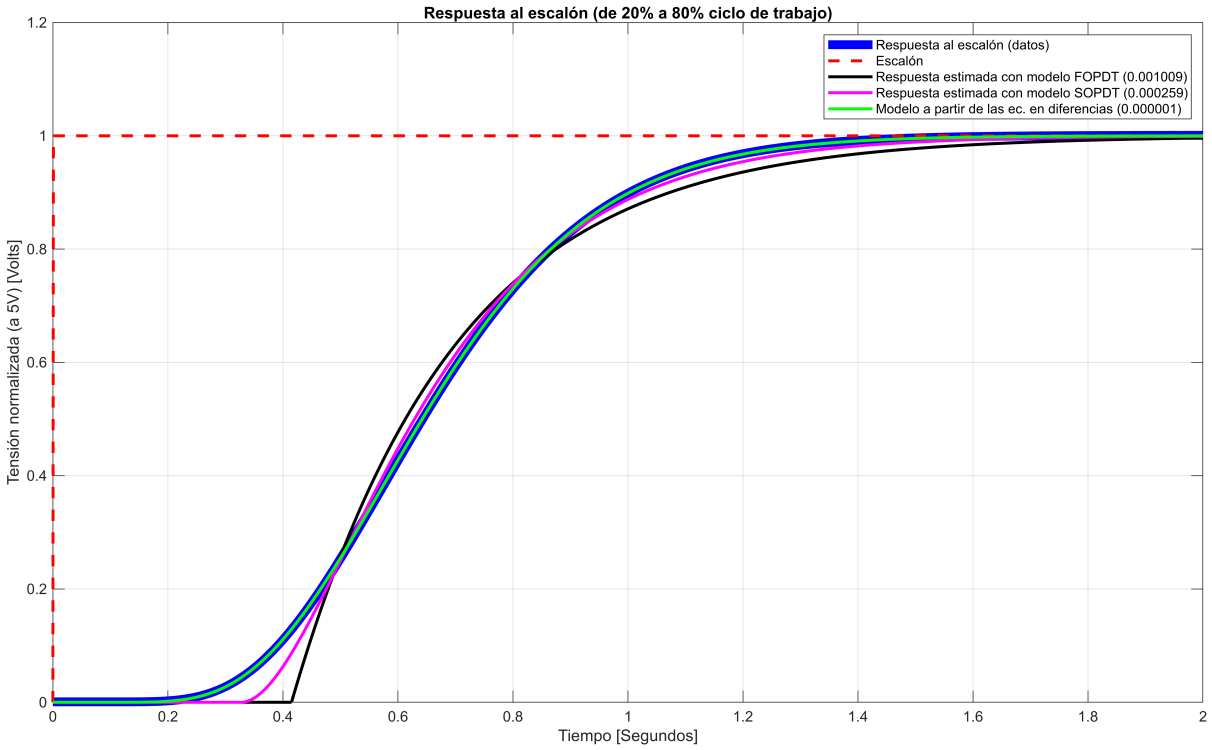
```
fprintf('La mejora del modelo SOPDT contra FOPDT es de "%f%%"\n', (error_fopdt-
error_sopdt)/error_fopdt);
```

```
La mejora del modelo SOPDT contra FOPDT es de "0.742895%"
```

```
% Modelo completo a partir de las ecuaciones en diferencias
% No entiendo por que tengo que agregarle InputDelay al modelo a partir de
% las ec. en diferencias
R1=10e3;
C1=10e-6;
complete_model = 1/((R1*C1*s+1)^6);
complete_model.InputDelay = 0.0753;

[ec_diff_y, ec_diff_t] = step(complete_model, step_time);
ec_diff_y=reshape(ec_diff_y, size(step_resp));
plot(ec_diff_t, ec_diff_y, 'g', 'linewidth', 2);

error_ec_diff = immse(step_resp, ec_diff_y);
fprintf('El error cuadrático medio (SOPDT) entre la estimada y los datos es: %f\n',
error_ec_diff);
```

```
El error cuadrático medio (SOPDT) entre la estimada y los datos es: 0.000001
```

```
legend('Respuesta al escalón (datos)', 'Escalón', sprintf('Respuesta estimada con
modelo FOPDT (%f)', error_fopdt), sprintf('Respuesta estimada con modelo SOPDT
(%f)', error_sopdt), sprintf('Modelo a partir de las ec. en diferencias (%f)\n',
error_ec_diff));
```



Modelo de alto orden con System Identification

```
% TODO
```

Señal PRBS

```
data_prbs = open('prot_sim_data_prbs.dat');
```

Extraigo los datos de la simulación

El ensaño se hace un 7 bits y considerando un tiempo de rising de 2 segundos. Por lo que el ensayo debería durar TEST_DURATION segundos

IMPORTANTE ARREGLAR LA FRECUENCIA DE MUESTREO, tiene que se como 10 veces más grande

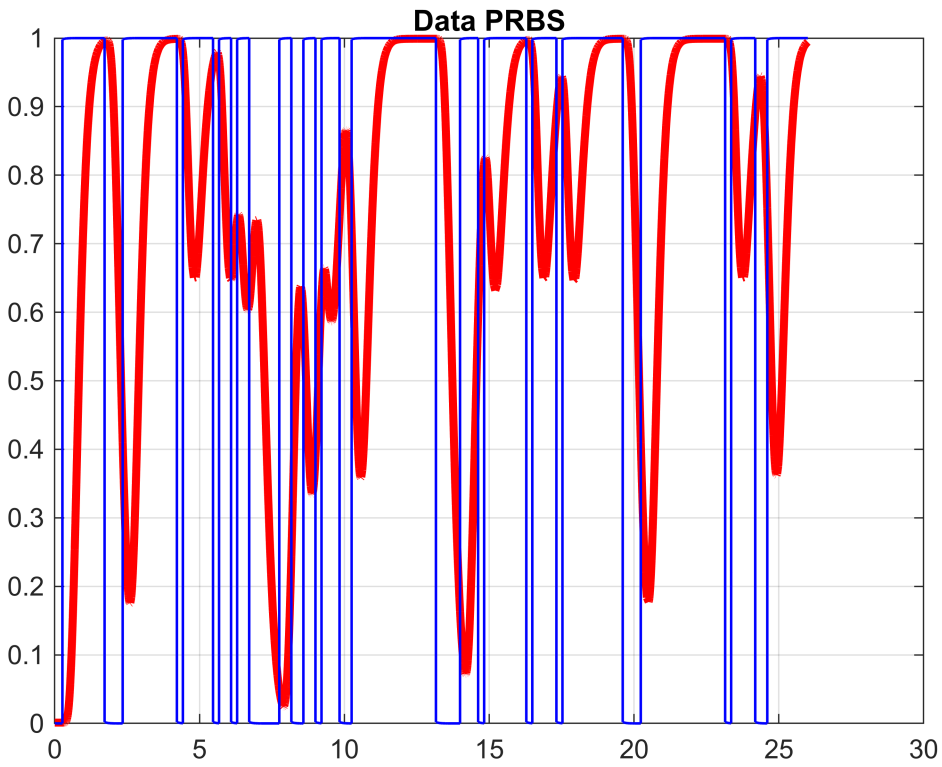
```
BITS=7;
SAMPLING_TIME=1.4/BITS
```

```
SAMPLING_TIME =
0.2000
```

```
TEST_DURATION=(2^BITS-1)*SAMPLING_TIME
```

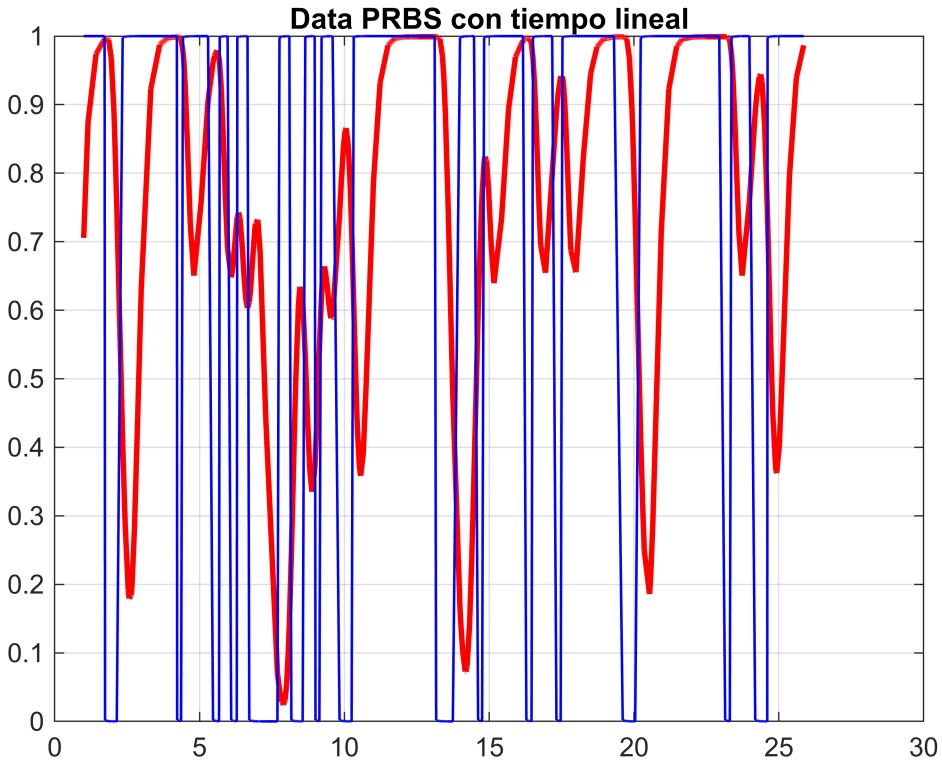
```
TEST_DURATION =
25.4000
```

```
MAX_TIME=26;

time_prbs = data_prbs.data(:,1);
pwm_prbs = data_prbs.data(:,2)/5;  % divido por 5 para normalizar los datos con
respecto al pico de 5v
y6_prbs = data_prbs.data(:,3)/5; % divido por 5 para normalizar los datos con
respecto al pico de 5v

figure();
plot(time_prbs, y6_prbs, 'r', 'linewidth', 3);
hold on;
grid on;
title('Data PRBS');
plot(time_prbs, pwm_prbs, 'b', 'linewidth', 1);
```



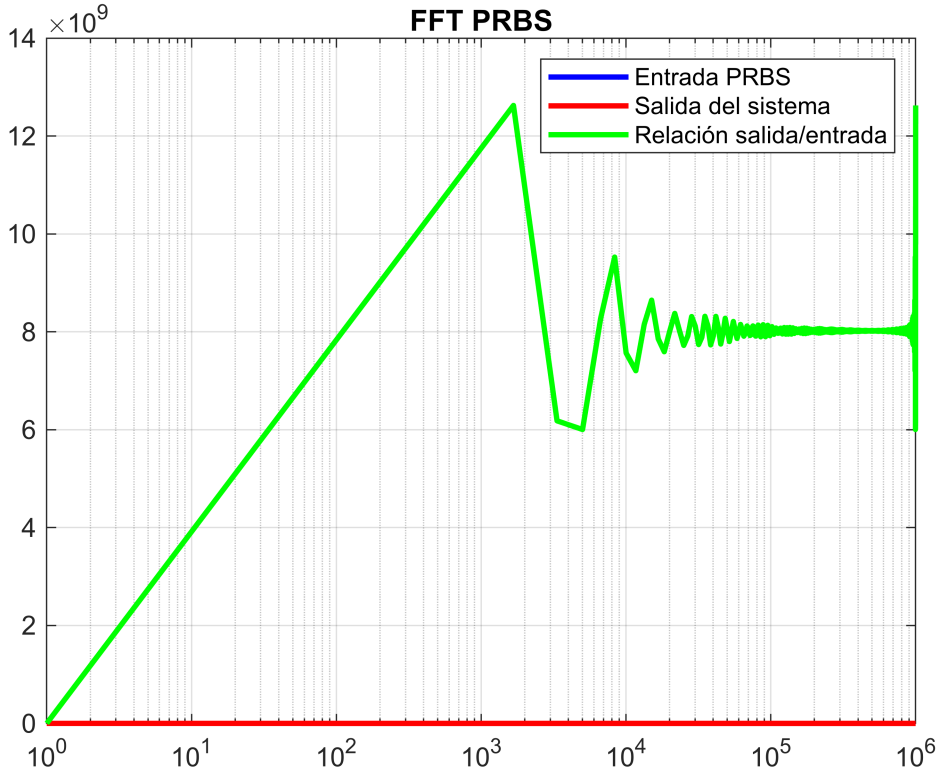

```
lineal_time_prbs = 1:TIME_DIFF:MAX_TIME;
lineal_pwm_prbs = interp1(time_prbs(1:10:length(time_prbs)),
pwm_prbs(1:10:length(time_prbs)), lineal_time_prbs);
lineal_y6_prbs = interp1(time_prbs(1:10:length(time_prbs)),
y6_prbs(1:10:length(time_prbs)), lineal_time_prbs);

figure;
plot(lineal_time_prbs, lineal_y6_prbs, 'r', 'linewidth', 2);
hold on;
grid on;
title('Data PRBS con tiempo lineal');
```

```
plot(lineal_time_prbs, lineal_pwm_prbs, 'b', 'linewidth', 1);
```



Transformada de fourier de la PRBS

```
L=600;
tf_prbs = abs(fft(lineal_pwm_prbs(1:L))).^2;
tf_sys = abs(fftn(lineal_y6_prbs(1:L))).^2;
w = linspace(1, 1e6, length(tf_prbs));

figure;
semilogx(w, tf_prbs, 'b', 'linewidth', 2);
hold on;
grid on;
title('FFT PRBS');
semilogx(w,tf_sys, 'r', 'linewidth', 2);

syst_out=tf_sys./tf_prbs;
semilogx(w, syst_out, 'g', 'linewidth', 2);
legend('Entrada PRBS', 'Salida del sistema', 'Relación salida/entrada');
```


FFT PRBS
×10^9
14
12
10
8
6
4
2
0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
Entrada PRBS
Salida del sistema
Relación salida/entrada